

2003

Effects of artificial neural networks characterization on prediction of diesel engine emissions

Azadeh Tehranian
West Virginia University

Follow this and additional works at: <https://researchrepository.wvu.edu/etd>

Recommended Citation

Tehranian, Azadeh, "Effects of artificial neural networks characterization on prediction of diesel engine emissions" (2003). *Graduate Theses, Dissertations, and Problem Reports*. 1298.
<https://researchrepository.wvu.edu/etd/1298>

This Thesis is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Thesis has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.

Effects of Artificial Neural Networks Characterization on Prediction of Diesel Engine Emissions

By

Azadeh Tehranian

**Thesis submitted to the
College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements
for the degree of**

**Master of Science
in
Mechanical Engineering**

**Nigel N. Clark, Ph.D., Chair
Richard Turton, Ph.D.
Gregory J. Thompson, Ph.D.**

**Department of Mechanical and Aerospace Engineering
Morgantown, West Virginia
2003**

Keywords: Artificial Neural Networks, Diesel Emissions, Emissions Prediction

ABSTRACT

Effects of Artificial Neural Networks Characterization on Prediction of Diesel Engine Emissions

Azadeh Tehranian

More than a century after its invention, diesel remains the fuel of choice for buses and freight trucks. Diesel exhaust contains three gases that are regulated by the United States Environmental Protection Agency (EPA), as well as particulate matter (PM). There is a societal need both to lower emissions and to predict or model emissions more accurately for inventory purposes. Engine modeling, and real time control are the most indispensable steps towards lowering engine emissions, and it is argued that this modeling can be achieved by implementation of Artificial Neural Networks (ANN). Effects of ANN design, architecture, and learning parameters on the accuracy of emissions predictions were studied along with the variation of embedded activation functions. An optimization strategy was followed to attain the most suitable network in the defined framework for five emissions of NO_x, PM, HC, CO, and CO₂. The emissions data were obtained from five engine transient test schedules, namely the E-CSHVR, ETC, FTP, E-Highway and E-WVU-5 Peak schedules. These were performed on a 550 hp General Electric DC engine dynamometer-testing unit at the West Virginia University Alternative Fuels, Engine and Emissions Research Center. The 3-Layer and Jump Connection networks were the most promising architectures and it was found that the radial basis functions such as the Gaussian and Gaussian Complement functions outperform the sigmoidal functions in all of the examined architectures. The accuracy of an excellent typical instance of CO₂ prediction was as good as 0.009% error of accumulated value over the course of a FTP cycle.

DEDICATION

To Afsaneh, for her profound empathy,
To Nick and Roya, for their remarkable nobility,
To Nasser, for his exclusive sincerity.

ACKNOWLEDGMENTS

Mostly as one of the rare documented accomplishments during ones life, the thesis is an exceptional opportunity to express special gratitude. Not as a custom but as a fortune, I take this chance to mention names of individuals to who I owe great appreciations.

I am sincerely grateful to Dr. Nigel Clark for his unique excellency in all he does. I acknowledge him for his brilliant beautiful mind. There is no way that I can thank him enough with feeble words, and I hope that I would be able to someday bequeath his goodwill to someone else.

I thank Dr. Gary Morris for his exceptional helpfulness.

I thank Ron for being such a nice patient coach.

I thank my dearly loved mom, Farzaneh, Pedar-Joon, Amme-Parvaneh, Heshmat-Khanoom and Khaleh-Soheyla. Also my very special thanks go to Brenda, Mori, Massoud, Reza and Hamid, for their generous invaluable support.

TABLE OF CONTENTS

ABSTRACT	II
DEDICATION	III
ACKNOWLEDGMENTS	IV
TABLE OF FIGURES	VIII
TABLE OF TABLES	XV
ACRONYMS	XIX
NOMENCLATURE	XXI
1. INTRODUCTION	1
1.1. MOTIVATION AND PURPOSE.....	1
1.2. PROBLEM DESCRIPTION	3
1.3. CAPABILITIES OF ARTIFICIAL NEURAL NETWORKS.....	3
1.4. LITERATURE REVIEW	6
2. ARTIFICIAL NEURAL NETWORKS	10
2.1. DEFINITION	10
2.2. HISTORY	12
2.3. VARIETIES.....	14
3. TRANSIENT CYCLES	18
3.1. CITY-SUBURBAN HEAVY VEHICLE ROUTE	18
3.2. EUROPEAN TRANSIENT CYCLE	19
3.3. FEDERAL TEST PROCEDURE.....	20
3.4. HIGHWAY CYCLE.....	21
3.5. WEST VIRGINIA UNIVERSITY-5 PEAK CYCLE	22
4. EXPERIMENTAL EQUIPMENT	24
4.1. ENGINE	26
4.2. DYNAMOMETER.....	26
4.3. DILUTION TUNNEL.....	26
4.4. SAMPLING AND MEASUREMENT SYSTEM.....	27
4.4.1. Hydrocarbon (HC) Analyzers.....	27
4.4.2. Oxides of Nitrogen (NO _x) Analyzer	28
4.4.3. Carbon Monoxide (CO) and Carbon Dioxide (CO ₂) Analyzer	28
4.4.4. PM Measurement	28
4.4.5. Data Acquisition	29
5. NEURAL NETWORK MODELING	30
5.1. PREPROCESSING OF DATA	30
5.1.1. Dispersion	30
5.1.2. Time Delay.....	31
5.1.3. Moisture Correction	34
5.2. GENERAL REGRESSION NEURAL NETWORKS (GRNN)	36

5.2.1.	Distribution Function	36
5.2.2.	Probability Density Function	36
5.2.3.	GRNN structure	37
5.3.	GROUP METHOD OF DATA HANDLING (GMDH)	39
5.4.	LEAST MEAN SQUARE (LMS).....	40
5.5.	BP DESIGN	41
5.5.1.	Bias and Threshold	43
5.5.2.	Practical Considerations.....	43
5.5.3.	Error Minimization	45
5.5.4.	Gradient Descent.....	46
5.5.5.	Momentum.....	46
5.6.	INPUTS	47
5.6.1.	Standardization	48
5.7.	ACTIVATION FUNCTIONS	49
5.7.1.	Sigmoid.....	50
5.7.2.	Radial Basis Functions.....	53
5.8.	APPLIED ARCHITECTURE	54
5.8.1.	3-Layer.....	55
5.8.2.	Ward1 and Ward2.....	56
5.8.3.	Jump Connection	57
5.8.4.	GRNN	58
5.8.5.	GMDH	58
6.	RESULTS	60
6.1.	PEARSON’S LINEAR CORRELATION COEFFICIENT	60
6.2.	COEFFICIENT OF DETERMINATION	60
6.3.	COEFFICIENT OF MULTIPLE DETERMINATION	61
6.4.	PREDICTIONS ASSESSMENT.....	61
6.4.1.	Different Runs of a test schedules	62
6.5.	NO _x PREDICTIONS	63
6.5.1.	Self-Predicting Test Schedules	63
6.5.2.	Predicting Other Test Schedules.....	74
6.5.3.	Deduction.....	78
6.6.	PM PREDICTIONS.....	79
6.6.1.	Self-Predicting Test Schedules	79
6.6.2.	Predicting Other Test Schedules.....	82
6.6.3.	Deduction.....	85
6.7.	CO PREDICTIONS.....	87
6.7.1.	Self-Predicting Test Schedules	87
6.7.2.	Predicting Other Test Schedules.....	90
6.7.3.	Deduction.....	92
6.8.	CO ₂ PREDICTIONS.....	93
6.8.1.	Self Predicting Test Schedules.....	93
6.8.2.	Predicting Other Test Schedules.....	96
6.8.3.	Deduction.....	98
6.9.	HC PREDICTIONS.....	99
6.9.1.	Self-Predicting Test Schedules	99

6.9.2.	Predicting Other Test Schedules.....	102
6.9.3.	Deduction.....	104
7.	CONCLUSIONS AND RECOMMENDATIONS.....	106
8.	REFERENCES.....	110
9.	APPENDIX A: ADDED FIGURES.....	117
10.	APPENDIX B: GMDH FORMULAS	189
10.1.	GMDH NO _x PREDICTION FORMULA	189
10.2.	GMDH TEOM PREDICTION FORMULA.....	191
10.3.	GMDH CO PREDICTION FORMULA.....	193
10.4.	GMDH CO ₂ PREDICTION FORMULA.....	195
10.5.	GMDH HC PREDICTION FORMULA.....	197

TABLE OF FIGURES

Figure 2.1. Schematic of biological and artificial neuron.....	11
Figure 2.2. Illustration of a conventional 2-Layer ANN.	14
Figure 2.3. Schematic of a Back-Propagation Neural Network (BPN).	17
Figure 3.1. Vehicle target speed versus distance for a CSHVR chassis test schedule.	19
Figure 3.2. Vehicle target speed versus time for an ETC chassis cycle.	20
Figure 3.3. Percentage of engine speed and torque versus time for a FTP engine dynamometer test.	21
Figure 3.4. Vehicle target speed versus time for a Highway chassis cycle.	22
Figure 3.5. Vehicle target speed versus time for a WVU 5-Peak chassis cycle.	23
Figure 4.1. Simplified sketch of the West Virginia University Engine and Emissions Research Center (WVU-ERC).	25
Figure 5.1. Power and rate of change of power and CO emissions versus time for an E- Highway cycle.	33
Figure 5.2. Power and rate of change of power and CO emissions versus time for an E- WVU-5 Peak cycle.	34
Figure 5.3. Schematic of a GRNN (General Regression Neural Network).	38
Figure 5.4. Schematic of a GMDH network [10].	40
Figure 5.5. Schematic of a BPN (Back-Propagation Neural Network).	42
Figure 5.6. Logistic function in the range of -1 and 1	50
Figure 5.7. Symmetric Logistic function in the range of -1 and 1	51
Figure 5.8. Sine function in the range of -1 and 1	51
Figure 5.9. Hyperbolic tangent function in the range of -1 and 1	52
Figure 5.10. Hyperbolic tangent function of $1.5x$ in the range of -1 and 1	52
Figure 5.11. Gaussian function in the range of -1 and 1	53
Figure 5.12. Gaussian Complement function in the range of -1 and 1	54
Figure 5.13. Schematic of connections between different layers in a 3-Layer back- propagation network.	56
Figure 5.14. Schematic of connections between different layers in a Ward1 back- propagation network.	56
Figure 5.15. Schematic of connections between different layers in a Ward2 back- propagation network.	57
Figure 5.16. Schematic of connections between different layers in a Jump Connection back-propagation network.	57
Figure 5.17. Schematic of connections between different layers in a GRNN network.	58
Figure 5.18. Schematic of connections between different layers in a GMDH network.	58
Figure 6.1. Power of FTP run 2 versus FTP run 1 performed on the Series 60 Detroit Diesel engine, model year 2000 dynamometer test.	62
Figure 6.2. Second by second prediction of NO_x emissions in a FTP test schedule by a 3- Layer network of Gaussian Complement activation function. The ANN was trained on all the transient test schedules combined.	77
Figure 6.3. Second by second prediction of PM (corrected TEOM for moisture content) in an E-CSHVR test schedule by a Jump Connection network of Gaussian Complement activation function. The ANN was trained on all the transient test schedules combined.	84

Figure 6.4. Second by second prediction of CO in an E-CSHVR test schedule by a Jump Connection network of Gaussian Complement activation function. The ANN was trained on all the transient test schedules combined.....	91
Figure 6.5. Second by second prediction of CO ₂ in a FTP test schedule by a 3-Layer network of both Gaussian and Gaussian Complement activation functions. The ANN was trained on all the transient test schedules combined.....	97
Figure 6.6. Second by second prediction of HC in an E-CSHVR test schedule by a Jump Connection network of both Gaussian and Gaussian Complement activation functions. The ANN was trained on all the transient test schedules combined.....	103
Figure 9.1. Variation of averaged r^2 of eight different cases for each architecture predicting NO _x emissions of an E-CSHVR in the plane of training criteria and architecture (see section 6.5.1.1).....	117
Figure 9.2. Variation of r^2 in the plane of training criterion and activation function for a GRNN architecture predicting NO _x emissions of an E-CSHVR (see section 6.5.1.1).	118
Figure 9.3. Variation of r^2 in the plane of activation function and architecture for the selected learning criteria Comb2. The ANN was trained on an E-CSHVR predicting NO _x emissions of the same test schedule. Cases are described in Table 6.9.....	119
Figure 9.4. Variation of r^2 and R^2 with all cases for the selected learning criteria Comb2. The ANN was trained on an E-CSHVR predicting NO _x emissions of the same test schedule. Cases are described in Table 6.9.....	120
Figure 9.5. Variation of r^2 in the plane of activation function and architecture for the selected learning criteria Comb2. The ANN was trained on an ETC cycle predicting NO _x emissions of the same test schedule. Cases are described in Table 6.10.....	121
Figure 9.6. Variation of r^2 and R^2 with all cases for the selected learning criteria Comb2. The ANN was trained on an ETC predicting NO _x emissions of the same test schedule. Cases are described in Table 6.10.....	122
Figure 9.7. Variation of r^2 in the plane of activation function and architecture for the selected learning criteria Comb2. The ANN was trained on a FTP cycle predicting NO _x emissions of the same test schedules. Cases are described in Table 6.11.	123
Figure 9.8. Variation of r^2 and R^2 with all cases for the selected learning criteria Comb2. The ANN was trained on a FTP predicting NO _x emissions of the same test schedule. Cases are described in Table 6.11.....	124
Figure 9.9. Variation of r^2 in the plane of activation function and architecture for the selected learning criteria Comb2. The ANN was trained on an E- Highway cycle predicting NO _x emissions the same test schedules. Cases are described in Table 6.12.....	125
Figure 9.10. Variation of r^2 and R^2 with all cases for the selected learning criteria Comb2. The ANN was trained on an E-Highway predicting NO _x emissions of the same test schedule. Cases are described in Table 6.12.....	126
Figure 9.11. Variation of r^2 in the plane of activation function and architecture for the selected learning criteria Comb2. The ANN was trained on an E-WVU-5 Peak cycle predicting NO _x emissions of the same test schedules. Cases are described in Table 6.13.....	127

Figure 9.12. Variation of r^2 and R^2 with all cases for the selected learning criteria Comb2. The ANN was trained on an E-WVU- 5 Peak predicting NO_x emissions of the same test schedule. Cases are described in Table 6.13.	128
Figure 9.13. Second by second prediction of NO_x in an E-CSHVR test schedule by a 3-Layer network of Gaussian Complement activation function. The ANN was trained on all the transient test schedules combined.	129
Figure 9.14. Second by second prediction of NO_x in an ETC test schedule by a 3-Layer network of Gaussian Complement activation function. The ANN was trained on all the transient test schedules combined.	130
Figure 9.15. Second by second prediction of NO_x in an E-Highway test schedule by a 3-Layer network of Gaussian Complement activation function. The ANN was trained on all the transient test schedules combined.	131
Figure 9.16. Second by second prediction of NO_x in an E-WVU-5 Peak test schedule by a 3-Layer network of Gaussian Complement activation function. The ANN was trained on all the transient test schedules combined.	132
Figure 9.17. Variation of r^2 in the plane of activation function and architecture. The ANN was trained on an E-CSHVR predicting TEOM emissions of the same test schedule. Cases are described in Table 6.22.	133
Figure 9.18. Variation of r^2 and R^2 with all cases. The ANN was trained on an E-CSHVR predicting TEOM emissions of the same test schedule. Cases are described in Table 6.22.	134
Figure 9.19. Variation of r^2 in the plane of activation function and architecture. The ANN was trained on an ETC predicting TEOM values of the same test schedule. Cases are described in Table 6.23.	135
Figure 9.20. Variation of r^2 and R^2 with all cases. The ANN was trained on an ETC predicting TEOM values of the same test schedule. Cases are described in Table 6.23.	136
Figure 9.21. Variation of r^2 in the plane of activation function and. The ANN was trained on a FTP predicting TEOM values of the same test schedule. Cases are described in Table 6.24.	137
Figure 9.22. Variation of r^2 and R^2 with all cases. The ANN was trained on a FTP predicting TEOM values of the same test schedule. Cases are described in Table 6.24.	138
Figure 9.23. Variation of r^2 in the plane of activation function and architecture. The ANN was trained on an E-Highway predicting TEOM values of the same test schedule. Cases are described in Table 6.25.	139
Figure 9.24. Variation of r^2 and R^2 with all cases. The ANN was trained on an E-Highway predicting the same test schedule. Cases are described in Table 6.25.	140
Figure 9.25. Variation of r^2 in the plane of activation function and. The ANN was trained on an E-WVU-5 Peak predicting TEOM values of the same test schedule. Cases are described in Table 6.26.	141
Figure 9.26. Variation of r^2 and R^2 with all cases. The ANN was trained on an E-WVU-5 Peak predicting TEOM values of the same test schedule. Cases are described in Table 6.26.	142
Figure 9.27. Second by second prediction of PM (corrected TEOM for moisture content) in an ETC test schedule by a Jump Connection network of Gaussian Complement	

activation function. The ANN was trained on all the transient test schedules combined.....	143
Figure 9.28. Second by second prediction of PM (corrected TEOM for moisture content) in a FTP test schedule by a Jump Connection network of Gaussian Complement activation function. The ANN was trained on all the transient test schedules combined.....	144
Figure 9.29. Second by second prediction of PM (corrected TEOM for moisture content) in an E-Highway test schedule by a Jump Connection network of Gaussian Complement activation function. The ANN was trained on all the transient test schedules combined.....	145
Figure 9.30. Second by second prediction of PM (corrected TEOM for moisture content) in an E-WVU-5 Peak test schedule by a Jump Connection network of Gaussian Complement activation function. The ANN was trained on all the transient test schedules combined.....	146
Figure 9.31. Variation of r^2 in the plane of activation function and. The ANN was trained on an E-CSHVR predicting CO emissions of the same test schedule. Cases are described in Table 6.31.....	147
Figure 9.32. Variation of r^2 and R^2 with all cases. The ANN trained on an E-CSHVR predicting CO emissions of the same test schedule. Cases are described in Table 6.31.....	148
Figure 9.33. Variation of r^2 in the plane of activation function and architecture. The ANN was trained on an ETC predicting CO emissions of the same test schedule. Cases are described in Table 6.32.....	149
Figure 9.34. Variation of r^2 and R^2 with all cases. The ANN trained on an ETC predicting CO emissions of the same test schedule. Cases are described in Table 6.32.	150
Figure 9.35. Variation of r^2 in the plane of activation function and architecture. The ANN was trained on a FTP predicting CO emissions of the same test schedule. Cases are described in Table 6.33.....	151
Figure 9.36. Variation of r^2 and R^2 with all cases. The ANN trained on a FTP predicting CO emissions of the same test schedule. Cases are described in Table 6.33.	152
Figure 9.37. Variation of r^2 in the plane of activation function and architecture. The ANN was trained on an E-Highway predicting CO emissions of the same test schedule. Cases are described in Table 6.34.....	153
Figure 9.38. Variation of r^2 and R^2 with all cases. The ANN was trained on an E-Highway predicting CO emissions of the same test schedule. Cases are described in Table 6.34.....	154
Figure 9.39. Variation of r^2 in the plane of activation function and. The ANN was trained on an E-WVU-5 Peak predicting CO emissions of the same test schedule. Cases are described in Table 6.35.....	155
Figure 9.40. Variation of r^2 and R^2 with all cases. The ANN was trained on an E-WVU-5 Peak predicting CO emissions of the same test schedule. Cases are described in Table 6.35.	156
Figure 9.41. Second by second prediction of CO in an ETC test schedule by a Jump Connection network of Gaussian Complement activation function. The ANN was trained on all the transient test schedules combined.....	157

Figure 9.42. Second by second prediction of CO in a FTP test schedule by a Jump Connection network of Gaussian Complement activation function. The ANN was trained on all the transient test schedules combined.....	158
Figure 9.43. Second by second prediction of CO in an E-Highway test schedule by a Jump Connection network of Gaussian Complement activation function. The ANN was trained on all the transient test schedules combined.....	159
Figure 9.44. Second by second prediction of CO in an E-WVU-5 Peak test schedule by a Jump Connection network of Gaussian Complement activation function. The ANN was trained on all the transient test schedules combined.....	160
Figure 9.45. Variation of r^2 in the plane of activation function and architecture. The ANN was trained on an E-CSHVR predicting CO ₂ emissions of the same test schedule. Cases are described in Table 6.38.....	161
Figure 9.46. Variation of r^2 and R^2 with all cases. The ANN was trained on an E-CSHVR predicting CO ₂ emissions of the same test schedule. Cases are described in Table 6.38.....	162
Figure 9.47. Variation of r^2 in the plane of activation function and architecture. The ANN was trained on an ETC predicting CO ₂ emissions of the same test schedule. Cases are described in Table 6.39.....	163
Figure 9.48. Variation of r^2 and R^2 with all cases. The ANN was trained on an ETC predicting CO ₂ emissions of the same test schedule. Cases are described in Table 6.39.....	164
Figure 9.49. Variation of r^2 in the plane of activation function and architecture. The ANN was trained on a FTP predicting CO ₂ emissions of the same test schedule. Cases are described in Table 6.40.....	165
Figure 9.50. Variation of r^2 and R^2 with all cases. The ANN was trained on a FTP predicting CO ₂ emissions of the same test schedule. Cases are described in Table 6.40.....	166
Figure 9.51. Variation of r^2 in the plane of activation function and architecture. The ANN was trained on an E-Highway predicting CO ₂ emissions of the same test schedule. Cases are described in Table 6.41.....	167
Figure 9.52. Variation of r^2 and R^2 with all cases. The ANN was trained on an E-Highway predicting CO ₂ emissions of the same test schedule. Cases are described in Table 6.41.....	168
Figure 9.53. Variation of r^2 in the plane of activation function and architecture. The ANN was trained on an E-WVU-5 Peak predicting CO ₂ emissions of the same test schedule. Cases are described in Table 6.42.....	169
Figure 9.54. Variation of r^2 and R^2 with all cases. The ANN was trained on an E-WVU-5 Peak predicting CO ₂ emissions of the same test schedule. Cases are described in Table 6.42.....	170
Figure 9.55. Second by second prediction of CO ₂ in an E-CSHVR test schedule by a 3-Layer network of both Gaussian and Gaussian Complement activation functions. The ANN was trained on all the transient test schedules combined.....	171
Figure 9.56. Second by second prediction of CO ₂ in an ETC test schedule by a 3-Layernetwork of both Gaussian and Gaussian Complement activation functions. The ANN was trained on all the transient test schedules combined.....	172

Figure 9.57. Second by second prediction of CO ₂ in an E-Highway test schedule by a 3-Layer network of both Gaussian and Gaussian Complement activation functions. The ANN was trained on all the transient test schedules combined.....	173
Figure 9.58. Second by second prediction of CO ₂ in an E-WVU-5 Peak test schedule by a 3-Layer network of both Gaussian and Gaussian Complement activation functions. The ANN was trained on all the transient test schedules combined.....	174
Figure 9.59. Variation of r^2 in the plane of activation function and architecture. The ANN was trained on an E-CSHVR predicting HC emissions of the same test schedule. Cases are described in Table 6.45.....	175
Figure 9.60. Variation of r^2 and R^2 with all cases. The ANN was trained on an E-CSHVR predicting HC emissions of the same test schedule. Cases are described in Table 6.45.....	176
Figure 9.61. Variation of r^2 in the plane of activation function and architecture. The ANN was trained on an ETC predicting HC emissions of the same test schedule. Cases are described in Table 6.46.....	177
Figure 9.62. Variation of r^2 and R^2 with all cases. The ANN was trained on an ETC predicting HC emissions of the same test schedule. Cases are described in Table 6.46.....	178
Figure 9.63. Variation of r^2 in the plane of activation function and architecture. The ANN was trained on a FTP predicting HC emissions of the same test schedule. Cases are described in Table 6.47.....	179
Figure 9.64. Variation of r^2 and R^2 with all cases. The ANN was trained on a FTP predicting HC emissions of the same test schedule. Cases are described in Table 6.47.....	180
Figure 9.65. Variation of r^2 in the plane of activation function and architecture. The ANN was trained on an E-Highway predicting HC emissions of the same test schedule. Cases are described in Table 6.48.....	181
Figure 9.66. Variation of r^2 and R^2 with all cases. The ANN was trained on an E-Highway predicting HC emissions of the same test schedule. Cases are described in Table 6.48.....	182
Figure 9.67. Variation of r^2 in the plane of activation function and architecture. The ANN was trained on an E-WVU-5 Peak predicting HC emissions of the same test schedule. Cases are described in Table 6.49.....	183
Figure 9.68. Variation of r^2 and R^2 with all cases. The ANN was trained on an E-WVU-5 Peak predicting HC emissions of the same test schedule. Cases are described in Table 6.49.....	184
Figure 9.69. Second by second prediction of HC in an ETC test schedule by a Jump Connection network of both Gaussian and Gaussian Complement activation functions. The ANN was trained on all the transient test schedules combined.....	185
Figure 9.70. Second by second prediction of HC in a FTP test schedule by a Jump Connection network of both Gaussian and Gaussian Complement activation functions. The ANN was trained on all the transient test schedules combined.....	186
Figure 9.71. Second by second prediction of HC in an E-Highway test schedule by a Jump Connection network of both Gaussian and Gaussian Complement activation functions. The ANN was trained on all the transient test schedules combined.....	187

Figure 9.72. Second by second prediction of HC in an E-WVU-5 Peak test schedule by a Jump Connection network of both Gaussian and Gaussian Complement activation functions. The ANN was trained on all the transient test schedules combined..... 188

TABLE OF TABLES

Table 5.1. Time delay estimation for CO emissions.....	33
Table 5.2. Activation function assortment for different architectures.....	59
Table 5.3. Activation function assortment for different architectures.....	59
Table 5.4. Adopted activation functions for the GRNN network architecture.....	59
Table 6.1. Average rate of change of speed over the total duration for all the test schedules.....	63
Table 6.2. Different combinations of learning criteria used for all of the examined networks.....	64
Table 6.3. r^2 values obtained from different combinations of learning criteria and activation function assortment in a 3-Layer architecture predicting NO _x emissions of an E-CSHVR.....	65
Table 6.4. r^2 values for different combinations of learning criteria and activations function assortment in a Ward1 architecture predicting NO _x emissions of an E- CSHVR.....	65
Table 6.5. r^2 values for different combinations of learning criteria and activation function assortment in a Jump Connection architecture predicting NO _x emissions of an E- CSHVR.....	65
Table 6.6. r^2 values for different combinations of learning criteria and activations function assortment in a Ward2 architecture predicting NO _x emissions of an E- CSHVR.....	65
Table 6.7. Averaged r^2 values of each architecture predicting NO _x emissions of an E- CSHVR for different combinations of learning criteria. The time is also the averaged over all of the training durations in each architecture.....	66
Table 6.8. r^2 values for different method of finding smoothing factor in a GRNN architecture predicting NO _x emissions of an E-CSHVR. r^2 and training time are both averaged in the last two columns.....	66
Table 6.9. Summary of results for all cases, using the best combination of training criteria. Trying to predict NO _x emissions of an E-CSHVR, the ANN was trained on the same test schedule.....	69
Table 6.10. Summary of results for all cases, using the best combination of training criteria. Trying to predict NO _x emissions of an ETC, the ANN was trained on the same test schedule.....	70
Table 6.11. Summary of results for all cases, using the best combination of training criteria. Trying to predict NO _x emissions of an FTP, the ANN was trained on the same test schedule.....	71
Table 6.12. Summary of results for all cases, using the best combination of training criteria. Trying to predict NO _x emissions of an E-Highway, the ANN was trained on the same test schedule.....	72
Table 6.13. Summary of results for all cases, using the best combination of training criteria. Trying to predict NO _x emissions of an E-WVU-5 Peak, the ANN was trained on the same test schedule.....	73
Table 6.14. Results obtained by predicting NO _x emissions of other transient test schedules using a network trained on an E-CSHVR. The architecture is the best E- CSHVR predictor (3-Layer, Lin-tanh15-tanh15-tanh15).....	74

Table 6.15. Results obtained by predicting NO _x emissions of other transient test schedules using a network trained on an ETC. The architecture is the best ETC self-predictor (3-Layer, Lin-tanh15-tanh15-tanh15).....	74
Table 6.16. Results obtained by predicting NO _x emissions of other transient test schedules using a network trained on a FTP. The architecture is one of the best FTP self-predictors (3-Layer, Lin-tanh15-tanh15-tanh15).....	75
Table 6.17. Results obtained by predicting NO _x emissions of other transient test schedules using a network trained on an E-Highway. The architecture is the best E-Highway self-predictor (3- Layer, Lin-Sin-Sin-Sin).	75
Table 6.18. Results obtained by predicting NO _x emissions of other transient test schedules using a network trained on an E-WVU-5 Peak. The architecture is the best E-WVU-5 Peak self-predictor (3-Layer, Lin-tanh15-tanh15-tanh15).....	75
Table 6.19. Results of NO _x predictions using a network trained on all the transient test schedules combined. The choice of architecture and activation function is based on their performance in self-prediction of each test schedule individually.	76
Table 6.20. Results of NO _x predictions obtained using a 3-Layer network with activation function of Gaussian Complement trained on all the transient test schedules combined and predicting each test schedules individually.	76
Table 6.21. Results of NO _x predictions obtained by using a GMDH network trained on all the transient test schedules combined and predicting each test schedules individually.	78
Table 6.22. Summary of results for TEOM predictions of an E-CSHVR. The ANN was trained on the same test schedule.....	80
Table 6.23. Summary of results for TEOM predictions of an ETC. The ANN was trained on the same test schedule.....	80
Table 6.24. Summary of results for TEOM predictions of a FTP. The ANN was trained on the same test schedule.....	81
Table 6.25. Summary of results for TEOM predictions of an E-Highway. The ANN was trained on the same test schedule.....	81
Table 6.26. Summary of results for TEOM predictions of an E-WVU-5 Peak. The ANN was trained on the same test schedule.....	82
Table 6.27. Results for prediction of TEOM (PM not corrected for moisture content) by using a Jump Connection network with activation function of Gaussian Complement. The ANN was trained on all the transient test schedules combined and predicting each test schedules individually.....	83
Table 6.28. Results for prediction of PM (corrected TEOM for moisture content) by using a Jump Connection network with activation function of Gaussian Complement. The ANN was trained on all the transient test schedules combined and predicting each test schedules individually.	83
Table 6.29. Results of TEOM (PM not corrected for moisture content) predictions obtained using a GMDH. The ANN was trained on all the transient test schedules combined and predicting each test schedules individually.	85
Table 6.30. Results of PM (TEOM corrected for moisture content) predictions obtained using a GMDH. The ANN was trained on all the transient test schedules combined and predicting each test schedules individually.....	85

Table 6.31. Summary of results for CO predictions of an E-CSHVR. The ANN was trained on the same test schedule.....	88
Table 6.32. Summary of results for CO predictions of an ETC. The ANN was trained on the same test schedule.....	88
Table 6.33. Summary of results for CO prediction of a FTP. The ANN was trained on the same test schedule.....	89
Table 6.34. Summary of results for CO prediction of an E-Highway. The ANN was trained on the same test schedule.....	89
Table 6.35. Summary of results CO prediction of an E-WVU-5 Peak. The ANN was trained on the same test schedule.....	90
Table 6.36. Results for prediction of CO emissions by using a Jump Connection network with activation function of Gaussian Complement. The ANN was trained on all transient test schedules combined and predicting each test schedules individually.	90
Table 6.37. Results of CO predictions obtained using a GMDH. The ANN was trained on all the transient test schedules combined and predicting each test schedules individually.....	92
Table 6.38. Summary of results for CO ₂ predictions of an E-CSHVR. The ANN was trained on the same test schedule.....	94
Table 6.39. Summary of results for CO ₂ predictions of an ETC. The ANN was trained on the same test schedule.....	94
Table 6.40. Summary of results for CO ₂ prediction of a FTP. The ANN was trained on the same test schedule.....	95
Table 6.41. Summary of results for CO ₂ prediction of an E-Highway. The ANN was trained on the same test schedule.....	95
Table 6.42. Summary of results CO ₂ prediction of an E-WVU-5 Peak. The ANN was trained on the same test schedule.....	96
Table 6.43. Results for prediction of CO ₂ emissions by using a 3-Layer network with both activation functions of Gaussian and Gaussian Complement. The ANN was trained on all the transient test schedules combined and predicting each test schedules individually.....	96
Table 6.44. Results of CO ₂ predictions obtained using a GMDH. The ANN was trained on all the transient test schedules combined and predicting each test schedules individually.....	98
Table 6.45. Summary of results HC predictions of an E-CSHVR. The ANN was trained on the same test schedule.....	99
Table 6.46. Summary of results for HC predictions of an ETC. The ANN was trained on the same test schedule.....	100
Table 6.47. Summary of results for HC prediction of a FTP. The ANN was trained on the same test schedule.....	100
Table 6.48. Summary of results for HC prediction of an E-Highway. The ANN was trained on the same test schedule.....	101
Table 6.49. Summary of results HC prediction of an E-WVU-5 Peak. The ANN was trained on the same test schedule.....	101
Table 6.50. Results for prediction of HC emissions by using a Jump Connection network with activation functions of both Gaussian and Gaussian Complement. The ANN	

was trained on all the transient test schedules combined and predicting each test schedules individually.....	102
Table 6.51. Results of HC predictions obtained using a GMDH. The ANN was trained on all the transient test schedules combined and predicting each test schedules individually.	104

ACRONYMS

ADC	Analogue to Digital Conversion
ADALINE	ADaptive LINear Element
AF	Activation Function Assortment
ANN	Artificial Neural Network
BP	Back-Propagation
BPN	Back-Propagation Networks
BSFC	Brake Specific Fuel Consumption
CO	Carbon Monoxide
CO ₂	Carbon Dioxide
Comb	Combination of Learning Criteria
CSHVR	City Suburban Heavy Vehicle Route
DC	Direct Current
E-CSHVR	Engine-City Suburban Heavy Vehicle Route
EGR	Exhaust Gas Recirculation
E-Highway	Engine-Highway Cycle
EPA	Environmental Protection Agency
ETC	European Transient Cycle
E-WVU-5 Peak	Engine-West Virginia University-5 Peak Cycle
exp	Exponential Function
FID	Flame Ionization Detector
FSN	Filter Smoke Number
FTP	Federal Test Procedure
G	Gaussian Function
GC	Gaussian Complement Function
GE	General Electric
GMDH	Group Method of Data Handling
GRNN	General Regression Neural Networks
HC	Hydrocarbons
Highway	Highway Cycle

LAF	Los Angeles Freeway Segment of FTP Cycle
LANF	Los Angeles Non Freeway Segment of FTP Cycle
Lin	Linear Function
LMS	Least Mean Square
Log	Logistic Function
NDIR	Non Dispersive Infra Red Analyzer
NNF	New York Non Freeway Segment of the FTP Cycle
NO	Nitrogen Monoxide
NO ₂	Nitrogen Dioxide
NO _x	Oxides of Nitrogen
PM	Particulate Matter
PNN	Parabolistic Neural Networks
PPM	Parts Per Million
Sin	Sine Function
SymLog	Symmetric Logistic Function
tanh	Hyperbolic Tangent Function of Variable x
tanh15	Hyperbolic Tangent Function of Variable 1.5x
TDNN	Time Delay Neural Networks
TEOM	Tapered Element Oscillating Microbalance
VDN	Vessel Dispersion Number (D/uL)
Ward1	Ward1 Network Design
Ward2	Ward2 Network Design
WVU ERC	West Virginia University Engine and Emissions Research Center
WVU-5 Peak	West Virginia University-5 Peak Cycle

NOMENCLATURE

α	Learning rate
$\Xi_j(x_i)$	Arbitrary function of variable x
$\Psi_{ij}(x_i)$	Arbitrary function of variable x
σ	Width of the bell curve
θ_i	Dimensionless time at i^{th} increment
τ	Mean value of time t
B	Bias matrix
C_1	Mass transfer constant with units of (1/s)
C_2	Unitless constant representing the state of balance between moisture concentration in the air and the TEOM filter
C_i	Concentration of gas measured by the analyzer at time t_i
C_θ	Estimated concentration of the gas in each interval of t_i
D	Equivalent diameter
$D(X)$	Distribution function of variable X
$D_{xy}(x, y)$	Joint distribution function of x and y
$E_{y x}(X)$	Conditional expectation of y for a given x
$E(x)$	Error function of x
I^n	Unit hypercube
L	Equivalent length
$m_{H_2O_a}$	Water content of air
$m_{H_2O_f}$	Water content of TEOM filter
\dot{m}_{CO}	Mass concentration of CO
\dot{m}_{CO_2}	Mass concentration of CO ₂
$\dot{m}_{H_2O_f}$	Rate of moisture adsorption on the TEOM filter
P_n	Descent direction
$P(X)$	Probability density function of variable X
r^2	Coefficient of determination

R^2	Coefficient of multiple determination
S	Speed
SS	Sum of Squares
T	Torque
t_i	Time at i^{th} increment
u	Average velocity
W	Wight matrix
$W(x)$	Weight function of x
WBP	Weight associated with back propagation algorithm
\hat{y}	ANN prediction of variable y

1. INTRODUCTION

Since the invention of the first successful gasoline-powered internal-combustion four-stroke cycle engine in 1876 by Nicholas A. Otto, drawbacks of this industry have gone hand in hand with its impressive benefits. At the beginning it was nothing more than the noise, but during the time the harm associated with exhaust odor have been considered more than a simple unpleasant scent. However, it was not until 1940's that it appeared as a pollution crisis. In the summer of 1943 the first known incident of smog happened. The haze in the city of Los Angeles thickened and the visibility of nearby mountains reduced. Crops began to show bronzing of their foliage, car tires had premature aging and health-related problems increased drastically. It was termed "gas attack" [1].

More than a century after its invention, diesel remains the preferred fuel for buses and freight trucks. The wide environmental disturbance due to diesel emissions includes direct health concerns for humans, animals and plants, as well as long-term health problems caused by phenomena such as smog, acid deposition, and toxic pollution. The diesel pollution contains more than 40 substances that are listed by the United States Environmental Protection Agency (EPA) as hazardous air pollutants and more than 30 epidemiological research have related diesel exhaust to cancer [2].

The emerging technology towards cleaner engines requires detailed analysis of correlations between engine parameters and exhaust production. The application of Artificial Neural Networks (ANN) seems to be very promising in different aspects of this issue.

1.1. Motivation and Purpose

Diesel exhaust contribution to ambient sulfur oxides, low-level ozone originators, and aerosols is seen to be a cause of chronic respiratory morbidity and mortality, and most likely contributes to the cancer risk of urban pollution. Over 30 human epidemiological studies have found a link between diesel exhaust and lung cancer. Several international studies have proved the association between particulate matters and lung diseases. A

number of studies have found that living, going to school, or working in proximity of high traffic concentration, is related to asthma, chronic bronchitis and allergies [3]. Smog causes over 6 million asthma attacks and 150,000 emergency room visits each year. Results presented by Brauer et al. [4] show that even if individuals living in areas of comparatively low ambient ozone, ozone exposure of specific groups is extensive enough to cause decline in lung function, and this decline persists into the following day. While oxides of nitrogen contribute to acid rain, research on children from communities that receive a high amount of acidic pollution show enhanced occurrences of chest cold, allergies and coughs. The toxic metals released from the rocks as a result of acid rain, can then end up in drinking water, crops and fish and if ingested by humans in great quantities can have toxic effects. For instance, Aluminum is believed to be related to Alzheimer's disease [5].

Regarding the growing pollution crisis emission regulation has been a striking topic for researchers over the last two decades. In May 2000, the Environmental Protection Agency (EPA) proposed new emission standards for heavy-duty vehicles. New gasoline heavy-duty engines used in a vehicle with a gross vehicle weight rating above 14,000 pounds must meet a combined HC (Hydrocarbons) and NO_x (Oxides of Nitrogen) standard of 1.0 g/bhp-hr. The current NO_x and HC standards are 4.0 and 1.9 g/bhp-hr, respectively [6]. Also new emission standards in Europe are as follows for the year 2005 (Euro IV): 55 g/kW-hr for HC, 3.5 g/kW-hr for NO_x, 0.03 g/kW-h for PM and 4.0 g/kW-h for CO (Carbon Monoxide) [7].

Meeting the new standards requires a good understanding of engine and vehicles features contributing to unsafe emissions. The ANN have been employed for modeling different incidents in the engines. They are widely used in areas like modeling of combustion chemical process, as well as emissions prediction and automatic control. This work is an attempt to improve the implementation of ANN in emissions prediction. Identifying the relations between engine parameters and emissions may lead to practical ways of reducing harmful emissions. The application of ANN in emission modeling has about one decade long history and is explained thorough, in the "Literature Review".

1.2. Problem Description

The role of the ANN's design and characterization in emissions prediction task was assessed along with the appliance of different transfer functions. Other determining parameters in an optimal network design such affecting features in the learning algorithm were changed in an attempt to approach the optimal network for each of the five emissions of NO_x (Oxides of Nitrogen), PM (Particulate Matter), CO (Carbon Monoxide), CO₂ (Carbon Dioxide) and HC (Hydrocarbons).

It is fairly obvious that considering all the possible variations in architecture, design, learning parameters, training set choices, activation functions and so on, it is impossible to have a comprehensive frame work to achieve an inclusive optimal choice for a particular task in emissions prediction. Regarding the previous experiments on the emissions prediction task, firstly a strategy was adopted to decide on a fair combination of learning criteria, including learning rate and momentum (for those architectures that these criteria are applicable). Afterwards the main axis of the framework were chosen to be:

- Different network architectures.
- Different combinations of activation functions.
- The measure of goodness, which is the coefficient of determination and the square of Pearson's linear correlation coefficient r^2 .

1.3. Capabilities of Artificial Neural Networks

Practical applications of ANN are in problems with abundant quantities of data including both inputs and the expected results that can be used for training the ANN. Besides, ANN are the most useful when it is hard to approach the problem with conventional algorithms; where the relationship between inputs and output cannot be defined closely by certain rules. The topic of "Artificial Intelligence" has been developed since the early 1950's as a possible solution to this category of problems, which are widely present in all branches of science and technology. Particularly in an emissions prediction task these are the

circumstances. In a book on GMDH (Group Method of Data Handling) edited by Farlow [10] the main problems associated with modeling are distinguished as:

- Analysis of interfaces between variables of system.
- Identifying of configurations of the system.
- Durable prediction of system.

Completion of these tasks requires many assumptions made by the model developer, which may lead complications and biased results. Particularly in an emissions prediction task these are the circumstances. Indeed it is realistic to identify ANN as the other way around in comparison to with traditional inductive logic. As early as 4th century B.C. Aristotle configured inductive reasoning and deductive reasoning as two different types of logic. Historically human beings have helped themselves to interpret surroundings by simplifying the phenomena and classifying them. In other words using a number of known instances to present a general conclusion. Artificial Intelligence is one step beyond all these possible classifications to trace the specifics, which do not fall into the existing categories.

Besides tracking a realistic modeling of combustion to achieve better designs, the main appealing advantage of ANN emissions prediction is associated with real-time control [11].

Capabilities of ANN can be briefly theorized through Kolmogrov's and Fourier's theorems. At the end of the 19th century the mathematician Hilbert compiled a list of 23 problems for the next century's challenge. His 13th problem was defined as if functions of several variables can be represented by the superposition of factions with fewer variables. In 1957, Kolmogrov proved that every continuous function defined on the unit hypercube $I^n (I = [0,1])$, and $n \geq 2$ can be represented as the superposition of a number of functions of one variable [12, 13].

$$f(x) = \sum_{j=1}^{2n+1} \Xi_j \left(\sum_{i=1}^d \Psi_{ij}(x_i) \right) \quad \text{Equation 1.1}$$

In an arbitrary problem it is possible to rescale the region in a hypercube I^n . In terms of ANN architecture, this means that for each input feature x_i , the sum of d nonlinear functions is needed for each $2n+1$ hidden neurons to emit the output as the summation of hidden neurons [13]. Naturally, there are some practical issues considering his theorem, for instance, smoothness of the functions is essential for gradient descent learning and very important in generalization performance of a network, while the functions Ξ_j , Ψ_{ij} are not smooth by definition. Another concern is that the process of finding an output is exactly converse of what is implied by the Kolmogorov theorem. In his theorem the number of hidden neurons are predetermined and the activation functions depend on the particular function to be approximated, while in ANN problems, the activation functions are fixed and the values of weights and biases are adjusted. In other terms, while a general continuous function has infinite degrees of freedom, it is not exactly possible to be characterized by a finite number of adjustable parameters, standing for finite number of degrees of freedom [12].

Fourier's theorem is another confirmation of approximation power of three-layer networks. It conveys that any continuous function can be approximated by an infinite number of harmonic functions. In terms of ANN modeling this can be restated that sufficient number of amplitudes and signs can be superposed to derive a desired function. However, this explanation is not that valuable from a practical viewpoint. In pattern recognition tasks the desired function is not known. Besides even if it was identified, the number of hidden nodes or proper weights are not suggested by Fourier's theorem [13].

Knowing the theoretical strengths and limitations of ANN, following comes a concise outlook of other trials in emissions modeling.

Rakopoulos et al. [14] built up a 3-D multi-zone combustion mode for a direct injection diesel engine. In general error was up to 10% for NO_x emissions prediction.

A nonlinear transient engine software along with an adaptable quasi-2-D multi-zone combustion-emission model was presented by Bazari [15] and the deviation seemed to be up to 50% for NO_x predictions.

Another predictive 3-D method was presented by Pitsch et al. [16] based on the kinetics of chemical reactions of combustion, it calculates NO_x emissions for instance by about 5% error.

Clark et al. [17], in an effort to predict NO_x missions for sixteen different chassis test schedules using three different methods, showed that the ANN models trained by axle torque and axle speed as input variables were able to predict NO_x emissions within 5 % error.

Reviewing the above mentioned models and considering the accuracy and calculation time, it is reasonable to believe in ANN are the most considerable model.

1.4. Literature Review

Plentiful applications of ANN in engine research area have started since the 90's. Earliest instances to be mentioned here are monitoring the life cycle of engine components and diagnosis of engine failures [18], and aircraft engine control modeling [19]. Since then encouraging results of ANN approaches in different fields of engine studies have been magnificent including prediction of mechanical efficiency [20], real-time assessment of vehicle drivability [21], estimating cylinder pressure and engine torque [22] and modeling human subjective responses [23]. Although during the time the most extended area of applications came out to be the engine control [24, 25].

Atkinson et al. [26] developed an ANN based engine performance, fuel efficiency and emissions prediction system in order to predict instantaneous torque and power output along with engine exhaust emissions and other outputs.

Quenou et al. [27] proposed a diesel exhaust emission modeling based on a systematic method to find ANN structural parameters. As one of the earlier works on ANN

modeling of the emissions, they chose the fuel flow, air flow and the engine speed as the inputs to a multiplayer feed forward ANN to predict the opacity of diesel exhaust emissions. They utilized the mean square error to chose the optimum hidden layer size of 8 neurons and cross correlation functions to determine the delay between inputs and output.

Traver et al. [28] investigated the possibility of using in-cylinder, pressure-based variables to predict gaseous exhaust emissions levels from a Navistar T444 direct-injection diesel engine through the use of ANN. They concluded that NO_x and CO_2 responded very well to the method. NO_x in particular gave good results, because its production is a direct result of high temperature in the cylinder and that associates directly with high pick pressure, which was their main input. They also pointed out that due to direct relation between each individual cylinder combustion event and the emission production process, dilution tunnels might create difficulties in prediction.

Thompson et al. [29] examined the possibility of predicting emission and performance of a modern heavy-duty diesel engine with stock controller by employing 3-Layer ANN. To determine the exact time delay tapped in the input, they used visual inspection and their criteria for node selection was based on the smallest possible size of the network that meets the target accuracy. Applying 10 input variables consisting of engine speed, intake air temperature, exhaust temperature, engine oil temperature, engine coolant temperature, intake air pressure, injection pressure, injection pulse width, start of injection and acceleration position, they declared that through limited, transient dynamometer testing, they were able to closely predict emissions of HC, CO, CO_2 , NO_x and PM of the engine exercised through a FTP (Federal Test Procedure) cycle.

To meet the more and more strict emission regulation, Khatri et al. [30] have attempted to use an ANN in engine management system by controlling the air fuel ratio and ignition timing.

In an attempt to show that the ANN are fast and accurate tools to predict the NO_x emission from diesel engines, Krijnsen et al. [11] developed a network and they

compared the result of its application to estimate the amount of reductant to be added to exhaust gas flow with that of engine map and linear fit. They used engine speed, rack position (as an indication of torque), charge air pressure and temperature along with a determined history of each, as inputs to a multi-layer perceptron network. The optimal number architecture was found to be one hidden layer and 30 nodes. The measurements were performed on a standard 6-cylinder, 12 L heavy-duty EURO-2 diesel engine with direct injection, turbocharger and intercooler. Their conclusion was that the high accuracy of ANN together with its capability to act as a real-time control is very promising.

Steyskal et al. [31] have developed an ANN Parametric Emissions Monitoring System (PEMS) based on general relationships between NO_x emissions and engine parameters. They have successfully used this model for predicting NO_x from large bore natural gas engine.

Clark et al. [32] used an ANN based emission model for incorporation with a software package called ADVISOR (Advanced Vehicle SimulatOR) to simulate conventional and hybrid vehicles and predict their emissions. They compared the emissions prediction results for NO_x and CO_2 with the actual emission obtained from similar vehicles tested on the WVU THDVETL (Heavy Duty Vehicle Emission Test Laboratory) and they claimed there is a good correlation between their model and emissions data from a chassis test.

In-car digital measurement data processing system along with a dynamic ANN was made by Hentschel et al. [33] as a vehicle emission modeling, employing the advantage of vehicle-specific parameters. They implanted a constant volume sampling by using the FSN (Filter Smoke Number) method that represented an integral method of measurement. Their claim was to consider the vehicle-specific parameters in a diesel engine car by using the dynamic ANN. In their result, the difference between integral values of measured NO_x and opacity was observed as a consequence of smoothing behavior of

small-sized network. Neither the detailed description of their network, nor the input variables they used was mentioned in their paper.

Desantes et al. [34] tried to form a mathematical approach that correlated NO_x and PM as a function of engine operating parameters, then to simultaneously optimize a number of operating parameters to lower the emissions. They implemented a wide range of inputs to their ANN including engine speed, fuel mass, air mass, fuel injection pressure, start of injection, exhaust gas recirculation (EGR) percentage and nozzle diameter (in the paper it is not specified which nozzle) to predict NO_x , PM and BSFC (Brake Specific Fuel Consumption) for a single-cylinder direct injection diesel engine turbocharged and after cooled with common rail injection. They used a multi layer Perceptron with a back-propagation learning algorithm and they concluded that EGR rate, fuel mass and start of injection are the most relevant parameters for NO_x , PM and BSFC. They claimed that their suggested objective function performed successfully in the task of minimizing BSFC and maintaining the emission values below the required level.

2. ARTIFICIAL NEURAL NETWORKS

2.1. Definition

Different terminologies have been used interchangeably such as: connectionism, neuro-computing or parallel distributed processing, however the most common is known to be the Artificial Neural Network (ANN). Perhaps, the simplest way of defining ANN is to say: “the ANN, like people, learn by example” [35], though there is not a universally accepted definition.

Obviously the indication of the expression suggests similarity between the natural neural networks in the brain and the artificial ones. A very simple element in the brain called neuron in connection with up to 200,000 other neurons presents the functions such as thinking and remembering. The brain is a gathering of about 10 billion unified neurons. A neuron has three modules: “Axon”, “Soma” and “Dendrite”. These modules are the counterparts of “Inputs” (along with their relative weights), “Transfer Function” and “Outputs” in the artificial neuron. The schematic of both natural and artificial neurons is shown in Figure 2.1. The inputs of an artificial neuron are taken to account after being multiplied by a weight factor, representing the contribution of that input. Each neuron has a threshold value and in the simplest case if the sum of input values is greater than the threshold there would be an output.

Some other definitions are:

“A neural network is a system composed of many simple processing elements operating in parallel whose function is determined by network structure, connection strengths, and the processing performed at computing elements or nodes” [36].

“A computer program that operates in a manner analogous to the natural neural network in the brain. The primary appeal of neural networks is their ability to emulate the brain's pattern-recognition skills. A distinguishing feature of neural networks is that knowledge

is distributed throughout the network itself rather than being explicitly written into the program” [38].

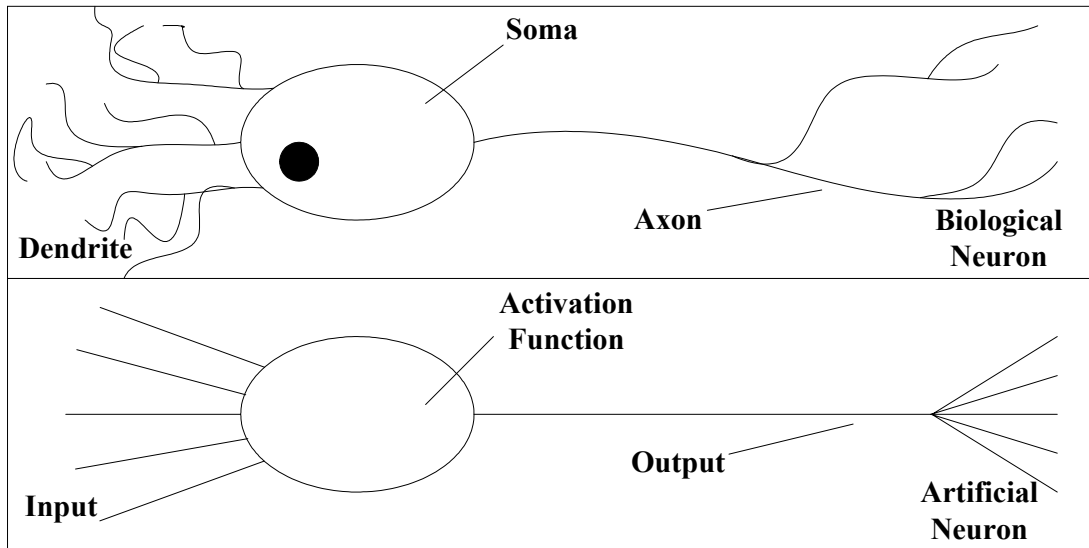


Figure 2.1. Schematic of biological and artificial neuron.

“A neural network is a massively parallel distributed processor that has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two respects:

1. Knowledge is acquired by the network through a learning process.
2. Interneuron connection strengths known as synaptic weights are used to store the knowledge” [37].

“A school of strong artificial intelligence is connectionism, which contends that cognition is distributed across a number of neural networks, or interconnective nodes. On this view, there is no central processing unit, symbols are not as important, and information is diverse and redundant. Perhaps most importantly, it is consistent with what we know about neurological arrangement. Unlike computational devices, devices made in the neural network fashion can execute commonsense tasks, recognize patterns efficiently, and learn” [39].

“A modeling technique based on the observed behavior of biological neurons and used to mimic the performance of a system. It consists of a set of elements that start out connected in a random pattern, and, based upon operational feedback, are molded into the pattern required to generate the required results. It is used in applications such as robotics, diagnosing, forecasting, image processing and pattern recognition” [40].

“An engineering and industrial field with a pragmatic approach to the use of the tools of neural networks to solve problems in the real world and build electronic devices to support these applications. Performance and ease of implementation are primary drivers, and many incremental improvements and hybrid approaches have evolved that combine neural networks, time-frequency feature extraction, and simple heuristics” [40].

2.2. History

As early as 1873, in a valuable review of body-mind study entitled “Mind and Body. The Theories of Their Relation”, Alexander Bain marked the first natural neural network of human’s body [42]. Diverse sketches of ANN can be found in Herbert Spencer's “Principles of Psychology”, published in 1872, Theodore Meynert's “Psychiatry”, published in 1884, William James’ “Principles of Psychology”, published in 1890 and Sigmund Freud's “Project for a Scientific Psychology”, composed in 1895 [43].

The earliest theoretical approach to ANN was developed in 1943 by McCulloch (neurophysiologist, University of Illinois) and Pitts (mathematician, University of Chicago). They created a logic circuit composed of simple elements mimicking the role of interconnected neurons. These elements were considered as binary tools resulting in Boolean phrases [44].

The first learning law for the ANN was designed by psychologist Donald Hebb of McGill University. In his book “The Organization of Behavior”, he stated that the correspondence of a simultaneous activity between two neurons should be performed by increasing the strength (weight) of the connection between them [45]. The usage of reinforcement in early ANN promoting feasible computer simulations was done by IBM

researchers Farley and Clark in 1954. Collaborating closely with neuroscientists at McGill University they used reinforcement learning to adjust the parameters of linear threshold functions [46]. As an alternative of Farley and Clark's method, Rochester et al. [47] modified Hebb's rule by normalizing the weights [48]. This idea was close to an expanded form of Hebb's learning rule developed by Kohonen and Anderson in 1972.

In 1958 the first ANN "Mark I Perceptron", that could actually learn and recognize the optical patterns was presented by Frank Rosenblatt [49] of the Cornell Aeronautical Laboratory. In 1959, Widrow and Hoff of Stanford University developed an analogue system called ADALINE (ADaptive LINear Element) with a different method of learning from Perceptron. Their method is still in commercial use [48]. One of the most famous unsupervised learning algorithms known to date in the ANN was developed by Kohonen and Anderson in 1972. They developed associative procedures independently, using matrix mathematics to illustrate their ANN [48].

To recognize all the pioneers of the ANN during 70's and 80's, one should refer to Henry Klopf, Steve Grossberg and Gail Carpenter, Paul Werbos, Bernard Widrow, Walter Freeman and Shun-Ichi Amari [50].

Today, the ANN are very promising for the problems that are not easy to be handled by conventional algorithms. Besides technology, they have proven to be useful from business to medicine. Their future appears to be very intense and bright. Companies have already started developing specialized microchips for ANN applications. In his recent book, "The Age of Spiritual Machines", Kurzweil [51] has gazed far to the future, believing that huge amounts of information in the form of ANN models can be exchanged between human brains. However this seems to be an illusive image of the future, the fact is that the United States is definitely the leading country for both theory and applications in ANN, and the related industry is estimated to be \$100 million per year and increasing [52]. It is expected that the ANN produce solutions to specific industrial problems in pattern recognition, control, design, logistics machinery monitoring, adaptive routing in power grids, financial predictions, credit-risk assessment, and autonomous

vehicle control. For real word problems, neural systems would likely consist of a number of ANN rather than a single identical ANN [40].

2.3. Varieties

New variations of ANN are invented often. ANN are a clustering of mainly three layers of neurons; input layer neurons communicates with the outer environment to obtain inputs. The output layer exports the output from the system and there are usually a number of hidden layers in between. The number of hidden layers is usually determined by the best possible results after trial and error. A network consisting of one input layer, one hidden layer, and one output layer by definition is called a two-layer network. Figure 2.2 illustrates a conventional 2-Layer network.

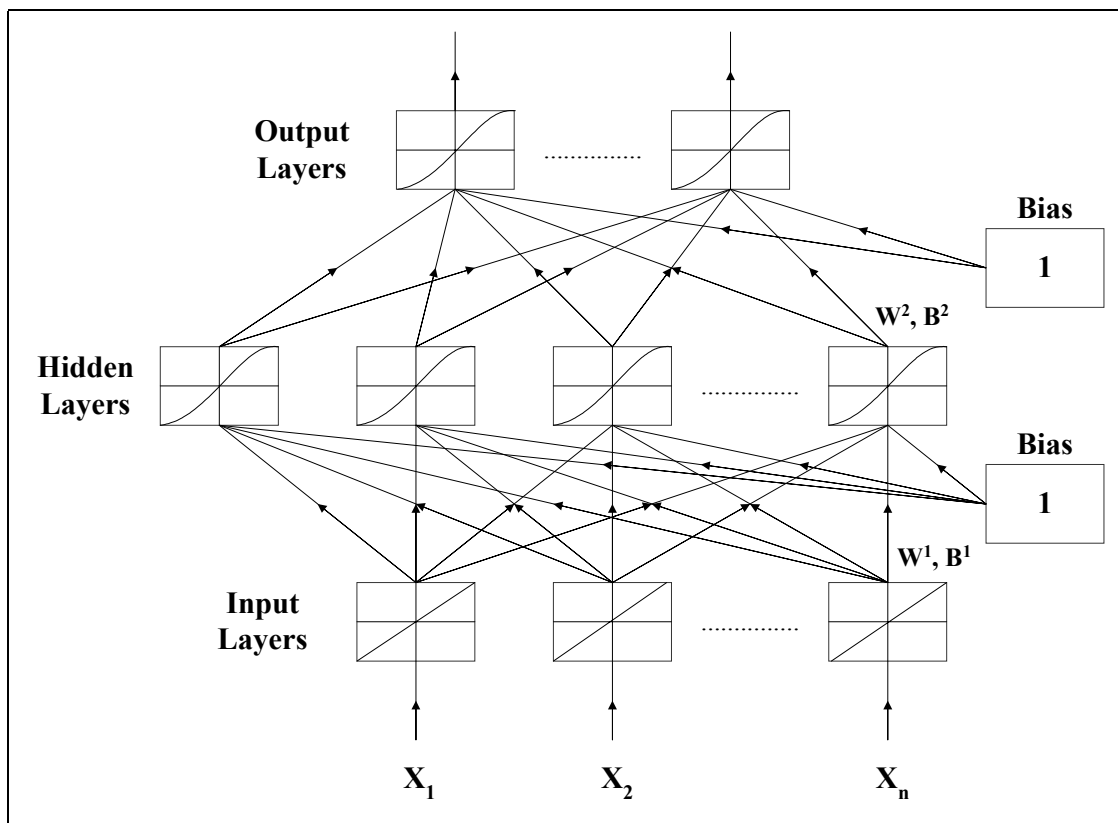


Figure 2.2. Illustration of a conventional 2-Layer ANN.

The ANN can be classified according to their learning algorithm, architecture, topology, data entry and some other labels. Some have considered the main classifications according to:

- The learning algorithm; this way the ANN can be divided to three groups of “supervised”, “unsupervised” and “reinforced”. Supervised learning algorithm is based on comparing the network’s result with the actual available values and then adjusting the weights consequently. This is perhaps the most typical network setting. Unsupervised learning works through utilizing the inputs just by classifying them into a set of possibilities; in this case there is no target value as in supervised learning. The reinforced algorithm uses the target values like the supervised method but it assigns a grade to different sequences of network’s outcome using different series of inputs. This grade quantifies how good the actual and target results conform.

- The structure architecture; there are not basically more than two types: “single-layer” and “multi-layer”. Single-layer network contains no hidden layer and consists of just input layer neurons and output layer neurons, while multi-layer contains one or more hidden layer.

- The topology of nodes and links between them; this label can group three categories of “feed-forward”, “recurrent” and “competitive”. In feed-forward networks there is no effect of output feedback from previous iteration; signals stream in forward direction from input nodes to output nodes. In a recurrent network, there is a circulation of the signal pathway from each neuron back to itself. A competitive network is the one in which all the neurons are interconnected and the information can pass through each in every direction. There could be another way of considering links; “fully connected”, “partially connected”, “hierarchical” and “resonance”. Fully connected is where each and every neuron on the first layer communicates with all neurons of the next layer, while in partially connected this is not essential. In hierarchical network the neurons of a lower layer are connected to the higher-

level neurons. Resonance is the kind in which there is definite condition preset and until reaching that condition neurons are in connection in both ways [53, 54].

“Back-propagation” is a general learning algorithm for supervised feed-forward multi-layer networks, based on minimizing the error of network’s output. It is also referred to as “generalized delta rule”, “Windrow-Haff learning rule” and “Least Mean Square learning rule”. It was first introduced in 1974 by Werbos and later by Parker in 1982 and also Rummelhurt and McClelland in 1986 [55]. It now stands among the most used types of ANN. Back-propagation is the short term for the backwards propagation of error. Once an ANN is faced with a data set, it generates a random estimate and then by confronting the actual desired output biases the respective weights to minimize the error an approach the answer. A gradient descent inside the consequent results converges to a “global minimum” which is indeed the smallest possible error. Irregularities in the solution space in practical problems may lead the network to swing in between pits and valleys of the surface and end up in a local minimum, which may not be the best overall solution. A schematic of back-propagation network is shown in Figure 2.3.

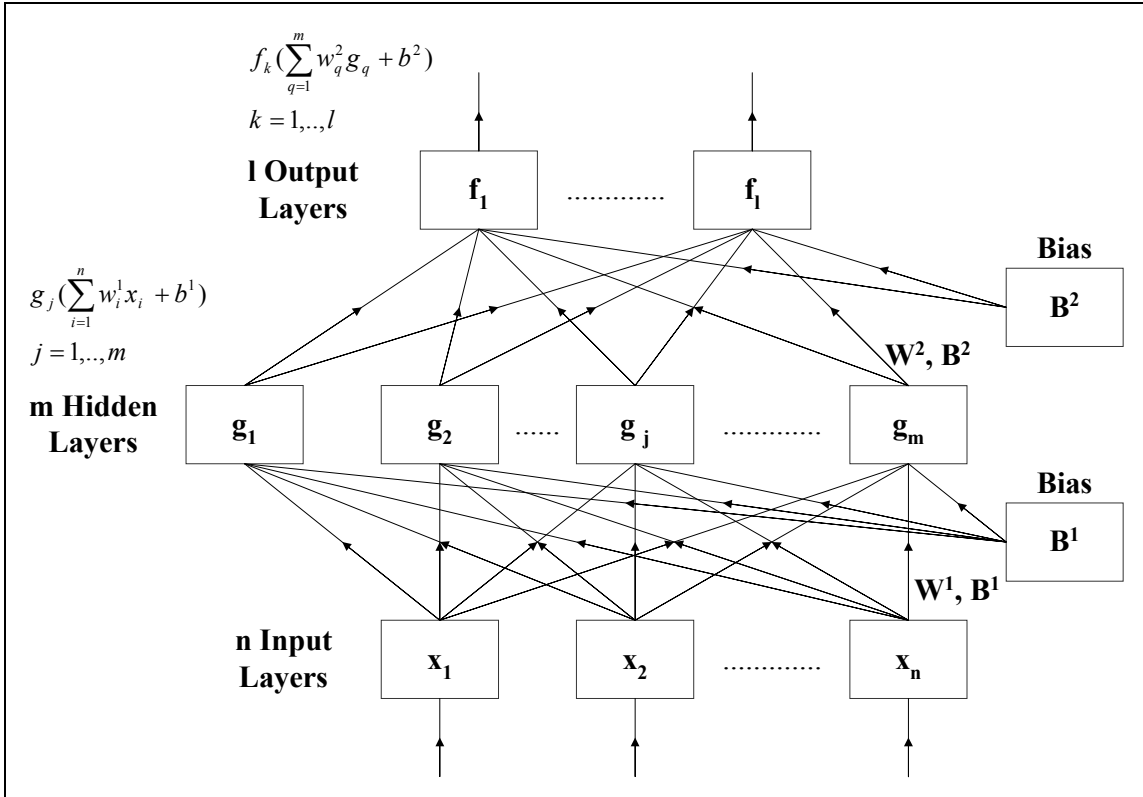


Figure 2.3. Schematic of a Back-Propagation Neural Network (BPN).

3. TRANSIENT CYCLES

In this research the emissions data were obtained from five transient test schedules performed on an engine dynamometer testing unit. Except the Federal Test Procedure (FTP) and European Transient Cycle (ETC) that are originally created for engine dynamometer tests, the City Suburban Heavy Vehicle Route (CSHVR), Highway and West Virginia University- 5 Peak (WVU-5 Peak), are originally chassis test schedules and were adapted to be performed on a dynamometer test and through this thesis they are denoted as E-CSHVR, E-Highway and E- WVU-5 Peak. This conversion has been an in-house work conducted at the West Virginia University Engine and Emissions Research Center (WVU-ETC). The basic idea has been using the good engineering judgment and experience of driving, to find out the engine speed for a known vehicle speed. The engine speed was worked out, taking the vehicle speed and load from the schedule, knowing the gear ratio of the transmission and rear-end of the vehicle. The exact gear in use was guessed using the experience of driving. The final set points were then ran through a simulation to be verified if they matched the chassis values reasonably [56].

3.1. City-Suburban Heavy Vehicle Route

The City Suburban Heavy Vehicle Route (CSHVR) was developed at West Virginia University [57] to mimic the real life driving in an urban environment. This test schedule was obtained as a route by converting the speed versus time to speed versus distance route with free acceleration ramps. The total traveled distance is 6.69 miles with maximum speed of 43.8 mph. The average speed is 14.6 mph and the nominal duration is 1700 seconds. The following Figure 3.1 shows the target speed versus distance for a chassis CSHVR test schedule.

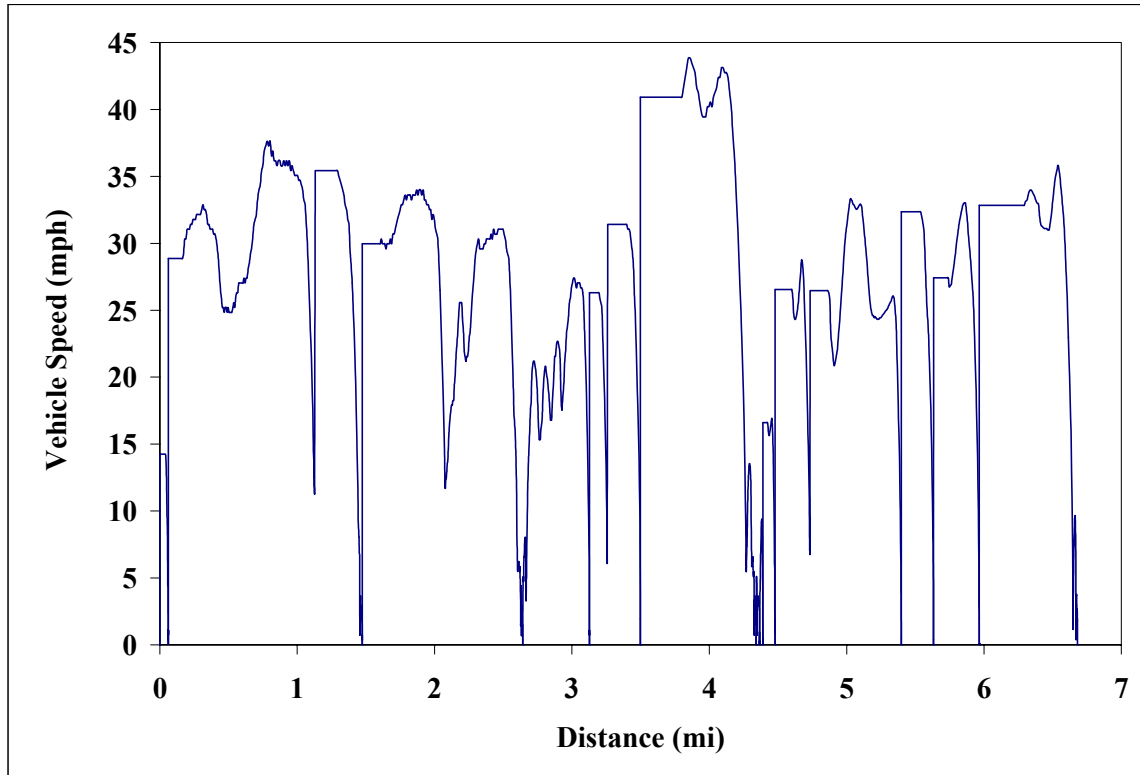


Figure 3.1. Vehicle target speed versus distance for a CSHVR chassis test schedule.

3.2. European Transient Cycle

The European Transient Cycle (ETC), also known as FIGE, is a cycle developed by the FIGE institute at Aachen, Germany in two variants of chassis and engine dynamometer, for emission certification of heavy-duty diesel engines in Europe. As an 1800-second test schedule, it consists of three 600-second segments representing city, rural and motorway driving. The first part simulates the driving condition in the city with recurrent stops and idling condition, and a maximum speed of 31.07 mph. The second part is a sudden acceleration with an average speed of 44.74 mph. The third segment is a freeway drive with 54.68 mph average speed. The total traveled distance is 18.37 miles [58]. Figure 3.2 shows the scheduled speed versus time for this cycle.

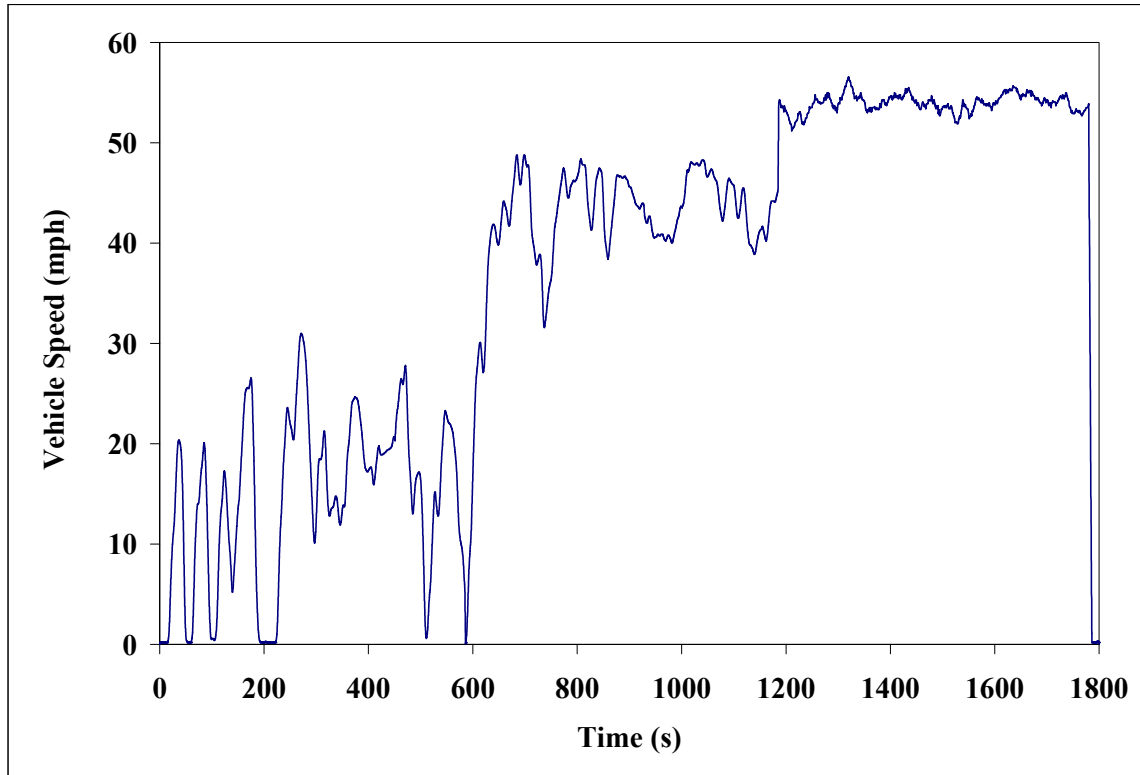


Figure 3.2. Vehicle target speed versus time for an ETC chassis cycle.

3.3. Federal Test Procedure

The FTP (Federal Test Procedure) heavy-duty transient cycle is designed to cover a wide range of operation for heavy-duty engines, inside and outside of cities in North America. It consists of four segments. The first and the last is New York Non Freeway (NNF) representing light city traffic, the second is Los Angeles Non Freeway (LANF) representing heavy city traffic, and the third is Los Angeles Freeway (LAF) representing packed highway traffic. With a total duration of 1200 seconds, the whole test is performed two times with a cold start and a warm start. After completion of the first run, which is representative of a run following an overnight stop, and a 1200 seconds pause, the second run is performed as a warm start. Figure 3.3 shows the speed and torque percentage of the engine versus time for this cycle [58].

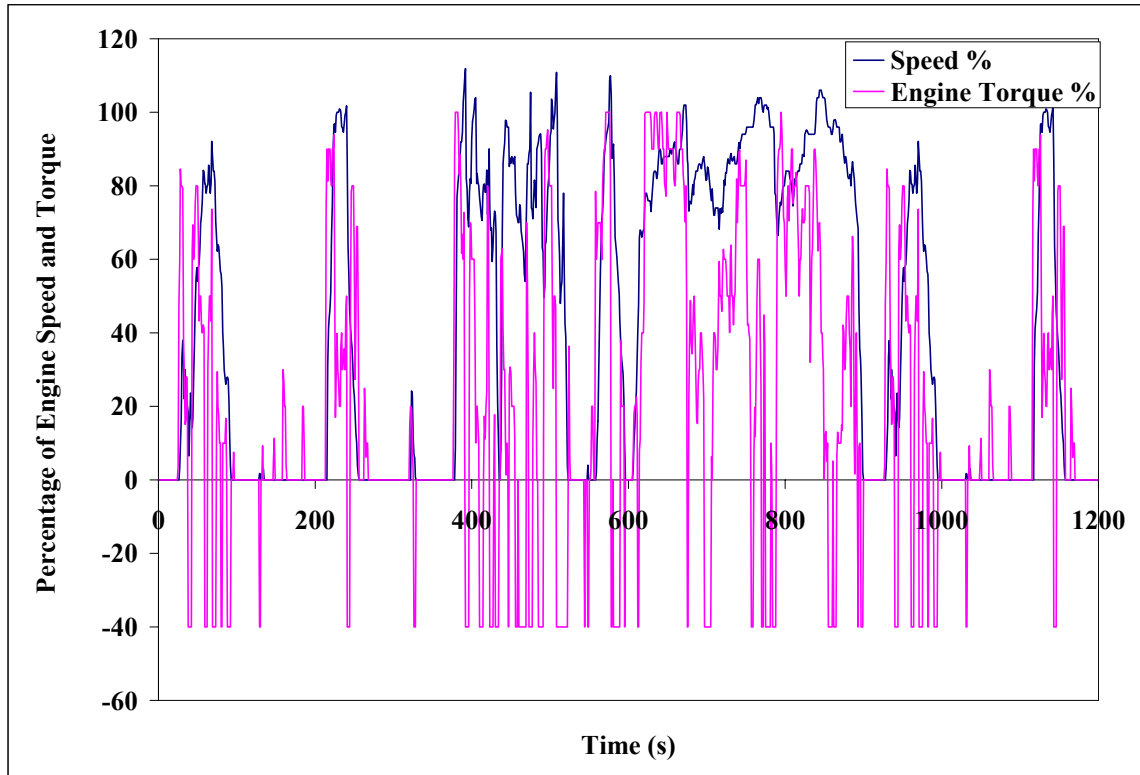


Figure 3.3. Percentage of engine speed and torque versus time for a FTP engine dynamometer test.

3.4. Highway Cycle

The Highway cycle was developed at West Virginia University [68] to simulate driving conditions in freeways. It is a high-speed cycle with an average speed of 34.05 mph and maximum speed of 60.7 mph. The total duration is 1640 seconds and the traveled distance is 15.51 miles. Figure 3.4 shows speed versus time for this cycle as a chassis test schedule.

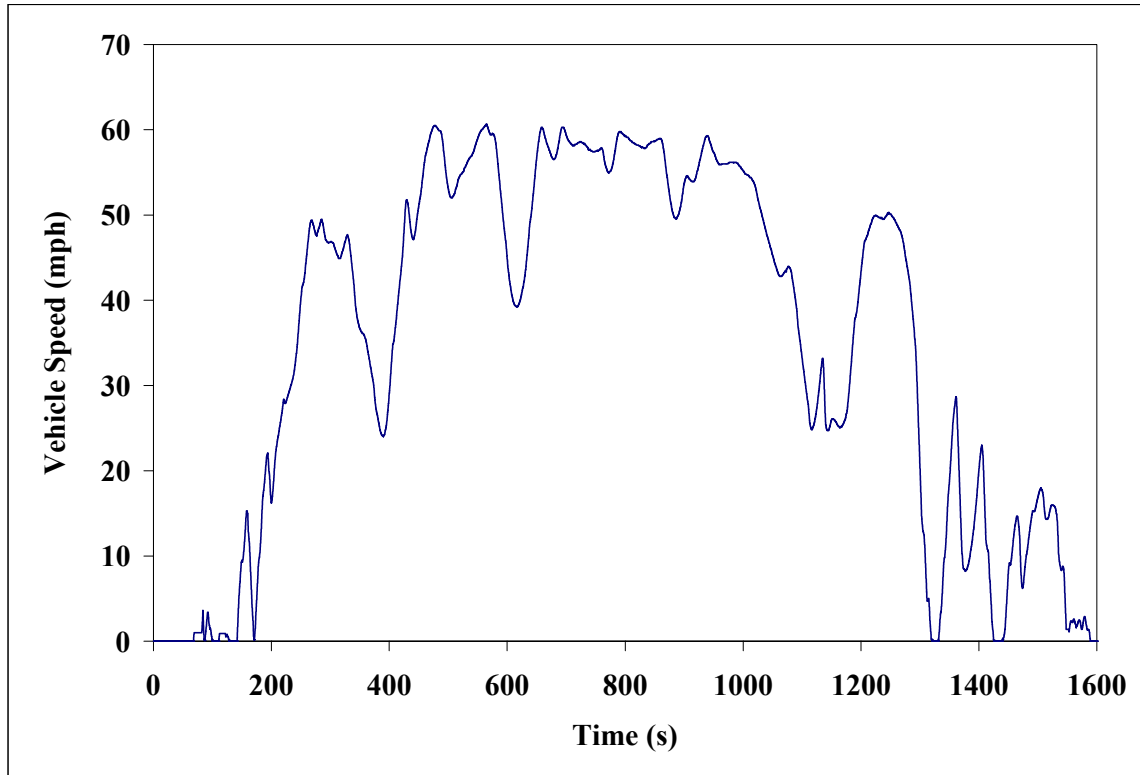


Figure 3.4. Vehicle target speed versus time for a Highway chassis cycle.

3.5. West Virginia University-5 Peak Cycle

The WVU-5 Peak cycle was developed at West Virginia University in a geometric form, consisting of 5 segments. Each segment consists of acceleration from idle to a peak speed, then maintaining that speed for a concise period, and deceleration back to the idle. As shown in Figure 3.5, the maximum speeds in 5 segments are respectively 20, 25, 30, 35, and 40 mph. The total duration of the cycle is 900 seconds, and the equivalent traveled distance is 5 miles [58].

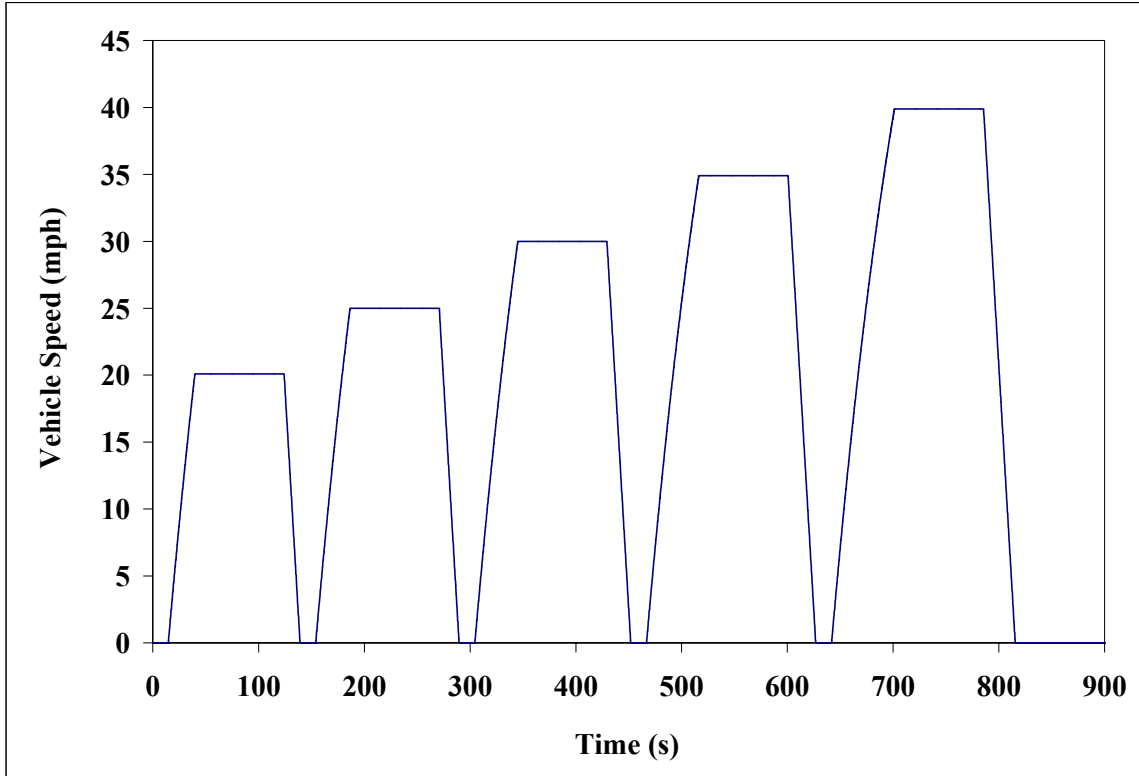


Figure 3.5. Vehicle target speed versus time for a WVU 5-Peak chassis cycle.

4. EXPERIMENTAL EQUIPMENT

The standard emission measurement for on-road heavy-duty engine emissions, according to the United States Code of Federal Regulations (CFR), Title 40, Part 86, Subpart N utilizes a test cell along with a dynamometer, a full scale dilution tunnel and gas analyzers [59]. During the test, a transient test schedule consisted of certain variations of engine speed and torque is performed on the engine. The dynamometer simulates the resistant load on the engine. In the dilution tunnel the exhaust is diluted by fresh air, trying to approach what happens in real world. At the end of the dilution tunnel, gas analyzers sample the diluted exhaust through the heated supply lines. The concentration of gaseous emissions is identified by the analyzers and reported to the data acquisition system to be recorded. The continuous data in grams per second are obtained by multiplying the concentration in parts per million (ppm) by density, volumetric flow rate, and some correction factors as instructed by the CFR, Title 40, Part 86, Subpart N.

All the data used in this research were taken at the WVU-ERC. Established in 1993 and in compliance with the CFR, Title 40, Part 86, Subpart N, this center utilizes a full flow dilution tunnel critical flow venturi constant volume sampling system. The schematic of equipment is illustrated in Figure 4.1. The performance of this particular set of tests was carried out by Bane [9] in completion of his thesis work.

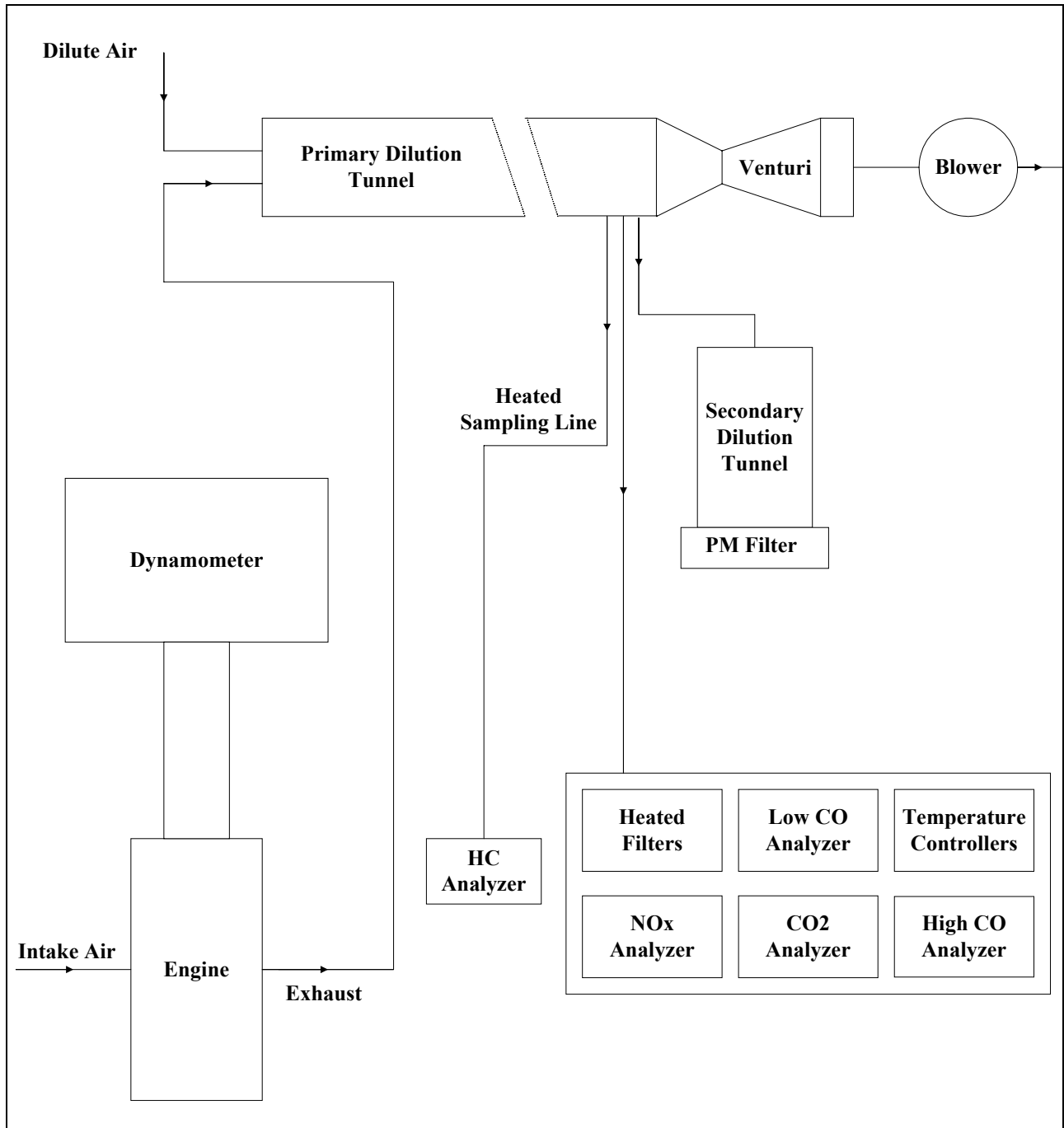


Figure 4.1. Simplified sketch of the West Virginia University Engine and Emissions Research Center (WVU-ERC).

4.1. Engine

The data used for this research was obtained from testing a Series 60 Detroit Diesel engine, model year 2000. The engine was a turbocharged, electronically controlled, heavy-duty diesel engine, with applications in both on and off-road trucks and marines. This is also the most popular heavy-duty engine in North America [60]. For this engine the maximum brake horsepower of 550 bhp is reached at 2100 rpm. According to the engine map the maximum torque is 1,670ft-lbs at 1450 rpm. The engine is a four-stroke, high speed, with 6 cylinders and compression ratio of 15:1. With a total displacement of 12.7 L, the engine has a bore of 130 mm and stroke of 160 mm.

4.2. Dynamometer

By definition, a dynamometer is a machine consisting of a motor, a cooling system, and a signal feedback representing torque, speed, current, etc. In a test arrangement the dynamometer is used to imitate the vehicle inertia, including, drag, friction, weight and grade. The engine flywheel is coupled to the dynamometer via a bonded-rubber coupling and a drive shaft. When the engine is running, the dynamometer exerts a braking force on the engine. A digital encoder inside the dynamometer measures the engine speed. The torque is calculated as the product of force measured at an arm of known length and that fixed length.

The dynamometer used in the WVU-ERC is a Direct Current (DC), General Electric (GE) model DC-243, capable of absorbing 550 hp and motoring up to 500 hp.

4.3. Dilution Tunnel

Similar to the real life exhaust emissions, the engine exhaust mixes with fresh air in a full-scale dilution tunnel. Cooling and mixing with air are the results of this mixing that lead to a condition of increased PM formation. This dilution with air also reduces the condensation of water as a primary product of combustion, which can be an interference to the analyzers. The flow rate in the dilution tunnel is much easier to determine compared with the flow rate of exhaust only. In addition, the reduction of gas

concentration into the measurable range of conventional analyzers is achieved by the means of dilution tunnel.

The exhaust pipe of 5 inches diameter from the Series 60 engine was connected to the 18 inches diameter stainless steel dilution tunnel. The tunnel is 40 feet long and utilizes a 75 hp blower fan to draw the diluted exhaust through the tunnel and venturies. Three feet from the entrance an orifice serves the purpose of mixing the fresh air and exhaust. Fifteen feet further down-stream, the sampling probes are place to draw a certain portion of the flow to analyzers, via heated sampling lines. One of the sampling lines draws to the secondary dilution tunnel in charge of PM collection. For the data used in this research, the PM data were measured continuously by the TEOM (Tapered Element Oscillating Microbalance). To control the flow in the tunnel and provide a constant volume sampling, a variation of critical flow venturies are used and can provide up to 3400 scfm. Increasing the velocity to the chocking condition at the venturi, results in the constant mass flow rate through the venturi.

4.4. Sampling and Measurement System

Through the stainless steel heated sampling probes and heated sampling lines, a portion of diluted exhaust is carried to gas analyzers. Heated pumps are also used to transfer the samples to the analyzers at a constant temperature maintained by the temperature controllers. The CO and CO₂ samples are dried by a chiller, since water is an interference to the detection of CO and CO₂. Prior to the test performance, all the analyzers are calibrated, using gases of known concentrations in ppm. To account for exhaust component present in the ambient air, bag sampling of the ambient air is done. The bags are analyzed by the same analyzers and the gas concentration is corrected for the background concentration.

4.4.1. Hydrocarbon (HC) Analyzers

HC emissions were measured using a Flame Ionization Detector (FID), Beckman analyzer Model 402. HC components of exhaust are ionized in a flame produced by a Hydrogen-Helium mixture. These ions collected by polarized electrodes induce current in

a circuit, proportional to the number of carbon atoms. Consequently by assuming a certain proportion of carbon atoms in a HC molecule, the concentration of HC is obtained [61].

4.4.2. Oxides of Nitrogen (NO_x) Analyzer

Oxides of Nitrogen (NO_x) in the exhaust emission include mainly Nitrogen monoxide (NO) and Nitrogen dioxide (NO₂). NO_x emissions were measured by a chemiluminescent analyzer, model 955 NO/NO_x Rosemount. The quantity of NO_x is measured through converting the NO₂ components of the sample gas to NO in a heated tube. In a chemiluminescent analyzer, the NO reacts with ozone. The product of this reaction is excited NO₂ that radiates energy in the form of photons. Detection of these photons is done by a photo-multiplier detector, which generates a DC current and finally an output voltage of 0 to 5 volts [62].

4.4.3. Carbon Monoxide (CO) and Carbon Dioxide (CO₂) Analyzer

CO emissions were measured using a Horiba AIA 210 LE, low CO analyzer for detecting low concentration of CO. CO₂ emissions were measured by a Model 868 Beckman Industrial analyzer. Both analyzers were Non-Dispersive Infrared (NDIR) analyzers. Element atoms are unique in their orbit spacing and can absorb or emit certain wavelengths of spectra. Therefore while observing a continuous spectrum, a missing wavelength or reduced intensity in a wavelength indicates the presence of a certain element in its pathway. Based on this phenomenon, the NDIR is a common method for measuring CO and CO₂. The method includes sensitizing ray path to the component and sensitizing ray path to the pure measuring gas. For this, the analyzer contains a sealed reference cell and a flow through sample cell. The difference between the total infrared energy absorption in the cells is constantly measured and represents the CO or CO₂ concentration. The overlap of water absorption band with CO and CO₂ makes it necessary, to dry the sample gas prior to measurement [63, 64].

4.4.4. PM Measurement

According to the Code of Federal Regulation (CFR) the standard way of measuring PM is through gravimetric measurement of PM collected by filters. However, this does not

carry any information regarding the transient formation of PM during the instantaneous combustion event. Although not accepted for certification purposes, the real-time PM measurement is done by Tapered Element Oscillating Microbalance (TEOM). Due to specific objective of this research the PM measured by TEOM was used as the raw data and corrected for moisture content (as detailed in Chapter 5), after the TEOM values were predicted by the ANN. A Rupprecht & Patashnick Co., Inc. Diesel Particulate Monitor TEOM Series 1105, connected to a secondary dilution tunnel with a 2 lpm flow was the analyzer used for PM data. In this method, the concentration of PM is determined by continuously weighing particles deposited on a replaceable Pallflex TX40 filter. The filter is attached to a hollow tapered element, vibrating at its natural frequency of oscillation. The natural frequency of oscillation changes by the mass change of the filter as particulate matters collect on the filter. The mass rate is calculated by subtracting the consequent readings of frequency and converting them to mass by a constant obtained during calibration. This is done every 0.42 seconds. This measurement can be very accurate in terms of small mass change measurements, given that frequency can be measured precisely. However, moisture and Volatile Organic Compounds (VOC) adsorption to and evaporation from the filter, exterior vibrations and other interferences can affect the accuracy of this system in terms of reflecting real transient PM data [65].

4.4.5. Data Acquisition

An RTI-815 digital to analog data acquisition board together with 3-B modules and a computer are the interface between the acquired signals and the recording system. The data obtained during a test are recorded in form of Analogue to Digital Conversion (ADC) and would change to conventional units during the data reduction process. Using Visual Basic software developed at WVU-ERC, the functional data are selected and calculated in conventional units.

5. NEURAL NETWORK MODELING

5.1. Preprocessing of Data

5.1.1. Dispersion

In very simple language, the axial dispersion in fluids is the result of relatively weak intermolecular forces in fluids that cause random fluctuations of velocity in each element of the fluid, namely molecular diffusion. Passing through free path or pipes, gaseous molecules collide due to their kinetic energy and at each collision they change direction and scatter. This phenomenon in turbulent flow was first observed in 1921 by Taylor. He tried to correlate the dispersion of particles to the turbulence of the fluid. Ever since then, there have been considerable efforts for detailed characterization of Taylor's works [66].

Standard emission measurement method requires full-scale dilution tunnel. Axial dispersion in the mixed flow of exhaust and fresh air occurs in dilution tunnel and sampling lines. Also the analyzers spread the signals in time. Considering the known relationship between power and emissions, in order to correct the errors associated with fluid diffusion in emission measurement Clark et al. [67], implemented a dispersion model to axle power. In this case power is also assumed to be subjected to diffusion and an improvement in correlation between power and emissions would be achieved easily. However research is being done for successful back transformation of measured emissions to yield instantaneous tailpipe emissions [68]. The original dispersion model was developed by Levenspiel et al. [69]. Introducing D as the equivalent diameter for the system with an average fluid velocity of u in a tunnel of length L , (D/uL) is the dimensionless group called Vessel Dispersion Number (VDN). C_θ , the estimated concentration of the gas in each interval of t_i , can be expressed in terms of these variables by the following equation:

$$C_\theta = \frac{1}{2\sqrt{\pi(D/uL)}} \exp\left[-\frac{(1-\theta_i)^2}{4(D/uL)}\right] \quad \text{Equation 5.1}$$

Where $\theta_i = t_i/\tau$, in which τ is the mean value and is calculated by the following formula:

$$\tau = \frac{\sum_{i \rightarrow n} t_i C_i}{\sum_{i \rightarrow n} C_i} \quad \text{Equation 5.2}$$

Here C_i is the concentration measured by the analyzer at time t_i . Jarrett [70] suggested a modification to this model by adding two variables ‘ a ’ and ‘ b ’ as shown in the following equation:

$$C_\theta = \frac{1}{2\sqrt{\pi\theta_i(D/uL)}} \exp\left[-\frac{(a-\theta_i)^2}{4\theta_i^b(D/uL)}\right] \quad \text{Equation 5.3}$$

He tried to optimized the error between the actual emissions measurements and the modified dispersion model predictions of the same emissions. By varying a , b and VDN ., there would be a set of variables that minimizes the discrepancy between actual measurements and the values offered by the dispersion model. Results showed an improvement in predicting the dispersed curve from a pulse injection, compared to the original model. The optimized result, was obtained by using: $a = 0.9324$, $b = 2.6624$, and $VDN = 0.0098$. It was considered in this work that the integral of Equation 5.3 from $-\infty$ to $+\infty$ does not equal 1, however the error associated with it, considering the time frame of gas traveling in the dilution tunnel is not more than 4%.

Another approach for modeling the emissions dispersion is done by Clark et al. [71], assuming a gamma function as the ultimate pattern in which the emissions diffuse. In this research, Levenspiel’s method of dispersion was applied to the engine data of torque and speed prior to their introduction to ANN. It was assumed that the effects are the same for all the emissions and the radial dispersion is negligible.

5.1.2. Time Delay

There is a time delay between measured engine parameters and the measured emissions. Transport time in the dilution tunnel (however, it is negligible) and sampling line, along with the analyzer’s response time add up to make a considerable time delay as mentioned. Analyzer’s respond time can bring up more complications, since it may change due to the change in the gas concentration. It is essential to consider the time delay in emissions data, prior to any further analysis. Because of the known correlation

between power and emissions of NO_x and CO₂, it is possible to use the instantaneous measured power to find the time delay in measured emissions of NO_x and CO₂.

One way to determine the exact time difference between the measured emissions and power is to find the maximum correlation region between these two with respect to a reference spectrum, which is time in this case. This is done by utilizing cross correlation envelope functions. In the simplest case, in an arbitrary time frame, this could be done by finding the maximum product of multiplying the two series. Basically, by assuming that the chosen time window contains at least one full period of each series, the maximum product of multiplication occurs, when the correct time shifting is done. This logic has been used in this research to find the time delay associated with HC, CO₂ and NO_x measurements. All the data series were cross-correlated against power and the resulted time shift was applied prior to ANN analysis.

There are additional considerations for CO and PM. These emissions are believed to be more affected by transient effects, leading them to be dependent on the rate of change of power as well as the power itself. This brings up the issue of contribution factors for power and rate of change of power versus time and deciding which one would be of a greater magnitude and consequently would be a better reference for cross correlating. The following strategy was pursued to find a practical approach. Assuming that CO for an instant is a function of power and rate of change of power, one could say:

$$\dot{m}_{CO} = aP + b \frac{dP}{dt} \quad \text{Equation 5.4}$$

Since a and b are dependent upon the engine specifications, for a series of transient cycles performed on a certain engine they should remain the same. Two quite different cycles in terms of power demand and rate of change of power, were chosen to assess the contribution of power and its first derivative in CO production. The E-Highway cycle has an approximately constant power derivative over a broad range of time, due to the maintained speed. The E-WVU-5 Peak cycle in contrast has a highly nonlinear transient power derivative versus time. For these two cycles cross correlation was performed between CO emissions and both power and rate of change of power. Table 5.1 shows

different results obtained for time delays. It would be informative to also review Figure 5.1 and Figure 5.2 presenting power, its first derivative and CO, versus time. The reasonable confirmation between power and rate of change of power results was the ground of choosing the power as the basis of cross correlation in all other data sets. In this research all emissions were time shifted, employing the described method.

	Peak to Peak (s)		Cross Correlation (s)	
	P and CO	dP/dt and CO	P and CO	dP/dt and CO
E-Highway	13	-	13	14
E-WVU-5 Peak	24	42	20	18

Table 5.1. Time delay estimation for CO emissions.

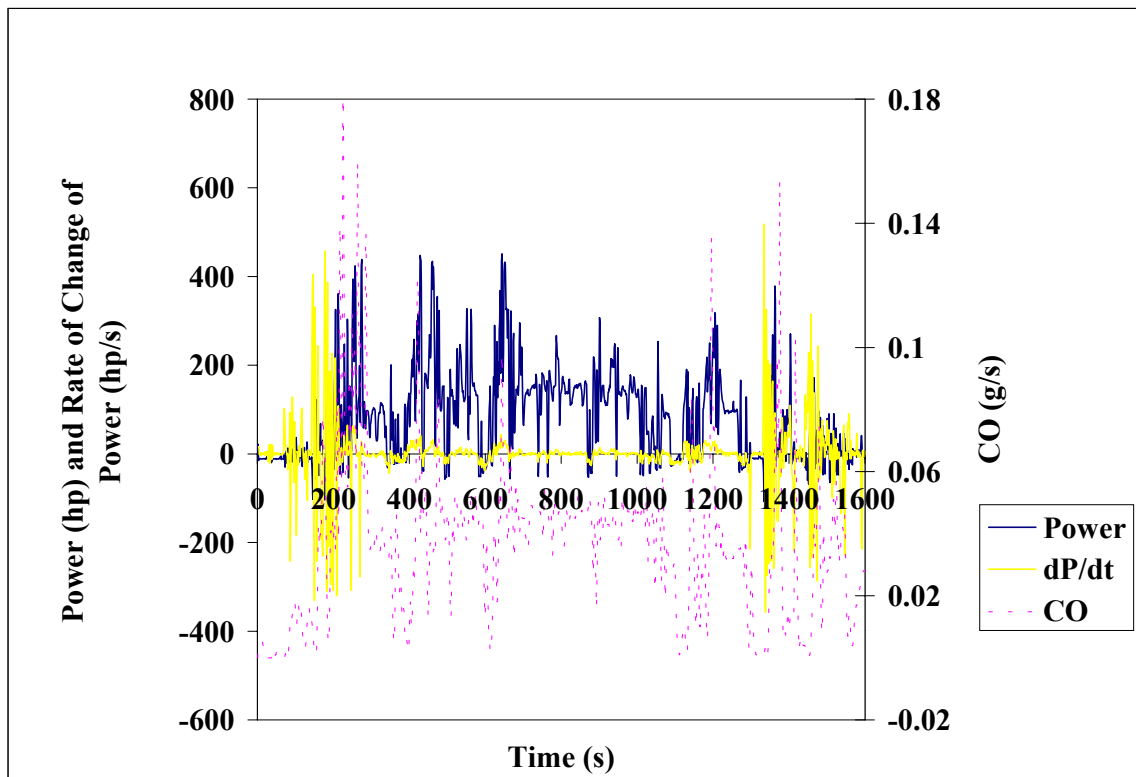


Figure 5.1. Power and rate of change of power and CO emissions versus time for an E-Highway cycle.

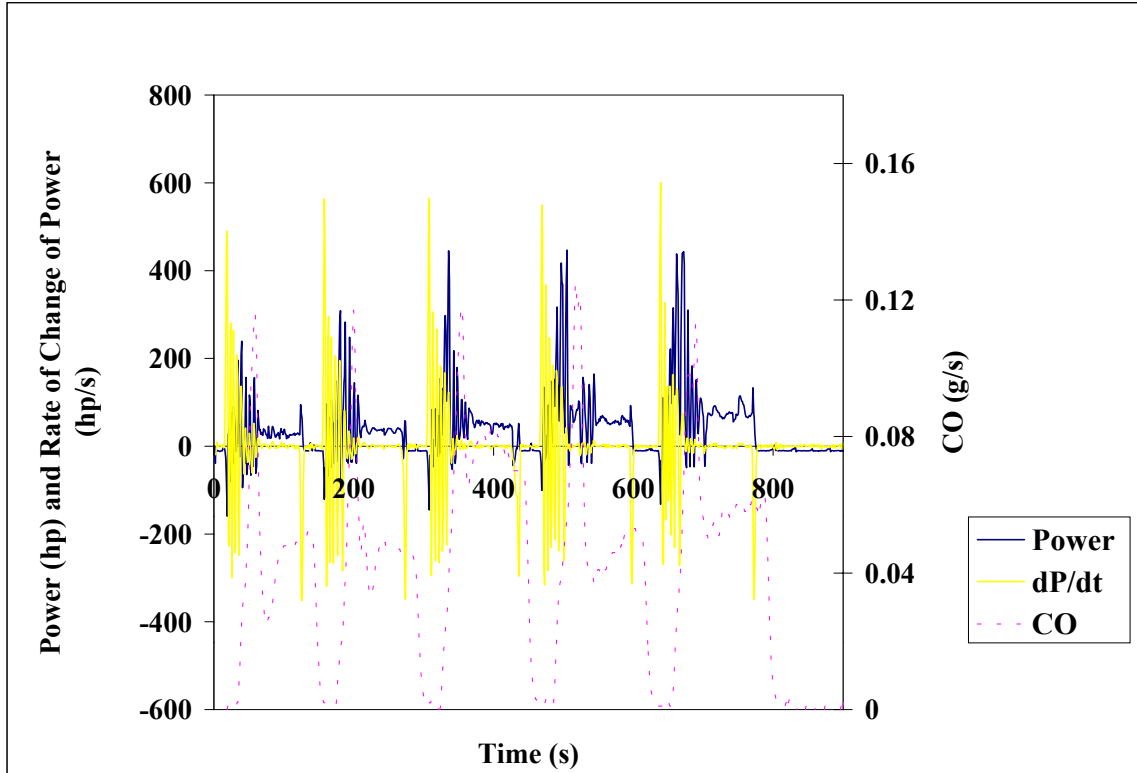


Figure 5.2. Power and rate of change of power and CO emissions versus time for an E-WVU-5 Peak cycle.

Regarding TDNN, there are certain structural configurations of an ANN that can lead to a network capable of accounting for time misalignments between data series. This is another approach that is highly recommended by the author for the future researchers of WVU-ERC, concerning emissions prediction. The main challenge in implementation of these networks might be the right sequences of training patterns.

5.1.3. Moisture Correction

To date, the only accepted standard way of PM measurement is the gravimetric method of filter sampling. This method provides the accumulated PM over the course of the test and the PM formation in terms of instant engine operation and condition remains unknown. There are some identified options of continuous PM measurement such as techniques relating PM to opacity, or different instruments counting the particles in a stream. Among them, the Tapered Element Oscillating Microbalance (TEOM) is believed to be the most promising, however it carries its own problems. The method of choice in

this research was the available TEOM at the West Virginia University Engine and Emissions Research Laboratory (WVU-ERC). The main trouble associated with the instantaneous readings of the TEOM data arises from the fact that the exhaust contains water and volatile hydrocarbons that are captured in the TEOM filter as well as the PM. This redundant residue biases the TEOM measurement. Particularly in an idling period that follows a heavy loading state there is a loss of mass from the filter, which might be addressed as the moisture and volatile HC leaving the filter. Another issue is the effect of accumulated PM on the filter, which increases the moisture retention of the filter. Jarrett et al. [57] suggested a model based on the relation of H₂O and CO₂ to determine the water content of the TEOM filter and deduct it from the original data approaching a more precise PM data. In their model, they neglected the added mass to TEOM filter by the volatile HC and they also assumed the water moisture remains constant and is neglectable compared with the water produced by combustion. Considering the combustion equation, they used the assumption that, for each mole of produced CO₂, a certain number of moles of H₂O are produced.

$$m_{H_2O_a} = 0.3542\dot{m}_{CO_2}\Delta t \quad \text{Equation 5.5}$$

Where, $m_{H_2O_a}$ is the water content of the air in the tunnel. This was their guide to calculate the H₂O content of air from the known mass rate of CO₂ in the tunnel. They correlated this to the rate of mass adsorption in the filter by the following formula [57]:

$$\dot{m}_{H_2O_f} = C_1(C_2\dot{m}_{H_2O_a} - \dot{m}_{H_2O_f}) \quad \text{Equation 5.6}$$

Where, $\dot{m}_{H_2O_f}$ is the rate of moisture adsorption on the TEOM filter, $m_{H_2O_f}$ is the TEOM filter water content, C_1 is a mass transfer constant with units of (1/s) and C_2 is a unitless constant representing the state of balance between moisture concentration in the air and the TEOM filter. They suggested the experimental values of $C_1 = 0.09201$ and $C_2 = 0.00979$, emphasizing that the model does not fully compensate for negative mass rates in the data. In this research the TEOM data was corrected for the moisture using the described model.

5.2. General Regression Neural Networks (GRNN)

Originally introduced in 1990 by Donald Spetch [75], general regression networks are developed from the statistic method named Nadaraya-Watson kernel regression. GRNN is indeed a modification to his novel formerly introduced Parabolistic Neural Networks (PNN). PNN are known to be the best classifiers, but they are not suitable for general mapping.

Like the standard regression statistical techniques, GRNN are used for estimation of continuous values. They are actually feed forward networks that utilize the probability density functions such as radial basis functions and they are proved to perform as universal approximators for the smooth functions. Some overview of “Distribution Function” and “Probability Density Function” seems to be helpful for understanding the basics of GRNN.

5.2.1. Distribution Function

Considering a random variable X , distribution function of X , for X equal or less than the real number x , is given by D as:

$$D(x) = P(X \leq x) = \int_{-\infty}^x P(x') dx' \quad \text{Equation 5.7}$$

When there are two variables involved, it can be redefined as “Joint Distribution Function” [74].

$$D[(x, y) \in C] = P(X \leq x, Y \leq y) = \iint_{(X, Y) \in C} P(X, Y) dXdY \quad \text{Equation 5.8}$$

5.2.2. Probability Density Function

For a continuous function, the probability density function is defined as follows [74].

$$D'(x) = [P(x)]_{-\infty}^x \quad \text{Equation 5.9}$$

For a standard (mean of 0, variable of 1) normal distribution, a general formula for the probability density function is:

$$f(x) = \frac{\exp\left(-\frac{x^2}{2}\right)}{\sqrt{2\pi}} \quad \text{Equation 5.10}$$

5.2.3. GRNN structure

The basic idea is that the regression of a dependant variable against an independent variable X is the estimation of the most probable values of dependant value for each value of the independent variable [75]. Statistically it is well known that the best predicted value for y , is its conditional expectation given x . The expectation is related to joint probability function $D_{XY}(x, y)$ as follows [76, 77].

$$E_{Y|X}(X) = \frac{\int_{-\infty}^{\infty} yD_{XY}(X, y)dy}{\int_{-\infty}^{\infty} D_{XY}(X, y)dy} \quad \text{Equation 5.11}$$

Actually, it is possible to use multivariate ‘‘Parzen’’ estimators to approximate the joint density function. The Rozenblatt-Parzen kernel density estimator, introduced by Rosenblatt in 1956 and Parzen in 1962, is known to be an excellent method for estimating an univariant probability function from a random sample. The Parzen estimator uses a weight function W , called a kernel, and converges to the true density as the sample size increases. Let us assume that the estimation of the density function D to be $g(x)$. For a sample size of n and scaling parameter of σ (defining the width of the bell curve surrounding each sample point), $g(x)$ is presented by the following formula:

$$g(x) = \frac{1}{n\sigma} \sum_{i=1}^n W\left(\frac{x-x_i}{\sigma}\right) \quad \text{Equation 5.12}$$

By using this approximation, and applying this to Equation 5.11, after simplification, the fundamental equation for GRNN is obtained as follows [76]:

$$\hat{y}(X) = \frac{\sum_{i=1}^n y_i \exp(-D(X, X_i))}{\sum_{i=1}^n \exp(-D(X, X_i))} \quad \text{Equation 5.13}$$

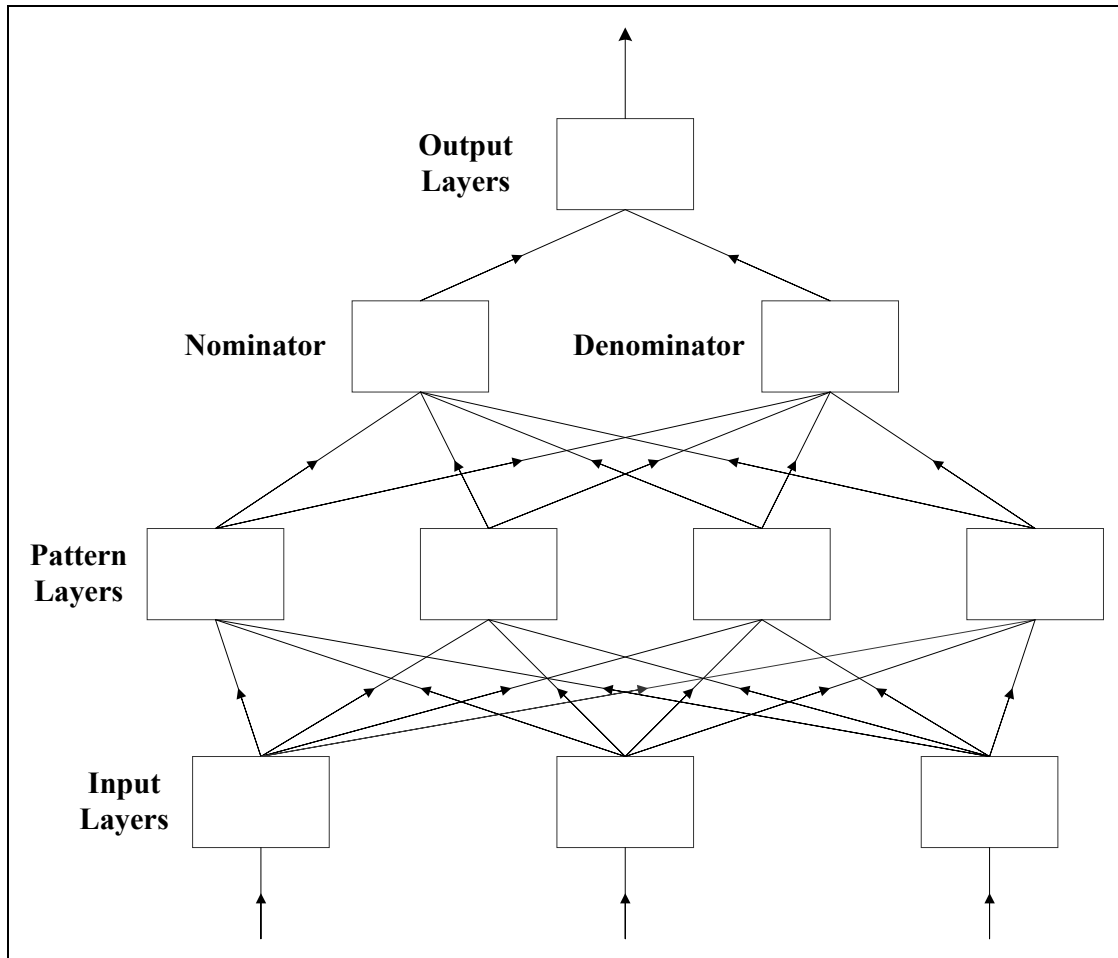


Figure 5.3. Schematic of a GRNN (General Regression Neural Network).

A regular GRNN consists of four layers. The first layer is the input layer that hands out the input pattern to the next layer known as the pattern layer. In a GRNN there must be one hidden neuron for each and every training pattern. This particular structure results in a network with a highly parallel structure that requires no iteration and is very fast. The pattern layer in GRNN is quite different from usual neurons in other networks. Instead of applying the input through a transform function after computing a weighted sum of them, it deducts an input pattern from a corresponding weight. Indeed a GRNN calculates the distance between the inputs and the known concentration, gained through the training set and applies the result to an activation function, which is the “Parzen”. The output of pattern layer is the evaluation of how close is the unknown pattern to that concentration. Finally, each pattern layer neuron is connected to two summation units, nominator and denominator of Equation 5.13. The output layer divides nominator by denominator. The

only unset variable in a GRNN is the smoothing factor for the kernel function. It is actually the coefficient of the interpolation between the training patterns in the training set and essentially can affect the performance of the network. The selection of smoothing factor can be done by different methods such as iterative adjustment or genetic algorithm [76, 77].

5.3. Group Method of Data Handling (GMDH)

First introduced in 1979 by A. G. Ivakhenkov, the Group Method of Data Handling (GMDH), is one of the inductive methods for handling complex object modeling such as pattern recognition. The ANN associated with this method are also known as “polynomial” ANN [78]. These networks are capable of highly nonlinear multivariable mapping. The final output of the network is a polynomial formula, which has the advantage of being simple and easy to understand.

GMDH networks are based on the simple regression analysis. The different layers of the network are generated by terms of a polynomial. For a given series of X_i with total of “ n ” inputs the regression equation for each pair of X_i and X_j is:

$$Y = A + BX_i + CX_j + DX_i^2 + EX_j^2 + FX_iX_j \quad \text{Equation 5.14}$$

So, for n inputs there would be $n(n-1)/2$ new generation of variables. The next layer after the input layer is generated through the selection of best regressions of the input layer. For all these $n(n-1)/2$ of new generation, the polynomials are evaluated and the goal is to keep the best of these variables and screen the ones with poor performance index. The selection criterion can be “mean square error”, “unbiased criterion” or “combined criterion” [10].

The consecutive layer is generated from regressions of the previous layer together with the input layer. In each step only certain, namely “survivors” are chosen for the next layer. In each step the linear combination of all polynomial terms are superposed together and in a similar way to BP algorithm, in a polynomial network, the algorithm adjusts the values of coefficient by judging them against the expected output values. In other words

here the performance index is defined as the mean square error and the whole idea is to decrease this value to a set minimum default. The process will stop when there is no improvement in the network's performance. Figure 5.4 shows a schematic of GMDH network.

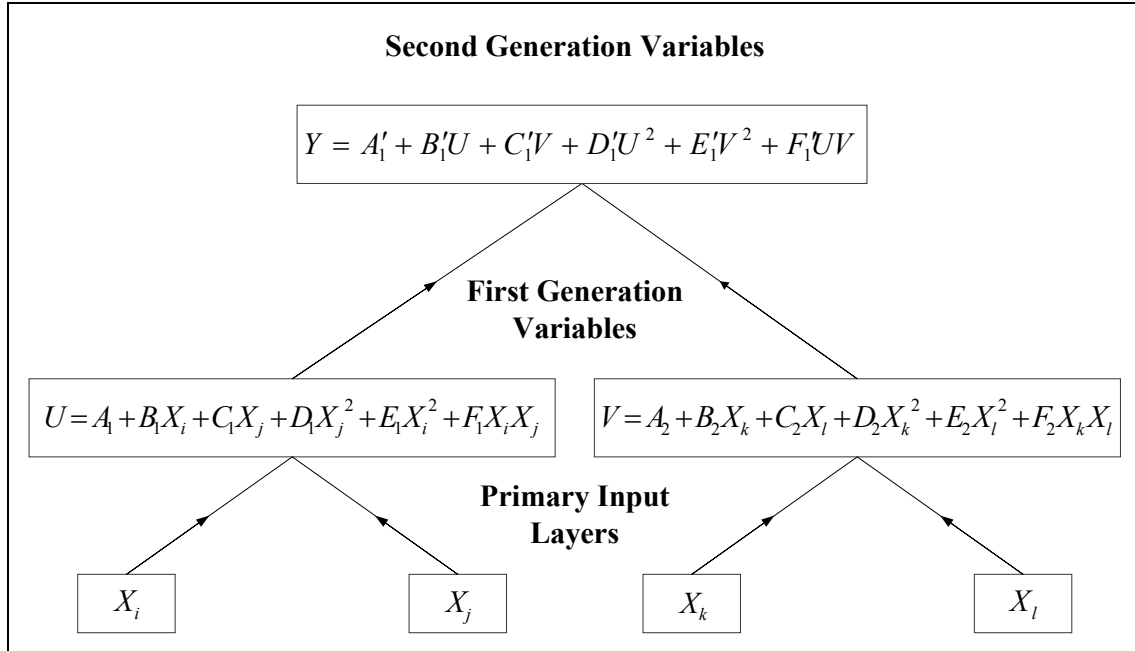


Figure 5.4. Schematic of a GMDH network [10].

A very sensitive issue here is the trade off between being exact and the generalization ability. “Over-complex” or “over-fit” networks result when there are too many terms included in the polynomial and this is equal to poor generalization. Avoiding this situation may be done by introducing as unseen set of data to the network, like a test set [79].

5.4. Least Mean Square (LMS)

During the process of training an ANN, by adjusting the weights and biases, the intention is to improve the “performance” of the network. In many fields of science and engineering the term “performance index” is used as a measure of performance, while the large numbers of this index stand for poor performance and small numbers represent

good performance. The LMS algorithm is a method in which, the performance index is mean square error. In this manner the error function is defined as:

$$E(x) = (T - N)^2 \quad \text{Equation 5.15}$$

Where T stands for the target value, and N stands for the network's output. A well known generalization of LMS is back-propagation.

5.5. BP Design

As described in the second chapter, back-propagation is a common supervised learning algorithm for feed-forward multilayer networks. It works on the basis of a proper error function, to be minimized by changing the weights and biases in the network. There are features specific to this particular network that makes it suitable for emissions prediction task, such as:

- Capability of being used in multi variable systems.
- Capability of being trained (on-line learning and off-line learning)
- Capability of being applied to non-linear systems.

There is also another feature that is important in the emissions prediction task. In real world situations, confusion of emission data set happens as a result of diffusion in the dilution tunnel. For instances of incomplete or noisy data, the BP networks may offer a solution. Although research is being done in order to back-transform diluted emissions and retrieve undiluted emissions [68], this still remains an open problem.

Indicated by Hornik [80], a proper multilayer network can approximate any non-linear function to an arbitrary degree of accuracy. These entire characteristics seem quite persuasive to choose a BP network for emissions prediction task.

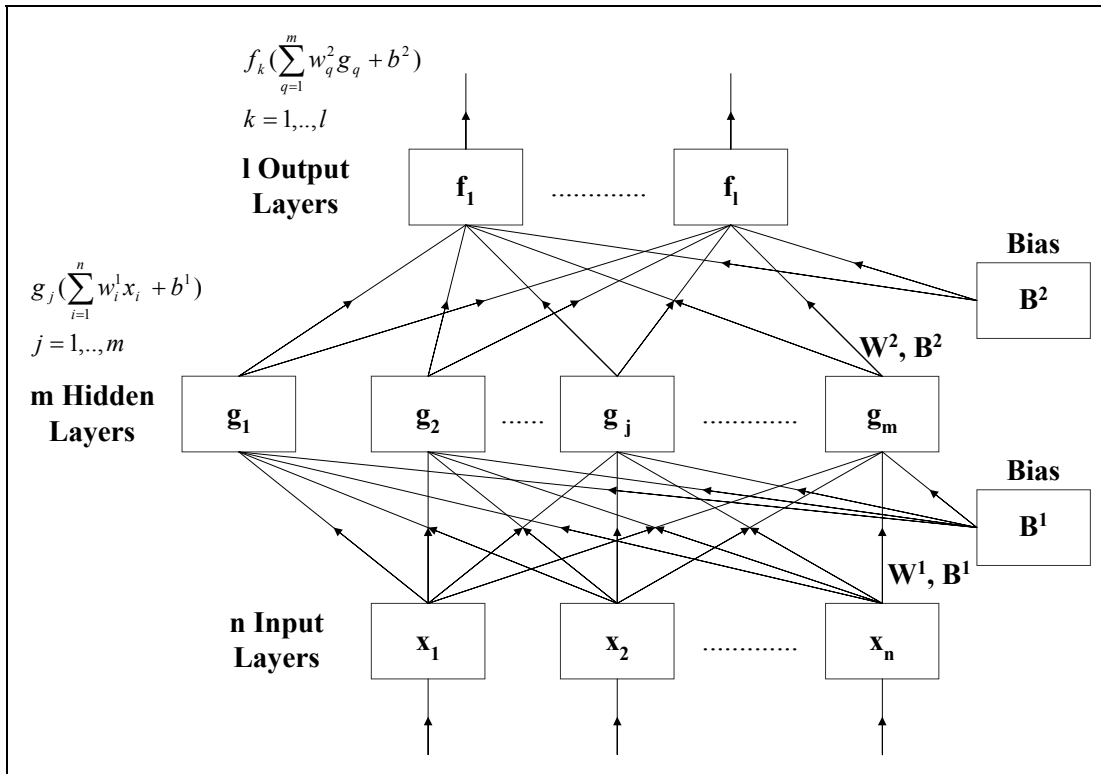


Figure 5.5. Schematic of a BPN (Back-Propagation Neural Network).

The basic algorithm [55] for the training the network in a BP design includes:

- Selection of initial weights.
- Applying the weights on the inputs through activation functions and calculating the corresponding output.
- Judging the actual outputs against the calculated outputs and determining the error.
- Deciding the direction of change in weights to achieve a smaller error.
- Deciding the amount of change in the weights.
- Applying the corrected weights to the network.
- Repeating all the steps until the error falls into the acceptable range.

There are definite imperative specifications to be held by the activation function of a BP network; “being continuous”, “being differentiable” and “being monotonically non-decreasing”. If calculation time is an issue, functions for which their first derivative is easy to calculate are preferred.

Two main classes of functions generally used are: “Sigmoidal or Logistic” basis functions and “Radial” basis functions. The more extended description of different activation functions can be found in the following chapters.

If W is the weight matrix and E is the error function and α is the learning rate, then the W is initialized with a random set of values and then changes in a direction to reduce the error:

$$\Delta W = -\alpha \frac{\partial E}{\partial W} \quad \text{Equation 5.16}$$

5.5.1. Bias and Threshold

Hidden units in a BP design, usually have a bias or threshold unit, which can be indeed treated as a constant weight. Term bias and threshold are pretty much used interchangeably, however bias is mostly indication of a constant value of 1 and threshold represents a constant value of -1 . A bias unit is connected to all hidden and output layers and in general each hidden and output layer has its own bias unit. The hyperplane defined by the weights, lounges in the input space and is restricted to bypass the origin of the space defined by inputs. In many problems it is more gainful to eliminate this constrain of passing the origin by facilitating a constant weight, namely bias [81].

5.5.2. Practical Considerations

A variety of factors are to be considered in a successful design of network for a given problem, such as:

- Regularization: The proper selection of network’s number of layers and size of the hidden layers is a critical feature in the suitable implication of the ANN techniques. Preferably three hidden layers are sufficient to represent any arbitrary function (Kolmogrov theorem). Most problems seem to be learned faster by the network if there are more than one hidden layers. In a particular problem the number of nodes in the input and output layer are defined by the problem, but there is no strict answer to the size of each hidden layer. However, there are offers based on experience stating the size

of hidden layers need to be a moderately small function of input layer size. When the Central Processing Unit (CPU) load is a concern it is desired to have as small number of nodes in the hidden layers as possible. In fact, too few hidden nodes, makes the network incapable of learning and too many nodes will cause poor generalization. May be the best approach is archived by trial and error; adding to the number of hidden layer nodes until the network converges to a solution, and also reducing the number of nodes by inspecting the weights of hidden layer nodes and eliminating the nodes that did not change the initial value of their weights from the start point of learning. The outcome of these two might be a set number of hidden layer nodes as the best performing design. Obviously there is no certain proof of the right choice for the number of hidden layers. Generally the number of hidden layers determines the total number of weights in the network, which can also represent the number of degrees of freedom. As a rule of thumb, it is considered for n training points, choose the size of hidden layer in a way to end up having the total number of weights to be $n/10$ [13].

- The training data: there is no unique answer that suits all the cases. Overall, it is possible to use all the available data, but usually there is a portion left to verify the network's performance. This subset of the data is called the "test" data set and the rest are the "training" data. There are two main features of BP networks that should be very well thought out; BPN's are good in generalization and in contrast they are not good extrapolators. The former means the unrelated data will be disregarded. The latter means that it is essential that the training data cover the whole input range.
- The learning parameters: The speed of learning is actually the rate of convergence between the current solution and the global minimum. "Momentum" helps the network to overcome obstacles (local minima) in the error surface and settle down at or near the global minimum. Considering the weight change value, we may add a fraction of previous change (namely

momentum) that helps keeping the weight changes going to the same direction. When a network reaches to an acceptable point there is no assurance of that it is the global minim, but if the solution is in an acceptable error range, this is not an issue at all. It is important to consider that too few training iterations make the network unable to extract important features from the training set; and too many, causes “over-training” a process in which the network will begin to learn the details of the training set and harm its ability to abstract common features.

- Static and dynamic networks: In order to consider system dynamics it is possible to use the current values of inputs along with their time-delayed value. This draws a line between static neural networks and Time Delay Neural Networks (TDNN). In a typical TDNN each hidden neuron is connected just to a limited array of input neurons. This results in a system in which the adjusted weights are independent of the position of the training pattern, or so called “weight sharing” [13]. In TDNN system there are concerns such as the right estimation of maximum delay time and the effects of this time delayed inputs on the feedback, which may result in higher errors beside extended calculation requirements. This issue has been addressed by Narendra et al. [82].

5.5.3. Error Minimization

In a network with differentiable activation function, the mean square error is also a differentiable function of outputs and the weights. In order to minimize the error function it is readily possible to differentiate the function with respect to the weights and find the values of the weights by means of optimization methods such as gradient descent method. In fact one of the most well known practical solutions to the problem of optimization of a W - dimensional weight function is gradient descent [12]. For an arbitrary network, the error function is a non-linear function of adaptive weights with several minimum points, while each of them satisfy the gradient of error function to be zero.

$$\nabla E = 0 \quad \text{Equation 5.17}$$

Among all these minimum points called “local minima”, there is a “global minimum” that makes the error function the smallest. One of the reasons of failing a network in convergence is being trapped in local minima.

5.5.4. Gradient Descent

Gradient descent or the steepest descent is a method to reduce the error function by taking a direction in which the error reduces the most rapidly. Considering the Taylor expansion:

$$E(x_{n+1}) = E(x_n) + \nabla E(x)^T \Big|_{x=x_n} \Delta x_n \quad \text{Equation 5.18}$$

In order to reduce the error in each iteration, the second term on the right hand side must be negative. If Δx_n is replaced with a “learning rate” α_n , multiplied by a “descent direction” P_n then we should have:

$$\Delta x_n = \alpha_n P_n < 0 \quad \text{Equation 5.19}$$

The steepest descent happens when this term is the most negative, meaning that the direction vector P_n is negative of gradient:

$$p_n = -\nabla E^T \Big|_{x=x_n} \quad \text{Equation 5.20}$$

So finally in order to achieve the steepest descent this is the iteration choice:

$$x_{n+1} = x_n - \alpha_n \nabla E^T \Big|_{x=x_n} \quad \text{Equation 5.21}$$

There should be a very careful choice of learning rate. The large learning rates can make the algorithm unstable. Besides the gradient descent, there are other error optimization methods, such as “Newton’s method” or “conjugate gradient”. However, the gradient descent method seems to be the most advantageous [83].

5.5.5. Momentum

Like its counterpart in physics, in the ANN terminology, the momentum term keeps the weight adjustments to continue changes in the same direction. It is actually a practical modification that by adding to the gradient descent formula induces inertia to the motion through the weight space. It is actually a combination of weight changes in the previous iteration and the present weight updates. Momentum learning can be represented as:

$$W_{n+1} = W_n + (1 - \alpha)\Delta WBP_n + \alpha\Delta W_{n-1} \quad \text{Equation 5.22}$$

Where α is the fraction of previous weight update, ΔWBP_n is the change in W_n due to the BP algorithm and $\Delta W_n = W_n - W_{n-1}$. For stability α should be in the range of (0, 1) [13].

5.6. Inputs

The explicit advantage of ANN is their capability of handling nonlinear models needless of search for the model type. However, it remains a challenge to find reasonable inputs to attain desirable output. The accuracy of the optimized output is reliant on the smart selection of relative inputs. In the emissions prediction task this comes along with a reasonable compensation of the time delays either by considering a time lag inside the network or in the data acquisition prior to network application. It is also imperative to have an intact viewpoint of the emission production procedure and identify the most relative features as the effective input choices. As mentioned in the “Practical Considerations” section there are other issues to be considered as the structure, learning algorithm, number of nodes, number of neurons, activation functions and momentum with an infinite number of possibilities to accomplish an optimized network.

In one of the earlier relatively successful modelings of emissions Bazari [15] used engine governor setting (as an indication of speed), engine break load and ambient conditions. In a 3-D multi-zone combustion model Rakopolos [14] validated the effects of both engine speed and load on NO_x .

Ramamurthy [85] in his master’s thesis correlated the continuous emissions data from trucks and buses with instantaneous axle power. He concluded that NO_x and CO_2 were reliably correlated, but CO was found to be more engine and fuel specific.

Kern [86] in his master thesis tried to categorize the emissions according to the vehicle speed and acceleration. He concluded that it shows acceptable results when compared to measured chassis emissions in units of grams per mile.

Krijnsen et al. [11] took engine speed, rack position (a measure of the engine torque), charge air inlet pressure, and charge air inlet temperature as the input variables for NO_x prediction.

In another paper on optimum NO_x abatement, Krijnsen et al. [87] found the optimum set of inputs for their ANN model to be intake air temperature, intake air pressure, intake air humidity at the current time and engine load not only at current time, but also 2, 5, 8, 11, 14 seconds in the past.

Based on engine speed and torque, Jiang et al. [88] developed a computer model consist of a quasi-steady-state engine combustion model, a dynamic engine model, and a dynamic turbocharger model to predict the transient particulate matter emissions. Over an FTP transient cycle they successfully predicted the brake-specific PM within 4.4% of the experimental data.

In one of the earlier efforts in CO emissions, Rao [89] used fuel parameters to model CO emissions. He found that CO is mainly affected by the Oxygen and Sulfur content of the fuel.

In this research dispersed torque and speed along with their first and second derivatives over 1, 5 and 10 seconds were the total of 14 inputs given to all networks.

5.6.1. Standardization

Standardizing a vector by definition is deducting a measure of location and dividing it by a measure of scale for a given vector (see Equation 5.23, Equation 5.24). A very common method of finding a standard normal of a random variable with mean of 0 and standard deviation of 1 is to subtract a mean from a value and then divide it by the standard deviation.

Although it is not a must, but there are benefits in standardizing input values in most multi-layer networks. Some believe that the best choice is to have the variables rescaled into the range of [-1, 1]. It is well known that the role of each input and its final weighting factor determination is established through its variability among other inputs. For instance, if one input ranges from 0 to 10 and the other ranges from 0 to 1000, the second input will dominate the first one and the network will end up ignoring it. Generally it is preferred to standardize all the inputs to the same range [81].

Rescaling of the data is usually linear, however there are certain cases that non-linear rescaling seems appropriate. In these cases the assumption of non-linear distribution of data is necessary. In the implemented networks in this research, the linear rescaling is done by a simple linear function, and nonlinear rescaling to the range of (0,1) is done by the logistic function and to the range of (-1, 1) by tanh as follows [79].

$$f(X_i) = \frac{1}{1 + \exp\left(\frac{mean - X_i}{sd}\right)} \quad \text{Equation 5.23}$$

$$f(X_i) = \tanh\left(\frac{X_i - mean}{sd}\right) \quad \text{Equation 5.24}$$

5.7. Activation Functions

Activation functions should possess the following characteristics:

- Being continuous.
- Being differentiable and preferably the derivatives being easy to calculate.

A BP network can work with nearly any activation function having these features, however prior insight into each specific problem can greatly benefit a better choice of activation functions. In general, following characteristics are towards functional choice of activation functions:

- Saturation; meaning that the function should have extremum points. This will limit the training time.

- Monotonicity; meaning that the derivative of the function has the same sign through out the scope of argument of function. This will prevent local minima in the error surface [13].

5.7.1. Sigmoid

Sigmoid or S shaped class of functions are smooth, differentiable, nonlinear and saturating and it closely represents the linear function when the weights are small.

5.7.1.1. Binary (Logistic) Sigmoid

This function is expressed with the following equation:

$$f(x) = \frac{1}{1 + \exp(-x)} \quad \text{Equation 5.25}$$

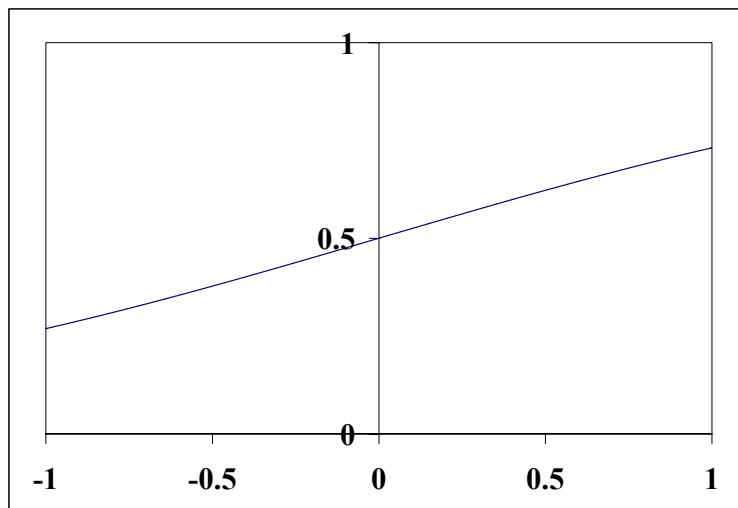


Figure 5.6. Logistic function in the range of -1 and 1 .

5.7.1.2. Bipolar (Symmetric) Sigmoid

This function is expressed with the following equation:

$$f(x) = \frac{1 - \exp(-x)}{1 + \exp(-x)} \quad \text{Equation 5.26}$$

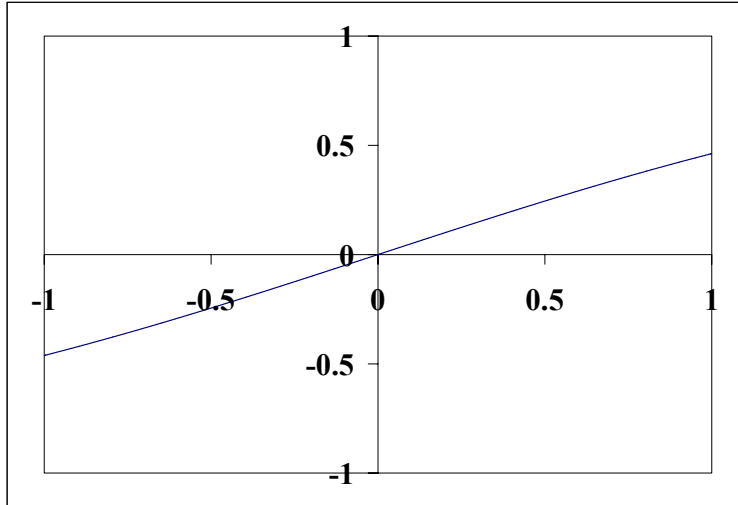


Figure 5.7. Symmetric Logistic function in the range of -1 and 1.

5.7.1.3. Sine

This function is expressed with the following equation:

$$f(x) = \text{Sin}(x) \quad \text{Equation 5.27}$$

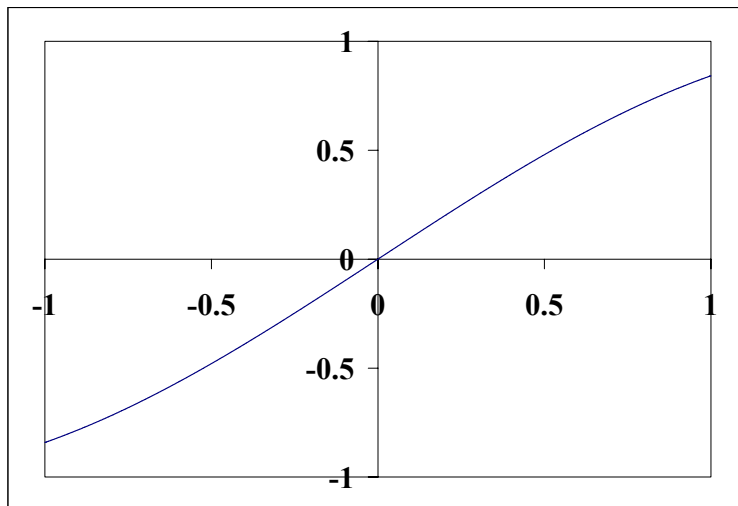


Figure 5.8. Sine function in the range of -1 and 1.

5.7.1.4. tanh

This function is expressed with the following equation:

$$f(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(x)} \quad \text{Equation 5.28}$$

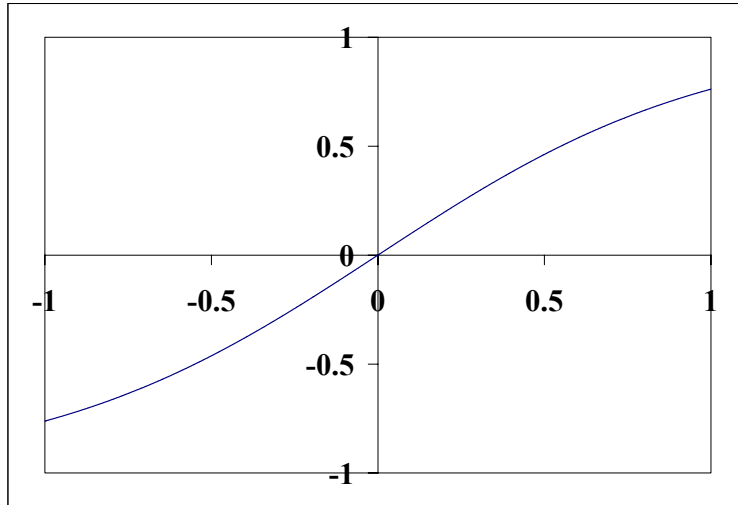


Figure 5.9. Hyperbolic tangent function in the range of -1 and 1 .

5.7.1.5. tanh15

This function is expressed with the following equation:

$$f(x) = \tanh(1.5x) \quad \text{Equation 5.29}$$

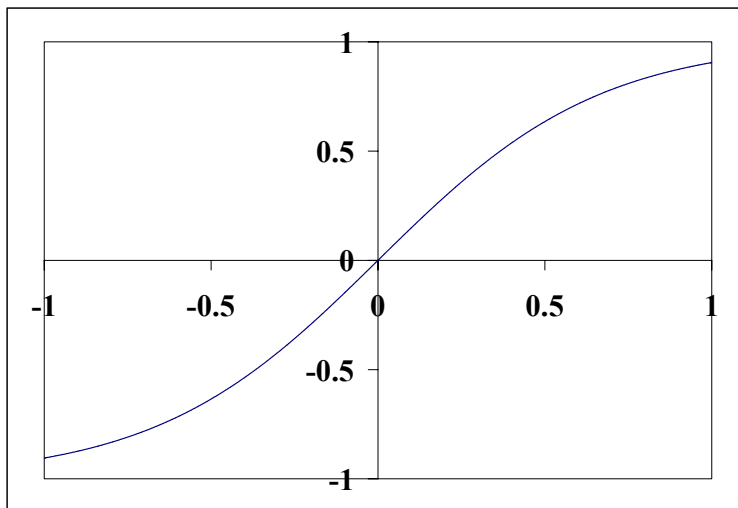


Figure 5.10. Hyperbolic tangent function of $1.5x$ in the range of -1 and 1 .

5.7.2. Radial Basis Functions

The history of radial basis functions dates back to their application in exact interpolation. In ANN applications, generalization is desired and some modifications in the interpolating functions are needed to achieve generalization. Overall the better generalization needs smoother functions. The output of these functions are positive or zero throughout the scope of argument. Radial Basis functions are non-monotonic.

5.7.2.1. Gaussian

This function is expressed with the following equation:

$$f(x) = \exp(-x^2) \quad \text{Equation 5.30}$$

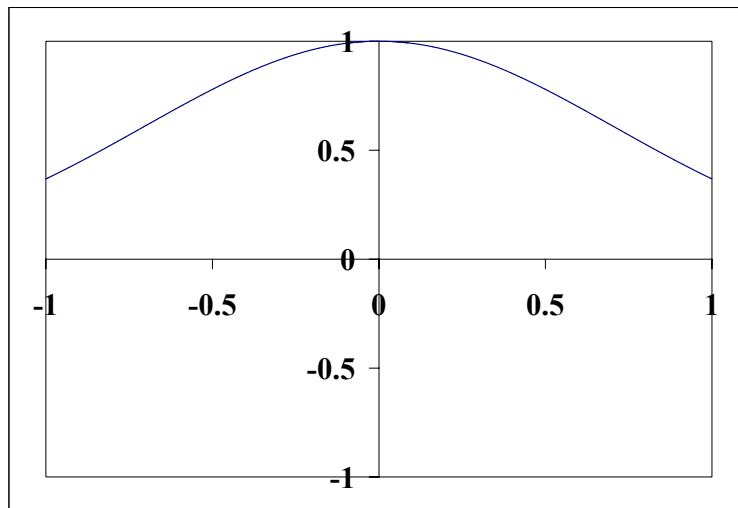


Figure 5.11. Gaussian function in the range of -1 and 1 .

5.7.2.2. Gaussian Complement

This function is expressed with the following equation:

$$f(x) = 1 - \exp(-x^2) \quad \text{Equation 5.31}$$

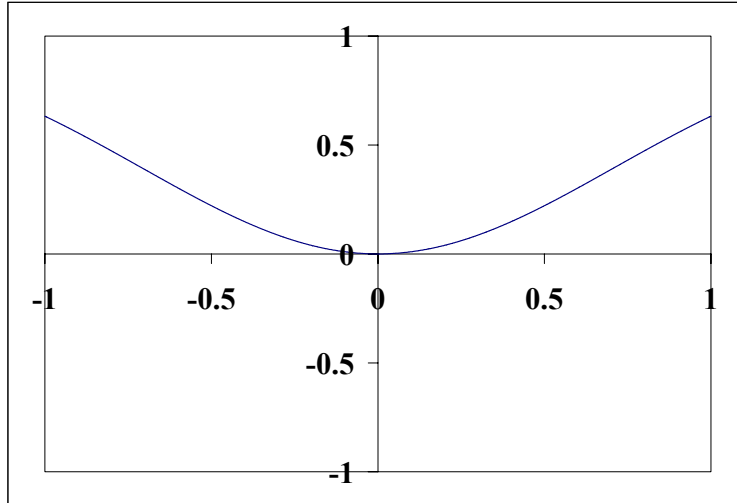


Figure 5.12. Gaussian Complement function in the range of -1 and 1 .

5.8. Applied Architecture

NeuroShell2 [79] was the software used in this study as the ANN software. There were two concerns prior to applying the network; the time delay and the dispersion. Data were time aligned and dispersed before their introduction to the ANN, as explained earlier in this chapter.

“Essentially any continuous functional mapping can be represented to arbitrary accuracy by a network having two layers of weights with sigmoidal hidden units” [12]. This statement was the base for one of the applied architectures.

In 1990 Hartman et al. [84] proved that a linear superposition of Gaussian basis functions, under the condition of the Gaussian width to be adjustable is capable of universal approximation. This was the base of other applied architecture.

According to the descriptions of different network architectures and the performance expected from them, six different architectures were utilized in this work namely 3-Layer, Ward1, Jump Connection, Ward2, GRNN and GMDH. Among these architectures, the first four, are considered as back-propagation networks that are well

known for their aptitude to handle a broad range of problems. The last two are in a way, combination of statistical methods and ANN and were chosen in the view of possibility to be implemented to this sort of problem. Particularly the GMDH was meant to reveal what might be the solid mathematical formula behind the correlations found between the inputs and output.

The number of input layer neurons is the same as the number of inputs, and was the set number of 14 throughout this study. The number of output layer neurons was the set number of 1, as each time only one emission was predicted. The number of hidden layer neurons for a 2-Layer network was computed with the following formula:

number of hidden layer neurons = $1/2$ (Inputs + Outputs) + square root of the number of training patterns.

For more hidden layers the number is divided by the number of hidden layers [79].

The choice of learning criteria is discussed later. A strategy was followed to acquire a suitable set of fixed values for learning rate and momentum. The final choice was applied to all of the networks throughout this work.

In all of the architectures except the GMDH, for all of the input patterns presented to the network 20% of them were randomly selected to be the test patterns and the rest 80% were used as training patterns. The training of the network was saved as the best test set, meaning that the weightings are fixed to the values that result in the lowest error while presented to the test set. Except for GMDH, the training would stop after certain number of events since minimum average error obtained in the test set.

5.8.1. 3-Layer

The 3-Layer network is indeed a customary kind of back-propagation network. In this type there are four layers of neurons, but in the common language of ANN, usually the input layer does not count. In such a network each layer is only connected to the immediate preceding layer. As mentioned earlier in section “5.5.2” there have been mathematical proofs behind the capabilities of a simple 3-Layer network.

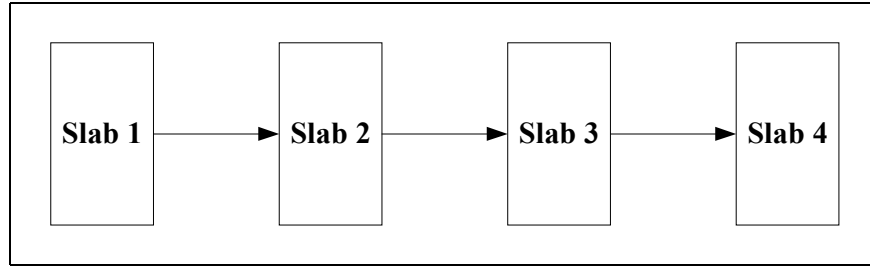


Figure 5.13. Schematic of connections between different layers in a 3-Layer back-propagation network.

5.8.2. Ward1 and Ward2

According to the developers of NeuroShell2 software they, developed these networks as feature detectors. For instance a Gaussian function may detect features in the mid-range data and a Gaussian Complement may detect features from the extreme sides of the data. These networks were used according to their relatively successful history in earlier research [28].

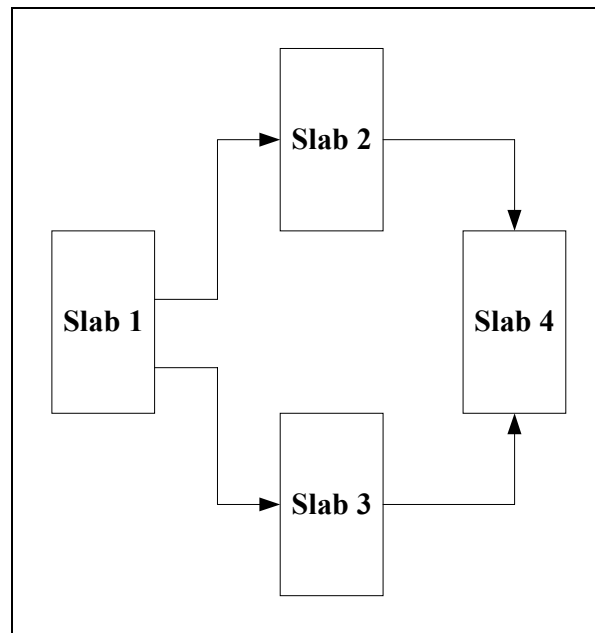


Figure 5.14. Schematic of connections between different layers in a Ward1 back-propagation network.

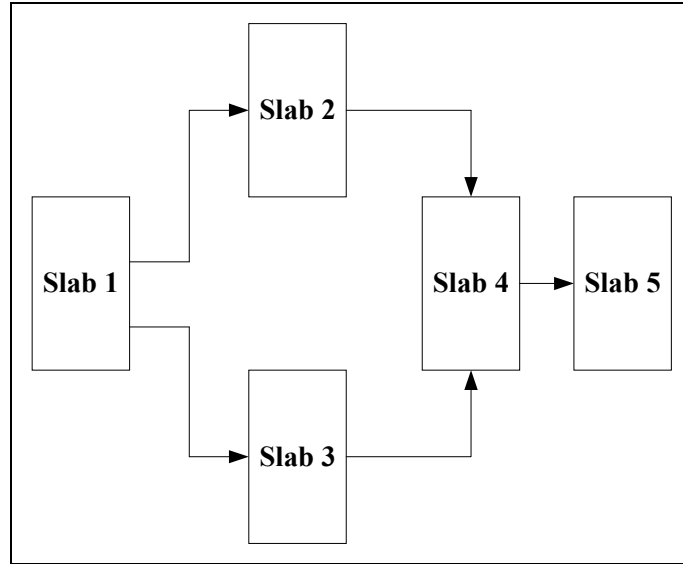


Figure 5.15. Schematic of connections between different layers in a Ward2 back-propagation network.

5.8.3. Jump Connection

In this architecture each layer is connected to all of the previous layers. The network used in this study was a 4-Layer Jump Connection, having total of 5 layers including input layer. Very complex patterns might be fairly detectable by this kind of network [79].

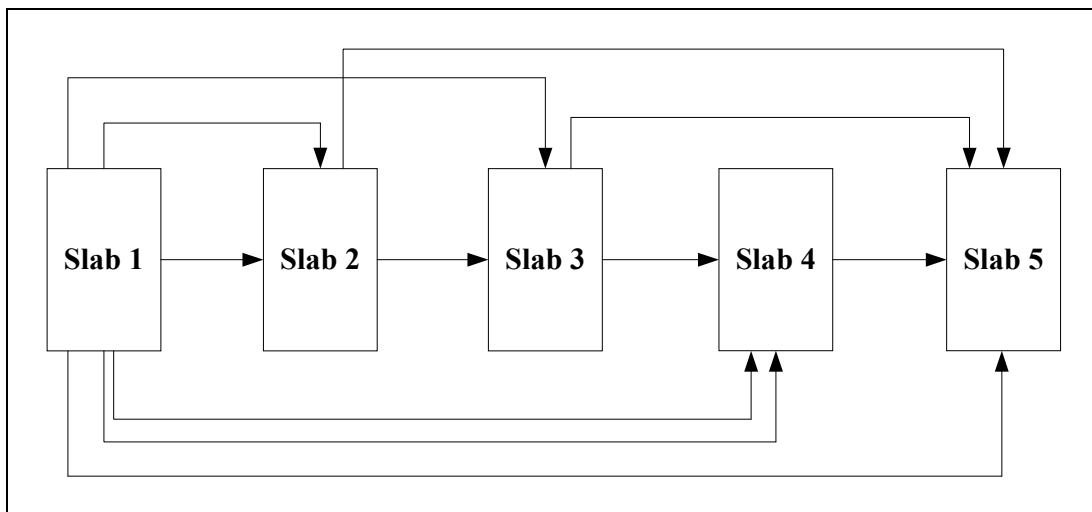


Figure 5.16. Schematic of connections between different layers in a Jump Connection back-propagation network.

5.8.4. GRNN

A GRNN network is a 3-layer network in which there is one hidden layer neuron for each and every training pattern. The training criterion here is just one smoothing factor that defines how closely the network predictions conform its training patterns.

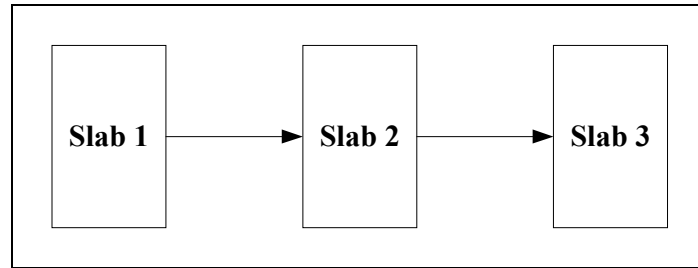


Figure 5.17. Schematic of connections between different layers in a GRNN network.

5.8.5. GMDH

Group Method of Data Handling (GMDH) also called polynomial networks, are not originally represented as ANN. The polynomial terms in the links are the producers of a final polynomial model and a genetic component settles on the number of layers to be built. The layer building procedure persists until the performance index stops increasing (mean square error stops decreasing).

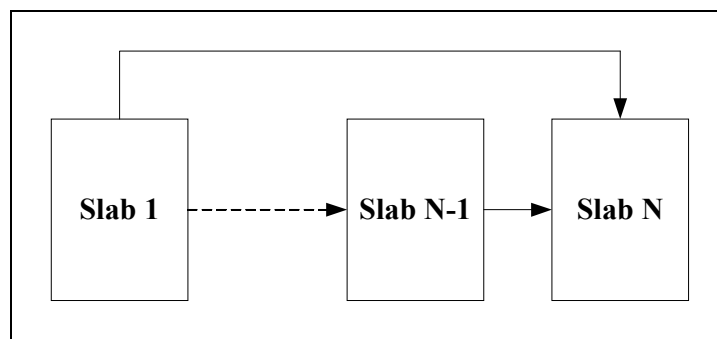


Figure 5.18. Schematic of connections between different layers in a GMDH network.

For all of the above choices of architectures, it has been attempted to include both radial basis and s-shaped functions in distinct and combined forms. The applied activation function assortment for each architecture are shown in Table 5.2 through Table 5.4.

Abbreviation	Functions in 3-Layer	Functions in Ward1
AF1	Lin-G-G-G	Lin-G-G-G
AF2	Lin-Log-Log-Log	Lin-Log-Log-Log
AF3	Lin-SymLog-SymLog-SymLog	Lin-SymLog-SymLog-SymLog
AF4	Lin-G-GC-Log	Lin-G-GC-Log
AF5	Lin-tanh-tanh-tanh	Lin-tanh-tanh-Log
AF6	Lin-GC-GC-GC	Lin-GC-GC-Log
AF7	Lin-Sin-Sin-Sin	Lin-Sin-Sin-Log
AF8	Lin-tanh15-tanh15-tanh15	Lin-tanh15-tanh15-Log

Table 5.2. Activation function assortment for different architectures.

Abbreviation	Functions in Jump Connection	Functions in Ward2
AF1	Lin-G-G-G-Log	Lin-G-tanh-GC-Log
AF2	Log-Log-Log-Log	Lin-tanh15-tan-Symlog-Log
AF3	Lin-SymLog-SymLog-SymLog	Lin-tanh-tanh-tanh-Log
AF4	Lin-G-GC-G-Log	Lin-Log-Log-Log-Log
AF5	Lin-tanh-tanh-tanh-tanh	Linsin-Sin-Sin-Log
AF6	Lin-GC-GC-GC-Log	Lin-G-G-G-Log
AF7	Lin-Sin-Sin-Sin-Sin	Lin-G-GC-G-Log
AF8	Lin-tanh15-tanh15-tanh15-tanh	Lin-tanh-tanh15-tanh-Log

Table 5.3. Activation function assortment for different architectures.

Abbreviation	Functions in GRNN
AF1	Lin
AF2	Log
AF3	tanh

Table 5.4. Adopted activation functions for the GRNN network architecture.

6. RESULTS

To measure the goodness of predictions two standard statistical quantities were considered: “Coefficient of Determination” denoted as r^2 and “Coefficient of Multiple Determination” denoted as R^2 . r^2 is mostly calculated by simply squaring r which is known as “Pearson’s Linear Correlation Coefficient”. To evaluate the goodness of predictions, both R^2 and r^2 were considered. However, it should be pointed out that the experimental error, does not allow the R^2 and r^2 to be very accurate, so their values to several places is accurate to several places only for this particular data set.

6.1. Pearson’s Linear Correlation Coefficient

This coefficient denoted as r represents the strength of the linear relationship between the actual and predicted outputs. Mostly for simplicity it is referred to as “correlation” between two variables. Considering actual values as y and predicted values as z , it can be expressed in terms of “Sum of Squares” of these two as follows:

$$r = \frac{SS_{yz}}{\sqrt{SS_{yy}SS_{zz}}} \quad \text{Equation 6.1}$$

Wherein:

$$SS_{yy} = \sum_{i=1}^n (y_i - \bar{y})^2 \quad \text{Equation 6.2}$$

$$SS_{yz} = \sum_{i=1}^n (y_i - \bar{y})(z_i - \bar{z}) \quad \text{Equation 6.3}$$

Regarding the linear relation it can vary between -1 and 1 , where -1 shows a perfect negative linearity, $+1$ stands for a perfect positive linear correlation, and 0 indicates no linear relationship.

6.2. Coefficient of Determination

As mentioned before, this coefficient denoted as r^2 is calculated by simply squaring “Pearson’s coefficient”.

$$r^2 = \frac{SS_{yz}^2}{SS_{yy}SS_{zz}} \quad \text{Equation 6.4}$$

This number also ranges between 0 and 1 and 1 describes a perfect linear correlation.

6.3. Coefficient of Multiple Determination

Coefficient of Multiple Determination denoted as R^2 is obtained through the following formula:

$$R^2 = 1 - \frac{SS_E}{SS_{yy}} \quad \text{Equation 6.5}$$

Where:

$$SS_E = \sum_{i=1}^n (y_i - z_i)^2 \quad \text{Equation 6.6}$$

R^2 of 1 indicates that all predicted values are equal to actual values, while negative values show a poor conformation of values.

6.4. Predictions Assessment

Complexity and abundance is essential to the nature of data analyzing for systems with plentiful effective parameters such as ANN. Therefore it is also necessary to narrow down the number of useful experiments. Considering all the possible combinations of learning criteria (learning rate, momentum, initial weight), the choices of architecture, and the possible variations of functions in each architecture there would be a huge number of variations for each emission in each transient cycle. Noting that each of learning rate, momentum and initial weight can be assigned to an arbitrary value between 0 and 1, directs the absolute solution of this problem into infinity.

The effect of learning criteria such as learning rate, momentum and initial weight was studied as three combinations of values obtained from previous experiments (other researches conducted by the author) and guidelines by the software developers for stability and convergence of the network solutions. The optimal r^2 value was the criteria

for the best design versus two variables of network architecture and the embedded functions in each architecture (eight combinations).

6.4.1. Different Runs of a test schedules

It might be questionable if the conclusions based on a particular run of a test schedule are applicable to its other runs of the same test schedule. To address this issue, the following graph of two different runs of an FTP cycle is presented for an instance. Data points of each axis are the engine power associated with an individual run of a FTP cycle. The good r^2 value supports the assumption of an adequate conformation between the two runs and this assumption is believed to be valid for all other test schedules.

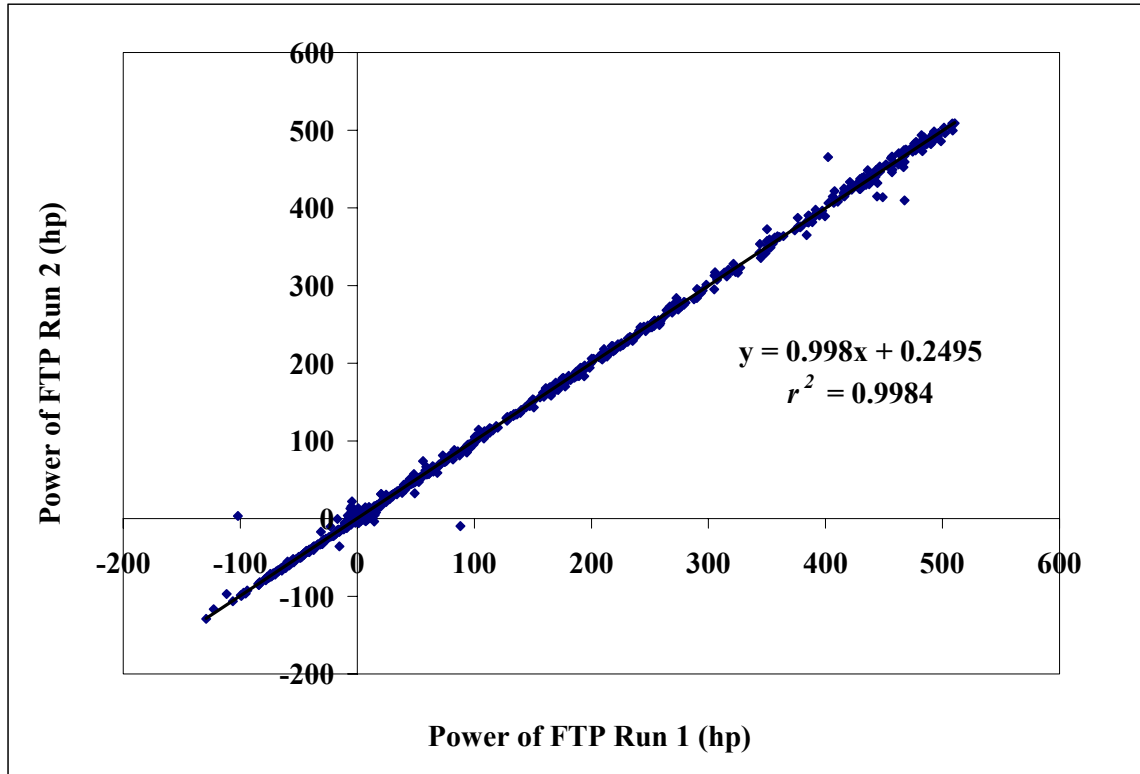


Figure 6.1. Power of FTP run 2 versus FTP run 1 performed on the Series 60 Detroit Diesel engine, model year 2000 dynamometer test.

6.5. NO_x Predictions

6.5.1. Self-Predicting Test Schedules

In this section all of the discussions are based on training an ANN on a particular test schedule and predicting emissions of the same test schedule.

6.5.1.1. Choice of learning Criteria

Transient effects are a serious influence on the emission measurement. To have a measure of how transient a test schedule is, the rate of change of speed was averaged over the duration of each test schedule for all of the five test schedules considered in this work. Evidently the E-CSHVR showed to be the most transient one. The following Table 6.1 shows the average rate of change of speed for all of the test schedules, though just by looking at the speed traces versus time for cycles, one can draw the same conclusion.

Test Schedule	Averaged Rate of Change of Speed (rpm/s)
E-CSHVR	75.71
ETC	37.49
FTP	37.27
E-Highway	24.19
E-WVU- 5 Peak	47.98

Table 6.1. Average rate of change of speed over the total duration for all the test schedules.

Recognizing the E-CSHVR as the most transient test schedule among all of the test schedules examined in this research, and having a huge number of possibilities by varying different features in a certain design, the following strategy was adopted to narrow down the optimization process. Different sets of combinations between training criteria and different assortments of activation functions were used to address the finest combination for a set architecture. These combinations of learning criteria were applied to 32 different networks, resulting in a total of 96 occurrences. Another 9 cases were considered for the GRNN architecture separately, and are discussed later. Table 6.2 shows the details of each combination (abbreviated as Comb). In all of combinations, the

initial weight was assigned the value of 0.3. The choice of three combinations was based on the roles of learning rate and momentum. Small learning rates lead to a steepest descent path that is always perpendicular to the contour lines, and eventually a very stable convergence in a lengthy time. Conversely a large learning rate means bigger steps towards convergence and less training time. In this research the calculation time was considered to be cheap and did not play a key factor in optimization process. However, the training times are presented for informative purposes. By increasing the learning rate the trajectory of steepest descent oscillates and too large learning rate leads to increasing oscillation and instability. For an arbitrary function it is not possible to determine an optimized point for learning rate. Adding a momentum adds stability in cases of large learning rate. If the steepest descent trajectory moves in a continuous track, momentum picks up the pace of convergence. It keeps the trajectory in the same direction [83].

In problems involved with predicting continuous values, smaller values of learning rate and momentum are recommended like 0.1, thus the first combination of learning rate and momentum was chosen to be (0.1, 0.1). For noisy set of data smaller values like 0.05 are recommended, so the second combination was accordingly chosen to be (0.05, 0.05). The third combination was chosen to have larger rate in compare to the first set (0.3, 0.1) to examine if it is feasible to achieve faster convergence while maintaining good predictions.

	Learning Rate	Momentum
Comb1	0.1	0.1
Comb2	0.05	0.05
Comb3	0.3	0.1

Table 6.2. Different combinations of learning criteria used for all of the examined networks.

Recalling Table 5.2 through Table 5.4 from the previous chapter for details of activation functions, Table 6.3 presents the r^2 values obtained from different combinations of learning criteria in a 3-Layer architecture predicting NO_x emissions of an E-CSHVR.

	AF1	AF2	AF3	AF4	AF5	AF6	AF7	AF8	Average
Comb1	0.9922	0.9876	0.9880	0.9930	0.9947	0.9933	0.9940	0.5980	0.9426
Comb2	0.9885	0.9721	0.9791	0.9925	0.9892	0.991	0.9905	0.9971	0.9875
Comb3	0.2204	0.9866	0.9897	0.9941	0.9486	0.9952	0.3471	0.3629	0.7306

Table 6.3. r^2 values obtained from different combinations of learning criteria and activation function assortment in a 3-Layer architecture predicting NO_x emissions of an E-CSHVR .

This comparison was conducted for all of the five different architectures considered in this research. Again recalling Table 5.2 through Table 5.4 from previous chapter, the following three tables present the r^2 values for three of the four remained architectures.

	AF1	AF2	AF3	AF4	AF5	AF6	AF7	AF8	Average
Comb1	0.9798	0.9884	0.9792	0.9890	0.9896	0.9894	0.9916	0.9902	0.9872
Comb2	0.9830	0.9884	0.9790	0.9867	0.9863	0.9884	0.9854	0.9884	0.9857
Comb3	0.074	0.9922	0.9883	0.988	0.9907	0.9915	0.9919	0.9922	0.8761

Table 6.4. r^2 values for different combinations of learning criteria and activations function assortment in a Ward1 architecture predicting NO_x emissions of an E-CSHVR.

	AF1	AF2	AF3	AF4	AF5	AF6	AF7	AF8	Average
Comb1	0.9934	0.9738	0.9753	0.9933	0.9908	0.9900	0.9915	0.9931	0.9877
Comb2	0.9919	0.9737	0.9738	0.9913	0.9895	0.9886	0.9912	0.9928	0.9866
Comb3	0.9926	0.9746	0.9898	0.9944	0.9943	0.9938	0.9929	0.9934	0.9907

Table 6.5. r^2 values for different combinations of learning criteria and activation function assortment in a Jump Connection architecture predicting NO_x emissions of an E-CSHVR.

	AF1	AF2	AF3	AF4	AF5	AF6	AF7	AF8	Average
Comb1	0.9893	0.9903	0.9902	0.9735	0.9859	0.9880	0.9891	0.9906	0.9871
Comb2	0.9889	0.9886	0.9861	0.9731	0.9848	0.9878	0.988	0.9883	0.9857
Comb3	0.991	0.993	0.9915	0.9852	0.9859	0.9891	0.9899	0.9928	0.9898

Table 6.6. r^2 values for different combinations of learning criteria and activations function assortment in a Ward2 architecture predicting NO_x emissions of an E-CSHVR.

The final conclusion would be drawn, after averaging r^2 values for all architectures as follows in Table 6.7. Figure 9.1 in Appendix A is the envision of data in Table 6.7 offering a quick visual assessment

	3-Layer	Ward1	Jump Connection	Ward2	Average	Time Average (s)
Comb1	0.9426	0.9872	0.9877	0.9871	0.9761	45.45
Comb2	0.9875	0.9857	0.9866	0.9857	0.9864	58.03
Comb3	0.7306	0.8761	0.9907	0.9898	0.8968	29.25

Table 6.7. Averaged r^2 values of each architecture predicting NO_x emissions of an E-CSHVR for different combinations of learning criteria. The time is also the averaged over all of the training durations in each architecture.

The training criterion has a different definition in a GRNN, due to its specific network design. Here there were three different approaches to converge on a smoothing factor that works best for the network. Using an iterative method (Comb1) or a genetic adaptive method (Comb3) to find the smoothing factor or simply using a default value as the smoothing factor (Comb2). Results are shown in Table 6.8 and Figure 9.2 in Appendix A is the envision of data in this table.

	AF1	AF2	AF3	Average r^2	Time Average (s)
Comb1 (Iterative Method)	0.9843	0.9683	0.9509	0.9678	8
Comb2 (Set Value)	0.9843	0.9713	0.9509	0.9688	0
Comb3 (Genetic Adaptive)	0.9826	0.9877	0.9805	0.9836	375

Table 6.8. r^2 values for different method of finding smoothing factor in a GRNN architecture predicting NO_x emissions of an E-CSHVR. r^2 and training time are both averaged in the last two columns.

After reviewing the graphs and also considering the average values showed in Table 6.7, it seems reasonable to make a rough conclusion on the best combination of learning criteria (learning rate, momentum). However the mathematics behind the convergence algorithm [87] insists on the unique characterization of each individual problem, but it is necessary to decide a relatively good combination that works fine for most cases.

Overall it is concluded that the second combination (Comb2) works for all the architectures implemented in predicting an E-CSHVR, with the average value of .9864 for r^2 . The average time is also presented in Table 6.7, and shows a trade off between accuracy of prediction and time, however time was not really a concern regarding the order of its largest values. Generally the computational time is considered to be cheap in this research. The results show that the input set seem to be noisy in terms of offering a clear reflection of NO_x emissions. This could be addressed as the influence of dilution tunnel and other interruptions occurred in the measuring system. However the larger learning rate did not provide a considerable advantage in terms of time saving, and reduced the total precision of predictions.

Given that E-CSHVR is the most transient test schedule and the toughest one to predict, all the other test schedules in the rest of this research used Comb2 as the training criteria, except for the GRNN. For the GRNN, according to Table 6.8, the best choice was genetic adaptive as the method of finding smoothing factor. The time is again the price paid for good prediction.

6.5.1.2. Different Architectures for a Set of Learning Criteria

To draw a more general conclusion from all the possible designs, after coming to a reasonable choice for training criteria, architectures and activation functions were varied to approach an optimal design. r^2 and R^2 were both considered as the measure of goodness in predictions. To explain why r^2 is not sufficient as a good measure of successful prediction, $y = 2x$ is an excellent example. In this case r^2 is 1, however x and y values are not identical. This is where R^2 comes in handy. Therefore a successful prediction should include both values of r^2 and R^2 close to 1. Using 0.05, 0.05, 0.3 for learning rate, momentum and initial weights, according to the procedures explained previously, and the genetic algorithm as the method of finding smoothing factor in the case of GRNN network, all the 35 cases were performed on all the transient cycles and following graphs are the results. The following five tables (Table 6.9 through Table 6.13) present results for all of the five transient test schedules.

Presenting a 3-D graph of optimized design in the plane of architectures and activation functions for each transient test schedules seems helpful. However it might be conceptually a bit out of place to make such a comparison, given that the activation function assortments cannot be identical in all architectures. For example speaking of a Gaussian (G) in a 3-Layer network means linear input, two Gaussian hidden layers and a Gaussian output layer. On the other hand in a Ward2 network, there are five layers and G would mean a linear input layer, three hidden layers of Gaussian and a logistic output layer. Interestingly, there can never be a G for a GRNN network, since it just uses S-shaped functions as its regression base. Having a view of all this inconsistencies, and yet to address the issue, it seemed appropriate to present the previously mentioned graph just for 3 architectures that are seemingly alike in terms of activation function assortment, and also give a 2-D graph presenting all the possible approaches and the best of them. Accordingly Figure 9.3 through Figure 9.12 are presented in Appendix A.

Case	Architecture	Abbreviation	Activation Functions	r^2	R^2	t (s)
1	3-Layer	AF1	Lin-G-G-G	0.9885	0.9884	37
2	3-Layer	AF2	Lin-Log-Log-Log	0.9721	0.9720	83
3	3-Layer	AF3	Lin-SymLog-SymLog-SymLog	0.9791	0.9780	42
4	3-Layer	AF4	Lin-G-GC-Log	0.9925	0.9925	56
5	3-Layer	AF5	Lin-tanh-tanh-tanh	0.9892	0.9889	24
6	3-Layer	AF6	Lin-GC-GC-GC	0.9910	0.9910	38
7	3-Layer	AF7	Lin-Sin-Sin-Sin	0.9905	0.9905	19
8	3-Layer	AF8	Lin-tanh15-tanh15-tanh15	0.9971	0.9970	260
9	Ward1	AF1	Lin-G-G-G	0.9830	0.9826	21
10	Ward1	AF2	Lin-Log-Log-Log	0.9884	0.9883	63
11	Ward1	AF3	Lin-SymLog-SymLog-SymLog	0.9790	0.9786	49
12	Ward1	AF4	Lin-G-GC-Log	0.9867	0.9867	40
13	Ward1	AF5	Lin-tanh-tanh-Log	0.9863	0.9861	62
14	Ward1	AF6	Lin-GC-GC-Log	0.9884	0.9884	47
15	Ward1	AF7	Lin-Sin-Sin-Log	0.9854	0.9854	56
16	Ward1	AF8	Lin-tanh15-tanh15-Log	0.9884	0.9883	68
17	Jump Connection	AF1	Lin-G-G-G-Log	0.9919	0.9919	62
18	Jump Connection	AF2	Lin-Log-Log-Log-Log	0.9737	0.9737	40
19	Jump Connection	AF3	Lin-SymLog-SymLog-SymLog-Log	0.9738	0.9738	24
20	Jump Connection	AF4	Lin-G-GC-G-Log	0.9913	0.9911	55
21	Jump Connection	AF5	Lin-tanh-tanh-tanh-tanh	0.9895	0.9895	67
22	Jump Connection	AF6	Lin-GC-GC-GC-Log	0.9886	0.9885	45
23	Jump Connection	AF7	Lin-Sin-Sin-Sin-Sin	0.9912	0.9912	85
24	Jump Connection	AF8	Lin-tanh15-tanh15-tanh15-Log	0.9928	0.9928	54
25	Ward2	AF1	Lin-G-tanh-GC-Log	0.9889	0.9889	57
26	Ward2	AF2	Lin-tanh15-tan-SymLog-Log	0.9886	0.9885	68
27	Ward2	AF3	Lin-tanh-tanh-tanh-Log	0.9861	0.9860	60
28	Ward2	AF4	Lin-Log-Log-Log-Log	0.9731	0.9730	51
29	Ward2	AF5	Lin-Sin-Sin-Sin-Log	0.9848	0.9847	50
30	Ward2	AF6	Lin-G-G-G-Log	0.9878	0.9877	54
31	Ward2	AF7	Lin-G-GC-G-Log	0.9880	0.9880	54
32	Ward2	AF8	Lin-tanh-tanh15-tanh-Log	0.9883	0.9882	66
33	GRNN	AF1	Lin	0.9826	0.9821	404
34	GRNN	AF2	Log	0.9877	0.9876	312
35	GRNN	AF3	tanh	0.9805	0.9804	409

Table 6.9. Summary of results for all cases, using the best combination of training criteria. Trying to predict NO_x emissions of an E-CSHVR, the ANN was trained on the same test schedule.

Reviewing Table 6.9 shows that for the E-CSHVR, case 8 (3-Layer, Lin-tanh15- tanh15- tanh15) is the best choice.

Case	Architecture	Abbreviation	Activation Functions	r^2	R^2	t (s)
1	3-Layer	AF1	Lin-G-G-G	0.9901	0.9899	59
2	3-Layer	AF2	Lin-Log-Log-Log	0.9685	0.9681	99
3	3-Layer	AF3	Lin-SymLog-SymLog-SymLog	0.9690	0.9686	66
4	3-Layer	AF4	Lin-G-GC-Log	0.9930	0.9929	102
5	3-Layer	AF5	Lin-tanh-tanh-tanh	0.9935	0.9934	56
6	3-Layer	AF6	Lin-GC-GC-GC	0.9925	0.9924	50
7	3-Layer	AF7	Lin-Sin-Sin-Sin	0.9791	0.9783	19
8	3-Layer	AF8	Lin-tanh15-tanh15-tanh15	0.9949	0.9947	33
9	Ward1	AF1	Lin-G-G-G	0.9748	0.9743	22
10	Ward1	AF2	Lin-Log-Log-Log	0.9654	0.9652	47
11	Ward1	AF3	Lin-SymLog-SymLog-SymLog	0.9690	0.9679	68
12	Ward1	AF4	Lin-G-GC-Log	0.9778	0.9773	49
13	Ward1	AF5	Lin-tanh-tanh-Log	0.9778	0.9777	57
14	Ward1	AF6	Lin-GC-GC-Log	0.9739	0.9737	27
15	Ward1	AF7	Lin-Sin-Sin-Log	0.9752	0.9751	47
16	Ward1	AF8	Lin-tanh15-tanh15-Log	0.9808	0.9799	52
17	Jump Connection	AF1	Lin-G-G-G-Log	0.9918	0.9918	69
18	Jump Connection	AF2	Lin-Log-Log-Log-Log	0.9709	0.9709	64
19	Jump Connection	AF3	Lin-SymLog-SymLog-SymLog-Log	0.9629	0.9613	40
20	Jump Connection	AF4	Lin-G-GC-G-Log	0.9948	0.9947	142
21	Jump Connection	AF5	Lin-tanh-tanh-tanh-tanh	0.9943	0.9942	89
22	Jump Connection	AF6	Lin-GC-GC-GC-Log	0.9823	0.9821	46
23	Jump Connection	AF7	Lin-Sin-Sin-Sin-Sin	0.9931	0.9929	50
24	Jump Connection	AF8	Lin-tanh15-tanh15-tanh15-Log	0.9780	0.9769	10
25	Ward2	AF1	Lin-G-tanh-GC-Log	0.9826	0.9822	46
26	Ward2	AF2	Lin-tanh15-tan-SymLog-Log	0.9806	0.9803	66
27	Ward2	AF3	Lin-tanh-tanh-tanh-Log	0.9781	0.9780	59
28	Ward2	AF4	Lin-Log-Log-Log-Log	0.9675	0.9674	63
29	Ward2	AF5	Lin-Sin-Sin-Sin-Log	0.9755	0.9752	50
30	Ward2	AF6	Lin-G-G-G-Log	0.9750	0.9748	35
31	Ward2	AF7	Lin-G-GC-G-Log	0.9775	0.9771	44
32	Ward2	AF8	Lin-tanh-tanh15-tanh-Log	0.9844	0.9843	59
33	GRNN	AF1	Lin	0.9846	0.9841	193
34	GRNN	AF2	Log	0.9785	0.9771	782
35	GRNN	AF3	tanh	0.974	0.9734	647

Table 6.10. Summary of results for all cases, using the best combination of training criteria. Trying to predict NO_x emissions of an ETC, the ANN was trained on the same test schedule.

Reviewing Table 6.10 shows that for the ETC, case 8 (3-Layer, Lin-tanh15-tanh15-tanh15) is the best choice.

Case	Architecture	Abbreviation	Activation Functions	r^2	R^2	t (s)
1	3-Layer	AF1	Lin-G-G-G	0.9939	0.9938	21
2	3-Layer	AF2	Lin-Log-Log-Log	0.9852	0.9851	59
3	3-Layer	AF3	Lin-SymLog-SymLog-SymLog	0.9833	0.9831	29
4	3-Layer	AF4	Lin-G-GC-Log	0.9957	0.9957	51
5	3-Layer	AF5	Lin-tanh-tanh-tanh	0.9955	0.9955	36
6	3-Layer	AF6	Lin-GC-GC-GC	0.9962	0.9961	40
7	3-Layer	AF7	Lin-Sin-Sin-Sin	0.9954	0.9954	34
8	3-Layer	AF8	Lin-tanh15-tanh15-tanh15	0.9943	0.9942	12
9	Ward1	AF1	Lin-G-G-G	0.9873	0.9873	13
10	Ward1	AF2	Lin-Log-Log-Log	0.9853	0.9852	54
11	Ward1	AF3	Lin-SymLog-SymLog-SymLog	0.9819	0.9816	26
12	Ward1	AF4	Lin-G-GC-Log	0.9925	0.9924	74
13	Ward1	AF5	Lin-tanh-tanh-Log	0.9877	0.9877	32
14	Ward1	AF6	Lin-GC-GC-Log	0.9955	0.9955	109
15	Ward1	AF7	Lin-Sin-Sin-Log	0.9863	0.9862	22
16	Ward1	AF8	Lin-tanh15-tanh15-Log	0.9942	0.994	73
17	Jump Connection	AF1	Lin-G-G-G-Log	0.9955	0.9954	52
18	Jump Connection	AF2	Lin-Log-Log-Log-Log	0.984	0.9839	34
19	Jump Connection	AF3	Lin-SymLog-SymLog-SymLog-Log	0.9845	0.9844	33
20	Jump Connection	AF4	Lin-G-GC-G-Log	0.9956	0.9955	77
21	Jump Connection	AF5	Lin-tanh-tanh-tanh-tanh	0.9948	0.9946	22
22	Jump Connection	AF6	Lin-GC-GC-GC-Log	0.9966	0.9965	75
23	Jump Connection	AF7	Lin-Sin-Sin-Sin-Sin	0.9948	0.9947	18
24	Jump Connection	AF8	Lin-tanh15-tanh15-tanh15-Log	0.9952	0.9951	28
25	Ward2	AF1	Lin-G-tanh-GC-Log	0.9936	0.9935	54
26	Ward2	AF2	Lin-tanh15-tan-SymLog-Log	0.9886	0.9886	41
27	Ward2	AF3	Lin-tanh-tanh-tanh-Log	0.9876	0.9876	36
28	Ward2	AF4	Lin-Log-Log-Log-Log	0.984	0.9839	44
29	Ward2	AF5	Lin-Sin-Sin-Sin-Log	0.9878	0.9878	51
30	Ward2	AF6	Lin-G-G-G-Log	0.9922	0.9922	46
31	Ward2	AF7	Lin-G-GC-G-Log	0.9917	0.9916	42
32	Ward2	AF8	Lin-tanh-tanh15-tanh-Log	0.9897	0.9897	39
33	GRNN	AF1	Lin	0.9962	0.9962	361
34	GRNN	AF2	Log	0.9963	0.9963	153
35	GRNN	AF3	tanh	0.997	0.997	323

Table 6.11. Summary of results for all cases, using the best combination of training criteria. Trying to predict NO_x emissions of an FTP, the ANN was trained on the same test schedule.

Reviewing Table 6.11 shows that for the FTP, case 35 (GRNN, tanh) is the best choice, however case 8 (3-Layer, Lin-tanh15- tanh15-tanh15) is also in a very good standing.

Case	Architecture	Abbreviation	Activation Functions	r^2	R^2	t (s)
1	3-Layer	AF1	Lin-G-G-G	0.9934	0.9927	54
2	3-Layer	AF2	Lin-Log-Log-Log	0.9708	0.9707	56
3	3-Layer	AF3	Lin-SymLog-SymLog-SymLog	0.9806	0.9794	70
4	3-Layer	AF4	Lin-G-GC-Log	0.9934	0.9934	70
5	3-Layer	AF5	Lin-tanh-tanh-tanh	0.9943	0.9934	38
6	3-Layer	AF6	Lin-GC-GC-GC	0.9933	0.9932	65
7	3-Layer	AF7	Lin-Sin-Sin-Sin	0.9957	0.9953	69
8	3-Layer	AF8	Lin-tanh15-tanh15-tanh15	0.9949	0.9943	28
9	Ward1	AF1	Lin-G-G-G	0.9893	0.9890	55
10	Ward1	AF2	Lin-Log-Log-Log	0.9771	0.9756	59
11	Ward1	AF3	Lin-SymLog-SymLog-SymLog	0.9799	0.9787	50
12	Ward1	AF4	Lin-G-GC-Log	0.9852	0.9851	26
13	Ward1	AF5	Lin-tanh-tanh-Log	0.9857	0.9852	42
14	Ward1	AF6	Lin-GC-GC-Log	0.9851	0.9850	26
15	Ward1	AF7	Lin-Sin-Sin-Log	0.9837	0.9835	41
16	Ward1	AF8	Lin-tanh15-tanh15-Log	0.9864	0.9861	24
17	Jump Connection	AF1	Lin-G-G-G-Log	0.9907	0.9904	36
18	Jump Connection	AF2	Lin-Log-Log-Log-Log	0.9746	0.9745	53
19	Jump Connection	AF3	Lin-SymLog-SymLog-SymLog-Log	0.9759	0.9757	54
20	Jump Connection	AF4	Lin-G-GC-G-Log	0.9877	0.9875	27
21	Jump Connection	AF5	Lin-tanh-tanh-tanh-tanh	0.9925	0.9922	28
22	Jump Connection	AF6	Lin-GC-GC-GC-Log	0.9909	0.9908	39
23	Jump Connection	AF7	Lin-Sin-Sin-Sin-Sin	0.9937	0.9935	31
24	Jump Connection	AF8	Lin-tanh15-tanh15-tanh15-Log	0.9902	0.9900	24
25	Ward2	AF1	Lin-G-tanh-GC-Log	0.9860	0.9857	34
26	Ward2	AF2	Lin-tanh15-tan-SymLog-Log	0.9863	0.9860	33
27	Ward2	AF3	Lin-tanh-tanh-tanh-Log	0.9844	0.9841	32
28	Ward2	AF4	Lin-Log-Log-Log-Log	0.9730	0.9728	56
29	Ward2	AF5	Lin-Sin-Sin-Sin-Log	0.9832	0.9828	32
30	Ward2	AF6	Lin-G-G-G-Log	0.9847	0.9844	36
31	Ward2	AF7	Lin-G-GC-G-Log	0.9850	0.9847	29
32	Ward2	AF8	Lin-tanh-tanh15-tanh-Log	0.9855	0.9851	36
33	GRNN	AF1	Lin	0.9922	0.9921	511
34	GRNN	AF2	Log	0.9916	0.9914	468
35	GRNN	AF3	tanh	0.9823	0.9818	502

Table 6.12. Summary of results for all cases, using the best combination of training criteria. Trying to predict NO_x emissions of an E-Highway, the ANN was trained on the same test schedule.

Reviewing Table 6.12 shows that for the E-Highway, case 7 (3-Layer, Lin-Sin-Sin-Sin) is the best choice. Case 8 (3-Layer, Lin-tanh15- tanh15-tanh15) is also among the best.

Case	Architecture	Abbreviation	Activation Functions	r^2	R^2	t (s)
1	3-Layer	AF1	Lin-G-G-G	0.9880	0.9878	17
2	3-Layer	AF2	Lin-Log-Log-Log	0.9687	0.9685	31
3	3-Layer	AF3	Lin-SymLog-SymLog-SymLog	0.9853	0.9851	35
4	3-Layer	AF4	Lin-G-GC-Log	0.9898	0.9896	22
5	3-Layer	AF5	Lin-tanh-tanh-tanh	0.9894	0.9893	16
6	3-Layer	AF6	Lin-GC-GC-GC	0.9900	0.9899	18
7	3-Layer	AF7	Lin-Sin-Sin-Sin	0.9908	0.9906	15
8	3-Layer	AF8	Lin-tanh15-tanh15-tanh15	0.9913	0.9912	11
9	Ward1	AF1	Lin-G-G-G	0.9889	0.9889	13
10	Ward1	AF2	Lin-Log-Log-Log	0.9707	0.9704	15
11	Ward1	AF3	Lin-SymLog-SymLog-SymLog	0.9858	0.9855	47
12	Ward1	AF4	Lin-G-GC-Log	0.9907	0.9906	29
13	Ward1	AF5	Lin-tanh-tanh-Log	0.9795	0.9791	15
14	Ward1	AF6	Lin-GC-GC-Log	0.9904	0.9904	27
15	Ward1	AF7	Lin-Sin-Sin-Log	0.9736	0.9735	7
16	Ward1	AF8	Lin-tanh15-tanh15-Log	0.9894	0.9893	19
17	Jump Connection	AF1	Lin-G-G-G-Log	0.9898	0.9898	48
18	Jump Connection	AF2	Lin-Log-Log-Log-Log	0.9704	0.9703	13
19	Jump Connection	AF3	Lin-SymLog-SymLog-SymLog-Log	0.9619	0.9581	18
20	Jump Connection	AF4	Lin-G-GC-G-Log	0.9897	0.9896	47
21	Jump Connection	AF5	Lin-tanh-tanh-tanh-tanh	0.9897	0.9897	18
22	Jump Connection	AF6	Lin-GC-GC-GC-Log	0.9908	0.9908	48
23	Jump Connection	AF7	Lin-Sin-Sin-Sin-Sin	0.9911	0.9911	10
24	Jump Connection	AF8	Lin-tanh15-tanh15-tanh15-Log	0.9880	0.9879	30
25	Ward2	AF1	Lin-G-tanh-GC-Log	0.9887	0.9887	32
26	Ward2	AF2	Lin-tanh15-tan-SymLog-Log	0.9798	0.9796	20
27	Ward2	AF3	Lin-tanh-tanh-tanh-Log	0.9694	0.9693	7
28	Ward2	AF4	Lin-Log-Log-Log-Log	0.9686	0.9685	15
29	Ward2	AF5	Lin-Sin-Sin-Sin-Log	0.9702	0.9702	7
30	Ward2	AF6	Lin-G-G-G-Log	0.9893	0.9892	30
31	Ward2	AF7	Lin-G-GC-G-Log	0.9904	0.9904	38
32	Ward2	AF8	Lin-tanh-tanh15-tanh-Log	0.9889	0.9888	34
33	GRNN	AF1	Lin	0.9755	0.9755	103
34	GRNN	AF2	Log	0.9688	0.9681	108
35	GRNN	AF3	tanh	0.9569	0.9555	168

Table 6.13. Summary of results for all cases, using the best combination of training criteria. Trying to predict NO_x emissions of an E-WVU-5 Peak, the ANN was trained on the same test schedule.

Reviewing Table 6.13 shows that for the E-WVU-5 Peak, case 8 (3-Layer, Lin-tanh15-tanh15-tanh15) is the best choice.

6.5.2. Predicting Other Test Schedules

In this section, all of the discussions are based on training an ANN on a particular test schedule or a combination of test schedules and predicting emissions of a different test schedule. The following five tables (Table 6.14 through Table 6.18) have used the best self-predictor network of each test schedule (results of previous section 6.5.1.2) to predict NO_x emissions of other test schedules. Given that the best self-predictor for FTP was a GRNN network, obviously due to architecture specifications, it was not feasible to use that as a predictor for other test schedules. Since from other test schedules it was practically concluded that the 3-Layer network with tanh15, gives fine predictions and furthermore for an FTP it gives an r^2 value of 0.994. Table 6.20 is presented for this architecture.

Best E-CSHVR (3-Layer, tanh15) Predicts	r^2	R^2
ETC	0.9803	0.0013
FTP	0.9572	0.9562
E-Highway	0.9773	0.962
E-WVU 5-Peak	0.9804	0.9779

Table 6.14. Results obtained by predicting NO_x emissions of other transient test schedules using a network trained on an E-CSHVR. The architecture is the best E-CSHVR predictor (3-Layer, Lin-tanh15-tanh15-tanh15).

Best ETC (3-Layer, tanh15) Predicts	r^2	R^2
E-CSHVR	0.9796	0.7493
FTP	0.9805	0.8402
E-Highway	0.9819	0.7493
E-WVU 5-Peak	0.9849	0.6741

Table 6.15. Results obtained by predicting NO_x emissions of other transient test schedules using a network trained on an ETC. The architecture is the best ETC self-predictor (3-Layer, Lin-tanh15-tanh15-tanh15).

FTP (3-Layer, tanh15) Predicts	r^2	R^2
E-CSHVR	0.9656	0.9632
ETC	0.9618	-0.2610
E-Highway	0.9750	0.9603
E-WVU 5-Peak	0.9584	0.9475

Table 6.16. Results obtained by predicting NO_x emissions of other transient test schedules using a network trained on a FTP. The architecture is one of the best FTP self-predictors (3-Layer, Lin-tanh15-tanh15-tanh15).

Best E-Highway (3-Layer, Sin) Predicts	r^2	R^2
E-CSHVR	0.9887	0.9860
ETC	0.9887	0.9860
FTP	0.9851	0.9667
E-WVU 5-Peak	0.9844	0.9761

Table 6.17. Results obtained by predicting NO_x emissions of other transient test schedules using a network trained on an E-Highway. The architecture is the best E-Highway self-predictor (3- Layer, Lin-Sin-Sin-Sin).

Best E-WVU- 5 Peak (3-Layer, tan15) Predicts	r^2	R^2
E-CSHVR	0.9600	0.9480
ETC	0.9466	-0.4353
FTP	0.9511	0.9472
E-Highway	0.9913	0.9912

Table 6.18. Results obtained by predicting NO_x emissions of other transient test schedules using a network trained on an E-WVU-5 Peak. The architecture is the best E-WVU-5 Peak self-predictor (3-Layer, Lin-tanh15-tanh15-tanh15).

Reviewing the five previous tables, remarks a simple 3-Layer network with tanh15 trained on an E-Highway test schedule as the best generalized network capable of predicting all other test schedules fairly well with both high r^2 and R^2 values.

To finalize the optimization process, it seemed reasonable to examine the ability of a network trained on patterns obtained from all the test schedules using the winning architecture of 3-Layer. The following table presents values of r^2 in such a case, where five of the activation function assortments with best results in 3-Layer architecture were used.

Combined Test Schedules Self-Prediction	r^2	R^2
3-Layer, Lin-G-GC-Log	0.9528	0.9527
3-Layer, Lin-tanh-tanh-tanh	0.9276	0.9226
3-Layer, Lin-GC-GC-GC	0.9599	0.9597
3-Layer, Lin-Sin-Sin-Sin	0.9525	0.9524
3-Layer, Lin-tanh15-tanh15-tanh15	0.9133	0.8922

Table 6.19. Results of NO_x predictions using a network trained on all the transient test schedules combined. The choice of architecture and activation function is based on their performance in self-prediction of each test schedule individually.

Table 6.19 remarks that a 3-Layer with Gaussian Complement activation function in hidden and output layers gives higher values of prediction accuracy, therefore a network with this architecture, trained on combination of test schedules was used to predict each of them individually and the following tables gives the results.

Combined Test Schedules Predicts	3-Layer, Lin-GC-GC-GC		
	r^2	R^2	Accumulative Prediction Error Percentage
E-CSHVR	0.9790	0.9728	3.8
ETC	0.9780	0.7288	27.3
FTP	0.9862	0.9856	1.8
E-Highway	0.9777	0.9666	7.7
E-WVU- 5 Peak	0.9784	0.9411	12.5
Average	0.9799	0.9190	10.62

Table 6.20. Results of NO_x predictions obtained using a 3-Layer network with activation function of Gaussian Complement trained on all the transient test schedules combined and predicting each test schedules individually.

The following graph presents the second by second predictions of this 3-Layer network with Gaussian Complement activation function for NO_x emissions of a FTP test schedule. Graphs for other four test schedules are available in Appendix A (Figure 9.13 through Figure 9.16).

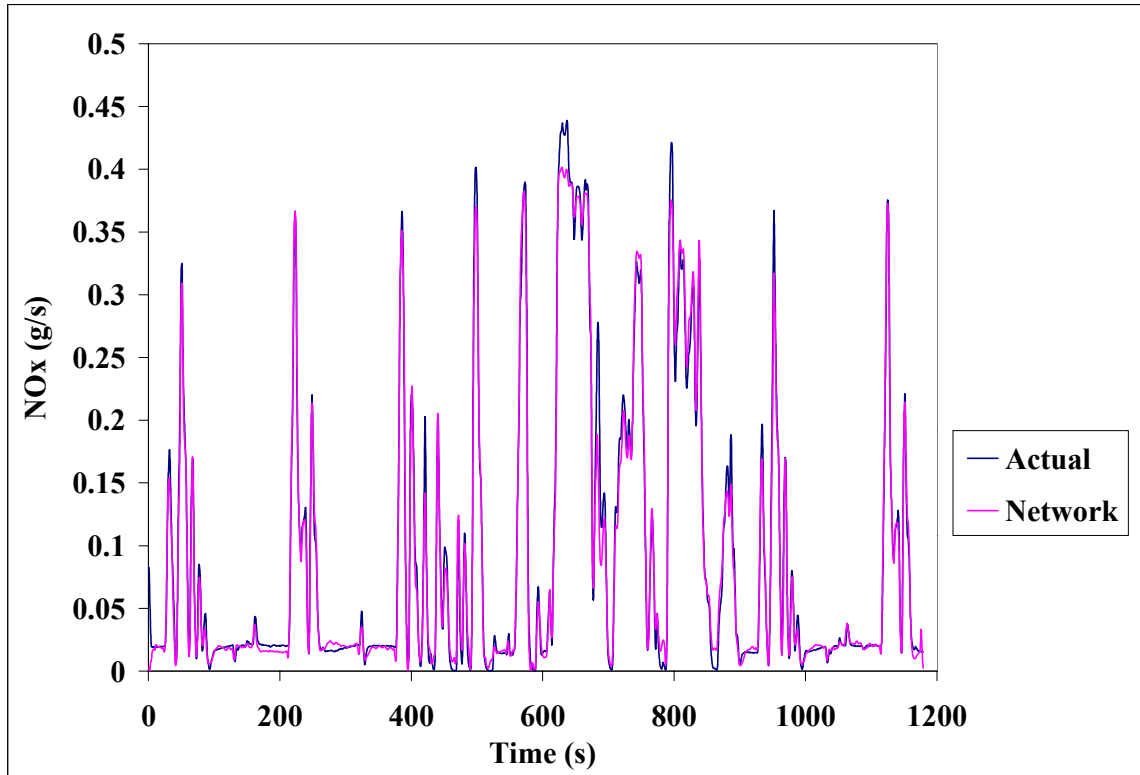


Figure 6.2. Second by second prediction of NO_x emissions in a FTP test schedule by a 3-Layer network of Gaussian Complement activation function. The ANN was trained on all the transient test schedules combined.

6.5.2.1. GMDH

In the ultimate approach in generalizing the correlation between engine variables and emissions, the Group Method of Data Handling (GMDH) networks were employed. Due to specifications of these networks, they seem to perform the best when they give a view of most and least significant variables in a polynomial consisting of all input variables expressing the output variable. A GMDH network was trained on all the transient test schedules combined, and the network produced a final formula with medium non-linearity. This is a lengthy formula and is presented in Appendix B. The following table shows the results of predictions made by this network. Interestingly values are very close to those in Table 6.20, however, these are a bit lower, emphasizing that expressing the complicated relations between variables in terms of an exact mathematical model cannot improve on the capabilities of black-box modeling as in a 3-Layer network.

Combined Test Schedules Predicts	GMDH	
	r^2	R^2
E-CSHVR	0.9768	0.9690
ETC	0.9718	0.6425
FTP	0.9850	0.9808
E-Highway	0.9759	0.9633
E-WVU- 5 Peak	0.9655	0.9127
Average	0.9750	0.8937

Table 6.21. Results of NO_x predictions obtained by using a GMDH network trained on all the transient test schedules combined and predicting each test schedules individually.

6.5.3. Deduction

Considering Table 6.20 and the second by second predictions, it is concluded that a 3 - Layer network of Gaussian Complement activation function seems promising in predicting NO_x emissions in most driving conditions, being trained on an available data base consisted of variety of test schedules. In all cases the network predicts higher accumulative values. Interestingly the best prediction is for FTP, which is the synthetic cycle emerged from a matrix of different operation modes, and the worst prediction with is for ETC, which is the second most transient cycle. The transient effects, that influence the diffusion of each second's emission in a dilution tunnel resulting in a more complicated data pattern would severely harm the network and reduce the accuracy of predictions. These effects are addressed in more details in reference [68].

Overall the author highly recommends simple 3-Layer architecture with just one activation function either S shaped or radial basis in all layers for prediction of NO_x in transient operations. Yet, the problem of complications associated with interactions of consequent emissions in highly transient operations remains open and improvements in predictions would be achieved once it is solved.

6.6. PM Predictions

Using the same choice of training criteria, as in NO_x predictions (learning rate of 0.05, momentum of 0.05 and initial weights of 0.3), the PM predictions were done for all of the transient test schedules. Experienced in NO_x predictions, the author observed that although GRNN are promising in cases of self-predictions in all test schedules, due to their unique way of finding smoothing factor they are not desirable where a generalization based on all deriving conditions is preferred. Also Ward1 and Ward2 were severely outperformed by 3-Layer and Jump Connection architectures in accuracy. For these reasons the author decided to carry on the optimization process by the best two architectures which were 3-Layer and Jump Connection.

It is to be pointed out that the instantaneous PM measurements is done by Tapered Element Oscillating Microbalance (TEOM) and as mentioned previously, there are other sources that interfere with its measurements. In this research the ANN dealt directly with the TEOM values and at the very end the outputs were corrected according to the moisture correction model for their moisture content. At this stage they were called PM, however it is still questionable if they truly reflect the real values of PM.

6.6.1. Self-Predicting Test Schedules

In this section all of the discussions are based on training an ANN on a particular test schedule and predicting emissions of the same test schedule. The following five tables (Table 6.22 through Table 6.26) show the results for the five studies test schedules. In addition, for a better visual impression for each test schedule there comes in the Appendix A, a 3-D graph that presents the variation of r^2 in the plane of activation function and architecture for the two architectures of choice, and a 2-D graph of both r^2 and R^2 for all cases as another staging of the best architecture and activation function assortment (Figure 9.17 through Figure 9.26).

Case	Architecture	Abbreviation	Activation Functions	r^2	R^2	t (s)
1	3-Layer	AF1	Lin-G-G-G	0.9167	0.9149	6
2	3-Layer	AF2	Lin-Log-Log-Log	0.9018	0.9013	27
3	3-Layer	AF3	Lin-SymLog-SymLog-SymLog	0.9084	0.9064	17
4	3-Layer	AF4	Lin-G-GC-Log	0.9352	0.9351	33
5	3-Layer	AF5	Lin-tanh-tanh-tanh	0.9482	0.9467	53
6	3-Layer	AF6	Lin-GC-GC-GC	0.9404	0.9401	30
7	3-Layer	AF7	Lin-Sin-Sin-Sin	0.9212	0.9196	11
8	3-Layer	AF8	Lin-tanh15-tanh15-tanh15	0.9065	0.9048	8
9	Jump Connection	AF1	Lin-G-G-G-Log	0.9412	0.9393	10
10	Jump Connection	AF2	Lin-Log-Log-Log-Log	0.8925	0.8918	6
11	Jump Connection	AF3	Lin-SymLog-SymLog-SymLog	0.8966	0.8959	5
12	Jump Connection	AF4	Lin-G-GC-G-Log	0.9434	0.9412	36
13	Jump Connection	AF5	Lin-tanh-tanh-tanh-tanh	0.9114	0.9112	7
14	Jump Connection	AF6	Lin-GC-GC-GC-Log	0.9470	0.9468	55
15	Jump Connection	AF7	Lin-Sin-Sin-Sin-Sin	0.9057	0.9039	5
16	Jump Connection	AF8	Lin-tanh15-tanh15-tanh15-tanh	0.7258	0.6192	4

Table 6.22. Summary of results for TEOM predictions of an E-CSHVR. The ANN was trained on the same test schedule.

Case	Architecture	Abbreviation	Activation Functions	r^2	R^2	t (s)
1	3-Layer	AF1	Lin-G-G-G	0.1329	0.1279	14
2	3-Layer	AF2	Lin-Log-Log-Log	0.0734	0.0724	53
3	3-Layer	AF3	Lin-SymLog-SymLog-SymLog	0.0742	0.0737	8
4	3-Layer	AF4	Lin-G-GC-Log	0.1295	0.1282	15
5	3-Layer	AF5	Lin-tanh-tanh-tanh	0.0852	0.0804	4
6	3-Layer	AF6	Lin-GC-GC-GC	0.1477	0.1455	7
7	3-Layer	AF7	Lin-Sin-Sin-Sin	0.0904	0.0880	4
8	3-Layer	AF8	Lin-tanh15-tanh15-tanh15	0.0696	0.0670	5
9	Jump Connection	AF1	Lin-G-G-G-Log	0.1107	0.1076	5
10	Jump Connection	AF2	Lin-Log-Log-Log-Log	0.0723	0.0688	10
11	Jump Connection	AF3	Lin-SymLog-SymLog-SymLog	0.0710	0.0668	5
12	Jump Connection	AF4	Lin-G-GC-G-Log	0.1094	0.1049	5
13	Jump Connection	AF5	Lin-tanh-tanh-tanh-tanh	0.0930	0.0867	7
14	Jump Connection	AF6	Lin-GC-GC-GC-Log	0.1145	0.1142	6
15	Jump Connection	AF7	Lin-Sin-Sin-Sin-Sin	0.1020	0.0993	9
16	Jump Connection	AF8	Lin-tanh15-tanh15-tanh15-tanh	0.0191	-0.5161	4

Table 6.23. Summary of results for TEOM predictions of an ETC. The ANN was trained on the same test schedule.

Case	Architecture	Abbreviation	Activation Functions	r^2	R^2	t (s)
1	3-Layer	AF1	Lin-G-G-G	0.7018	0.7010	67
2	3-Layer	AF2	Lin-Log-Log-Log	0.1334	0.1323	21
3	3-Layer	AF3	Lin-SymLog-SymLog-SymLog	0.6358	0.6348	80
4	3-Layer	AF4	Lin-G-GC-Log	0.7031	0.7027	63
5	3-Layer	AF5	Lin-tanh-tanh-tanh	0.6073	0.6032	33
6	3-Layer	AF6	Lin-GC-GC-GC	0.7093	0.7081	44
7	3-Layer	AF7	Lin-Sin-Sin-Sin	0.6505	0.6486	43
8	3-Layer	AF8	Lin-tanh15-tanh15-tanh15	0.3712	0.2049	12
9	Jump Connection	AF1	Lin-G-G-G-Log	0.6497	0.6467	29
10	Jump Connection	AF2	Lin-Log-Log-Log-Log	0.1373	0.1277	15
11	Jump Connection	AF3	Lin-SymLog-SymLog-SymLog	0.5619	0.5467	78
12	Jump Connection	AF4	Lin-G-GC-G-Log	0.7149	0.7118	64
13	Jump Connection	AF5	Lin-tanh-tanh-tanh-tanh	0.0630	-0.1779	9
14	Jump Connection	AF6	Lin-GC-GC-GC-Log	0.7042	0.7004	43
15	Jump Connection	AF7	Lin-Sin-Sin-Sin-Sin	0.0606	-0.4437	8
16	Jump Connection	AF8	Lin-tanh15-tanh15-tanh15-tanh	0.0901	-0.6019	9

Table 6.24. Summary of results for TEOM predictions of a FTP. The ANN was trained on the same test schedule.

Case	Architecture	Abbreviation	Activation Functions	r^2	R^2	t (s)
1	3-Layer	AF1	Lin-G-G-G	0.9301	0.9301	109
2	3-Layer	AF2	Lin-Log-Log-Log	0.8189	0.8187	256
3	3-Layer	AF3	Lin-SymLog-SymLog-SymLog	0.9055	0.9050	180
4	3-Layer	AF4	Lin-G-GC-Log	0.9157	0.9134	114
5	3-Layer	AF5	Lin-tanh-tanh-tanh	0.9396	0.9382	139
6	3-Layer	AF6	Lin-GC-GC-GC	0.9365	0.9355	101
7	3-Layer	AF7	Lin-Sin-Sin-Sin	0.9444	0.9438	79
8	3-Layer	AF8	Lin-tanh15-tanh15-tanh15	0.9279	0.9190	91
9	Jump Connection	AF1	Lin-G-G-G-Log	0.9513	0.9504	137
10	Jump Connection	AF2	Lin-Log-Log-Log-Log	0.6556	0.6547	25
11	Jump Connection	AF3	Lin-SymLog-SymLog-SymLog	0.9265	0.9258	271
12	Jump Connection	AF4	Lin-G-GC-G-Log	0.9529	0.9526	148
13	Jump Connection	AF5	Lin-tanh-tanh-tanh-tanh	0.6699	0.5753	11
14	Jump Connection	AF6	Lin-GC-GC-GC-Log	0.9620	0.9615	145
15	Jump Connection	AF7	Lin-Sin-Sin-Sin-Sin	0.6301	0.5904	10
16	Jump Connection	AF8	Lin-tanh15-tanh15-tanh15-tanh	0.3483	-1.6389	13

Table 6.25. Summary of results for TEOM predictions of an E-Highway. The ANN was trained on the same test schedule.

Case	Architecture	Abbreviation	Activation Functions	r^2	R^2	t (s)
1	3-Layer	AF1	Lin-G-G-G	0.5306	0.5236	25
2	3-Layer	AF2	Lin-Log-Log-Log	0.1761	0.1725	15
3	3-Layer	AF3	Lin-SymLog-SymLog-SymLog	0.1703	0.1654	6
4	3-Layer	AF4	Lin-G-GC-Log	0.2000	0.1980	8
5	3-Layer	AF5	Lin-tanh-tanh-tanh	0.1799	0.1716	6
6	3-Layer	AF6	Lin-GC-GC-GC	0.5833	0.5752	28
7	3-Layer	AF7	Lin-Sin-Sin-Sin	0.4655	0.4631	23
8	3-Layer	AF8	Lin-tanh15-tanh15-tanh15	0.2202	0.1850	9
9	Jump Connection	AF1	Lin-G-G-G-Log	0.4544	0.4494	28
10	Jump Connection	AF2	Lin-Log-Log-Log-Log	0.1706	0.1701	8
11	Jump Connection	AF3	Lin-SymLog-SymLog-SymLog	0.1550	0.1536	2
12	Jump Connection	AF4	Lin-G-GC-G-Log	0.1843	0.1831	3
13	Jump Connection	AF5	Lin-tanh-tanh-tanh-tanh	0.1679	0.1458	3
14	Jump Connection	AF6	Lin-GC-GC-GC-Log	0.2104	0.2086	6
15	Jump Connection	AF7	Lin-Sin-Sin-Sin-Sin	0.1737	0.1019	7
16	Jump Connection	AF8	Lin-tanh15-tanh15-tanh15-tanh	0.1323	-0.4571	7

Table 6.26. Summary of results for TEOM predictions of an E-WVU-5 Peak. The ANN was trained on the same test schedule.

6.6.2. Predicting Other Test Schedules

Reviewing all of the previous results for five transient test schedules teaches that the Gaussian Complement function is the best function. Overall case 14 is the best or the second best in four of the test schedules and in the fifth test schedule it is among first five good predictors. Accordingly it was used to be trained on the combination test schedules and predict each test schedule individually.

There could be different schemes to move towards an optimized network that overall works fair for all of the possibilities. Here, unlike the NO_x case, the author decided to train the combined test schedules on the most promising network and predict each test schedule one by one. Accordingly a Jump Connection network with a Gaussian Complement function in all of its three hidden layers, Linear function at input layer and logistic in output layer was trained on the combined test schedules and the following Table 6.27 presents the result of predictions done by this network. These are the comparison between TEOM values that are not corrected for their moisture content. Later both actual and network values were corrected based on the model described before in the

“Moisture Correction” section. Results for these corrected TEOM values that are supposedly a better representative of PM are shown in Table 6.28. With the intention of avoiding superposition of errors, the network values of TEOM were corrected using the actual CO₂ values. While another option would be correction the TEOM using the prediction of CO₂ values obtained from the very same architecture, where those values would carry the burden of their own errors.

Combined Test Schedules Predicts	Jump Connection, Lin-GC-GC-GC	
	r^2	R^2
E-CSHVR	0.6078	0.6023
ETC	0.0450	-0.2642
FTP	0.2431	0.2118
E-Highway	0.0450	-0.2642
E-WVU- 5 Peak	0.0249	-0.8296
Average	0.1932	-0.1088

Table 6.27. Results for prediction of TEOM (PM not corrected for moisture content) by using a Jump Connection network with activation function of Gaussian Complement. The ANN was trained on all the transient test schedules combined and predicting each test schedules individually.

Combined Test Schedules Predicts	Jump Connection, Lin-GC-GC-GC	
	r^2	R^2
E-CSHVR	0.6278	0.6271
ETC	0.0674	-0.2315
FTP	0.1758	0.1496
E-Highway	0.6104	0.7764
E-WVU- 5 Peak	0.0000	-0.6234
Average	0.2963	0.1396

Table 6.28. Results for prediction of PM (corrected TEOM for moisture content) by using a Jump Connection network with activation function of Gaussian Complement. The ANN was trained on all the transient test schedules combined and predicting each test schedules individually.

The following graphs present the second by second predictions of this Jump Connection network with Gaussian Complement activation function for the PM emissions of an E-CSHVR test schedule. Graphs for other four test schedules are available in Appendix A (Figure 9.27 through Figure 9.30).

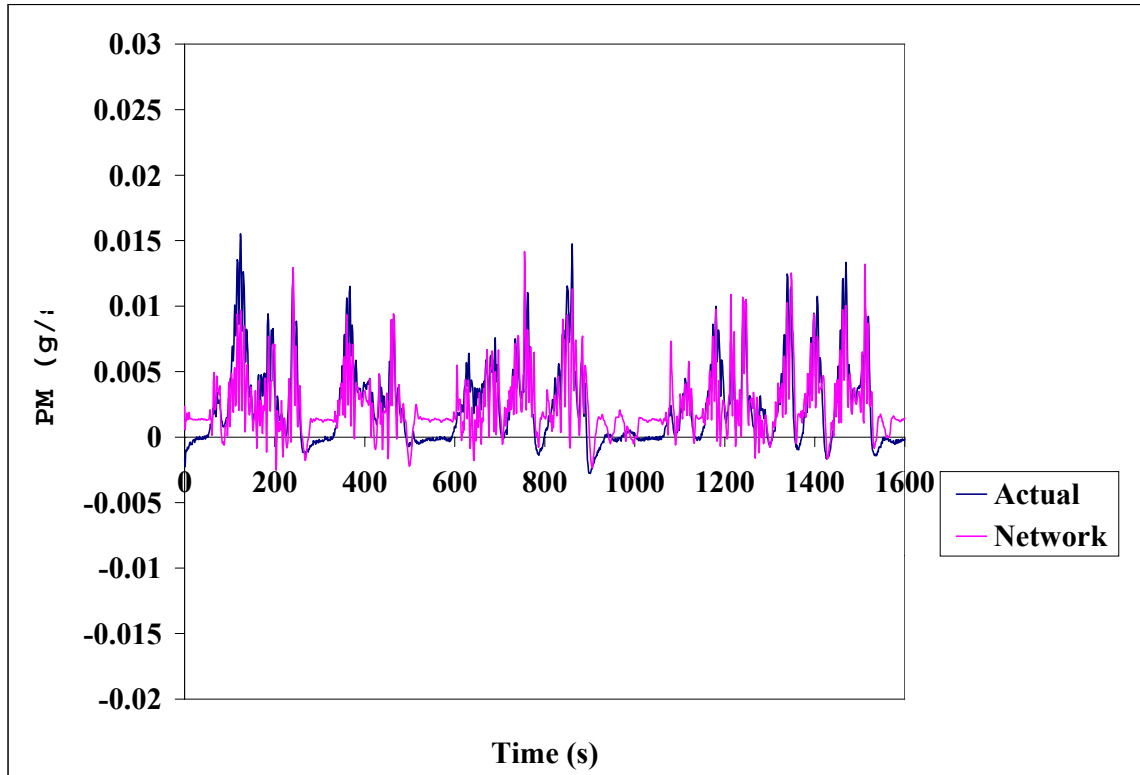


Figure 6.3. Second by second prediction of PM (corrected TEOM for moisture content) in an E-CSHVR test schedule by a Jump Connection network of Gaussian Complement activation function. The ANN was trained on all the transient test schedules combined.

6.6.2.1. GMDH

As in NO_x predictions in the last approach in generalizing the correlation between engine variables and emissions, the Group Method of Data Handling (GMDH) networks were employed. The lengthy final formula with medium nonlinearity is presented in Appendix B. The following Table 6.29 shows the results of predictions made by this network.

Combined Test Schedules Predicts	GMDH	
	r^2	R^2
E-CSHVR	0.5940	0.5937
ETC	0.5940	0.5937
FTP	0.2399	0.2068
E-Highway	0.5516	0.4916
E-WVU- 5 Peak	0.0132	-0.7044
Average	0.3985	0.2363

Table 6.29. Results of TEOM (PM not corrected for moisture content) predictions obtained using a GMDH. The ANN was trained on all the transient test schedules combined and predicting each test schedules individually.

The TEOM values were corrected for their moisture content to represent a better approach to real PM values. The following Table 6.30 shows the results after correction.

Combined Test Schedules Predicts	GMDH	
	r^2	R^2
E-CSHVR	0.6193	0.6190
ETC	0.6272	0.6270
FTP	0.1735	0.1442
E-Highway	0.7432	0.5738
E-WVU- 5 Peak	0.0004	-0.5123
Average	0.4327	0.2903

Table 6.30. Results of PM (TEOM corrected for moisture content) predictions obtained using a GMDH. The ANN was trained on all the transient test schedules combined and predicting each test schedules individually.

Contrary to NO_x case it is evident that GMDH has an average better prediction compared to the Jump Connection network that was meant to have the optimized configurations for the best prediction. This might be addressed as a result of information lack to the network, particularly in areas of negative TEOM values leading to a confused network.

6.6.3. Deduction

Considering Table 6.28 and the second by second prediction graphs, it is concluded that a Jump Connection of Gaussian Complement activation function gives the impression of the best architecture among all those were examined here. Yet, it is obvious that predictions are nowhere near thriving predictions of NO_x . Besides the nature of PM

formation, more complications arise from the inaccuracy of TEOM values. Although the moisture correction was an attempt to achieve more accurate data, other factors like volatile compounds and frequency resonance from the ambient could significantly reduce the accuracy of TEOM values, while these features are not eliminated in that model. Since there has been no information provided to the ANN regarding how these features could affect the TEOM data, it is well expected that the model would have poor generalization. Reviewing the weighting factors also suggest that because of irrelevant data, the network treats all the inputs equally and this is indeed the ultimate attempt of the network to ascribe a connection that hardly exists.

Reviewing second by second graphs of a generalized network predicting different test schedules turns that the network mainly diverges from the actual values where the values are negative and associated with the evaporation from the TEOM filter. This is exactly where there is no information provided to network. The author believes that if in future works some level of information, perhaps in the form of temperature and relative humidity of the exhaust stream is provided to the network, improved prediction would be achieved.

Overall the author concludes that once more, here the basic definition of ANN is proven to be true. They learn from example and the factual reason of their poor performance in PM predictions arises from the lack of adequate inputs. For the most part in the areas of negative TEOM measurements, there is no information in the speed, torque and their derivatives associated with these values. Even the moisture correction model did not eliminate these negative areas and it was confirmed verbally by one of the authors (Jarrett) of the PM correction model [70] that this model is in its early stages and needs further developments to reflect the true second by second PM emissions and work for all of the test schedules using proper constants that are yet to be found. For that reason the instantaneous PM predictions remains an open problem depending on a precise measurement of PM. Prior to that stage, the author believes that adding some appropriate input variables such as temperature of the flow, relative humidity, pressure or any other

factor that might affect the negative values in TEOM data, seem to be very practical for the future trials.

6.7. CO Predictions

Using the same choice for training criteria, as in NO_x prediction (learning rate of 0.05, momentum of 0.05 and initial weight of 0.3), the CO predictions were done for all of the transient test schedules. Similar to PM predictions, the optimization process was carried out by the best two architectures of 3-Layer and Jump Connection.

6.7.1. Self-Predicting Test Schedules

In this section all of the discussions are based on training an ANN on a particular test schedule and predicting emissions of the same test schedule. The following five tables (Table 6.31 through Table 6.35) show the results for the five studies test schedules. In addition, for a better visual impression for each test schedule the following graphs are presented in Appendix A: a 3-D graph that presents the variation of r^2 in the plane of activation function and architecture for the two architectures of choice, and a 2-D graph of both r^2 and R^2 for all cases as another staging of the best architecture and activation function assortment (Figure 9.31 through Figure 9.40).

Case	Architecture	Abbreviation	Activation Functions	r^2	R^2	t (s)
1	3-Layer	AF1	Lin-G-G-G	0.9039	0.9037	13
2	3-Layer	AF2	Lin-Log-Log-Log	0.8632	0.8628	28
3	3-Layer	AF3	Lin-SymLog-SymLog-SymLog	0.8633	0.8617	7
4	3-Layer	AF4	Lin-G-GC-Log	0.8959	0.8952	11
5	3-Layer	AF5	Lin-tanh-tanh-tanh	0.8987	0.8985	15
6	3-Layer	AF6	Lin-GC-GC-GC	0.9604	0.9601	55
7	3-Layer	AF7	Lin-Sin-Sin-Sin	0.8913	0.8911	10
8	3-Layer	AF8	Lin-tanh15-tanh15-tanh15	0.8650	0.8535	4
9	Jump Connection	AF1	Lin-G-G-G-Log	0.9575	0.9573	65
10	Jump Connection	AF2	Lin-Log-Log-Log-Log	0.8584	0.8581	10
11	Jump Connection	AF3	Lin-SymLog-SymLog-SymLog	0.8600	0.8580	5
12	Jump Connection	AF4	Lin-G-GC-G-Log	0.9542	0.9540	68
13	Jump Connection	AF5	Lin-tanh-tanh-tanh-tanh	0.8716	0.8714	4
14	Jump Connection	AF6	Lin-GC-GC-GC-Log	0.9575	0.9573	62
15	Jump Connection	AF7	Lin-Sin-Sin-Sin-Sin	0.8697	0.8654	4
16	Jump Connection	AF8	Lin-tanh15-tanh15-tanh15-tanh	0.7927	0.6856	5

Table 6.31. Summary of results for CO predictions of an E-CSHVR. The ANN was trained on the same test schedule.

Case	Architecture	Abbreviation	Activation Functions	r^2	R^2	t (s)
1	3-Layer	AF1	Lin-G-G-G	0.8610	0.8609	35
2	3-Layer	AF2	Lin-Log-Log-Log	0.6185	0.6183	38
3	3-Layer	AF3	Lin-SymLog-SymLog-SymLog	0.7894	0.7882	58
4	3-Layer	AF4	Lin-G-GC-Log	0.8475	0.8453	39
5	3-Layer	AF5	Lin-tanh-tanh-tanh	0.8211	0.8189	27
6	3-Layer	AF6	Lin-GC-GC-GC	0.8462	0.8459	22
7	3-Layer	AF7	Lin-Sin-Sin-Sin	0.8185	0.8173	13
8	3-Layer	AF8	Lin-tanh15-tanh15-tanh15	0.7848	0.7724	11
9	Jump Connection	AF1	Lin-G-G-G-Log	0.8671	0.8645	47
10	Jump Connection	AF2	Lin-Log-Log-Log-Log	0.6215	0.6210	21
11	Jump Connection	AF3	Lin-SymLog-SymLog-SymLog	0.7952	0.7936	59
12	Jump Connection	AF4	Lin-G-GC-G-Log	0.8726	0.8722	48
13	Jump Connection	AF5	Lin-tanh-tanh-tanh-tanh	0.7171	0.7033	13
14	Jump Connection	AF6	Lin-GC-GC-GC-Log	0.8832	0.8824	51
15	Jump Connection	AF7	Lin-Sin-Sin-Sin-Sin	0.6638	0.6574	6
16	Jump Connection	AF8	Lin-tanh15-tanh15-tanh15-tanh	0.5627	0.2694	5

Table 6.32. Summary of results for CO predictions of an ETC. The ANN was trained on the same test schedule.

Case	Architecture	Abbreviation	Activation Functions	r^2	R^2	t (s)
1	3-Layer	AF1	Lin-G-G-G	0.9715	0.9713	20
2	3-Layer	AF2	Lin-Log-Log-Log	0.8242	0.8241	5
3	3-Layer	AF3	Lin-SymLog-SymLog-SymLog	0.9768	0.9766	81
4	3-Layer	AF4	Lin-G-GC-Log	0.9736	0.9733	33
5	3-Layer	AF5	Lin-tanh-tanh-tanh	0.9713	0.9709	19
6	3-Layer	AF6	Lin-GC-GC-GC	0.9814	0.9803	41
7	3-Layer	AF7	Lin-Sin-Sin-Sin	0.9812	0.9804	27
8	3-Layer	AF8	Lin-tanh15-tanh15-tanh15	0.9615	0.9599	14
9	Jump Connection	AF1	Lin-G-G-G-Log	0.9811	0.9808	41
10	Jump Connection	AF2	Lin-Log-Log-Log-Log	0.8299	0.8298	4
11	Jump Connection	AF3	Lin-SymLog-SymLog-SymLog	0.9796	0.9796	111
12	Jump Connection	AF4	Lin-G-GC-G-Log	0.9797	0.9792	42
13	Jump Connection	AF5	Lin-tanh-tanh-tanh-tanh	0.9223	0.9216	10
14	Jump Connection	AF6	Lin-GC-GC-GC-Log	0.9819	0.9815	41
15	Jump Connection	AF7	Lin-Sin-Sin-Sin-Sin	0.8525	0.8430	3
16	Jump Connection	AF8	Lin-tanh15-tanh15-tanh15-tanh	0.7268	0.6368	3

Table 6.33. Summary of results for CO prediction of a FTP. The ANN was trained on the same test schedule.

Case	Architecture	Abbreviation	Activation Functions	r^2	R^2	t (s)
1	3-Layer	AF1	Lin-G-G-G	0.9386	0.9381	33
2	3-Layer	AF2	Lin-Log-Log-Log	0.6221	0.6208	42
3	3-Layer	AF3	Lin-SymLog-SymLog-SymLog	0.8073	0.8073	48
4	3-Layer	AF4	Lin-G-GC-Log	0.9218	0.9188	50
5	3-Layer	AF5	Lin-tanh-tanh-tanh	0.9195	0.9177	30
6	3-Layer	AF6	Lin-GC-GC-GC	0.9421	0.9414	42
7	3-Layer	AF7	Lin-Sin-Sin-Sin	0.9242	0.9216	30
8	3-Layer	AF8	Lin-tanh15-tanh15-tanh15	0.8459	0.8395	10
9	Jump Connection	AF1	Lin-G-G-G-Log	0.9089	0.9076	33
10	Jump Connection	AF2	Lin-Log-Log-Log-Log	0.6116	0.6110	9
11	Jump Connection	AF3	Lin-SymLog-SymLog-SymLog	0.7948	0.7923	31
12	Jump Connection	AF4	Lin-G-GC-G-Log	0.9323	0.9316	48
13	Jump Connection	AF5	Lin-tanh-tanh-tanh-tanh	0.8669	0.8635	16
14	Jump Connection	AF6	Lin-GC-GC-GC-Log	0.9494	0.9483	58
15	Jump Connection	AF7	Lin-Sin-Sin-Sin-Sin	0.7423	0.7314	5
16	Jump Connection	AF8	Lin-tanh15-tanh15-tanh15-tanh	0.5761	0.5730	4

Table 6.34. Summary of results for CO prediction of an E-Highway. The ANN was trained on the same test schedule.

Case	Architecture	Abbreviation	Activation Functions	r^2	R^2	t (s)
1	3-Layer	AF1	Lin-G-G-G	0.9260	0.9259	9
2	3-Layer	AF2	Lin-Log-Log-Log	0.9000	0.8994	5
3	3-Layer	AF3	Lin-SymLog-SymLog-SymLog	0.9028	0.9025	4
4	3-Layer	AF4	Lin-G-GC-Log	0.9124	0.9122	4
5	3-Layer	AF5	Lin-tanh-tanh-tanh	0.9055	0.9053	3
6	3-Layer	AF6	Lin-GC-GC-GC	0.9176	0.9175	4
7	3-Layer	AF7	Lin-Sin-Sin-Sin	0.9016	0.9008	2
8	3-Layer	AF8	Lin-tanh15-tanh15-tanh15	0.9085	0.9076	3
9	Jump Connection	AF1	Lin-G-G-G-Log	0.9194	0.9193	9
10	Jump Connection	AF2	Lin-Log-Log-Log-Log	0.8971	0.8971	7
11	Jump Connection	AF3	Lin-SymLog-SymLog-SymLog	0.9036	0.9035	6
12	Jump Connection	AF4	Lin-G-GC-G-Log	0.9206	0.9205	8
13	Jump Connection	AF5	Lin-tanh-tanh-tanh-tanh	0.8850	0.8726	2
14	Jump Connection	AF6	Lin-GC-GC-GC-Log	0.9145	0.9145	6
15	Jump Connection	AF7	Lin-Sin-Sin-Sin-Sin	0.9081	0.9044	4
16	Jump Connection	AF8	Lin-tanh15-tanh15-tanh15-tanh	0.8441	0.7794	2

Table 6.35. Summary of results CO prediction of an E-WVU-5 Peak. The ANN was trained on the same test schedule.

6.7.2. Predicting Other Test Schedules

Reviewing all of the previous results shows that for three of the test schedules (ETC, FTP and E-Highway) case 14 (a Jump Connection network with Gaussian Complement function in hidden layers) was the best network. For other two, case 14 is among the first five best predictors. Thus this network was the choice for training the combined test schedules and predicting each test schedule individually.

Combined Test Schedules Predicts	Jump Connection, Lin-GC-GC-GC	
	r^2	R^2
E-CSHVR	0.9104	0.9077
ETC	0.7553	0.4211
FTP	0.8875	0.8842
E-Highway	0.8692	0.8584
E-WVU- 5 Peak	0.8916	0.8459
Average	0.8628	0.7835

Table 6.36. Results for prediction of CO emissions by using a Jump Connection network with activation function of Gaussian Complement. The ANN was trained on all transient test schedules combined and predicting each test schedules individually.

The following graph presents the second by second predictions of this Jump Connection network with Gaussian Complement activation function for the CO emissions of an E-CSHVR test schedule. Graphs for other four test schedules are available in Appendix A (Figure 9.41 through Figure 9.44).

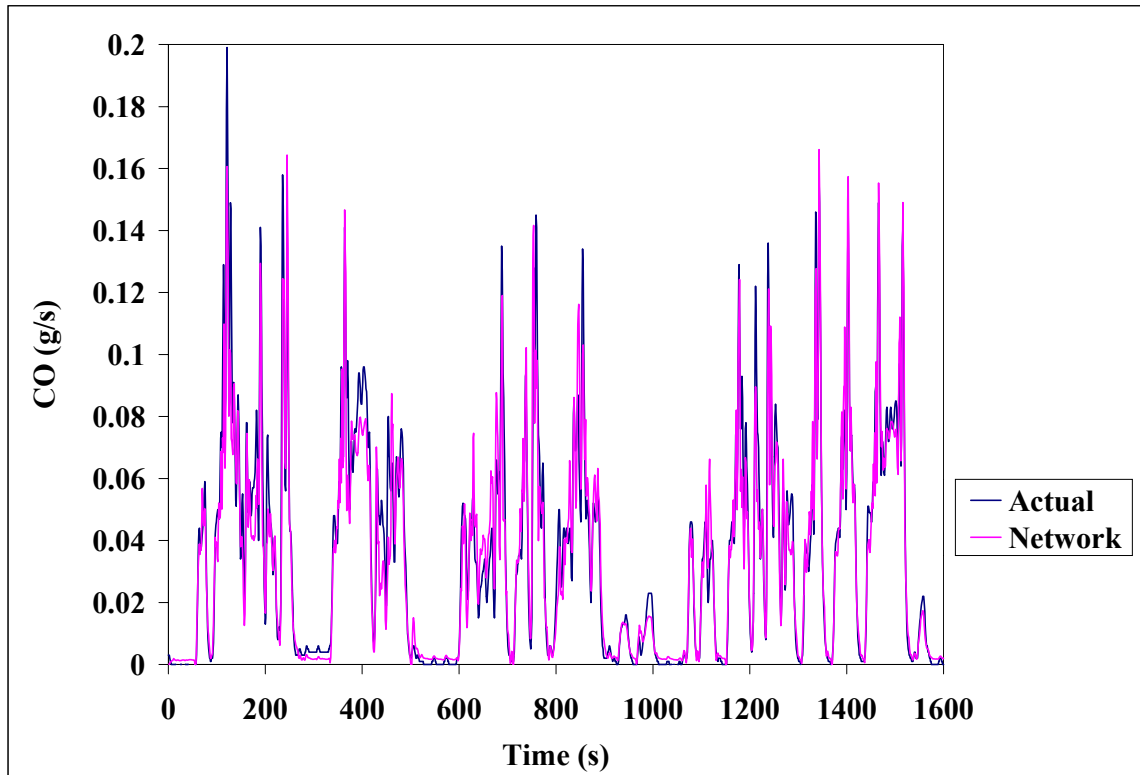


Figure 6.4. Second by second prediction of CO in an E-CSHVR test schedule by a Jump Connection network of Gaussian Complement activation function. The ANN was trained on all the transient test schedules combined.

6.7.2.1. GMDH

As in previous emissions predictions, in the last approach in generalizing the correlation between engine variables and emissions, the Group Method of Data Handling (GMDH) networks were employed. The lengthy final formula with medium nonlinearity is presented in Appendix . The following table shows the results of predictions made by this network.

Combined Test Schedules Predicts	GMDH	
	r^2	R^2
E-CSHVR	0.8586	0.8468
ETC	0.6981	0.2424
FTP	0.7959	0.7887
E-Highway	0.7333	0.7274
E-WVU- 5 Peak	0.8439	0.7145
Average	0.7860	0.6640

Table 6.37. Results of CO predictions obtained using a GMDH. The ANN was trained on all the transient test schedules combined and predicting each test schedules individually.

Similar to NO_x case, it is observed that GMDH has an average less accurate prediction compared to the Jump Connection network and this is in the same line with the basic idea behind this network. Yet, it is quite informative paying attention to the formula given by this network in terms of each individual input's role.

6.7.3. Deduction

Considering Table 6.36 and the second by second graphs, it is concluded that a Jump Connection of Gaussian Complement activation function give the best results for the E-CSHVR, and the worse results are for the high speed cycle ETC. The extreme divergence of the network from actual values happens in the higher speed end of the cycle, where the network over-predicts the emissions. To explain this matter, it is useful to consider that in the ending portion of ETC, speed remains constant and Torque has a high frequency and relatively low amplitude variation in the time frame. This results in the same pattern of power (high frequency), which seems to be over-learned by the network. Meaning that by almost giving the same weightings to the speed and torque as well as all their derivatives, the transient effects in power are over learned and the author believes this is the reason of over-prediction in case of ETC. In all other test schedules, looks like that network has smoothen the CO values. However the complicated formation of CO remains to be discovered and improved, the author believes that avoiding too many inputs from engine map and adding some other input as Oxygen concentration may lead to some improvements

The Jump Connection network of choice with Gaussian Complement activation function offers very good potentials in CO predictions in the case of E-CSHVR and is fairly acceptable for FTP, E-Highway and E-WVU-5 Peak, yet to be improved. Like other earlier attempts in predicting CO, the sophisticated formation of CO is emphasized here. Still, the author believes accurate predictions are very feasible at least compared with the HC case, by further attempts in finding the right choice of inputs to the network perhaps, by digging more into the formation mechanism of CO.

6.8. CO₂ Predictions

Using the same choice for training criteria, as in previous predictions (learning rate of 0.05, momentum of 0.05 and initial weight of 0.3), the CO₂ predictions were done for all of the transient test schedules. The optimization process was carried out by the two architectures of 3-Layer and Jump Connection.

6.8.1. Self Predicting Test Schedules

In this section all of the discussions are based on training an ANN on a particular test schedule and predicting emissions of the same test schedule. The following five tables (Table 6.38 through Table 6.42) show the results for the five studied test schedules. In addition, for a better visual impression for each test schedule there comes in the Appendix A, a 3-D graph that presents the variation of r^2 in the plane of activation function and architecture for the two architectures of choice, and a 2-D graph of both r^2 and R^2 for all cases as another staging of the best architecture and activation function assortment (Figure 9.45 through Figure 9.54).

Case	Architecture	Abbreviation	Activation Functions	r^2	R^2	t (s)
1	3-Layer	AF1	Lin-G-G-G	0.9952	0.9951	57
2	3-Layer	AF2	Lin-Log-Log-Log	0.9862	0.9862	70
3	3-Layer	AF3	Lin-SymLog-SymLog-SymLog	0.9893	0.9888	50
4	3-Layer	AF4	Lin-G-GC-Log	0.9961	0.9961	71
5	3-Layer	AF5	Lin-tanh-tanh-tanh	0.9959	0.9958	50
6	3-Layer	AF6	Lin-GC-GC-GC	0.9959	0.9959	54
7	3-Layer	AF7	Lin-Sin-Sin-Sin	0.9947	0.9947	31
8	3-Layer	AF8	Lin-tanh15-tanh15-tanh15	0.9960	0.9960	29
9	Jump Connection	AF1	Lin-G-G-G-Log	0.9955	0.9955	68
10	Jump Connection	AF2	Lin-Log-Log-Log-Log	0.9834	0.9832	54
11	Jump Connection	AF3	Lin-SymLog-SymLog-SymLog	0.9878	0.9873	53
12	Jump Connection	AF4	Lin-G-GC-G-Log	0.9962	0.9962	87
13	Jump Connection	AF5	Lin-tanh-tanh-tanh-tanh	0.9967	0.9967	86
14	Jump Connection	AF6	Lin-GC-GC-GC-Log	0.9951	0.9951	57
15	Jump Connection	AF7	Lin-Sin-Sin-Sin-Sin	0.9946	0.9946	74
16	Jump Connection	AF8	Lin-tanh15-tanh15-tanh15-tanh	0.9775	0.9774	5

Table 6.38. Summary of results for CO₂ predictions of an E-CSHVR. The ANN was trained on the same test schedule.

Case	Architecture	Abbreviation	Activation Functions	r^2	R^2	t (s)
1	3-Layer	AF1	Lin-G-G-G	0.9950	0.9950	75
2	3-Layer	AF2	Lin-Log-Log-Log	0.9790	0.9789	86
3	3-Layer	AF3	Lin-SymLog-SymLog-SymLog	0.9810	0.9801	22
4	3-Layer	AF4	Lin-G-GC-Log	0.9954	0.9954	97
5	3-Layer	AF5	Lin-tanh-tanh-tanh	0.9921	0.9918	32
6	3-Layer	AF6	Lin-GC-GC-GC	0.9927	0.9927	46
7	3-Layer	AF7	Lin-Sin-Sin-Sin	0.9938	0.9937	43
8	3-Layer	AF8	Lin-tanh15-tanh15-tanh15	0.9939	0.9939	17
9	Jump Connection	AF1	Lin-G-G-G-Log	0.9956	0.9956	94
10	Jump Connection	AF2	Lin-Log-Log-Log-Log	0.9769	0.9763	32
11	Jump Connection	AF3	Lin-SymLog-SymLog-SymLog	0.9811	0.9801	30
12	Jump Connection	AF4	Lin-G-GC-G-Log	0.9947	0.9947	69
13	Jump Connection	AF5	Lin-tanh-tanh-tanh-tanh	0.9920	0.9918	38
14	Jump Connection	AF6	Lin-GC-GC-GC-Log	0.9944	0.9943	66
15	Jump Connection	AF7	Lin-Sin-Sin-Sin-Sin	0.9954	0.9953	147
16	Jump Connection	AF8	Lin-tanh15-tanh15-tanh15-tanh	0.9706	0.9592	5

Table 6.39. Summary of results for CO₂ predictions of an ETC. The ANN was trained on the same test schedule.

Case	Architecture	Abbreviation	Activation Functions	r^2	R^2	t (s)
1	3-Layer	AF1	Lin-G-G-G	0.9986	0.9986	28
2	3-Layer	AF2	Lin-Log-Log-Log	0.9945	0.9945	78
3	3-Layer	AF3	Lin-SymLog-SymLog-SymLog	0.9941	0.9938	46
4	3-Layer	AF4	Lin-G-GC-Log	0.9991	0.9991	59
5	3-Layer	AF5	Lin-tanh-tanh-tanh	0.9988	0.9988	35
6	3-Layer	AF6	Lin-GC-GC-GC	0.9986	0.9985	33
7	3-Layer	AF7	Lin-Sin-Sin-Sin	0.9984	0.9983	15
8	3-Layer	AF8	Lin-tanh15-tanh15-tanh15	0.9828	0.9809	2
9	Jump Connection	AF1	Lin-G-G-G-Log	0.9986	0.9986	78
10	Jump Connection	AF2	Lin-Log-Log-Log-Log	0.9829	0.9829	18
11	Jump Connection	AF3	Lin-SymLog-SymLog-SymLog	0.9878	0.9876	7
12	Jump Connection	AF4	Lin-G-GC-G-Log	0.9988	0.9988	106
13	Jump Connection	AF5	Lin-tanh-tanh-tanh-tanh	0.9989	0.9989	55
14	Jump Connection	AF6	Lin-GC-GC-GC-Log	0.9992	0.9992	149
15	Jump Connection	AF7	Lin-Sin-Sin-Sin-Sin	0.9980	0.9979	27
16	Jump Connection	AF8	Lin-tanh15-tanh15-tanh15-tanh	0.9803	0.9764	3

Table 6.40. Summary of results for CO₂ prediction of a FTP. The ANN was trained on the same test schedule.

Case	Architecture	Abbreviation	Activation Functions	r^2	R^2	t (s)
1	3-Layer	AF1	Lin-G-G-G	0.9972	0.9971	40
2	3-Layer	AF2	Lin-Log-Log-Log	0.9817	0.9816	42
3	3-Layer	AF3	Lin-SymLog-SymLog-SymLog	0.9882	0.9872	66
4	3-Layer	AF4	Lin-G-GC-Log	0.9969	0.9968	49
5	3-Layer	AF5	Lin-tanh-tanh-tanh	0.9970	0.9969	40
6	3-Layer	AF6	Lin-GC-GC-GC	0.9965	0.9964	44
7	3-Layer	AF7	Lin-Sin-Sin-Sin	0.9979	0.9978	31
8	3-Layer	AF8	Lin-tanh15-tanh15-tanh15	0.9967	0.9964	16
9	Jump Connection	AF1	Lin-G-G-G-Log	0.9978	0.9978	80
10	Jump Connection	AF2	Lin-Log-Log-Log-Log	0.9733	0.9725	27
11	Jump Connection	AF3	Lin-SymLog-SymLog-SymLog	0.9818	0.9815	17
12	Jump Connection	AF4	Lin-G-GC-G-Log	0.9978	0.9978	73
13	Jump Connection	AF5	Lin-tanh-tanh-tanh-tanh	0.9978	0.9977	63
14	Jump Connection	AF6	Lin-GC-GC-GC-Log	0.9975	0.9974	75
15	Jump Connection	AF7	Lin-Sin-Sin-Sin-Sin	0.9969	0.9968	16
16	Jump Connection	AF8	Lin-tanh15-tanh15-tanh15-tanh	0.9660	0.9657	4

Table 6.41. Summary of results for CO₂ prediction of an E-Highway. The ANN was trained on the same test schedule.

Case	Architecture	Abbreviation	Activation Functions	r^2	R^2	t (s)
1	3-Layer	AF1	Lin-G-G-G	0.9946	0.9940	15
2	3-Layer	AF2	Lin-Log-Log-Log	0.9884	0.9884	22
3	3-Layer	AF3	Lin-SymLog-SymLog-SymLog	0.9872	0.9858	9
4	3-Layer	AF4	Lin-G-GC-Log	0.9961	0.9959	29
5	3-Layer	AF5	Lin-tanh-tanh-tanh	0.9941	0.9939	19
6	3-Layer	AF6	Lin-GC-GC-GC	0.9957	0.9956	23
7	3-Layer	AF7	Lin-Sin-Sin-Sin	0.9925	0.9923	6
8	3-Layer	AF8	Lin-tanh15-tanh15-tanh15	0.9945	0.9943	6
9	Jump Connection	AF1	Lin-G-G-G-Log	0.9953	0.9949	30
10	Jump Connection	AF2	Lin-Log-Log-Log-Log	0.9841	0.9840	8
11	Jump Connection	AF3	Lin-SymLog-SymLog-SymLog	0.9882	0.9871	12
12	Jump Connection	AF4	Lin-G-GC-G-Log	0.9953	0.9951	29
13	Jump Connection	AF5	Lin-tanh-tanh-tanh-tanh	0.9932	0.9931	13
14	Jump Connection	AF6	Lin-GC-GC-GC-Log	0.9954	0.9952	31
15	Jump Connection	AF7	Lin-Sin-Sin-Sin-Sin	0.9945	0.9942	9
16	Jump Connection	AF8	Lin-tanh15-tanh15-tanh15-tanh	0.9926	0.9919	10

Table 6.42. Summary of results CO₂ prediction of an E-WVU-5 Peak. The ANN was trained on the same test schedule.

6.8.2. Predicting Other Test Schedules

For CO₂ predictions every test schedule turns a different architecture as the best predictor. However, case 4 (3-Layer, Lin-G-GC-Log) was among the best five predictors for all of them, therefore this network was the choice for training the combined test schedules predicting each test schedule individually.

Combined Test Schedules Predicts	3-Layer, Lin-G-GC-G	
	r^2	R^2
E-CSHVR	0.9792	0.9700
ETC	0.9745	0.6265
FTP	0.9909	0.9906
E-Highway	0.9828	0.9665
E-WVU- 5 Peak	0.9787	0.9615
Average	0.9812	0.9030

Table 6.43. Results for prediction of CO₂ emissions by using a 3-Layer network with both activation functions of Gaussian and Gaussian Complement. The ANN was trained on all the transient test schedules combined and predicting each test schedules individually.

The following graph presents the second by second predictions of this 3-Layer network with Gaussian and Gaussian Complement activation functions for the CO₂ emissions of a FTP cycle. Graphs for other four test schedules are available in Appendix A (Figure 9.55 through Figure 9.58).

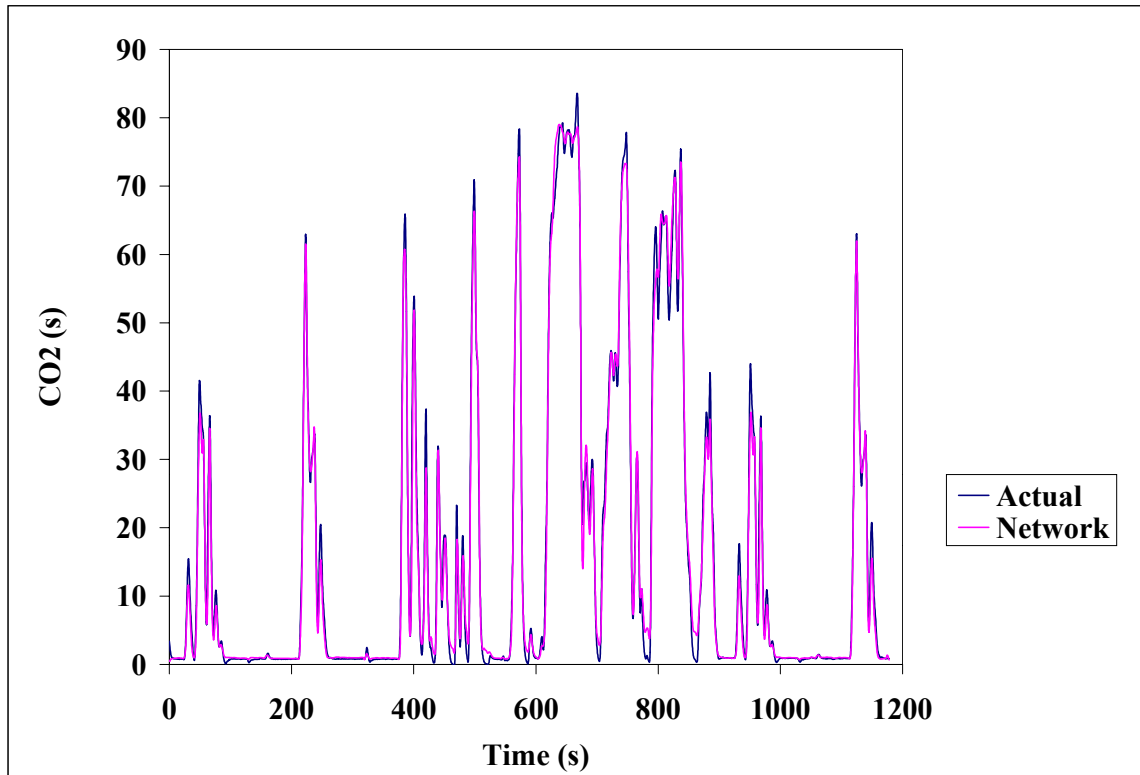


Figure 6.5. Second by second prediction of CO₂ in a FTP test schedule by a 3-Layer network of both Gaussian and Gaussian Complement activation functions. The ANN was trained on all the transient test schedules combined.

6.8.2.1. GMDH

The same as preceding emissions predictions, in the last move toward generalizing the correlation between engine variables and emissions, the Group Method of Data Handling (GMDH) networks were utilized. The extensive final formula with medium nonlinearity is presented in Appendix B. The following table shows the results of predictions made by this network.

Combined Test Schedules Predicts	GMDH	
	r^2	R^2
E-CSHVR	0.9792	0.9729
ETC	0.9750	0.6266
FTP	0.9911	0.9909
E-Highway	0.9882	0.9636
E-WVU- 5 Peak	0.9791	0.9596
Average	0.9825	0.9027

Table 6.44. Results of CO₂ predictions obtained using a GMDH. The ANN was trained on all the transient test schedules combined and predicting each test schedules individually.

Interestingly GMDH has a higher average of r^2 and almost the same average R^2 , though both 3-Layer and GMDH have given an impressive accuracy. This could be the result of a strong correlation between inputs and CO₂, and the correlation being straight forward enough to yield a precise mathematical formula.

6.8.3. Deduction

Considering Table 6.43 and the second by second predictions graphs, it is concluded that a 3-Layer network of both Gaussian and Gaussian Complement activation function gives out excellent results in CO₂ predictions in a similar way to NO_x predictions. For CO₂ being proportional to power and having all derivatives of power over time as inputs, provides outstanding information to the network, resulting in precise predictions.

The 3-Layer network of choice with Gaussian and Gaussian Complement activation functions provides accurate results in CO₂ predictions. Results are reliable both in the case of predicting a certain test schedule or any given driving condition using a broad data base previously seen by the network.

6.9. HC Predictions

Using the same choice for training criteria, as in previous predictions (learning rate of 0.05, momentum of 0.05 and initial weight of 0.3), the HC predictions were done for all of the transient test schedules. The optimization process was carried out by the two architectures of 3-Layer and Jump Connection.

6.9.1. Self-Predicting Test Schedules

In this section all of the discussions are based on training an ANN on a particular test schedule and predicting emissions of the same test schedule. The following five tables (Table 6.45 through Table 6.49) show the results for the five studies test schedules. To give a better visual impression, for each test schedule there comes in the Appendix A, a 3-D graph that presents the variation of r^2 in the plane of activation function and architecture for the two architectures of choice, and a 2-D graph of both r^2 and R^2 for all cases as another staging of the best architecture and activation function assortment (Figure 9.59 through Figure 9.68).

Case	Architecture	Abbreviation	Activation Functions	r^2	R^2	t (s)
1	3-Layer	AF1	Lin-G-G-G	0.9245	0.9244	27
2	3-Layer	AF2	Lin-Log-Log-Log	0.8243	0.8239	50
3	3-Layer	AF3	Lin-SymLog-SymLog-SymLog	0.9052	0.9040	70
4	3-Layer	AF4	Lin-G-GC-Log	0.9267	0.9264	36
5	3-Layer	AF5	Lin-tanh-tanh-tanh	0.9056	0.9026	24
6	3-Layer	AF6	Lin-GC-GC-GC	0.9188	0.9149	21
7	3-Layer	AF7	Lin-Sin-Sin-Sin	0.9215	0.9212	28
8	3-Layer	AF8	Lin-tanh15-tanh15-tanh15	0.8969	0.8966	12
9	Jump Connection	AF1	Lin-G-G-G-Log	0.9203	0.9194	35
10	Jump Connection	AF2	Lin-Log-Log-Log-Log	0.7507	0.7506	10
11	Jump Connection	AF3	Lin-SymLog-SymLog-SymLog	0.8918	0.8910	37
12	Jump Connection	AF4	Lin-G-GC-G-Log	0.9395	0.9392	54
13	Jump Connection	AF5	Lin-tanh-tanh-tanh-tanh	0.8948	0.8897	21
14	Jump Connection	AF6	Lin-GC-GC-GC-Log	0.9350	0.9345	48
15	Jump Connection	AF7	Lin-Sin-Sin-Sin-Sin	0.8741	0.8672	11
16	Jump Connection	AF8	Lin-tanh15-tanh15-tanh15-tanh	0.6652	0.5046	4

Table 6.45. Summary of results HC predictions of an E-CSHVR. The ANN was trained on the same test schedule.

Case	Architecture	Abbreviation	Activation Functions	r^2	R^2	t (s)
1	3-Layer	AF1	Lin-G-G-G	0.6633	0.6514	12
2	3-Layer	AF2	Lin-Log-Log-Log	0.4719	0.4694	35
3	3-Layer	AF3	Lin-SymLog-SymLog-SymLog	0.4630	0.4587	20
4	3-Layer	AF4	Lin-G-GC-Log	0.7586	0.7576	87
5	3-Layer	AF5	Lin-tanh-tanh-tanh	0.7083	0.7040	37
6	3-Layer	AF6	Lin-GC-GC-GC	0.7768	0.7754	54
7	3-Layer	AF7	Lin-Sin-Sin-Sin	0.7514	0.7500	44
8	3-Layer	AF8	Lin-tanh15-tanh15-tanh15	0.7083	0.7040	37
9	Jump Connection	AF1	Lin-G-G-G-Log	0.8074	0.8069	141
10	Jump Connection	AF2	Lin-Log-Log-Log-Log	0.4640	0.4624	12
11	Jump Connection	AF3	Lin-SymLog-SymLog-SymLog	0.4921	0.4869	10
12	Jump Connection	AF4	Lin-G-GC-G-Log	0.7068	0.7061	41
13	Jump Connection	AF5	Lin-tanh-tanh-tanh-tanh	0.6150	0.6004	17
14	Jump Connection	AF6	Lin-GC-GC-GC-Log	0.7002	0.7001	38
15	Jump Connection	AF7	Lin-Sin-Sin-Sin-Sin	0.6569	0.6556	25
16	Jump Connection	AF8	Lin-tanh15-tanh15-tanh15-tanh	0.5085	0.2862	4

Table 6.46. Summary of results for HC predictions of an ETC. The ANN was trained on the same test schedule.

Case	Architecture	Abbreviation	Activation Functions	r^2	R^2	t (s)
1	3-Layer	AF1	Lin-G-G-G	0.7742	0.7637	19
2	3-Layer	AF2	Lin-Log-Log-Log	0.5560	0.5481	26
3	3-Layer	AF3	Lin-SymLog-SymLog-SymLog	0.6288	0.6056	18
4	3-Layer	AF4	Lin-G-GC-Log	0.7471	0.7415	18
5	3-Layer	AF5	Lin-tanh-tanh-tanh	0.7019	0.6867	10
6	3-Layer	AF6	Lin-GC-GC-GC	0.8238	0.8205	20
7	3-Layer	AF7	Lin-Sin-Sin-Sin	0.5984	0.5726	4
8	3-Layer	AF8	Lin-tanh15-tanh15-tanh15	0.5745	0.5686	3
9	Jump Connection	AF1	Lin-G-G-G-Log	0.6163	0.6036	6
10	Jump Connection	AF2	Lin-Log-Log-Log-Log	0.5548	0.5452	15
11	Jump Connection	AF3	Lin-SymLog-SymLog-SymLog	0.5823	0.5758	12
12	Jump Connection	AF4	Lin-G-GC-G-Log	0.6799	0.6689	11
13	Jump Connection	AF5	Lin-tanh-tanh-tanh-tanh	0.5784	0.5728	4
14	Jump Connection	AF6	Lin-GC-GC-GC-Log	0.6887	0.6770	11
15	Jump Connection	AF7	Lin-Sin-Sin-Sin-Sin	0.5567	0.5407	3
16	Jump Connection	AF8	Lin-tanh15-tanh15-tanh15-tanh	0.4930	0.2206	3

Table 6.47. Summary of results for HC prediction of a FTP. The ANN was trained on the same test schedule.

Case	Architecture	Abbreviation	Activation Functions	r^2	R^2	t (s)
1	3-Layer	AF1	Lin-G-G-G	0.5937	0.5695	37
2	3-Layer	AF2	Lin-Log-Log-Log	0.2730	0.2716	19
3	3-Layer	AF3	Lin-SymLog-SymLog-SymLog	0.2775	0.2757	4
4	3-Layer	AF4	Lin-G-GC-Log	0.5621	0.5553	44
5	3-Layer	AF5	Lin-tanh-tanh-tanh	0.2866	0.2809	3
6	3-Layer	AF6	Lin-GC-GC-GC	0.5390	0.5386	20
7	3-Layer	AF7	Lin-Sin-Sin-Sin	0.4010	0.4006	14
8	3-Layer	AF8	Lin-tanh15-tanh15-tanh15	0.2615	0.2612	3
9	Jump Connection	AF1	Lin-G-G-G-Log	0.6269	0.6254	57
10	Jump Connection	AF2	Lin-Log-Log-Log-Log	0.2653	0.2626	5
11	Jump Connection	AF3	Lin-SymLog-SymLog-SymLog	0.2776	0.2759	5
12	Jump Connection	AF4	Lin-G-GC-G-Log	0.6309	0.6297	63
13	Jump Connection	AF5	Lin-tanh-tanh-tanh-tanh	0.3006	0.2868	7
14	Jump Connection	AF6	Lin-GC-GC-GC-Log	0.6307	0.6303	53
15	Jump Connection	AF7	Lin-Sin-Sin-Sin-Sin	0.2829	0.2665	4
16	Jump Connection	AF8	Lin-tanh15-tanh15-tanh15-tanh	0.2695	0.1316	4

Table 6.48. Summary of results for HC prediction of an E-Highway. The ANN was trained on the same test schedule.

Case	Architecture	Abbreviation	Activation Functions	r^2	R^2	t (s)
1	3-Layer	AF1	Lin-G-G-G	0.9512	0.9499	14
2	3-Layer	AF2	Lin-Log-Log-Log	0.7929	0.7854	17
3	3-Layer	AF3	Lin-SymLog-SymLog-SymLog	0.9322	0.9317	17
4	3-Layer	AF4	Lin-G-GC-Log	0.9577	0.9574	20
5	3-Layer	AF5	Lin-tanh-tanh-tanh	0.9554	0.9547	16
6	3-Layer	AF6	Lin-GC-GC-GC	0.9583	0.9578	12
7	3-Layer	AF7	Lin-Sin-Sin-Sin	0.9570	0.9563	14
8	3-Layer	AF8	Lin-tanh15-tanh15-tanh15	0.9428	0.9411	12
9	Jump Connection	AF1	Lin-G-G-G-Log	0.9583	0.9581	20
10	Jump Connection	AF2	Lin-Log-Log-Log-Log	0.7335	0.7283	4
11	Jump Connection	AF3	Lin-SymLog-SymLog-SymLog	0.7628	0.7515	3
12	Jump Connection	AF4	Lin-G-GC-G-Log	0.9504	0.9497	16
13	Jump Connection	AF5	Lin-tanh-tanh-tanh-tanh	0.9466	0.9461	29
14	Jump Connection	AF6	Lin-GC-GC-GC-Log	0.9640	0.9639	26
15	Jump Connection	AF7	Lin-Sin-Sin-Sin-Sin	0.9514	0.9497	20
16	Jump Connection	AF8	Lin-tanh15-tanh15-tanh15-tanh	0.6696	0.3665	2

Table 6.49. Summary of results HC prediction of an E-WVU-5 Peak. The ANN was trained on the same test schedule.

6.9.2. Predicting Other Test Schedules

In HC predictions for two of the test schedules case 12 (Jump connection, Lin-G-GC-G-Log) turns the best results, and for three others it is among the best five predictors. Hence this network was the choice for training the combined test schedules.

Combined Test Schedules Predicts	Jump Connection, Lin-G-GC-G-Log	
	r^2	R^2
E-CSHVR	0.8221	0.7637
ETC	0.3880	-0.1623
FTP	0.4519	0.1587
E-Highway	0.2862	0.2389
E-WVU- 5 Peak	0.8089	0.7365
Average	0.5514	0.3471

Table 6.50. Results for prediction of HC emissions by using a Jump Connection network with activation functions of both Gaussian and Gaussian Complement. The ANN was trained on all the transient test schedules combined and predicting each test schedules individually.

The following graph presents the second by second predictions of this Jump Connection network with Gaussian and Gaussian Complement activation functions for the HC emissions of an E-CSHVR test schedule. Graphs for other four test schedules are available in Appendix A (Figure 9.69 through Figure 9.72).

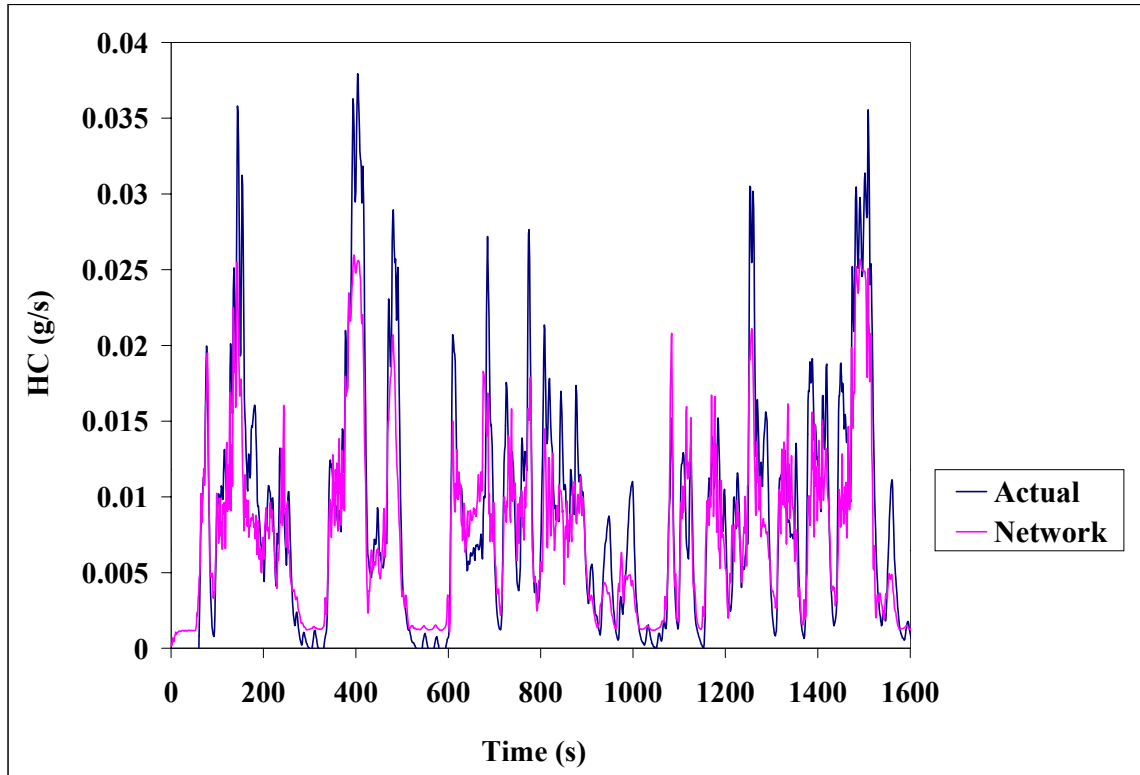


Figure 6.6. Second by second prediction of HC in an E-CSHVR test schedule by a Jump Connection network of both Gaussian and Gaussian Complement activation functions. The ANN was trained on all the transient test schedules combined.

6.9.2.1. GMDH

As in previous emissions predictions, in the last approach in generalizing the correlation between engine variables and emissions, the Group Method of Data Handling (GMDH) networks were employed. The lengthy final formula with medium nonlinearity is presented in Appendix B. The following table shows the results of predictions made by this network.

Combined Test Schedules Predicts	GMDH	
	r^2	R^2
E-CSHVR	0.7910	0.6952
ETC	0.4194	-0.2729
FTP	0.4047	-0.3272
E-Highway	0.2712	0.2131
E-WVU- 5 Peak	0.7445	0.6051
Average	0.5262	0.1827

Table 6.51. Results of HC predictions obtained using a GMDH. The ANN was trained on all the transient test schedules combined and predicting each test schedules individually.

Similar to NO_x and CO it is observed that GMDH has an average less accurate prediction compared to the Jump Connection network and this is quite expected from the characterization of this network. Yet, it is noticeable that in both cases the best results are obtained in the case of E-CSHVR and the worse results are for the E-Highway. This might emphasize that both networks lack the right input pattern associated with HC emissions and similar to PM emissions, in author's belief correct inputs are yet to be discovered.

6.9.3. Deduction

Considering Table 6.50 and the second by second predictions graphs, it is evident that the ANN under-predict the HC emissions in all cases except ETC and FTP. E-CSHVR and E-WVU-5 Peak are the ones that are predicted with a higher range of accuracy. It is noticeable in all of the five second by second graphs of HC predictions that there is a time delay in network predictions. This unforeseen time misalignment is evident, even though the issue of time delay between engine variables and emission measurements was addressed prior to introduction of input patterns to the network as mentioned in "Time Delay" section. As an attempt to figure out how much this time delay had affected the r^2 values, another time shifting was performed to align actual and network predictions, however this did not improve the r^2 considerably, since the limitations of predictions are originated from other bases. For instance 11 second of time shifting in the FTP cycle network predictions resulted in an improvement in r^2 value from 0.4519 to 0.5251.

The Jump Connection network of choice with both Gaussian and Gaussian Complement activation functions offers its best performance in prediction of E-CSHVR emissions, while trained on a database including a variety of engine operation conditions. As in CO predictions, the complicated formation mechanism of HC that has been an obstacle for other modeling attempts as well, is the major dilemma here. However, if this work is meant to approach a superb network architecture and characterization, Jump Connection with Gaussian and Gaussian Complement activation functions seems promising considering its relatively higher accuracy.

Anyhow not only for HC predictions, but in all emissions predictions, the possibility of employing time-delay ANN with added competence in dealing with time delay issues seems a reasonable recommendation for future efforts.

7. CONCLUSIONS AND RECOMMENDATIONS

Among the examined combinations of learning criteria in different architectures of BP design, a set of 0.05, 0.05 and 0.3 for learning rate, momentum and initial weights gave the best-averaged accuracy for NO_x predictions of the examined transient test schedules. Also considering the computational time to be cheap, the trade off between time and accuracy by choosing this learning rate and momentum was justified.

Between all of the architectures applied for the NO_x predictions of the five transient schedules, the 3-layer BP networks were the most accurate, in self-predictions. A 3-Layer network with tanh15 activation function was the best or the second best in the entire considered transient test schedules. While training the network on a database consisted of all test schedules combined, a 3-Layer network with Gaussian Complement function in hidden layers, showed to give the best results. Overall 3-Layer and Jump Connection networks with just Gaussian or Gaussian Complement functions or both of them combined, had outstanding performance. Although GRNN offered good capabilities in self-predictions and are highly recommended for particular cases of self-predicting test schedules, they did not suite the data base training. In 3 of the test schedules and also the combined database, torque and its first derivative over time had the highest weighting factors in the trained networks. Speed was among the first five in the descending order of weights and this seems as a verification of relation between NO_x and power. In the database training, the GMDH also gave close results to those of the 3-layer Gaussian network

Relatively poor predictions of TEOM and consequently PM, was implied through the almost equal final weightings of the network, suggesting that all inputs were treated equally and most likely were not sufficiently informative to the network. However, speed was still among the first five weightings. Yet, the superior performance of Gaussian and Gaussian Complement activation functions was evident. The author believes that the negative values of TEOM measurements associated with the adsorption of moisture and volatile compounds are the main source of information lack to the network. Besides, the

complicated nature of PM formation does not seem to be completely reflected in the choice of inputs applied in the present modeling attempt.

CO predictions also enjoyed the best result by implementation of a Gaussian Complement Jump Connection network. Speed and torque were among the first three weightings. The performance of tanh, Gaussian and Gaussian Complement functions were noticeable in CO predictions.

CO₂ predictions were overall successful as NO_x predictions. Gaussian Complement, Gaussian, tanh, and Sin were the most suitable activation functions. Speed and torque had the highest weightings, indication of relation with power and sufficient information for good prediction.

In HC predictions a 3-Layer network with both Gaussian and Gaussian Complement functions outperformed the other networks. Speed and torque were among the highest weightings, while Gaussian and Gaussian Complement functions were the best activation functions embedded in both examined architectures.

Overall 3-Layer and Jump Connection with three hidden layers are the recommended architectures and radial bases functions are the recommended activation functions for emissions prediction tasks. The main feature of Gaussian function is mapping the mid range data to high values and the Gaussian Complement magnifies the extreme points of the data. This is opposed to other used functions, which are constantly increasing, mapping low to low and high to high. Hyperbolic tangents can also offer good capabilities in some cases. It is to be mentioned that the scope of this research was limited to certain inputs of speed, torque and their higher order derivatives over time. Considering the exceptional ability of ANN in black-box modeling, further attempts in the future should be focused on the search for the right inputs. That would be an indispensable urge in particular for CO, HC and PM emissions. It is apparent that for these emissions the formation mechanism is definitely not confined to be only expressed in terms of torque, speed and their variations over time. As discussed earlier, there are

certain prefaces to a successful ANN application. For emissions prediction task, choosing the most proper parameters as inputs among all the available obtained data is an imperative step.

The number of nodes in each layer of a decided architecture could be another parameter to optimize. In the early stages of this research, while deciding to define a framework for optimization process, this also became considerable to the author, but changing the number of nodes did not seem to improve the results drastically. Therefore this factor was crossed out and replaced by the default number calculated according to the number of patterns introduced to the network. However there seems to be interest for the future researchers to seek for an optimal number of nodes in the recommended architectures here.

Further improvements in predictions could be achieved by alternating the method of propagation of the inputs to the network. According to the NeuroShell2 manual, “A back-propagation network with standard connections responds to a given input pattern with exactly the same output pattern every time the input pattern is presented. A recurrent network may respond to the same input pattern differently at different times, depending upon the patterns that have been presented as inputs just previously. Thus, the sequence of the patterns is as important as the input pattern itself” [79]. This could be a key to solving the dilemma of consecutive emissions interferences after being dispersed in the dilution tunnel. However, the right choice of test patterns is crucial to get reliable results from this architecture. This issue is also quite challenging for the future researchers particularly, while using a combined test schedule to train a network as a database.

There are further discussions about possible improvements of training data, prior to their implementation to the network; “Data Mining, Knowledge Discovery and Data-Knowledge Fusion” is a suggested term by Mohagheh [90]; the main idea is to extract the great deal of knowledge enveloped in the data and eliminate unnecessary confusing part containing noise and human/device errors. This is also somehow close to the concept of back transformation in the emissions data introduced by Clark et al. [68] as an attempt

to reach the instantaneous engine out emissions, before the dilution tunnel. There is obviously room in the future studies for developments in training data, leading to the perfection of pattern recognition in emissions application.

8. REFERENCES

1. <http://www.arb.ca.gov/html/brochure/history.htm>, accessed 7/10/02.
2. <http://www.pirg.org/reports/enviro/dangerousdiesel/index.html>, accessed 7/10/02.
3. <http://naftp.nrcce.wvu.edu/>, accessed 2/20/02.
4. Brauer, M., Brook, J. R., “Ozone Personal Exposure and Health Effects for Selected Group Residing in the Fraser Valley”, *Atmospheric Environment*, Vol. 31, No. 14, 1997, pp. 2113-2121.
5. <http://www.royal.okanagan.bc.ca/>, accessed 2/20/02.
6. <http://www.epa.gov/otaq/regs/hd-hwy/2000frm/f00026.htm>, accessed 7/10/02.
7. <http://www.dieselnet.com>, accessed 9/9/02.
8. Ramamurthy, R., Clark, N. N., Atkinson, C. M., Lyons, D. W., “Models for Predicting Transient Heavy Duty Vehicle Emissions”, SAE paper 982652.
9. Bane, B. R., “A Comparison of Steady State and Transient Emissions from a Heavy-Duty Diesel Engine”, Master's Thesis, West Virginia University, 2002.
10. Farlow, S. J., “Self-Organizing Methods in Modeling”, Marcel Dekker, Inc., New York and Basel, NY, 1984.
11. Krijnsen, H., van Kooten, W., Calis, H., Verbeek, R., van den Bleek, C., “Evaluation of an Artificial Neural Network for NO_x Emission Prediction from a Transient Diesel Engine as a Base for NO_x Control”, *The Canadian Journal of Chemical Engineering*, Vol. 78, April 2000, pp. 408-417.
12. Bishop, C. M., “Neural Networks for Pattern Recognition”, Oxford University Press, Inc., New York, NY, 1995.
13. Duda, R. O., Hart, P. E., Stork, D. G., “Pattern Classification”, John Wiley & Sons, Inc., New York, NY, 2001.
14. Rakopoulos, C., Hountalas, D., “Development and Validation of a 3-D Multi-Zone Combustion Model for Prediction of DI Diesel Engines Performance and Pollutants Emissions”, SAE paper 981021.
15. Bazari, Z., “Diesel Exhaust Emissions Prediction under Transient Operating Conditions”, SAE paper 940666.

16. Pitsch, H., Barths, H., Peters, N., "Three-Dimensional Modeling of NO_x and Soot Formation in DI-Diesel Engines Using Detailed Chemistry Based on the Interactive Flamelet Approach", SAE paper 962057.
17. Clark, N. N., Tehranian, A., Jarrett, R. P., Nine, R. D., "Translation of Distance Specific Emissions Rates between Different Heavy Duty Vehicle Chassis Test Schedules", SAE paper 2002-01-1754.
18. Duyar, A., Merrill, W., "A Failure Diagnosis System Based on a Neural Network Classifier for the Space Shuttle Main Engine", Proceedings of the 29th IEEE Conference on Decision and Control, Part 4 (of 6), Honolulu, HI, December 1990, pp. 2391-2400.
19. Shukla, K. K., "Neurocomputer Based Learning Controller for Critical Industrial Applications", Proceedings of the IEEE/IAS International Conference on Industrial Automation and Control Conference, Hyderabad, India, January 1995, pp. 43-48.
20. Arsie, I., Pianese, C., Rizzo, R., Serra, G., "Development and Validation of a Model for Mechanical Efficiency in a Spark-Ignition Engine", SAE paper 1999-01-0905.
21. Schoeggl, P., Ramschak, E., "Vehicle Drivability Assessment Using Neuralnetworks for Development, Calibration and Quality Tests", SAE paper 2000-01-072.
22. Winsel, T., Ayeb, M., Lichtenthaler, D., Theuerkauf, H.J., "A Neural Estimator for Cylinder Pressure and Engine Torque", SAE paper 1999-01-1165.
23. Moon, K., Osborne, M., Kuykendall, D., Poirier, W., "Modeling of Human Response from Vehicle Performance Characteristics Using Artificial Neural Networks", SAE paper 2002-01-1570.
24. Shayler, P., Goodman, M., Ma, T., "Transient Air/Fuel Ratio Control of an S.I. Engine Using Neural Networks", SAE paper 960326.
25. Weidong, Z., Guojun, C., "Neural Feed Forward Compensation for Closed Cycle Diesel System", SAE paper 2002-01-0478.

26. Atkinson, C., Long, T., Hanzevack, E., "Virtual Sensing; a Neural Network-Based Intelligent Performance and Emissions Prediction System for Onboard Diagnostics and Engine Control", SAE paper 980516.
27. Quenou Gamo, S., Ouladsine, M., Rachid, A., "Diesel Engine Exhaust Emission Modeling Using Artificial Neural Networks", SAE paper 1999-01-1163.
28. Traver, M., Atkinson, R., Atkinson, C., "Neural Network-Based Diesel Engine Emissions Prediction Using In-Cylinder Combustion Pressure", SAE paper 1999-01-1532.
29. Thompson, G. J., Atkinson, C. M., Clark, N. N., Long, T. W., Hanzevack, E., "Neural Network Modeling of the Emissions and Performance of a Heavy-Duty Diesel Engine", Proc. Inst. Mech. Engrs., Vol. 214, Part D, 2000, pp. 111-126.
30. Khatri, D., Kumar, S., "Feasibility Study on Neural Network Approach in Engine Management System in S.I. Engine", SAE paper 2000-01-1426.
31. Steyskal, M., Olsen, D., Willson, B., "Development of PEMS Models for Predicting NO_x Emissions from Large Bore Natural Gas Engines", SAE paper 2001-01-1914.
32. Clark, N. N., Conley, J., Jarrett, R. P., Nennelli, A., Toth-Nagy, C., "Emissions Modeling of Heavy-Duty Conventional and Hybrid Electric Vehicles", SAE paper 2001-01-3675.
33. Hentschel, R., Cernat, R., Varchmin, J., "In-Car Modeling of Emissions with Dynamic Artificial Neural Networks", SAE paper 2001-01-3383.
34. Desantes, J., Lopez, J., Garcia, J., Hernandez, L., "Application of Neural Networks for Prediction and Optimization of Exhaust Emissions in a H.D. Diesel Engine", SAE paper 2002-01-1144.
35. http://www.dse.doc.ic.ac.uk/~nd/surprise_96/journal/vol1/cs11/article1.html, accessed 4/29/02.
36. DARP Neural Network Study, AFCEA International Press, 1988, p. 60, ISBN 0-916159-17-5.
37. Haykin, S., "Neural networks: A Comprehensive Foundation", Macmillan College Publishing Company, Inc., New York, NY, 1994.

38. "Encyclopedia Britannica", <http://search.eb.com/eb/article?eu=139070>, accessed 06/03/02.
39. "The Internet Encyclopedia of Philosophy",
<http://www.utm.edu/research/iep/a/artintel.htm>, accessed 6/19/02.
40. "Techencyclopedia", <http://www.techweb.com/encyclopedia>, accessed 7/19/02.
41. Honavar, V., Uhr, L., "Artificial Intelligence and Neural Networks: Steps Toward Principled Integration", Academic Press, New York, NY, 1994.
42. Wilkes, A. L., Wade, N. J., "Bain on Neural Networks", *Brain and Cognition*, Vol.33, 1997, pp. 295-305.
43. <http://www.artsci.wustl.edu/~philos/MindDict/connectionismhistory.html>, accessed 6/21/02.
44. http://www.neurosciences.com/Technologies/nn_history.htm, accessed 6/20/02.
45. Hebb, D. O., "The Organization of Behavior: A Neuropsychological Theory", John Wiley & Sons, Inc., New York, NY, 1949.
46. Farley, B. G., Clark, W. A., "Simulations of Self-Organizing Systems by Digital Computer", *IRE Transaction of Information Theory*, IT-Vol. 4, No. 3, September 1954, pp. 76-84.
47. Rochester, N., Holland, J. H., Haibt L. H., Duda, W. L., "Tests on a Cell Assembly Theory of the Action of the Brain Using a Large Digital Computer", *IRE Transaction of Information Theory* IT-Vol. 2, 1956, pp. 80-93.
48. Olmsted, D. D., http://www.neurocomputing.org/History/body_history.html, accessed 6/21/02.
49. Rosenblatt, F., "Principles of Neurodynamics: Perceptions and the Theory of Brain Mechanisms", Spartan Books, Washington, D.C., 1962.
50. Crick, F., "The Astonishing Hypothesis: The Scientific Search for the Soul", Charles Scribner's Sons, New York, NY, 1994.
51. Kurzweil, R., "The Age of Intelligent Machines", MIT Press, Boston, MA, 1990.
52. NeuroNet Research Committee–Gallinari., P., Lansner, A., Babloyantz, A., Oja, E., Eckmiller, R., Kolas, S., Clarkson, T., Plumbley, M., Taylor, J., Garner, T., "A European Starategy for Neural Networks", NeuroNet, 1995. Also available on <http://www-connex.lip6.fr/Articles/E-S-NN.ps.gz>, accessed 6/24/02.

53. Fausett, L., "Fundamentals of Neural Networks: Architectures, Algorithms, and Applications", Prentice-Hal, Inc., Prentice Hall, NJ, 1994.
54. <http://hem.hj.se/~de96klda/NeuralNetworks.htm>, accessed 7/26/2002.
55. Freeman, J., Skapura, D., "Neural Networks: Algorithms, Applications, and Programming Techniques", Addison-Wesley publishing company, Inc., Reading, MA, 1991.
56. Nine, R. D., Personal Communication, West Virginia University, MAE Department, 4/30/03.
57. Clark, N. N., Daley, J. J., Nine, R. D., Atkinson, C. M., "Application of the New City-Suburban Heavy Vehicle Route (CSHVR) to Truck Emissions Characterizations", SAE paper 99011467.
58. <http://dieselnet.com/standards/cycles/>, accessed 12/9/02.
59. <http://www.epa.gov/epahome/cfr40toc.htm>, accessed 12/12/02.
60. <http://www.detroitdiesel.com/public/corp/pr031301.asp>, accessed 12/12/02.
61. Rosemount Analytical Inc., Model 402 Hydrocarbon Analyzer, Instruction Manual, April 1993.
62. Rosemount Analytical Inc., Model 955 NO/NO_x Analyzer, Instruction Manual, Literature No. 748190-D, February 1992.
63. Horiba Instrument Incorporated, AIA-210/220 Infrared Analyzer, Instruction Manual, Horiba Manual No. 091215, 1995.
64. Rosemount Analytical Inc., Model 868 Non-Dispersive Infrared Analyzer, Instruction Manual, Instruction 015-748003-J, March 1991.
65. Rupprecht & Patashnick Co., Inc., TEOM, Series 1105, Diesel Particulate Mass Monitor, Operating Manual, September 1997.
66. Olson, J., http://api.mech.ubc.ca/research/turb/turb_jfm_html/node2.html, revised 07/29/00.
67. Clark, N. N., Jarrett, R. P., Atkinson, C. M., "Field Measurements of Particulate Matter Emissions and Exhaust Opacity from Heavy Duty Vehicles", Proceedings of the Air and Waste Management Association, PM 2.5, A Fine Particle Standard Conference, Long Beach, CA, January 28-30, 1998.

68. Aravelli, A., Clark, N., Tehranian, A., "Backward-Transformation Techniques for NO_x Emission Prediction from Heavy-Duty Diesel Engines", Paper in preparation, August 2002.
69. Levenspiel, O., "Chemical Reaction Engineering", 3rd Edition, John Wiley & Sons, Inc., New York, NY, 1999.
70. Jarrett, R. P., "Evaluation of Opacity Particulate Matter, and Carbon Monoxide From Heavy Duty Diesel Transient Chassis Tests", Master's Thesis, West Virginia University, 2000.
71. Ganesan, B., Clark, N. N., "Relationships between Instantaneous and Measured Emissions in Heavy-Duty Applications", SAE paper 2001-01-3536.
72. Nine, R. D., Clark, N. N., Norton, P., "Effects on Emissions of Multiple Driving Test Schedules Performed on Two Heavy-Duty Vehicles", SAE paper 2000-01-2818.
73. Jarrett, P. R., Clark, N. N., Gilber, M., Ramamurthy, R., "Evaluation and Correction of Moisture Adsorption and Desorption from a Tapered Element Oscillating Microbalance", Powder Technology, No. 119, 2001, pp. 215-228.
74. <http://mathworld.wolfram.com/>, accessed 11/17/02.
75. Spetch, D. F., "A General Regression Neural Network", IEEE Transaction on Neural Networks, Vol. 2, No. 6, November 1991, pp. 568-576.
76. Masters, T., "Advanced Algorithms for Neural Networks", John Wiley & Sons, Inc., New York, NY, 1995.
77. Caudill, M., "GRNN and Bear It", AI Expert, Vol. 8, No. 5, May 1993, pp. 28-33.
78. <http://www.gmdh.net/>, accessed 10/10/02.
79. NeuroShell2 User's Manual, Ward Systems Group, Inc., Fredrick, MD, 1996.
80. Hornik, K., Stinchcombe, M., White, H., "Multilayer Feedforward Networks Are Universal Approximators", Neural Networks, Vol. 2, No. 5, 1989, pp. 359-366.
81. Sarle, S. W., <ftp://ftp.sas.com/pub/neural/FAQ7.html>, modified 5/31/02.
82. Narendra, K., Parthasarathy, K., "Identification and Control of Dynamical Systems Using Neural Networks", IEEE Transaction s on Neural Network, Vol. 1 No. 1, March 1990, pp. 4-27.

83. Hagan, M. T., Demuth, H. B., Beale, M., "Neural Network Design", PWS publishing Co., Boston, MA, 1996.
84. Hartman, E. J., Keeler, J. D., Kowalski, J. M., "Layered Neural Networks with Gaussian Hidden Units as Universal Approximations", Neural Computation, Vol. 2, No. 2, pp. 210-215.
85. Ramamurthy, R., "Heavy Duty Emissions Inventory and Prediction", Master's Thesis, West Virginia University, 1999.
86. Kern, J., "Inventory and Prediction of Heavy-Duty Diesel Vehicle Emissions", Master's Thesis, West Virginia University, 2000.
87. Krijnsen, H., van Leeuwen, J., Bakker, R., van den Bleek, C., Calis, H., "Optimum NO_x Abatement in Diesel Exhaust Using Inferential Feed Forward Reductant Control", Fuel 80, 2001, pp. 1001–1008.
88. Jiang, Q., Van Gerpen, J. H., "Prediction of Diesel Engine Particulate Emission During Transient Cycles", SAE paper 920466.
89. Rao, V., "Development of an Exhaust Carbon Monoxide Emissions Model", SAE paper 961214.
90. Mohagheh, S., "E&P "Gets Smart" with Virtual Intelligence", American Oil & Gas Reporter Journal, Kansas, July 2001, pp. 77-83.

9. APPENDIX A: ADDED FIGURES

In this appendix, 3-D and 2-D graphs are presented to provide easier visual evaluation of activation function's influence on performance of the ANN. Also, second by second predictions are presented to visualize the final results of the optimized networks. Detailed explanation for each figure is given in the "Results" chapter.

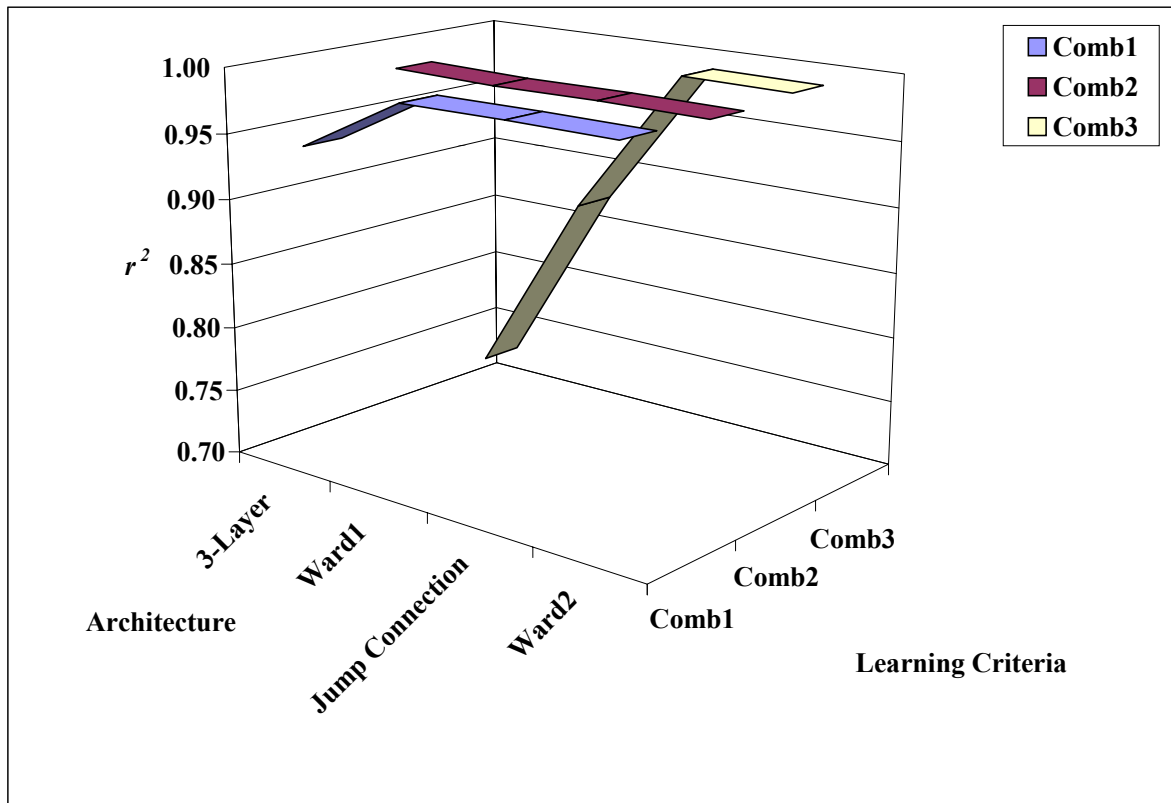


Figure 9.1. Variation of averaged r^2 of eight different cases for each architecture predicting NO_x emissions of an E-CSHVR in the plane of training criteria and architecture (see section 6.5.1.1).

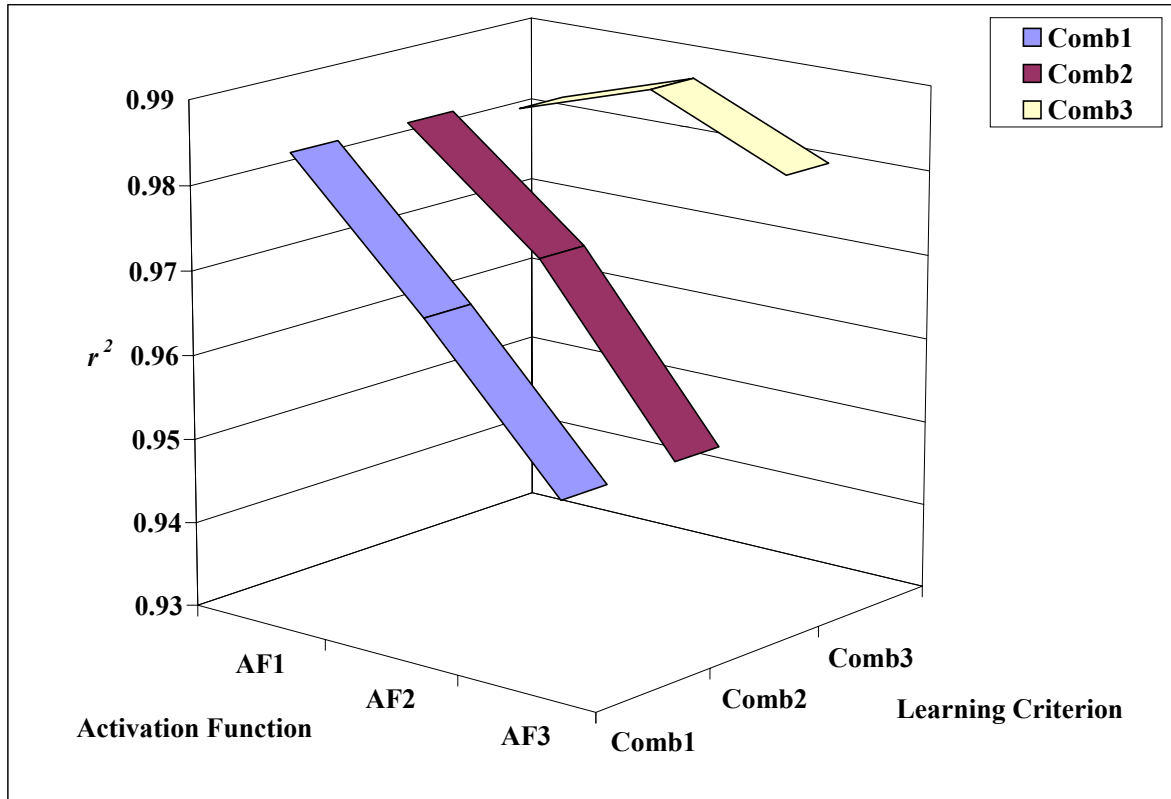


Figure 9.2. Variation of r^2 in the plane of training criterion and activation function for a GRNN architecture predicting NO_x emissions of an E-CSHVR (see section 6.5.1.1).

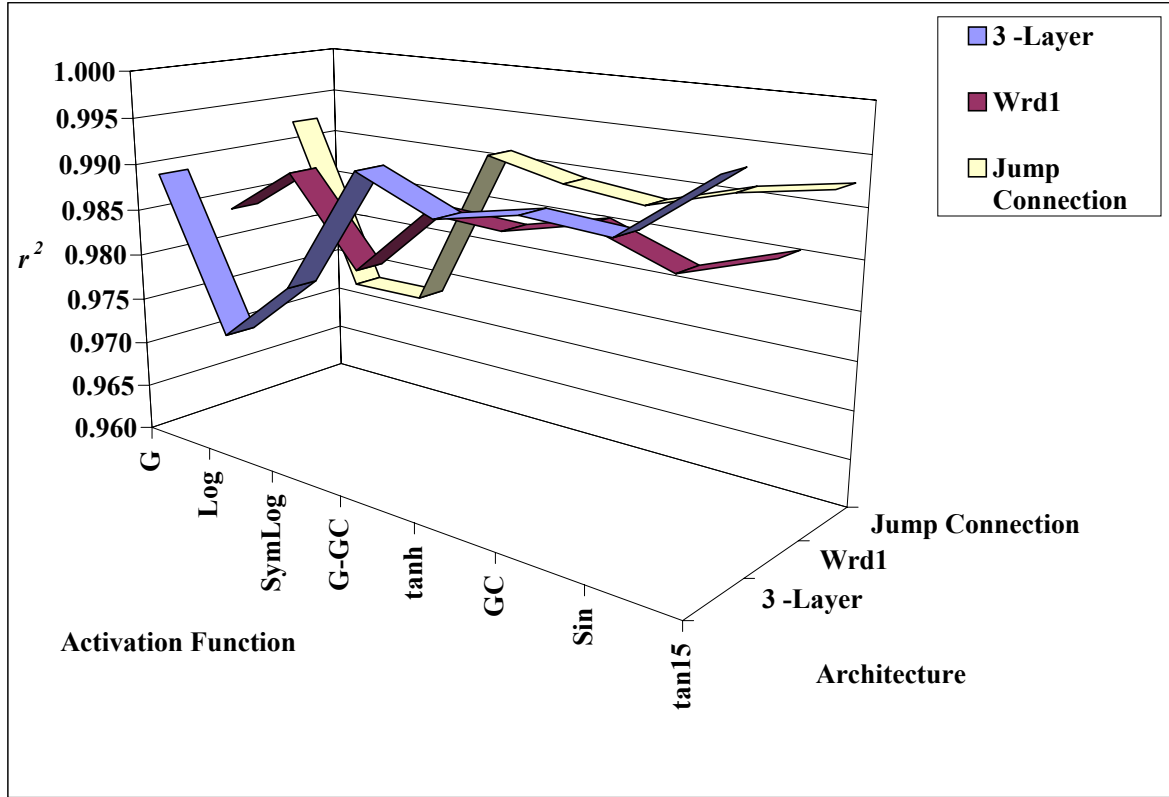


Figure 9.3. Variation of r^2 in the plane of activation function and architecture for the selected learning criteria Comb2. The ANN was trained on an E-CSHVR predicting NO_x emissions of the same test schedule. Cases are described in Table 6.9.

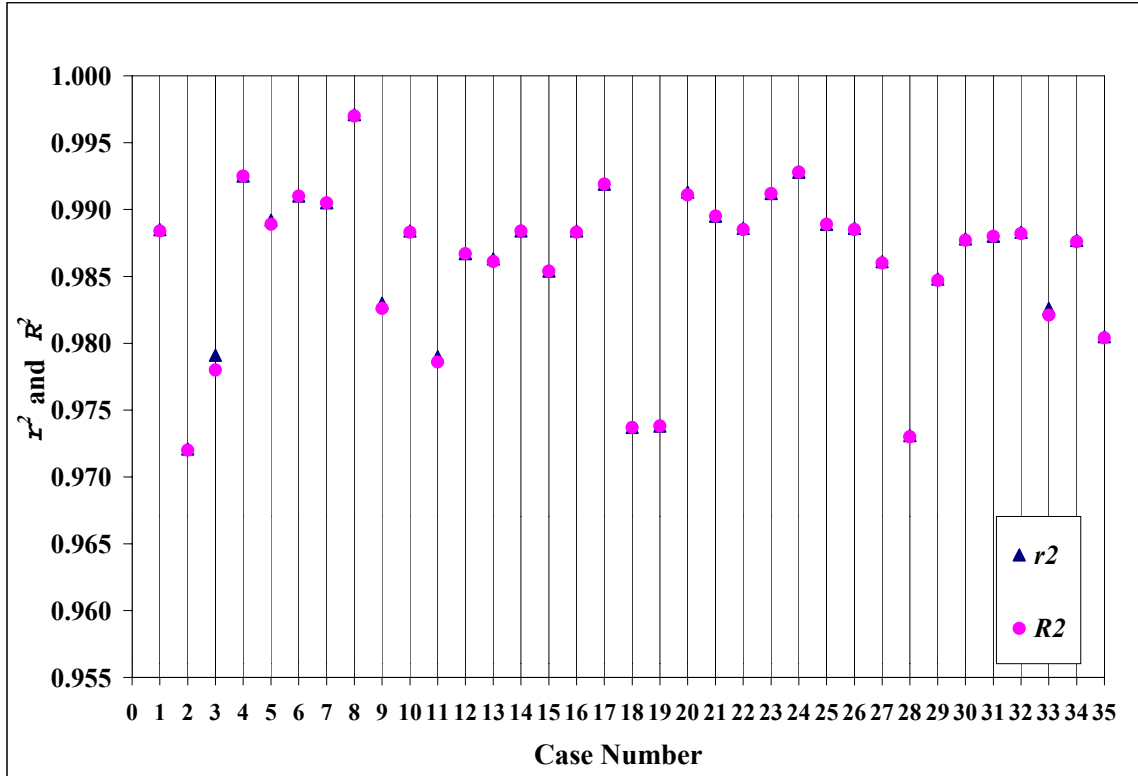


Figure 9.4. Variation of r^2 and R^2 with all cases for the selected learning criteria Comb2. The ANN was trained on an E-CSHVR predicting NO_x emissions of the same test schedule. Cases are described in Table 6.9.

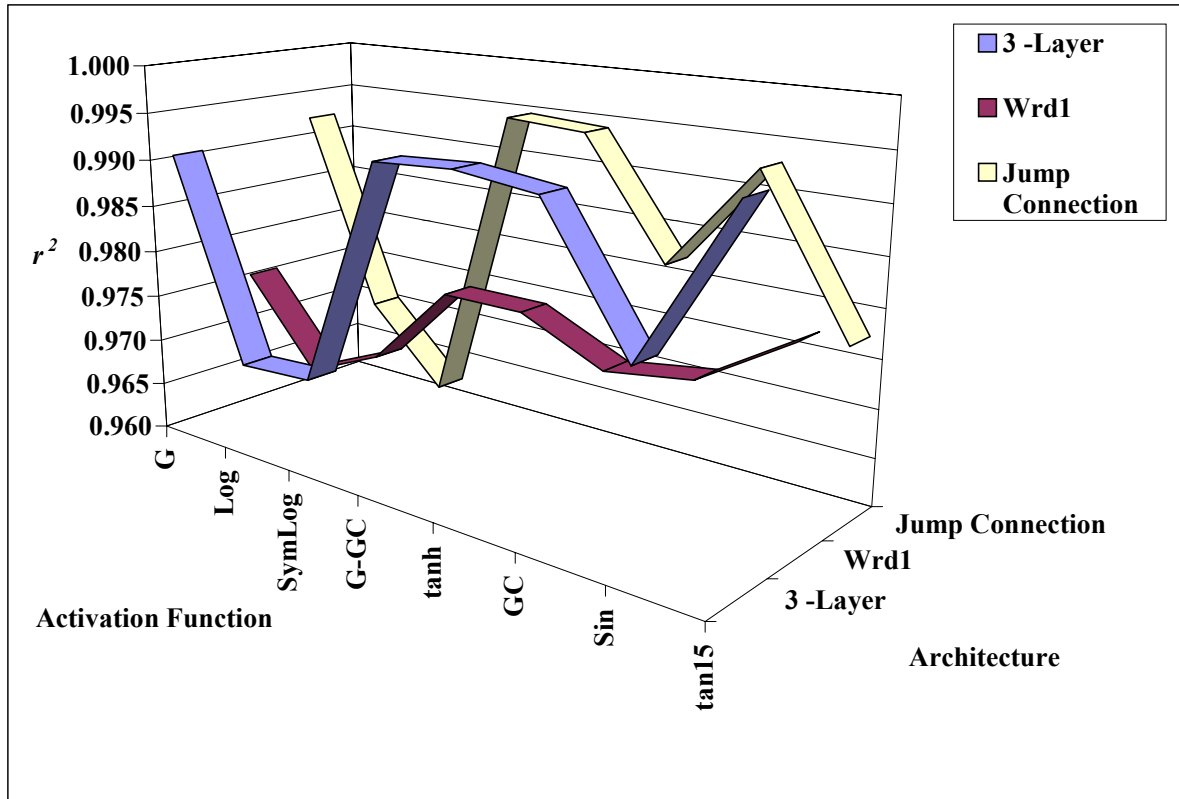


Figure 9.5. Variation of r^2 in the plane of activation function and architecture for the selected learning criteria Comb2. The ANN was trained on an ETC cycle predicting NO_x emissions of the same test schedule. Cases are described in Table 6.10.

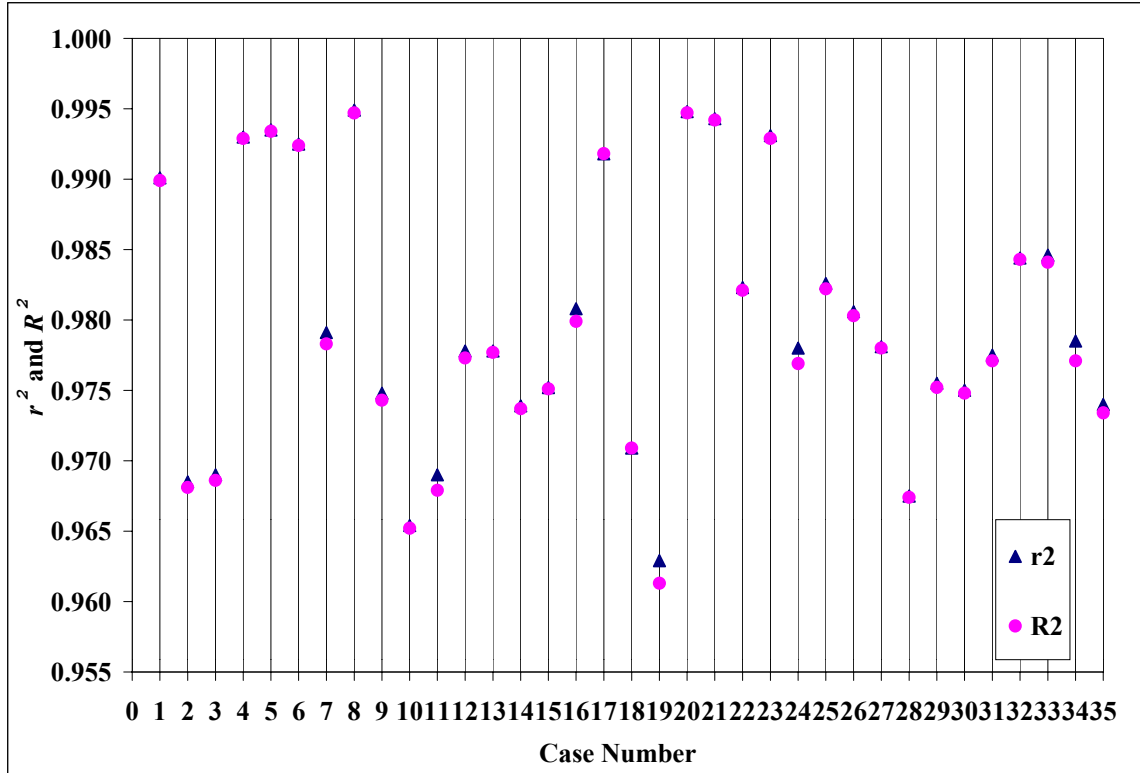


Figure 9.6. Variation of r^2 and R^2 with all cases for the selected learning criteria Comb2. The ANN was trained on an ETC predicting NO_x emissions of the same test schedule. Cases are described in Table 6.10.

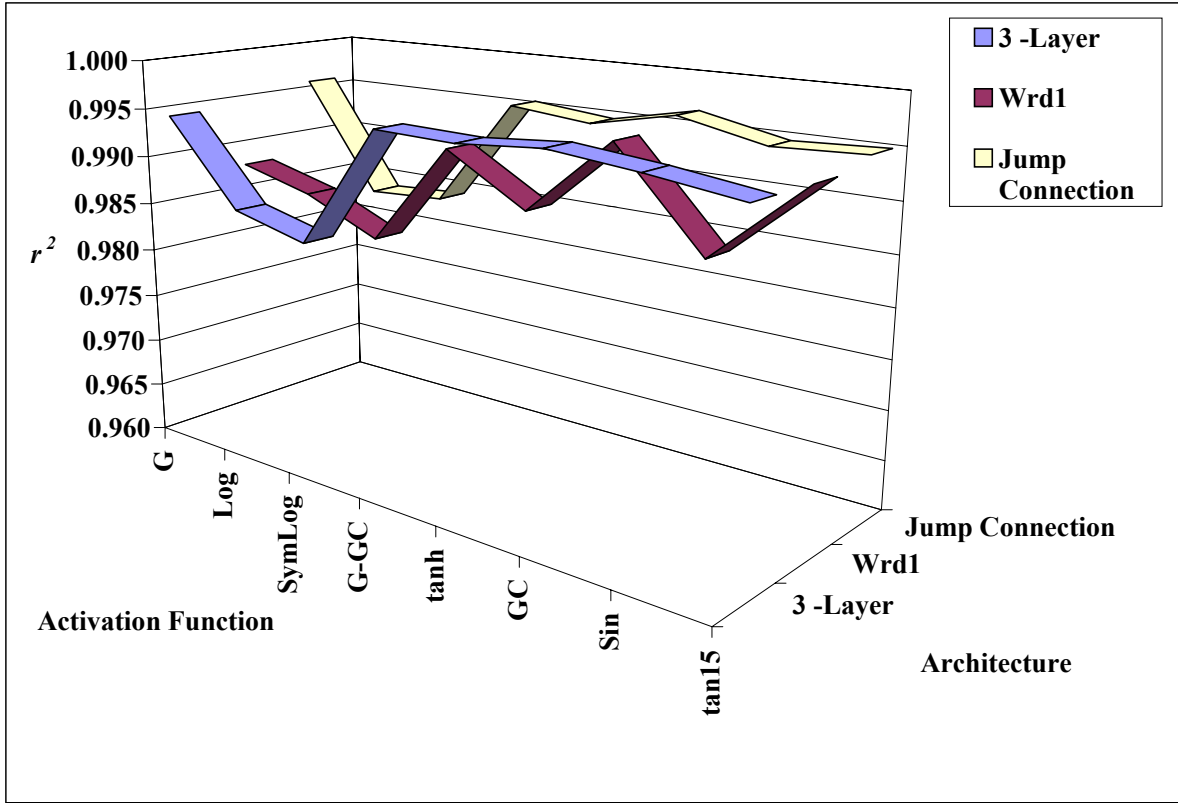


Figure 9.7. Variation of r^2 in the plane of activation function and architecture for the selected learning criteria Comb2. The ANN was trained on a FTP cycle predicting NO_x emissions of the same test schedules. Cases are described in Table 6.11.

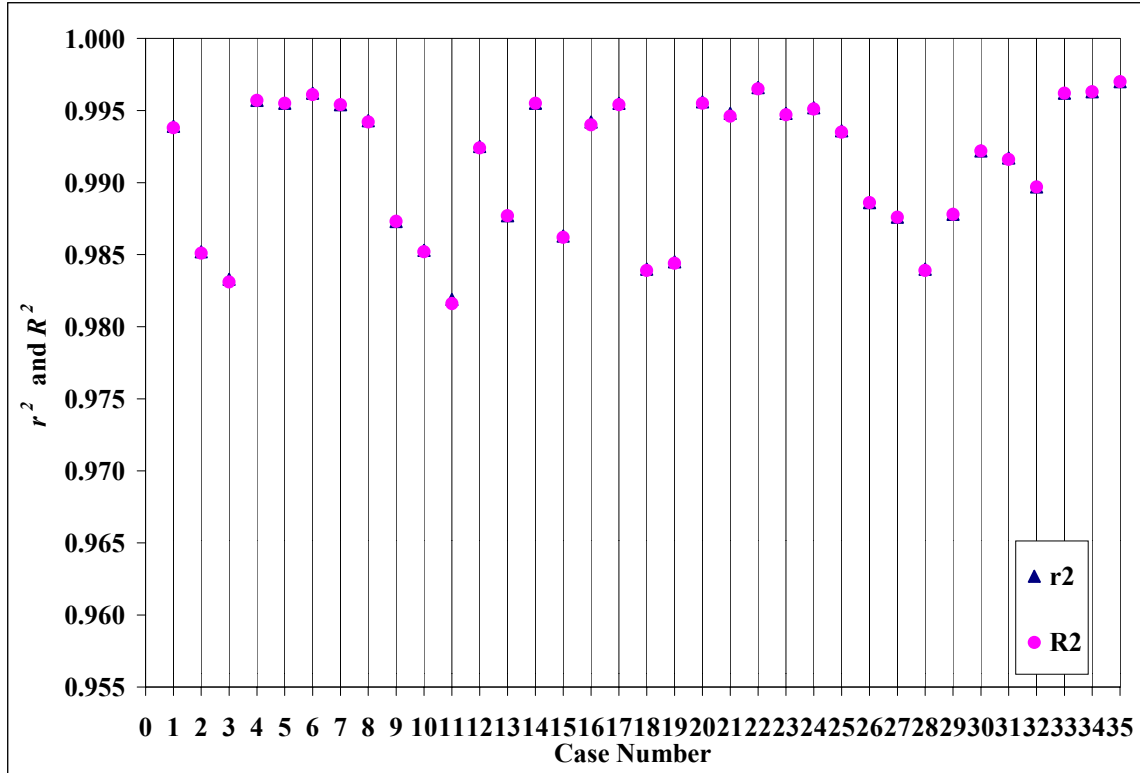


Figure 9.8. Variation of r^2 and R^2 with all cases for the selected learning criteria Comb2. The ANN was trained on a FTP predicting NO_x emissions of the same test schedule. Cases are described in Table 6.11.

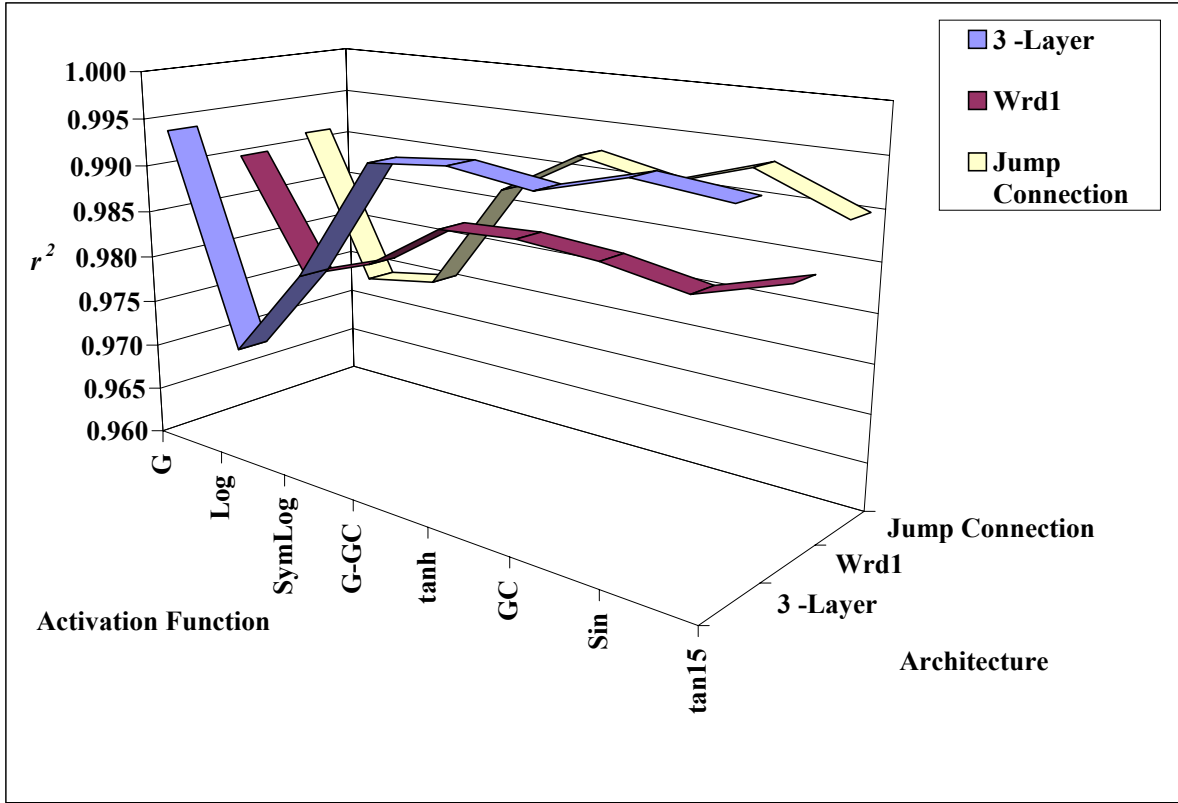


Figure 9.9. Variation of r^2 in the plane of activation function and architecture for the selected learning criteria Comb2. The ANN was trained on an E- Highway cycle predicting NO_x emissions the same test schedules. Cases are described in Table 6.12.

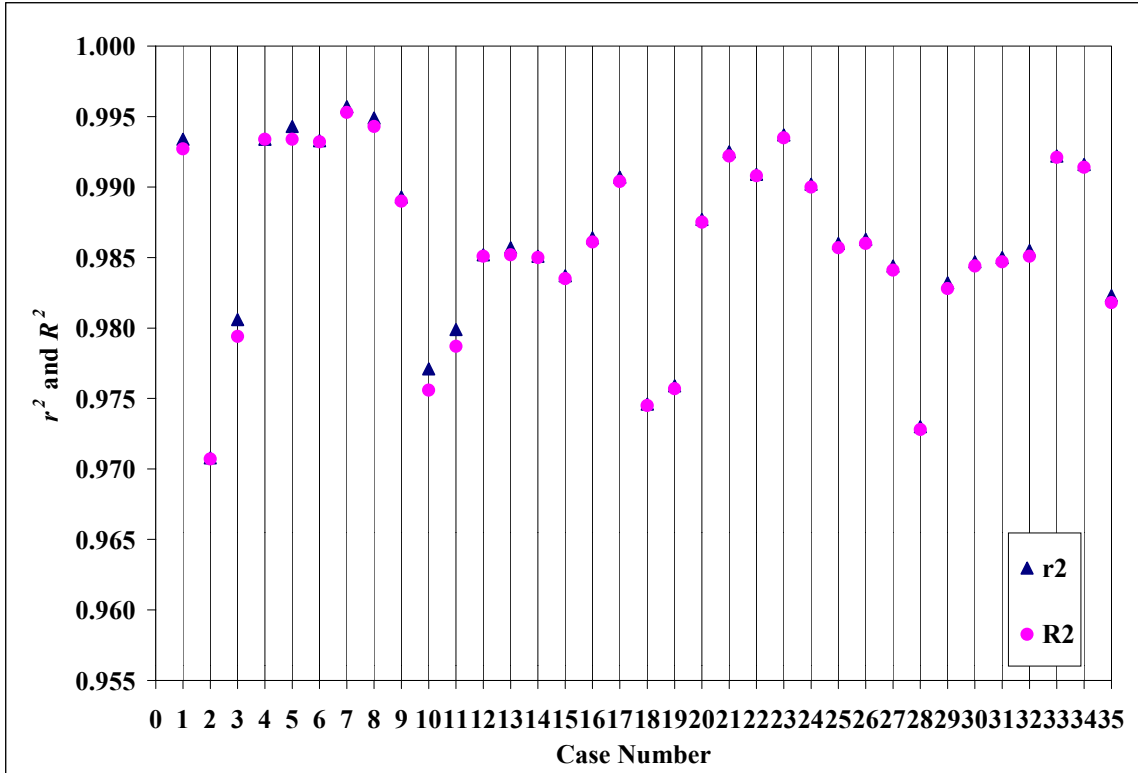


Figure 9.10. Variation of r^2 and R^2 with all cases for the selected learning criteria Comb2. The ANN was trained on an E-Highway predicting NO_x emissions of the same test schedule. Cases are described in Table 6.12.

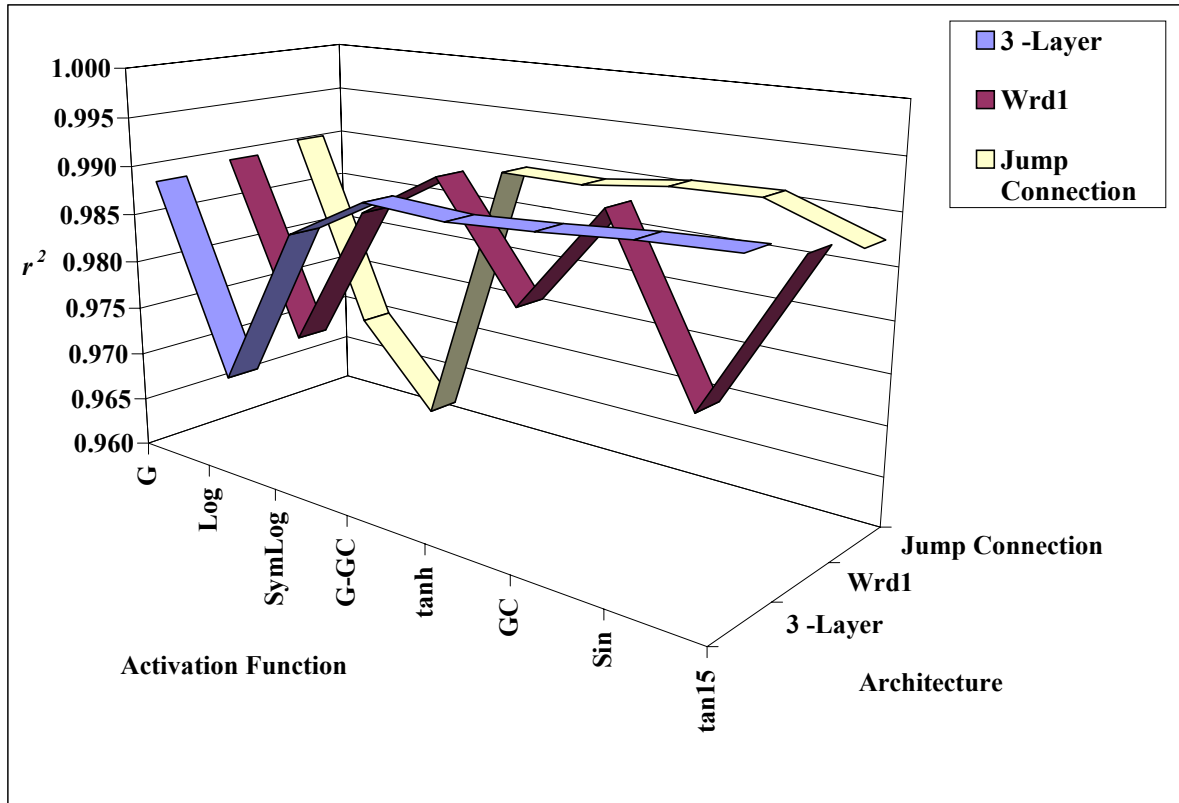


Figure 9.11. Variation of r^2 in the plane of activation function and architecture for the selected learning criteria Comb2. The ANN was trained on an E-WVU-5 Peak cycle predicting NO_x emissions of the same test schedules. Cases are described in Table 6.13.

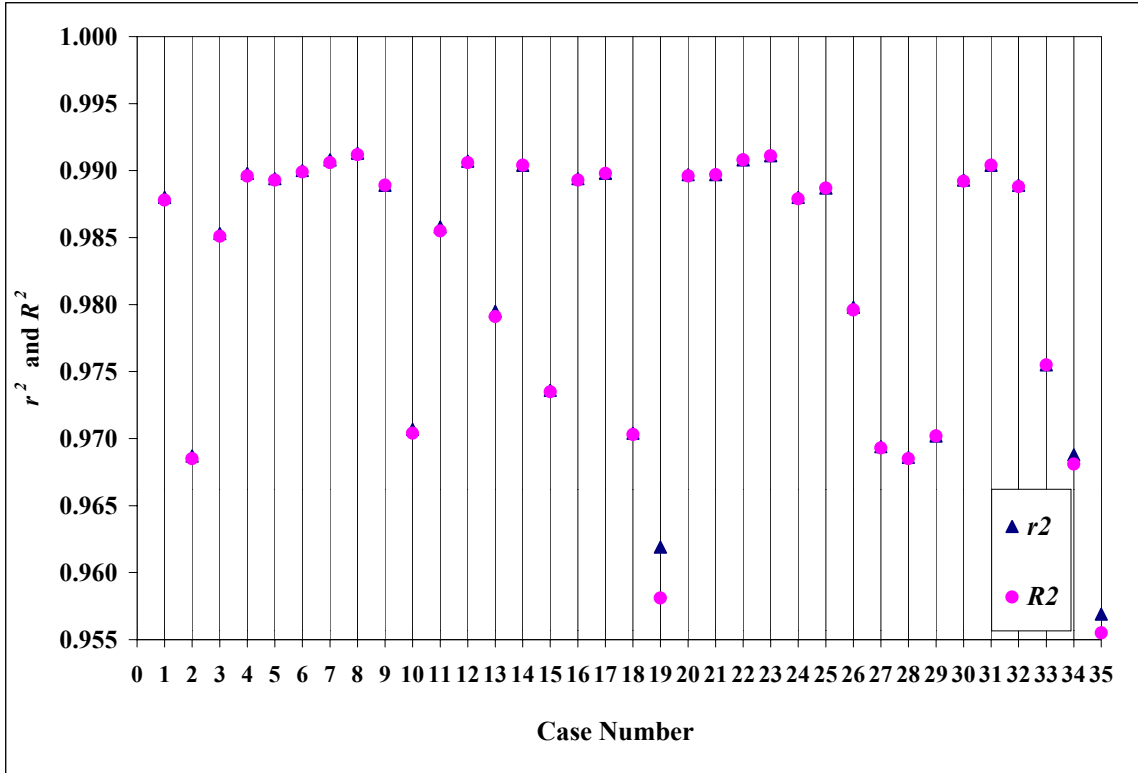


Figure 9.12. Variation of r^2 and R^2 with all cases for the selected learning criteria Comb2. The ANN was trained on an E-WVU- 5 Peak predicting NO_x emissions of the same test schedule. Cases are described in Table 6.13.

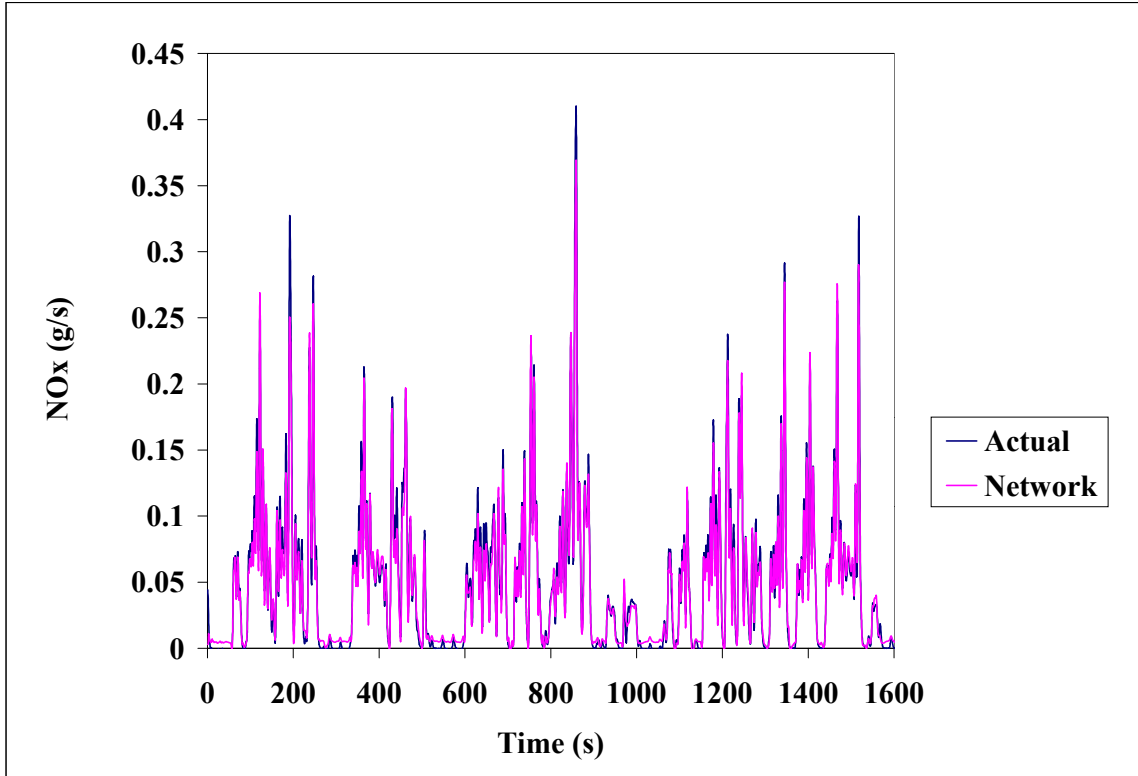


Figure 9.13. Second by second prediction of NO_x in an E-CSHVR test schedule by a 3-Layer network of Gaussian Complement activation function. The ANN was trained on all the transient test schedules combined.

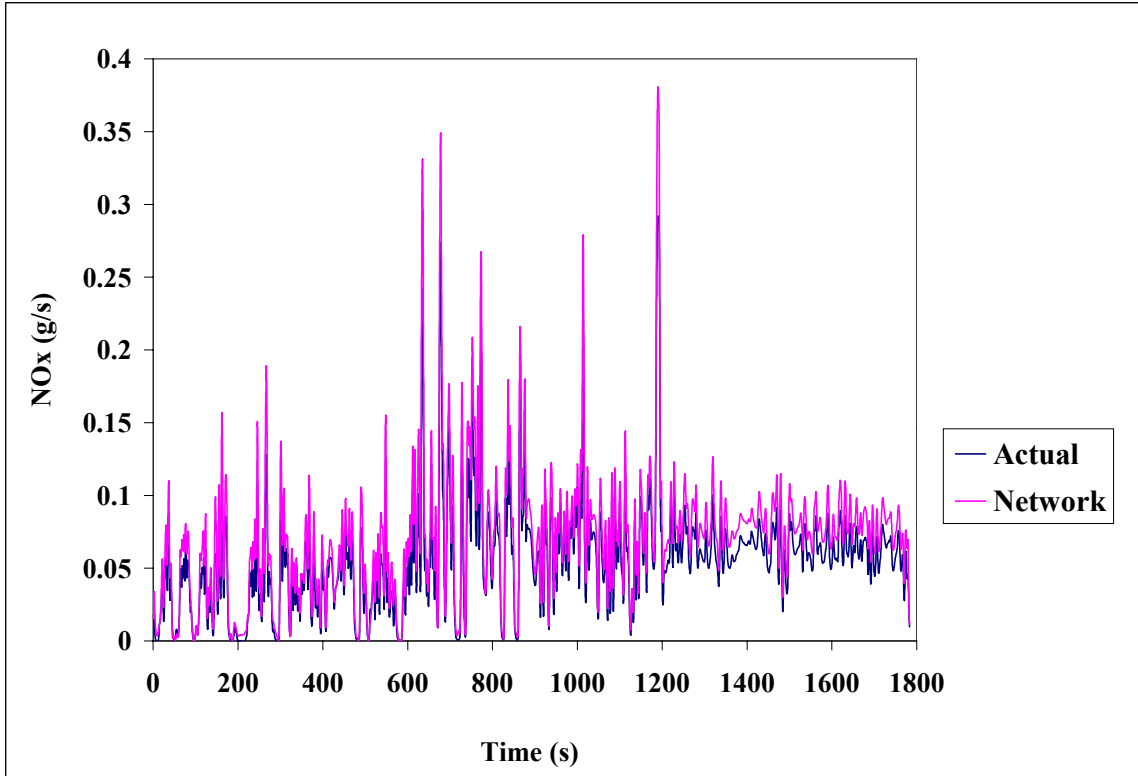


Figure 9.14. Second by second prediction of NO_x in an ETC test schedule by a 3-Layer network of Gaussian Complement activation function. The ANN was trained on all the transient test schedules combined.

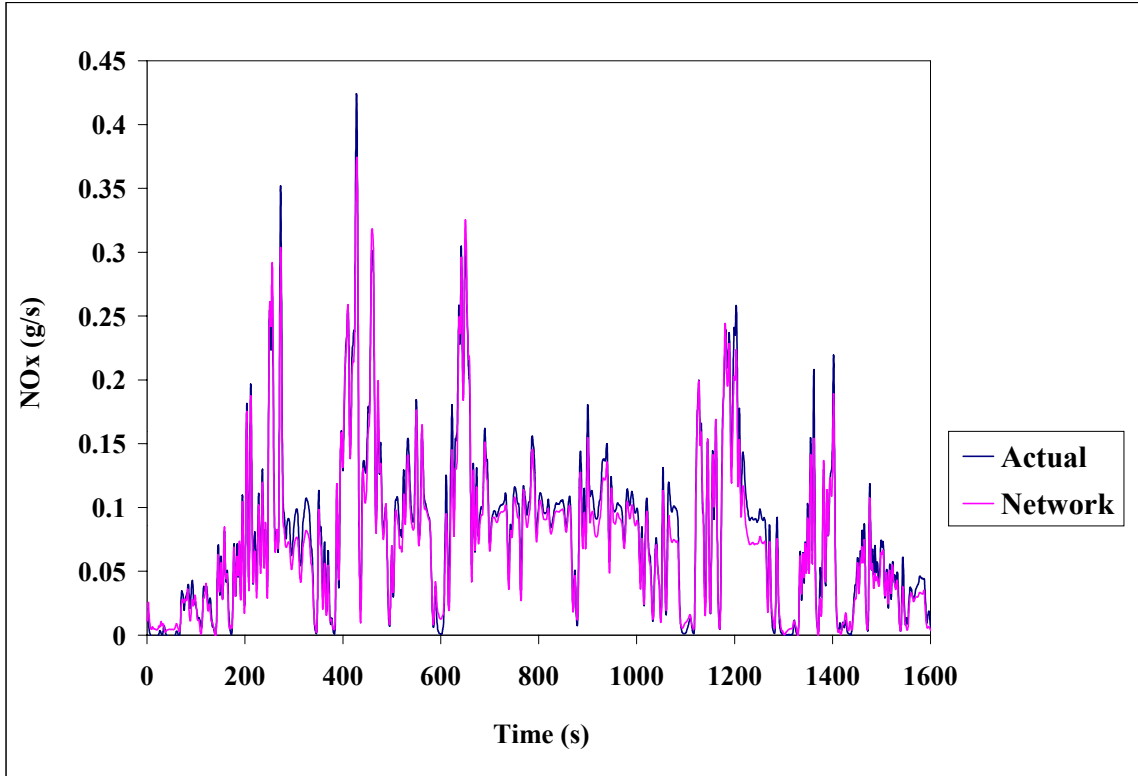


Figure 9.15. Second by second prediction of NO_x in an E-Highway test schedule by a 3-Layer network of Gaussian Complement activation function. The ANN was trained on all the transient test schedules combined.

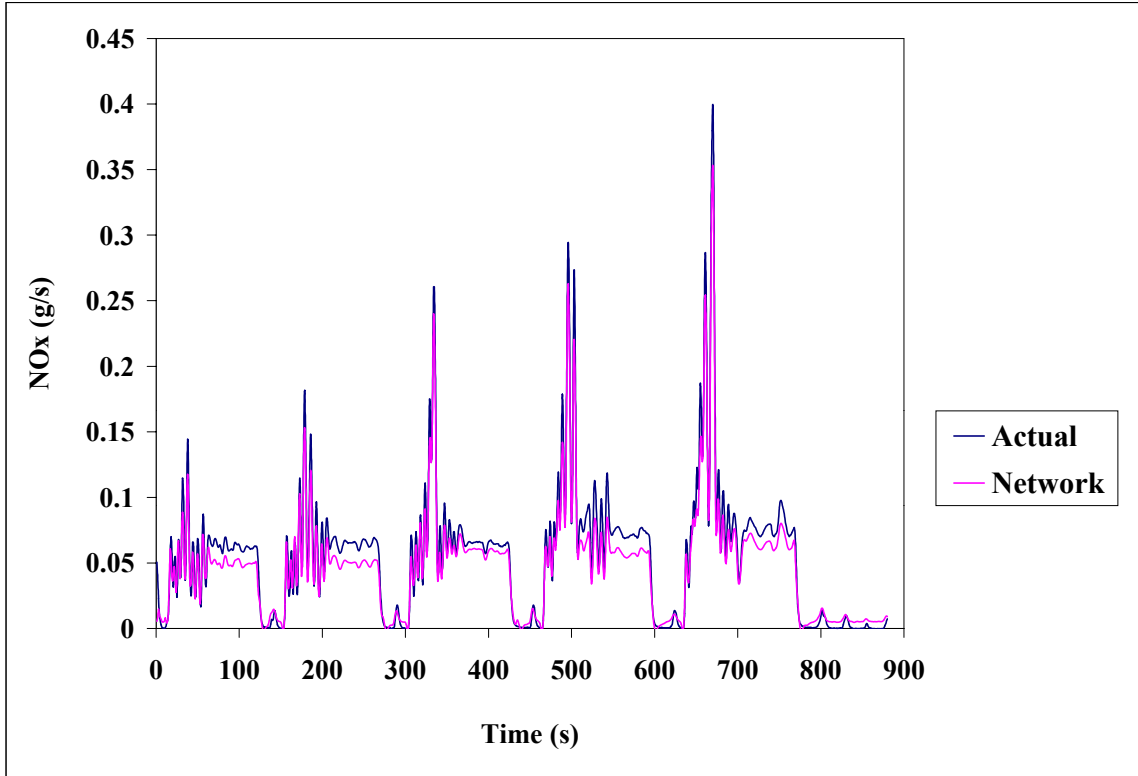


Figure 9.16. Second by second prediction of NO_x in an E-WVU-5 Peak test schedule by a 3-Layer network of Gaussian Complement activation function. The ANN was trained on all the transient test schedules combined.

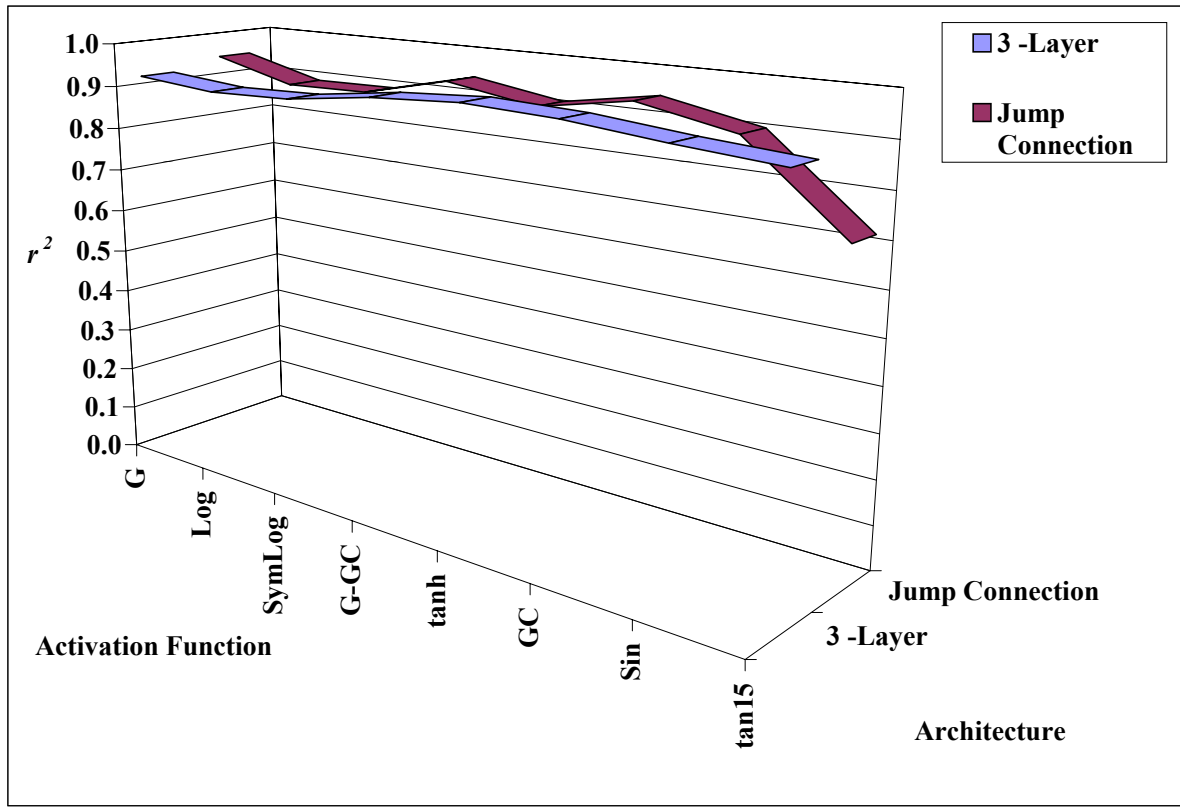


Figure 9.17. Variation of r^2 in the plane of activation function and architecture. The ANN was trained on an E-CSHVR predicting TEOM emissions of the same test schedule. Cases are described in Table 6.22.

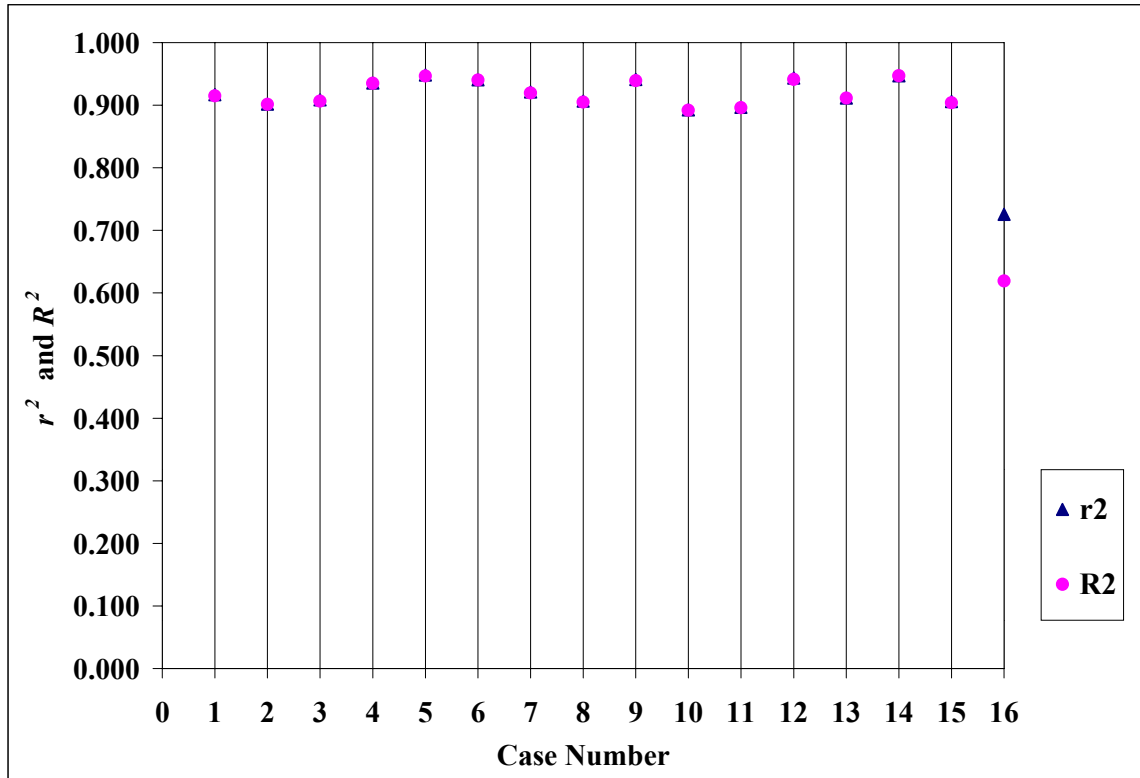


Figure 9.18. Variation of r^2 and R^2 with all cases. The ANN was trained on an E-CSHVR predicting TEOM emissions of the same test schedule. Cases are described in Table 6.22.

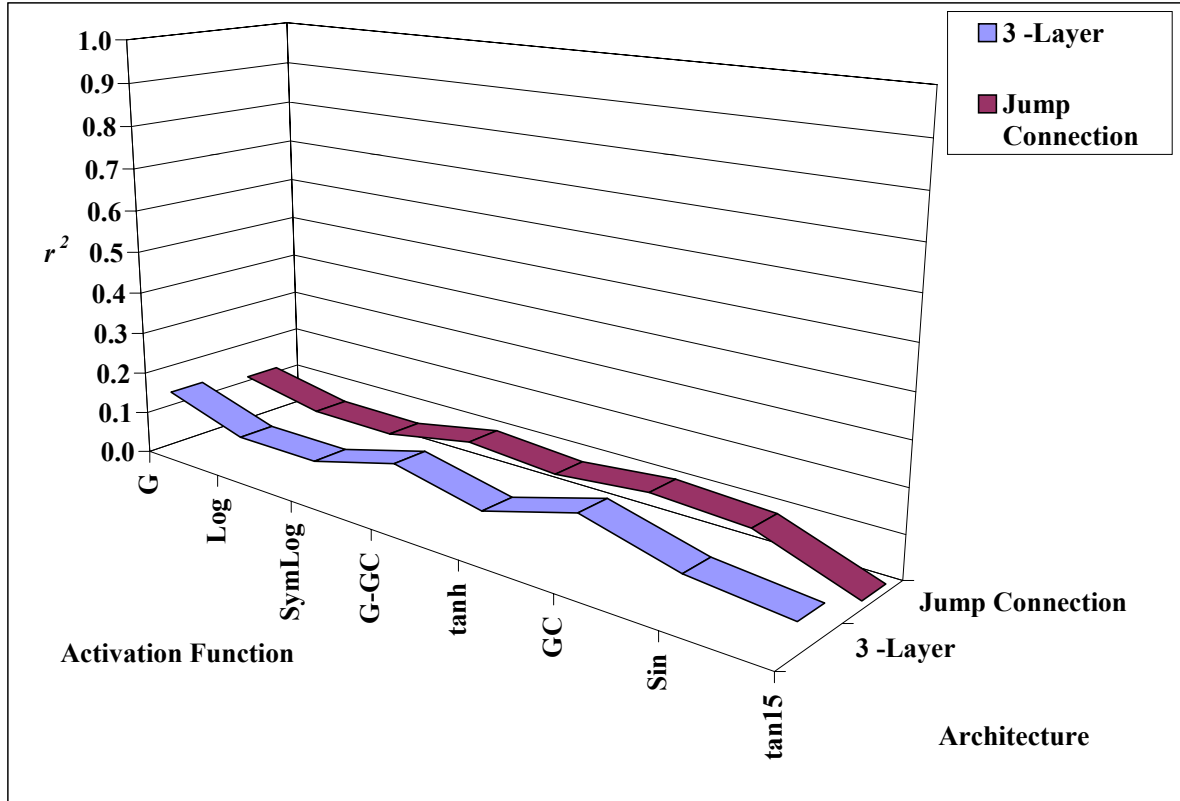


Figure 9.19. Variation of r^2 in the plane of activation function and architecture. The ANN was trained on an ETC predicting TEOM values of the same test schedule. Cases are described in Table 6.23.

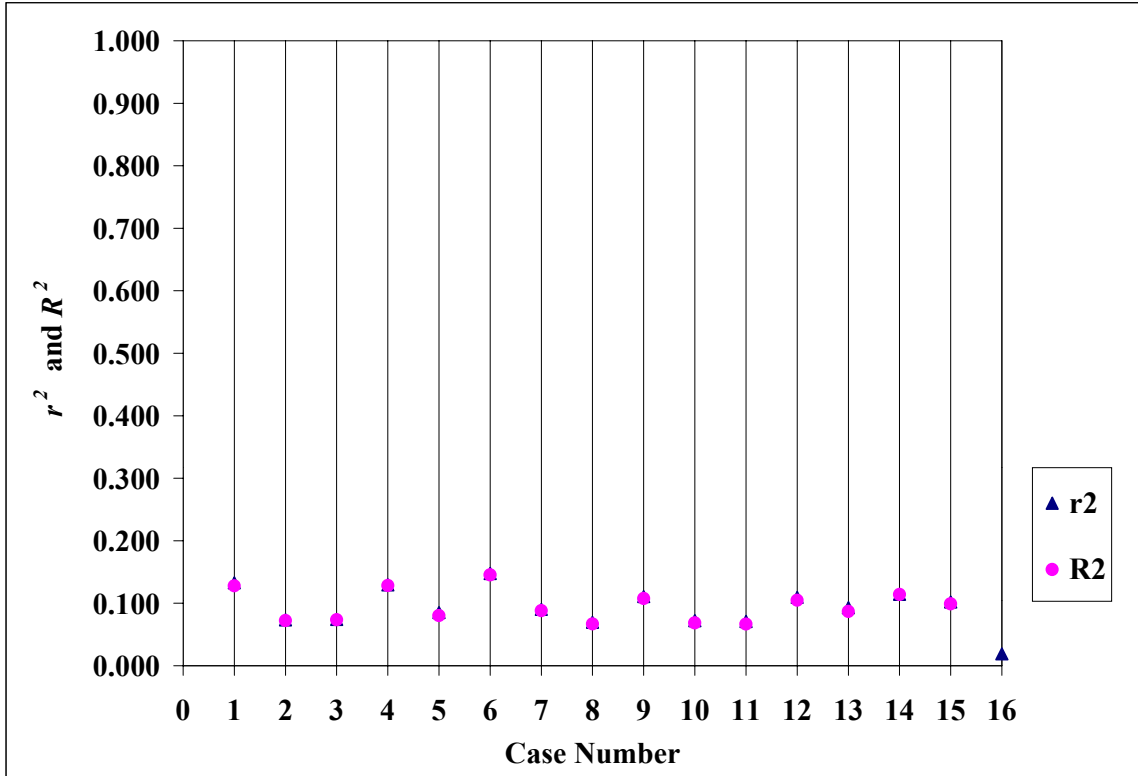


Figure 9.20. Variation of r^2 and R^2 with all cases. The ANN was trained on an ETC predicting TEOM values of the same test schedule. Cases are described in Table 6.23.

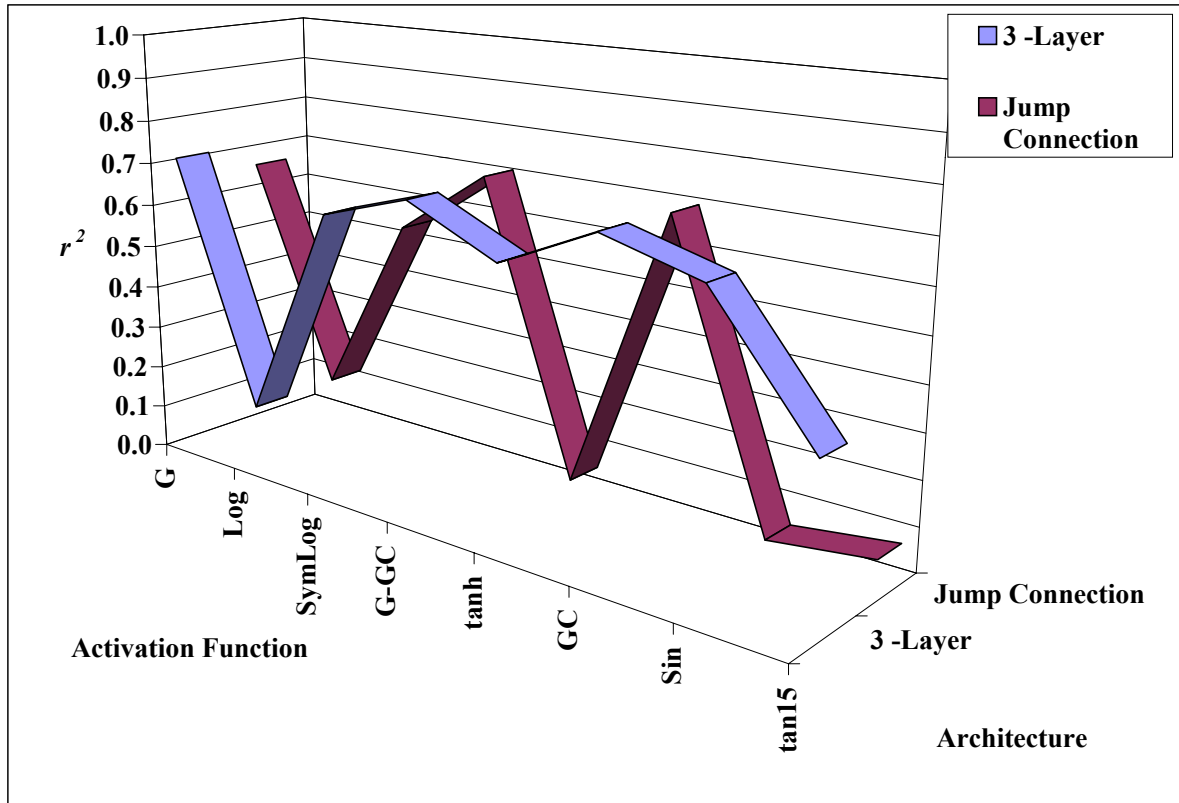


Figure 9.21. Variation of r^2 in the plane of activation function and. The ANN was trained on a FTP predicting TEOM values of the same test schedule. Cases are described in Table 6.24.

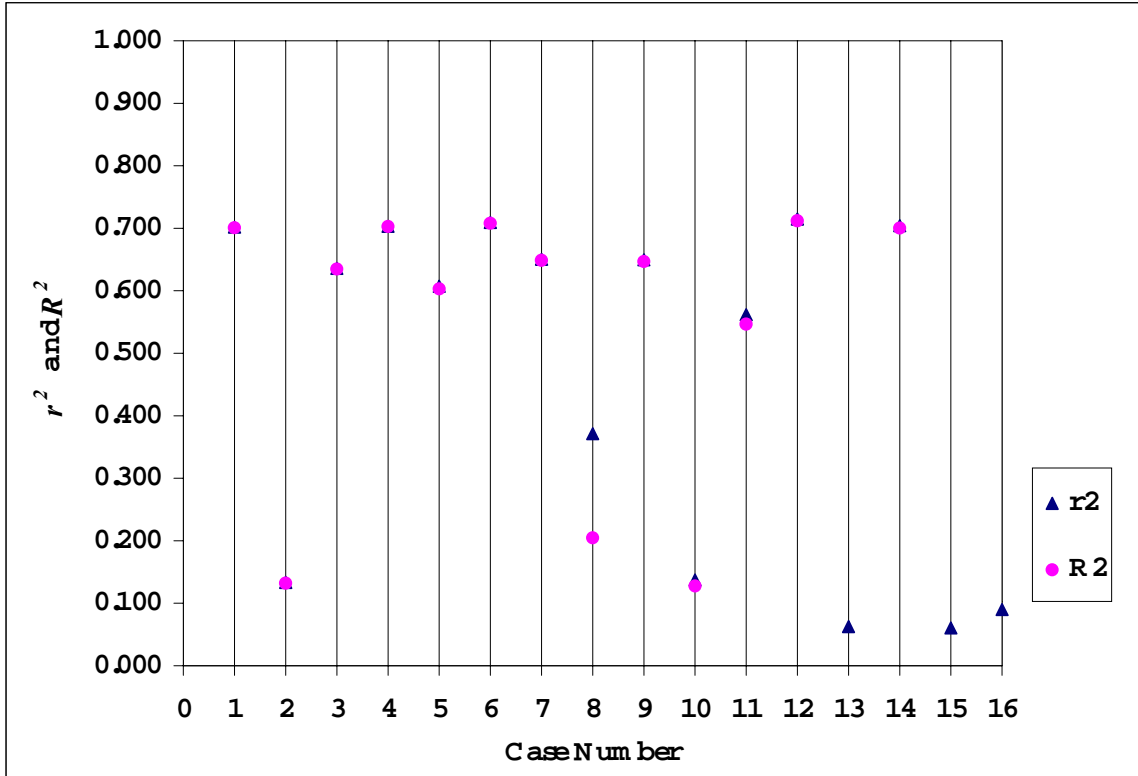


Figure 9.22. Variation of r^2 and R^2 with all cases. The ANN was trained on a FTP predicting TEOM values of the same test schedule. Cases are described in Table 6.24.

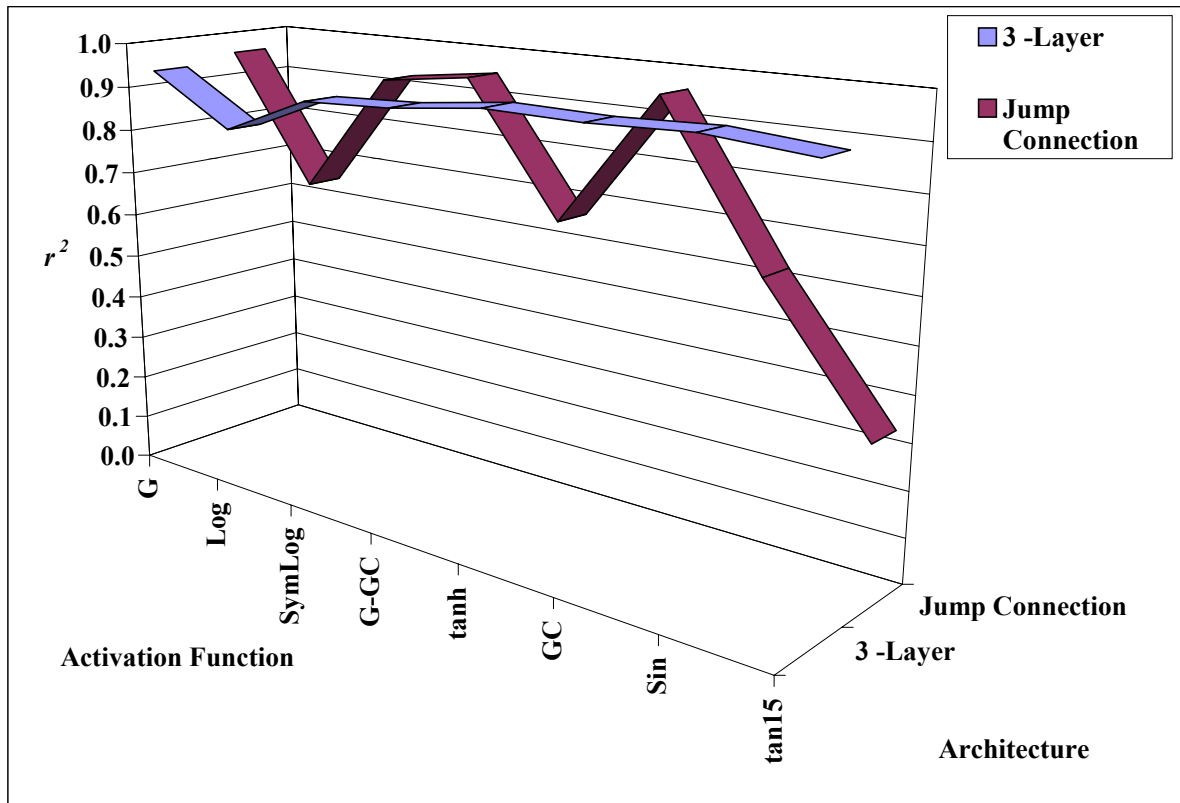


Figure 9.23. Variation of r^2 in the plane of activation function and architecture. The ANN was trained on an E-Highway predicting TEOM values of the same test schedule. Cases are described in Table 6.25.

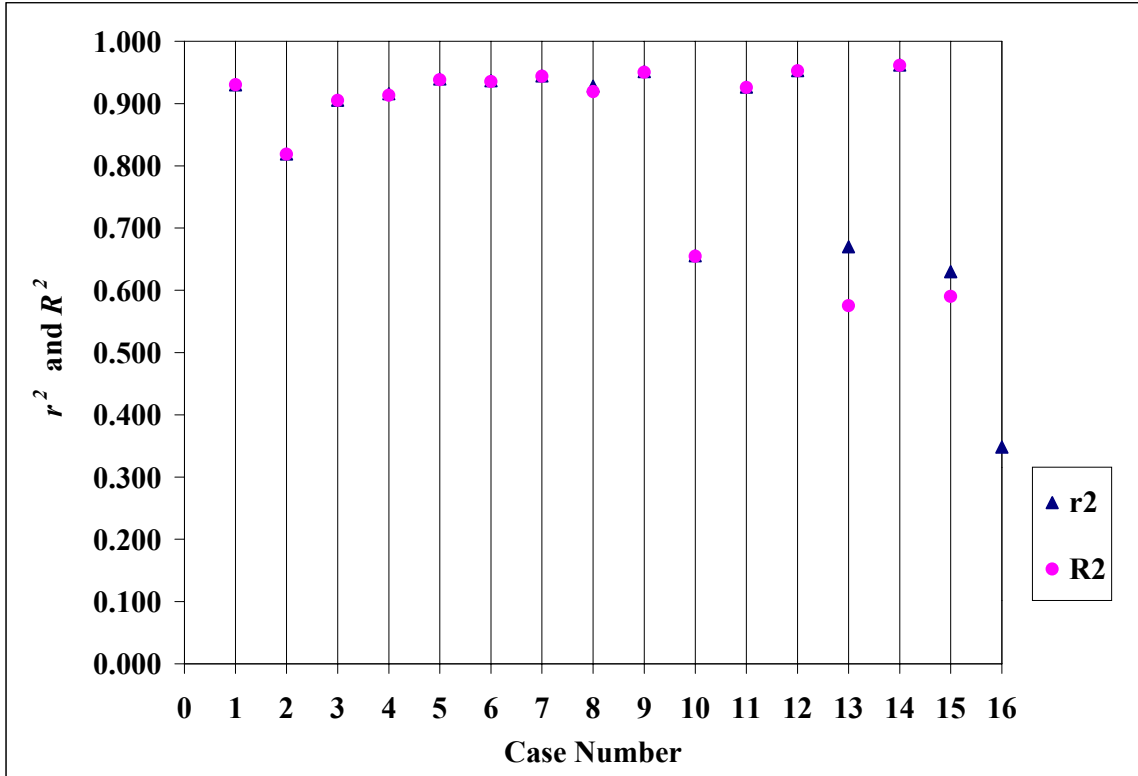


Figure 9.24. Variation of r^2 and R^2 with all cases. The ANN was trained on an E-Highway predicting the same test schedule. Cases are described in Table 6.25.

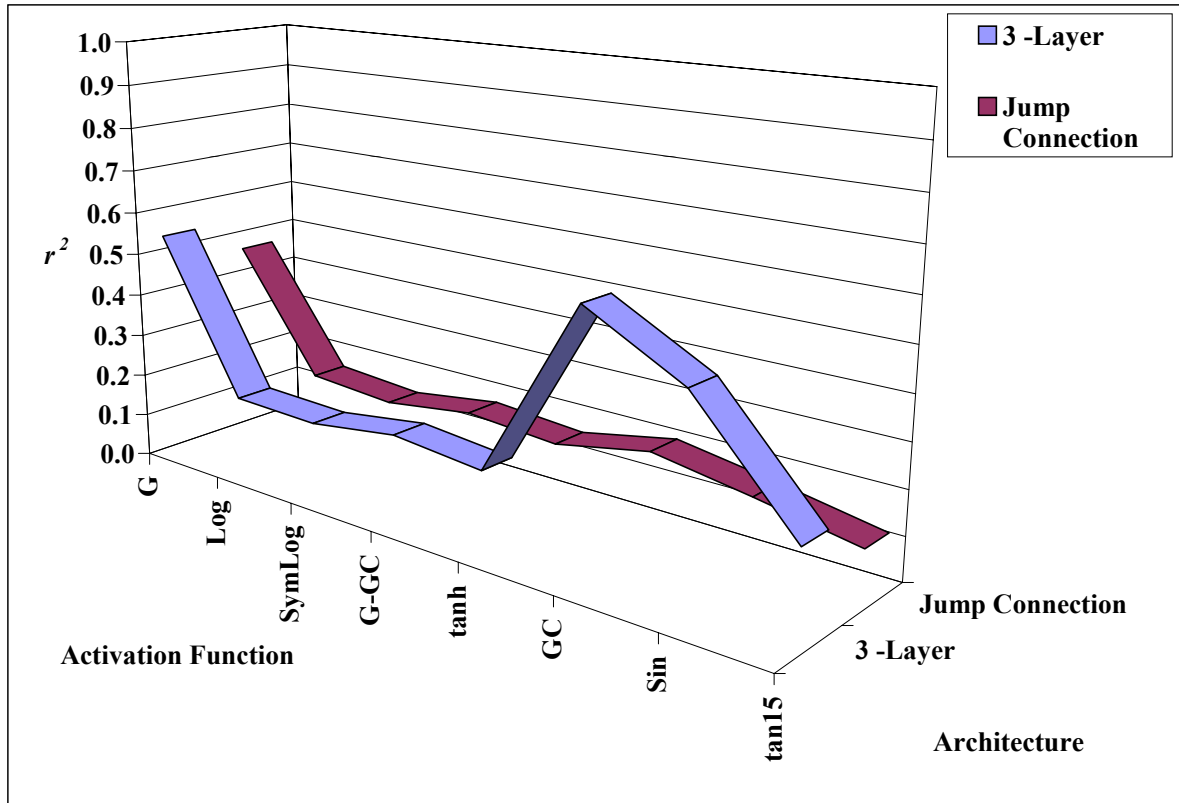


Figure 9.25. Variation of r^2 in the plane of activation function and. The ANN was trained on an E-WVU-5 Peak predicting TEOM values of the same test schedule. Cases are described in Table 6.26.

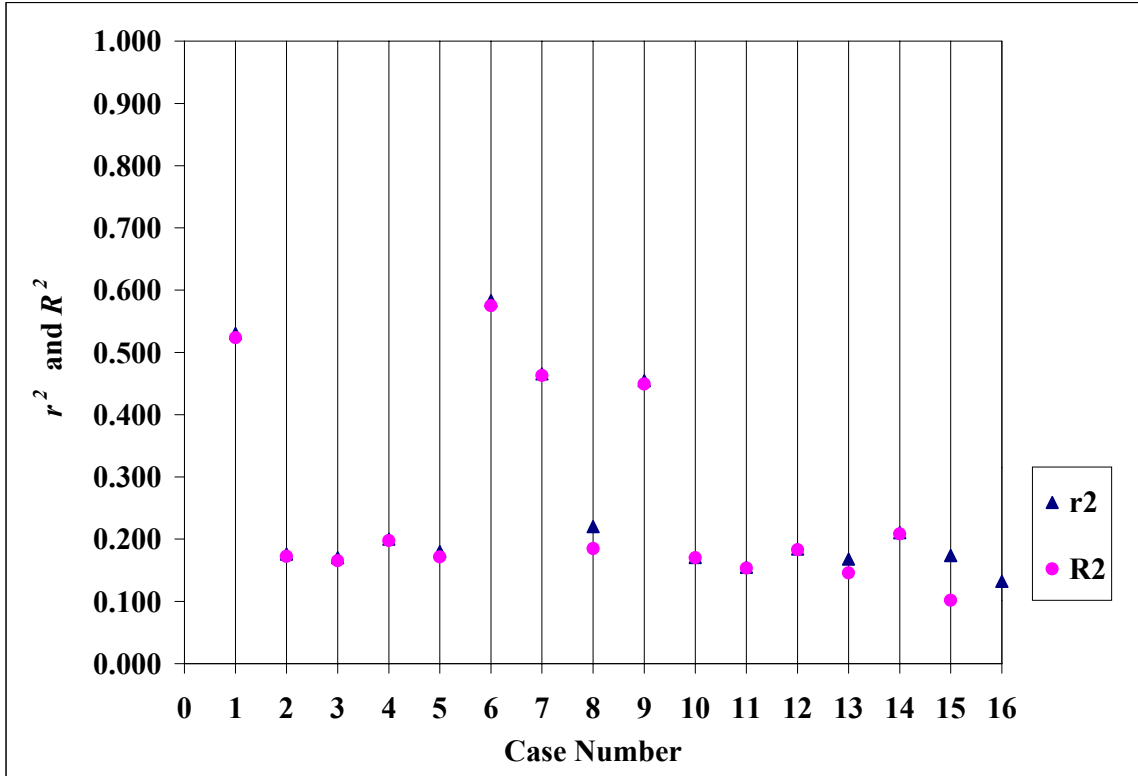


Figure 9.26. Variation of r^2 and R^2 with all cases. The ANN was trained on an E-WVU-5 Peak predicting TEOM values of the same test schedule. Cases are described in Table 6.26.

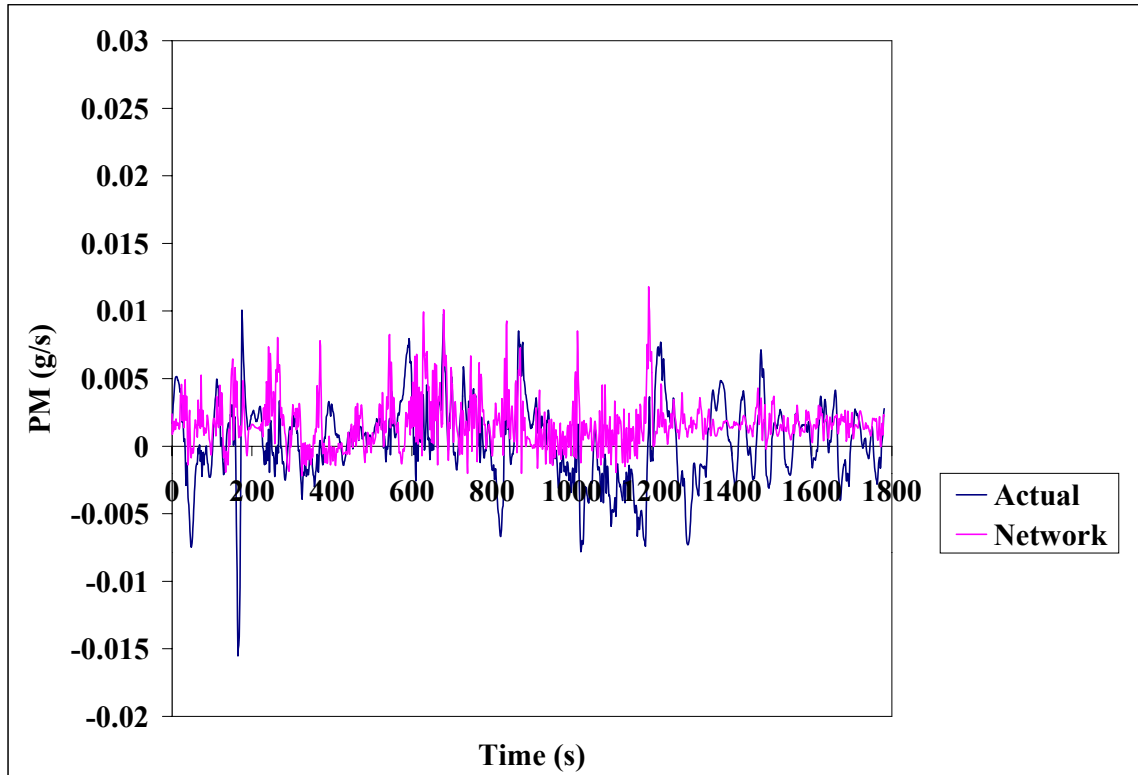


Figure 9.27. Second by second prediction of PM (corrected TEOM for moisture content) in an ETC test schedule by a Jump Connection network of Gaussian Complement activation function. The ANN was trained on all the transient test schedules combined.

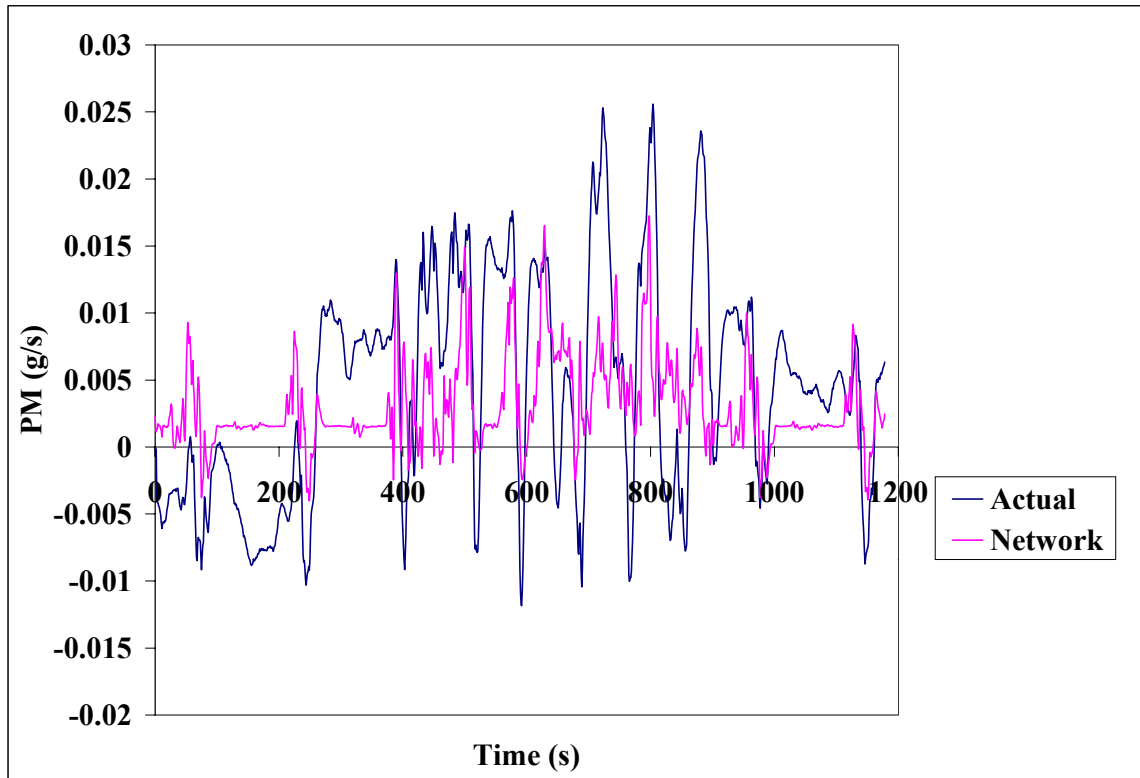


Figure 9.28. Second by second prediction of PM (corrected TEOM for moisture content) in a FTP test schedule by a Jump Connection network of Gaussian Complement activation function. The ANN was trained on all the transient test schedules combined.

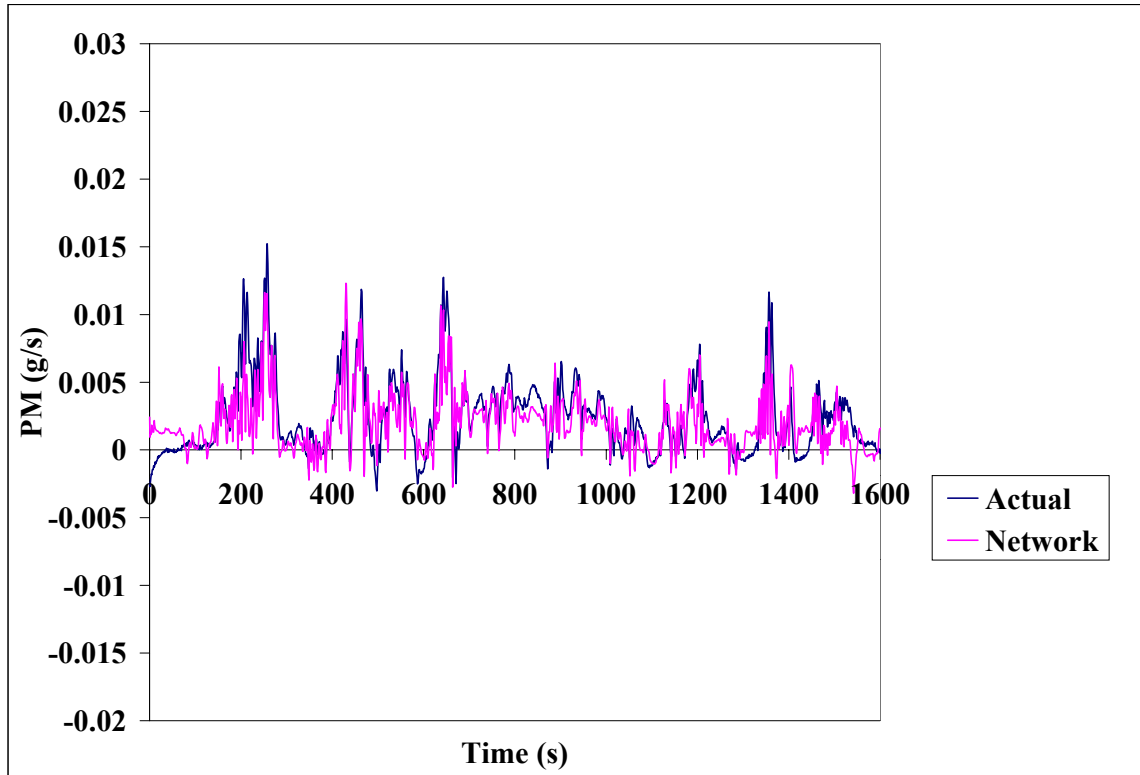


Figure 9.29. Second by second prediction of PM (corrected TEOM for moisture content) in an E-Highway test schedule by a Jump Connection network of Gaussian Complement activation function. The ANN was trained on all the transient test schedules combined.

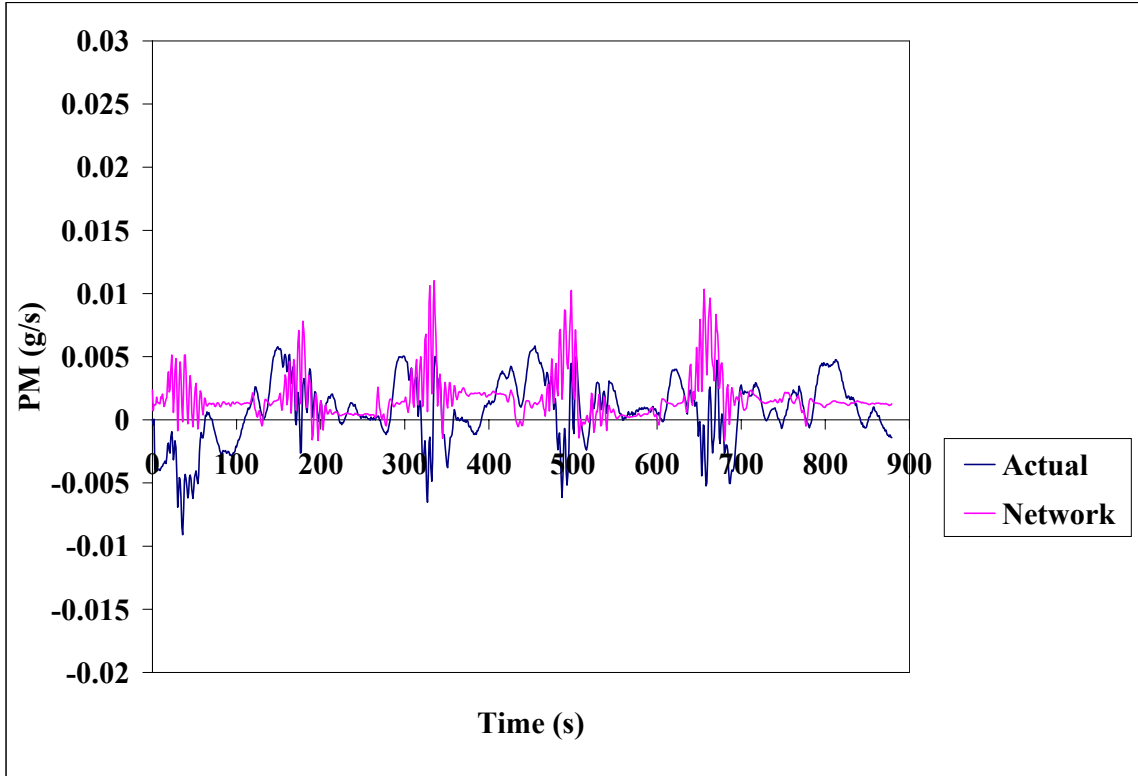


Figure 9.30. Second by second prediction of PM (corrected TEOM for moisture content) in an E-WVU-5 Peak test schedule by a Jump Connection network of Gaussian Complement activation function. The ANN was trained on all the transient test schedules combined.

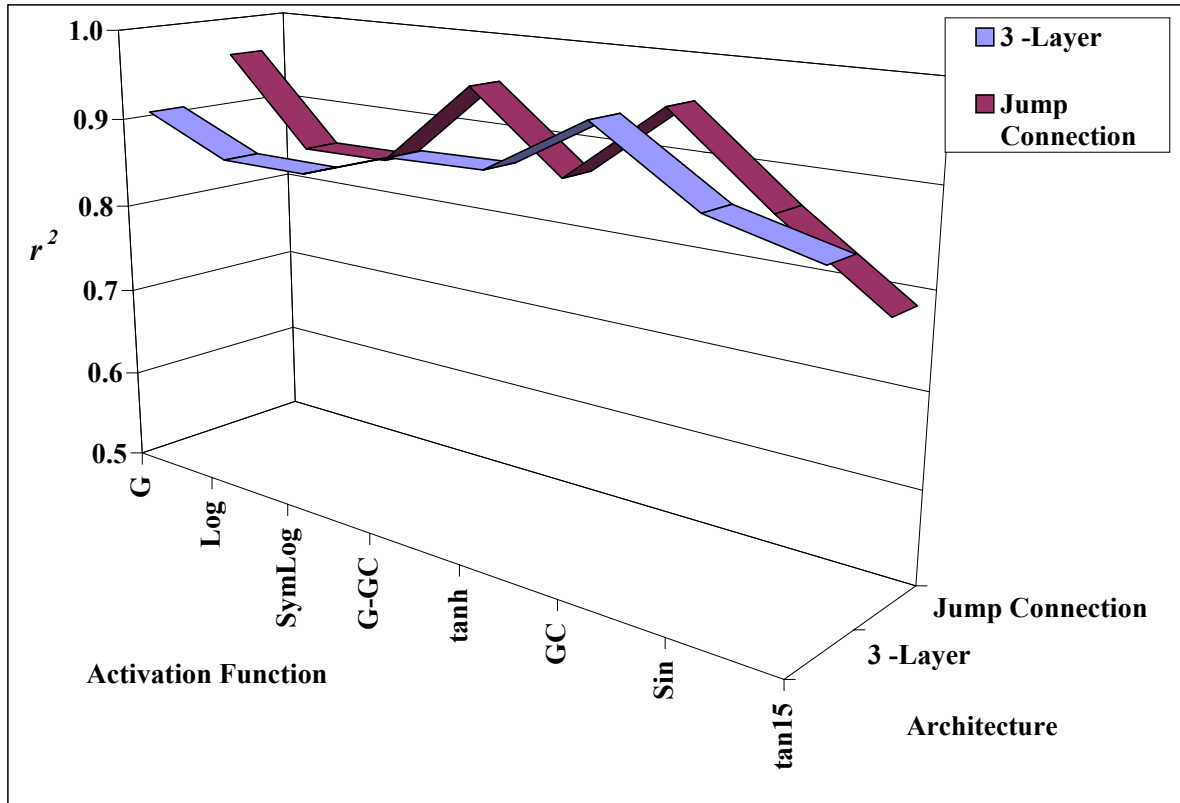


Figure 9.31. Variation of r^2 in the plane of activation function and. The ANN was trained on an E-CSHVR predicting CO emissions of the same test schedule. Cases are described in Table 6.31.

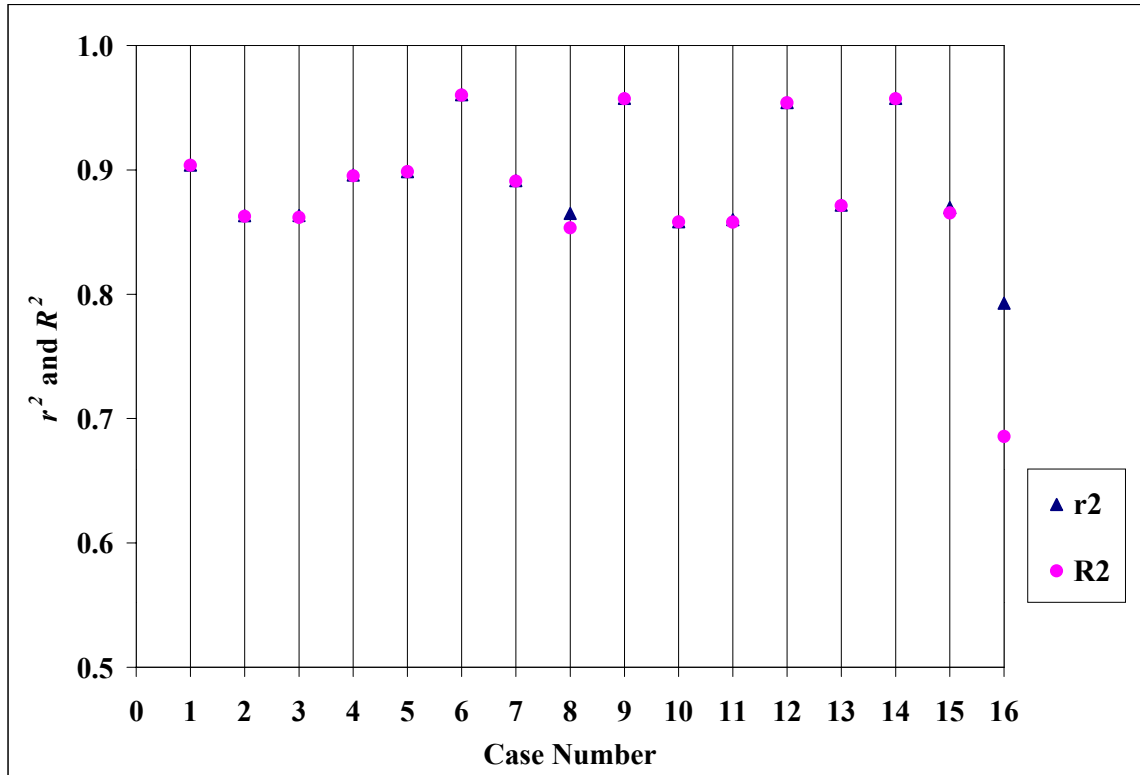


Figure 9.32. Variation of r^2 and R^2 with all cases. The ANN trained on an E-CSHVR predicting CO emissions of the same test schedule. Cases are described in Table 6.31.

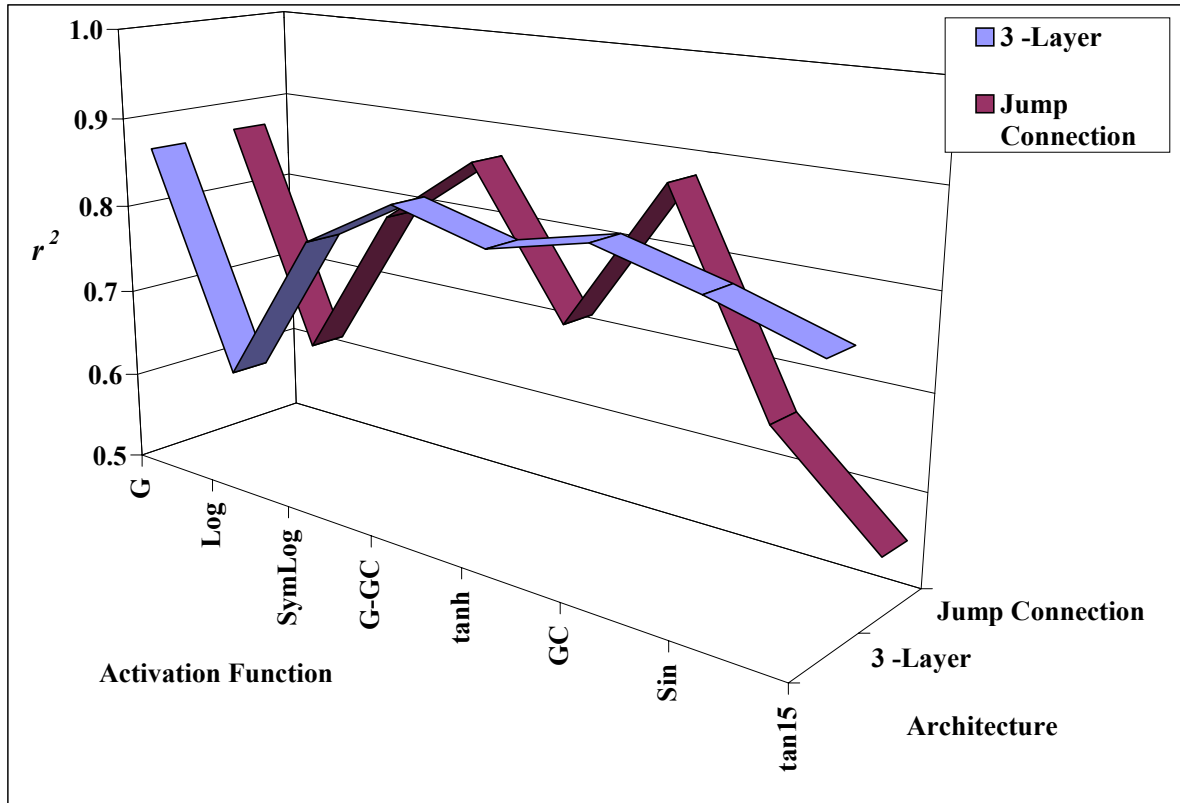


Figure 9.33. Variation of r^2 in the plane of activation function and architecture. The ANN was trained on an ETC predicting CO emissions of the same test schedule. Cases are described in Table 6.32.

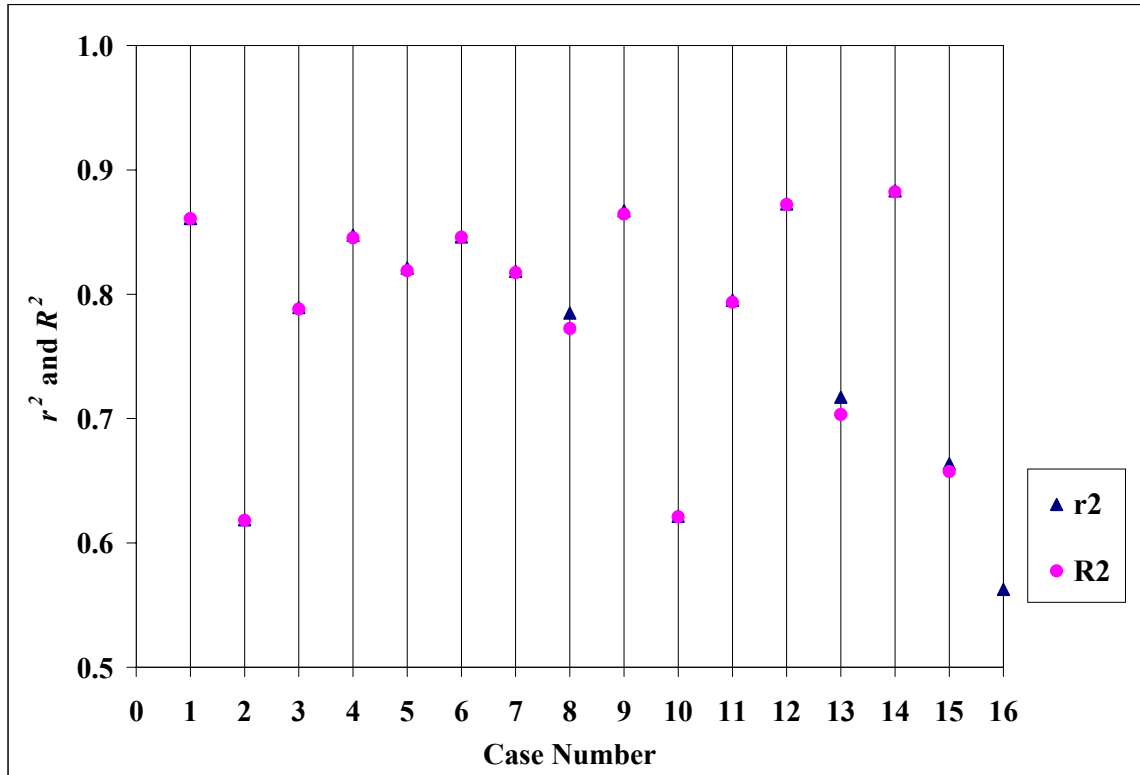


Figure 9.34. Variation of r^2 and R^2 with all cases. The ANN trained on an ETC predicting CO emissions of the same test schedule. Cases are described in Table 6.32.

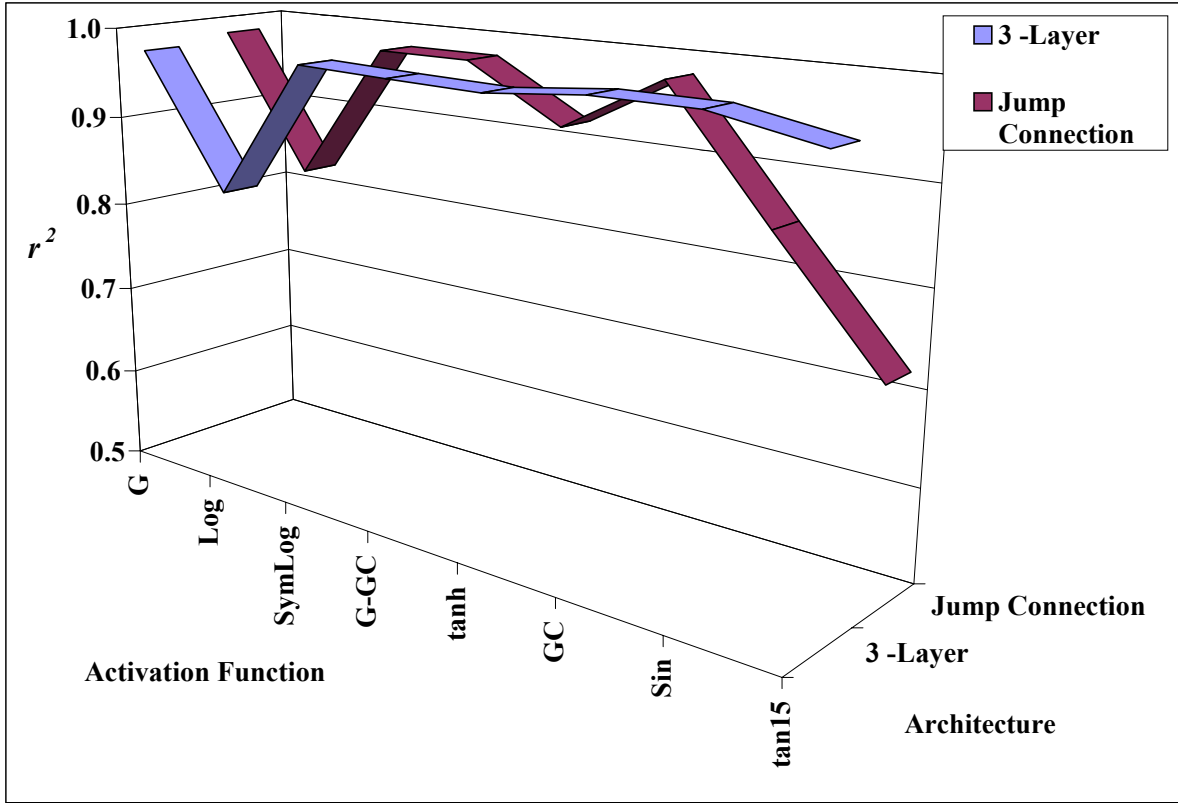


Figure 9.35. Variation of r^2 in the plane of activation function and architecture. The ANN was trained on a FTP predicting CO emissions of the same test schedule. Cases are described in Table 6.33.

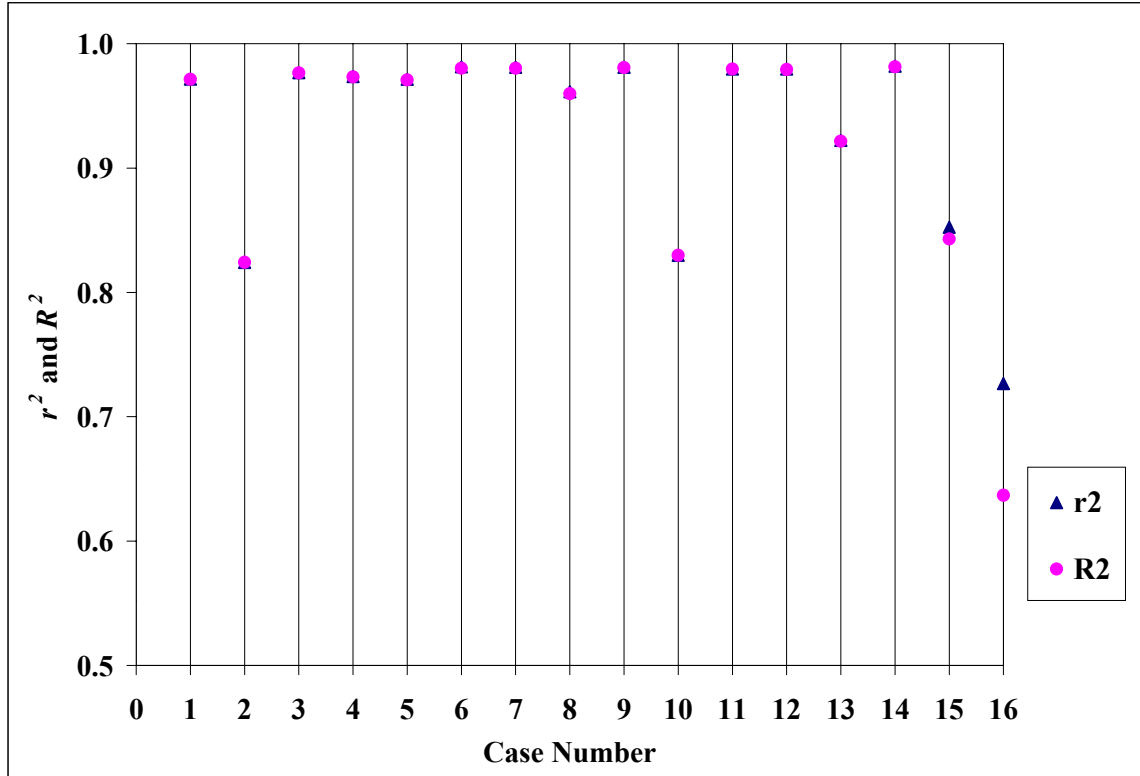


Figure 9.36. Variation of r^2 and R^2 with all cases. The ANN trained on a FTP predicting CO emissions of the same test schedule. Cases are described in Table 6.33.

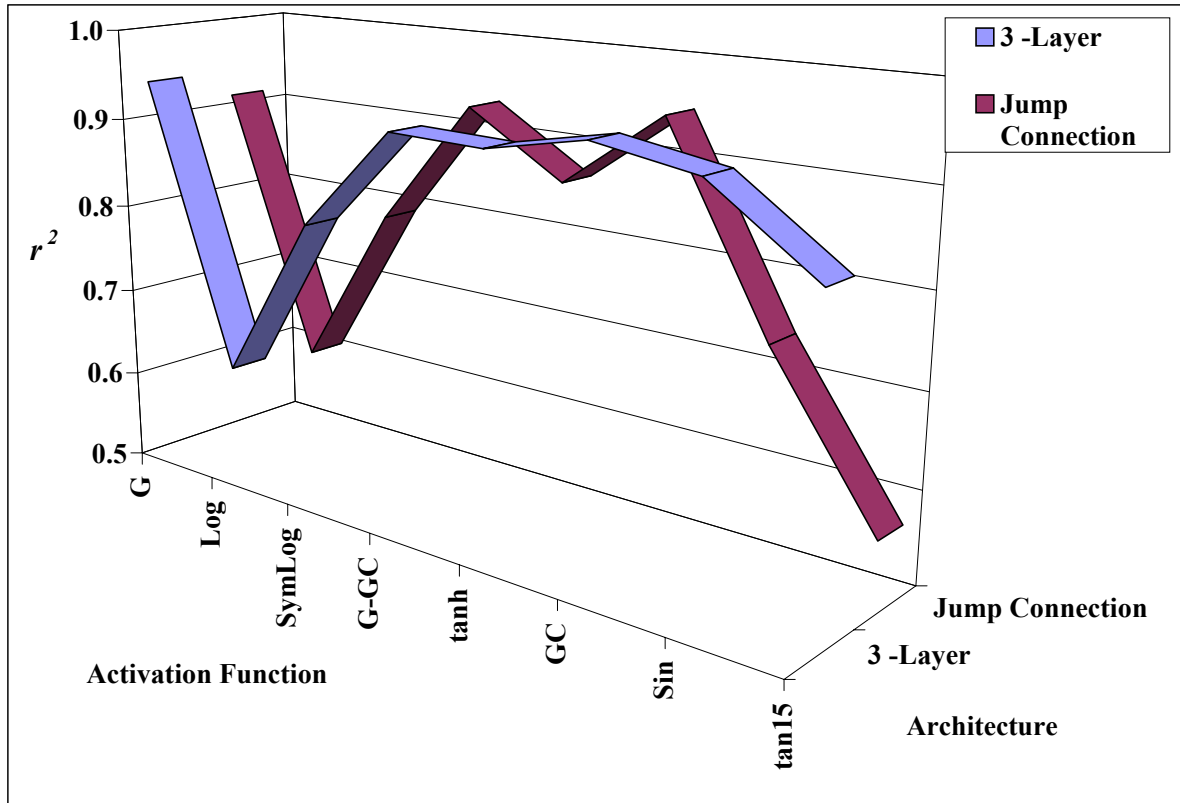


Figure 9.37. Variation of r^2 in the plane of activation function and architecture. The ANN was trained on an E-Highway predicting CO emissions of the same test schedule. Cases are described in Table 6.34.

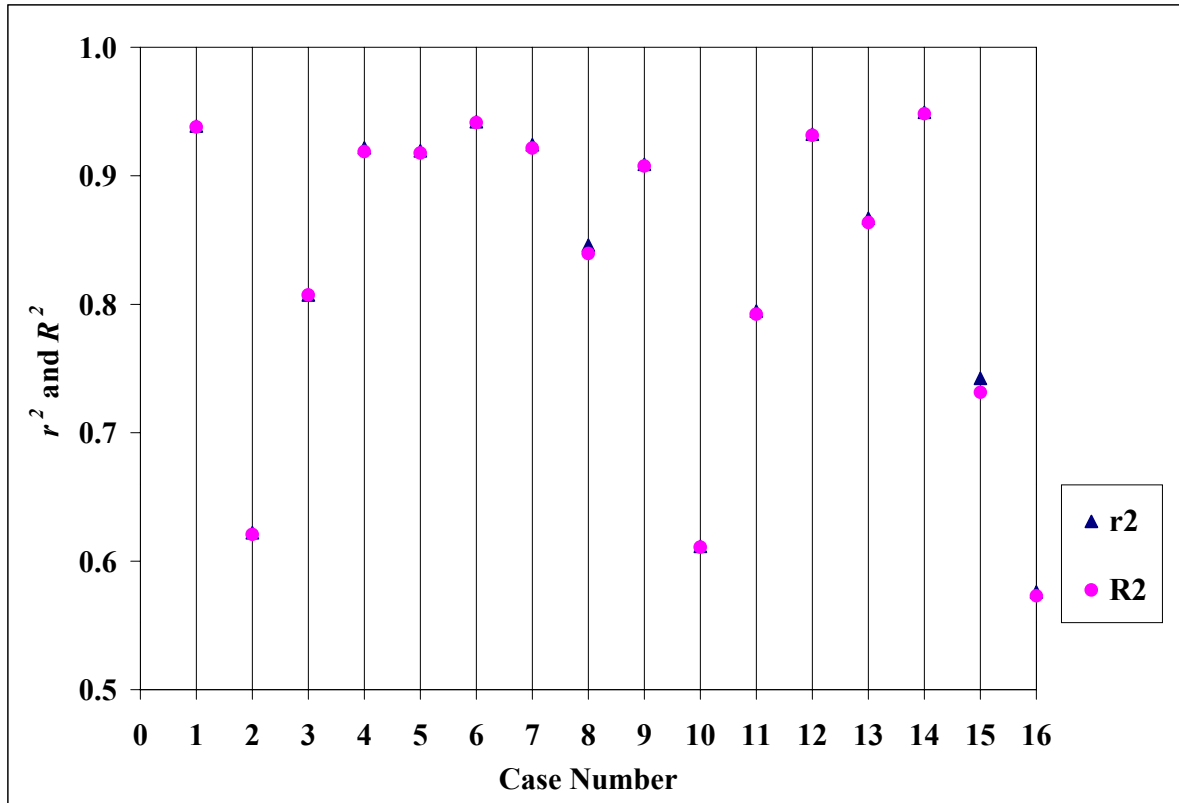


Figure 9.38. Variation of r^2 and R^2 with all cases. The ANN was trained on an E-Highway predicting CO emissions of the same test schedule. Cases are described in Table 6.34.

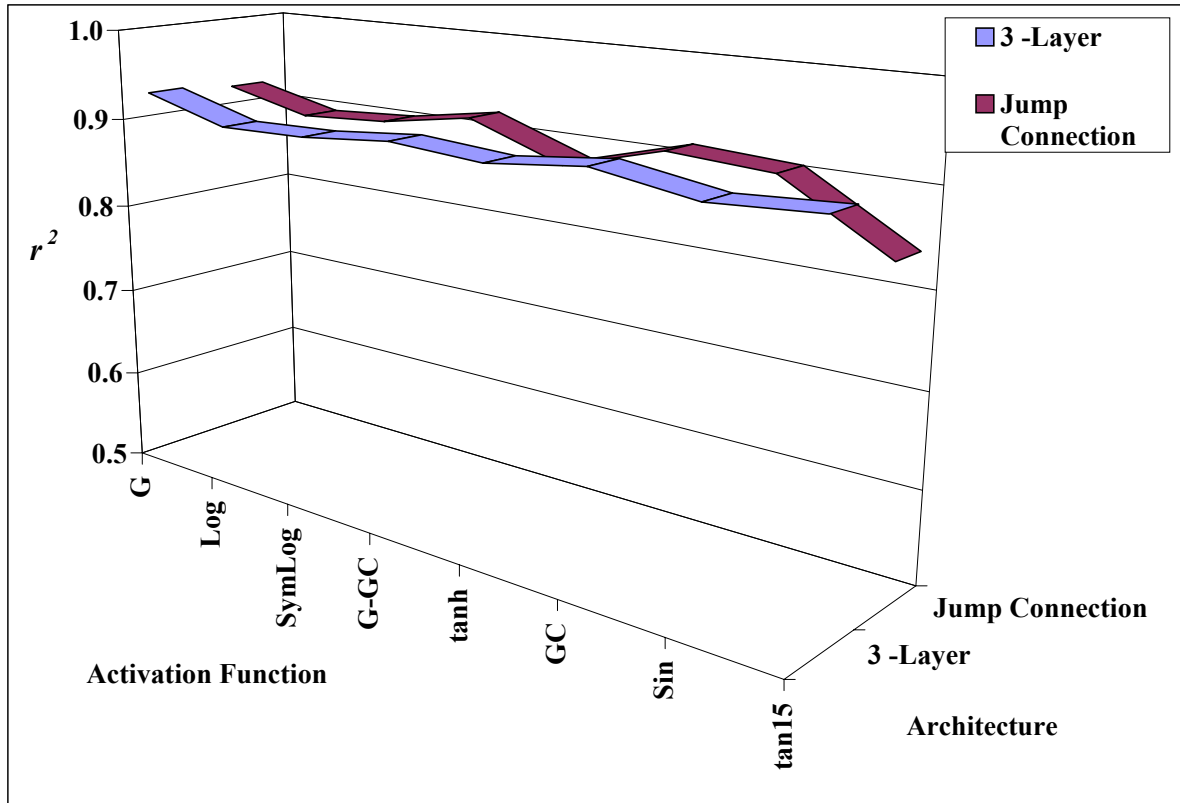


Figure 9.39. Variation of r^2 in the plane of activation function and. The ANN was trained on an E-WVU-5 Peak predicting CO emissions of the same test schedule. Cases are described in Table 6.35.

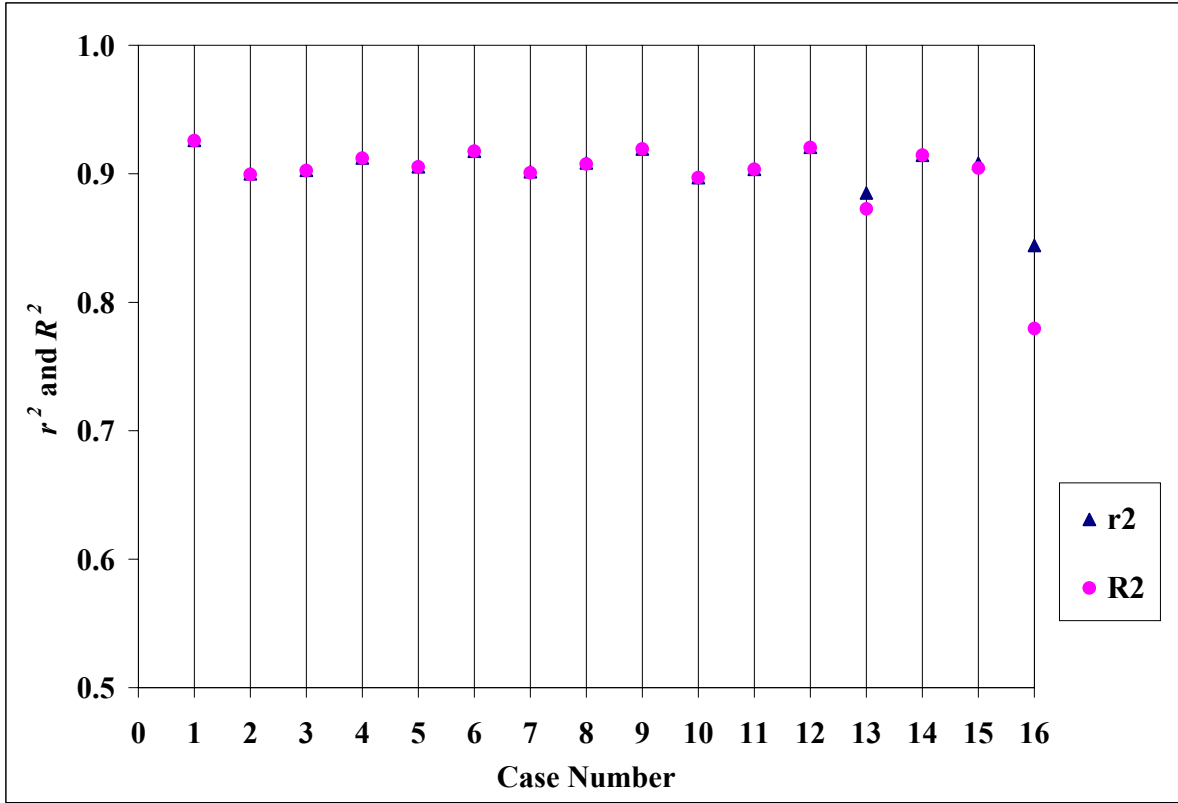


Figure 9.40. Variation of r^2 and R^2 with all cases. The ANN was trained on an E-WVU-5 Peak predicting CO emissions of the same test schedule. Cases are described in Table 6.35.

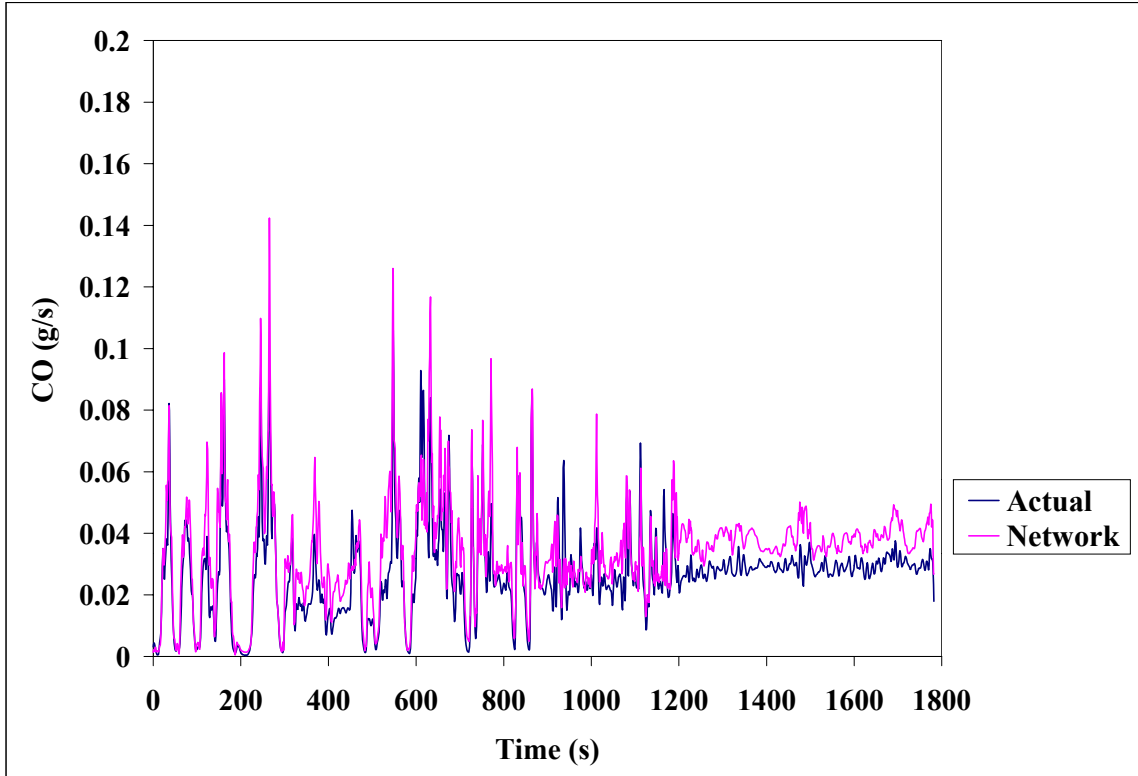


Figure 9.41. Second by second prediction of CO in an ETC test schedule by a Jump Connection network of Gaussian Complement activation function. The ANN was trained on all the transient test schedules combined.

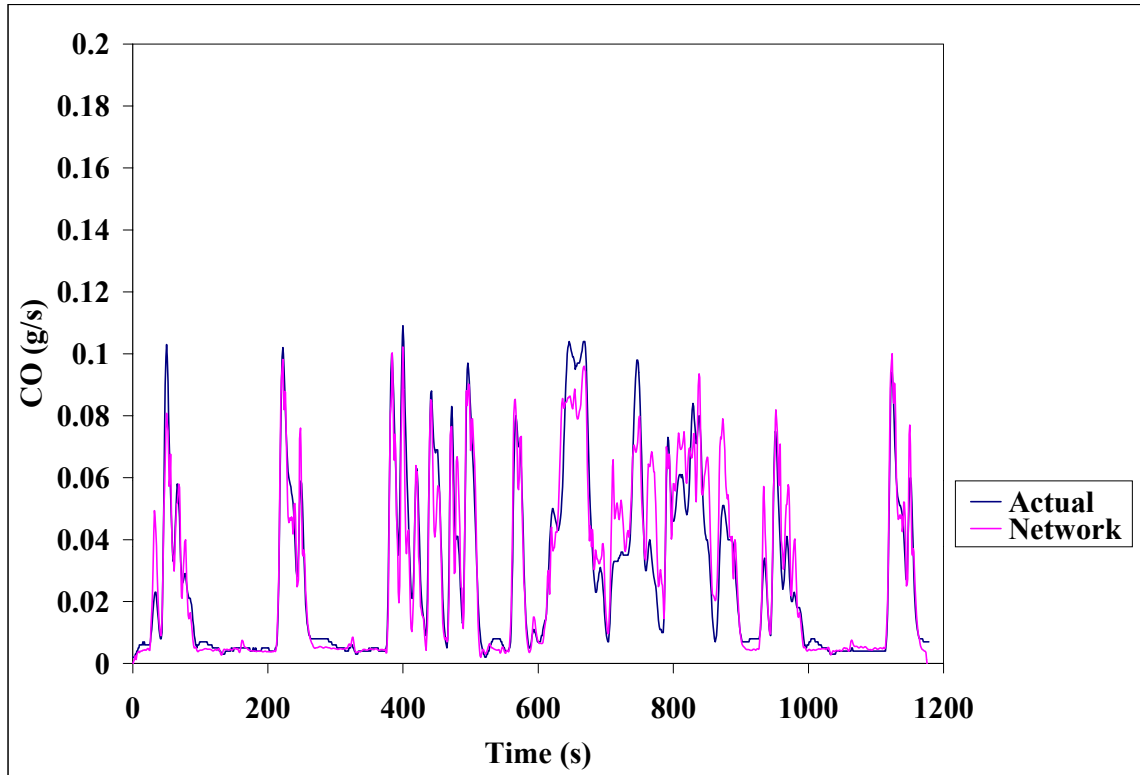


Figure 9.42. Second by second prediction of CO in a FTP test schedule by a Jump Connection network of Gaussian Complement activation function. The ANN was trained on all the transient test schedules combined.

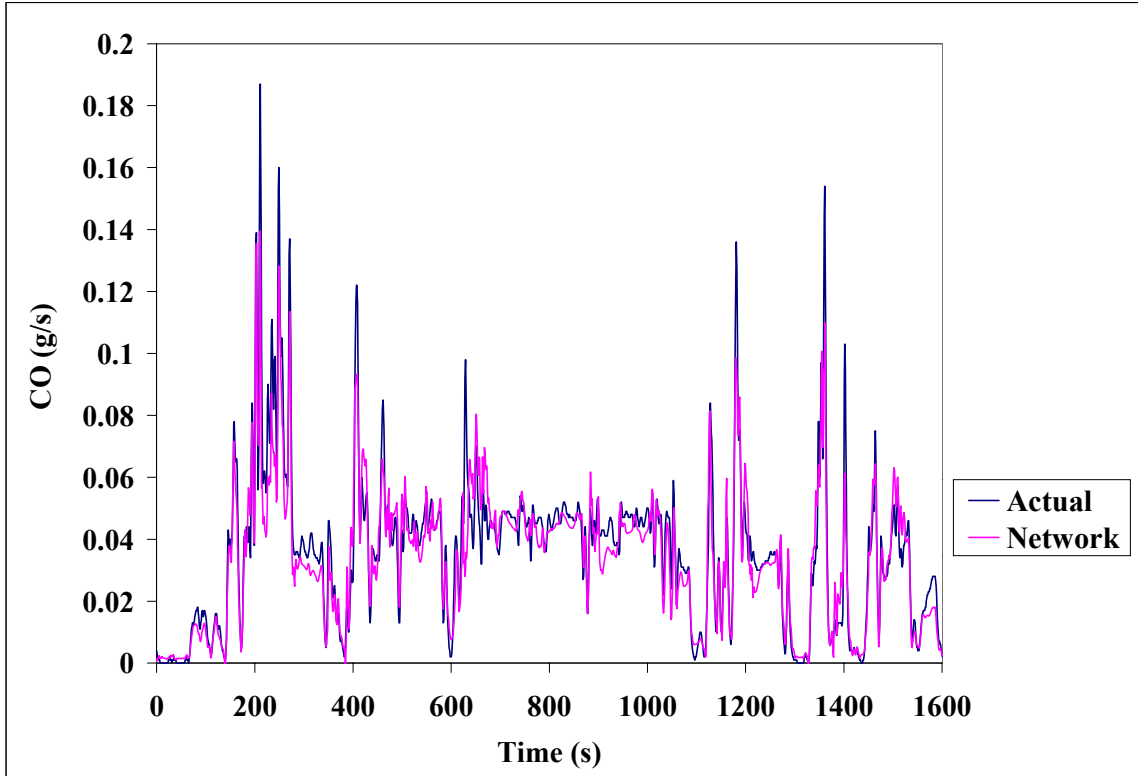


Figure 9.43. Second by second prediction of CO in an E-Highway test schedule by a Jump Connection network of Gaussian Complement activation function. The ANN was trained on all the transient test schedules combined.

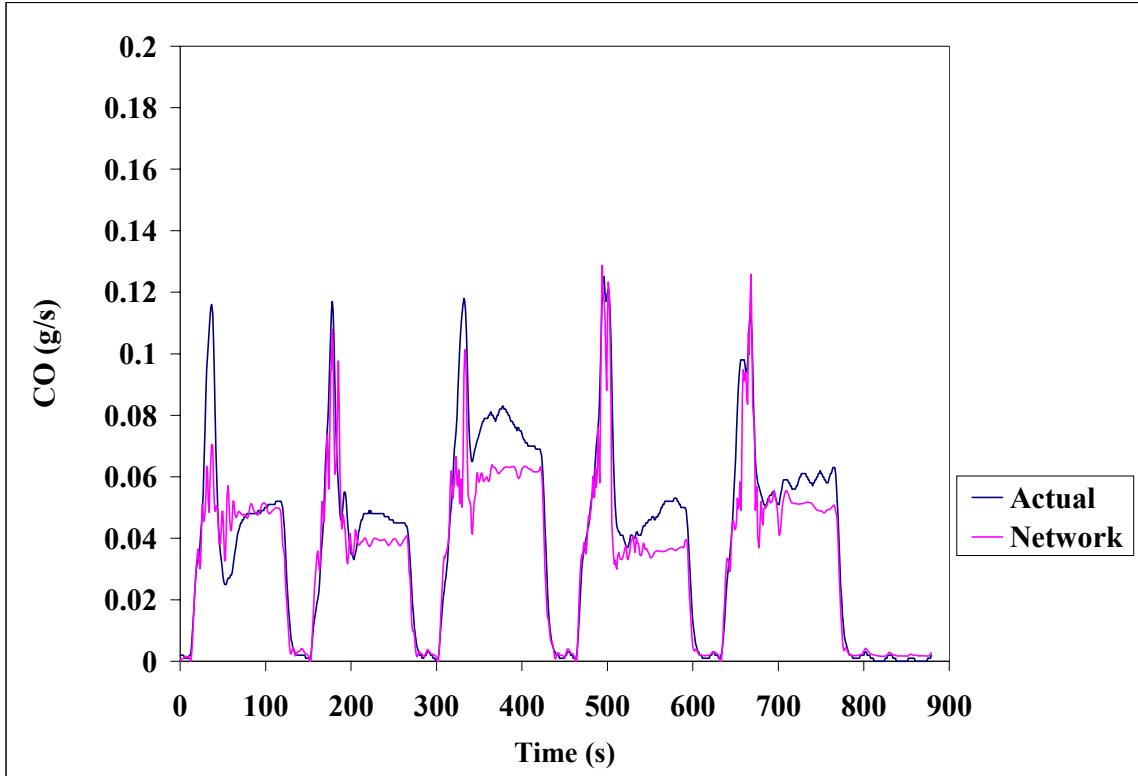


Figure 9.44. Second by second prediction of CO in an E-WVU-5 Peak test schedule by a Jump Connection network of Gaussian Complement activation function. The ANN was trained on all the transient test schedules combined.

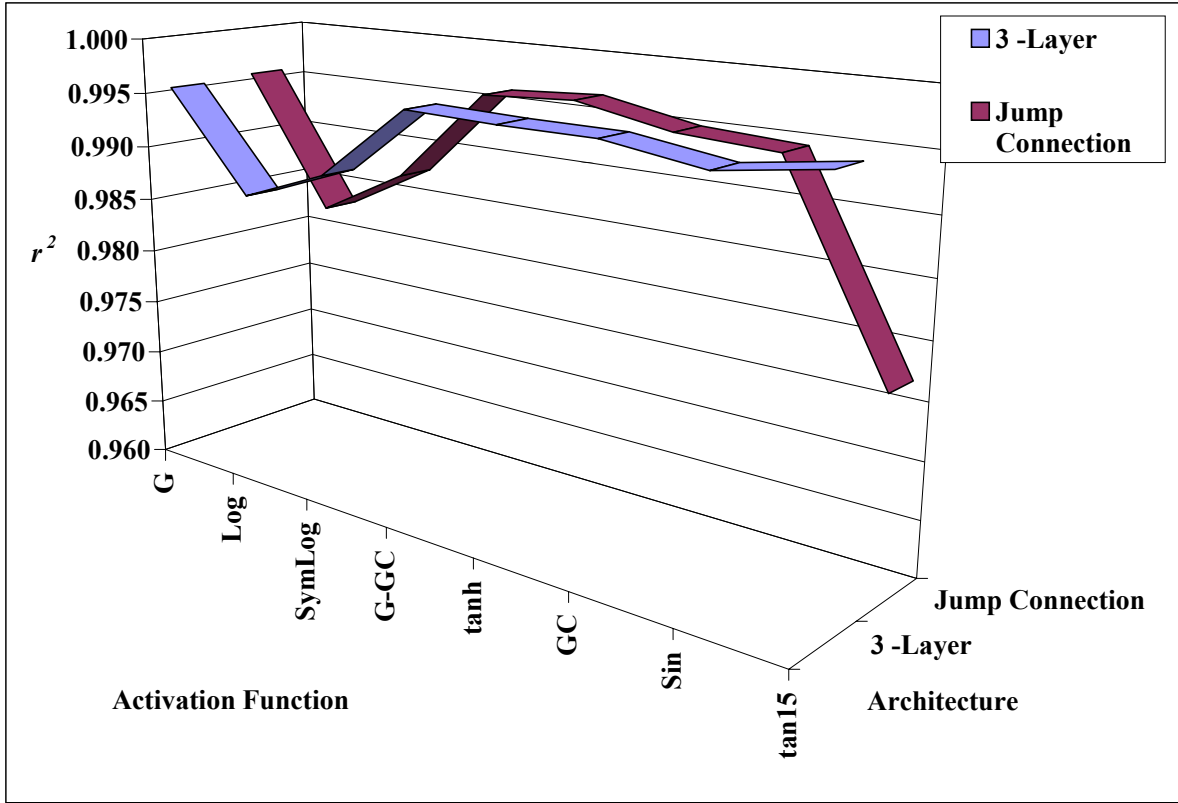


Figure 9.45. Variation of r^2 in the plane of activation function and architecture. The ANN was trained on an E-CSHVR predicting CO₂ emissions of the same test schedule. Cases are described in Table 6.38.

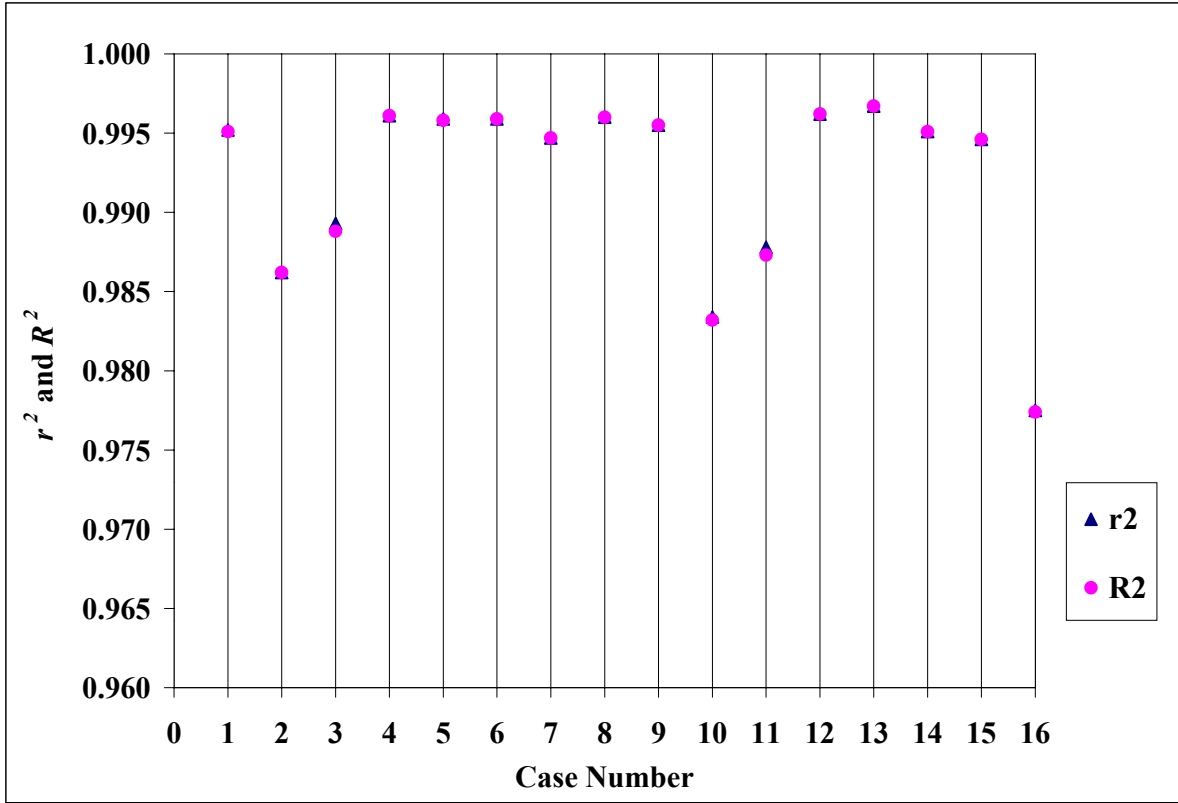


Figure 9.46. Variation of r^2 and R^2 with all cases. The ANN was trained on an E-CSHVR predicting CO₂ emissions of the same test schedule. Cases are described in Table 6.38.

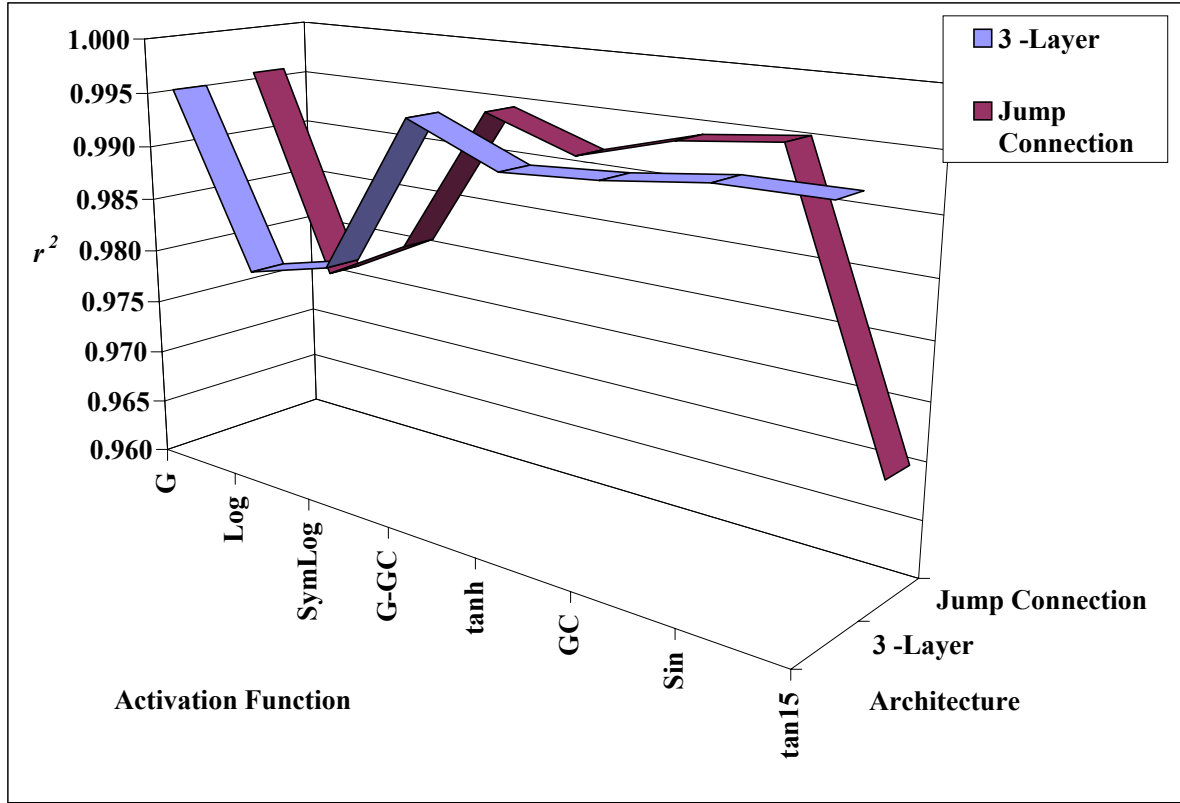


Figure 9.47. Variation of r^2 in the plane of activation function and architecture. The ANN was trained on an ETC predicting CO₂ emissions of the same test schedule. Cases are described in Table 6.39.

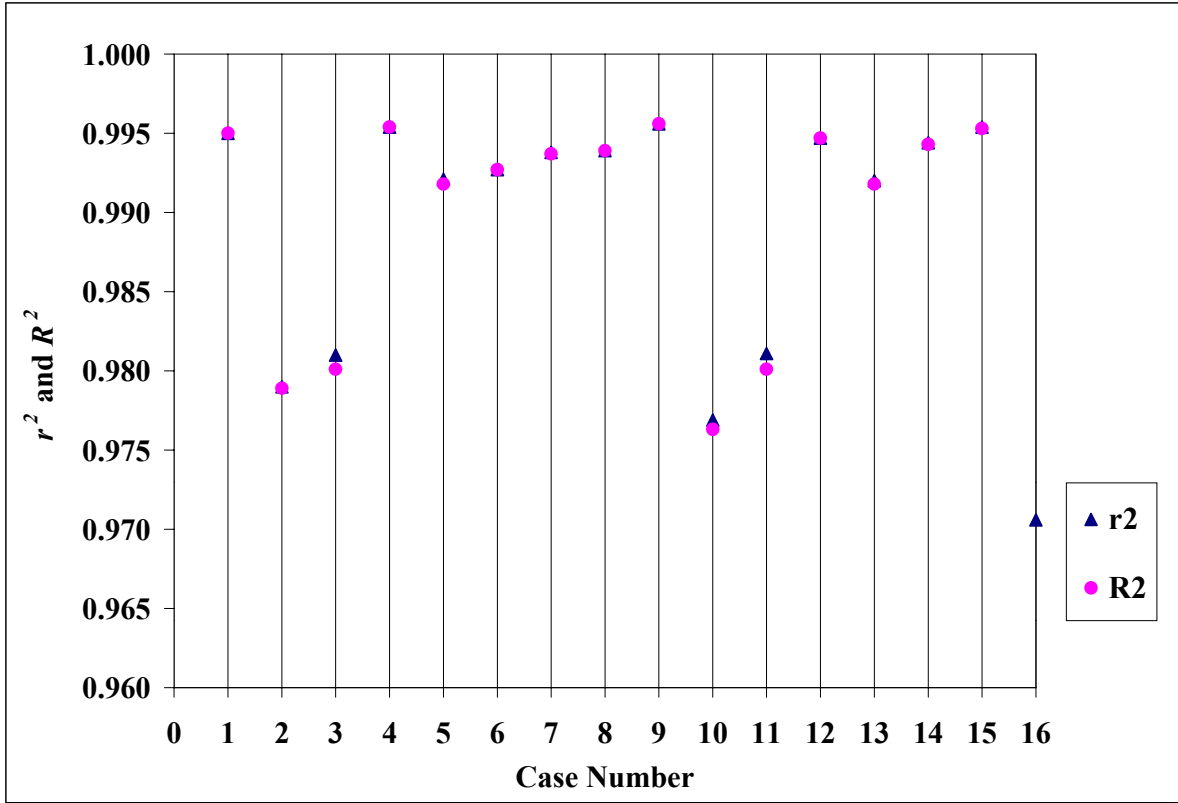


Figure 9.48. Variation of r^2 and R^2 with all cases. The ANN was trained on an ETC predicting CO₂ emissions of the same test schedule. Cases are described in Table 6.39.

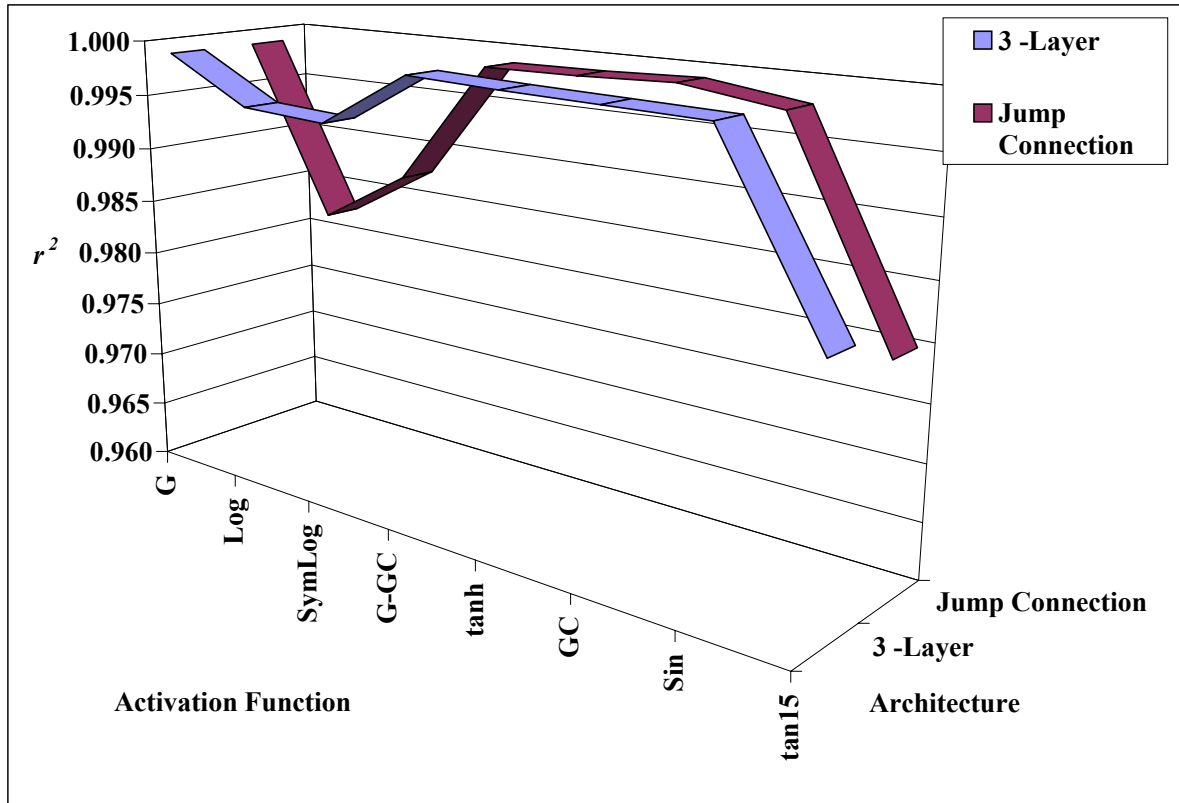


Figure 9.49. Variation of r^2 in the plane of activation function and architecture. The ANN was trained on a FTP predicting CO₂ emissions of the same test schedule. Cases are described in Table 6.40.

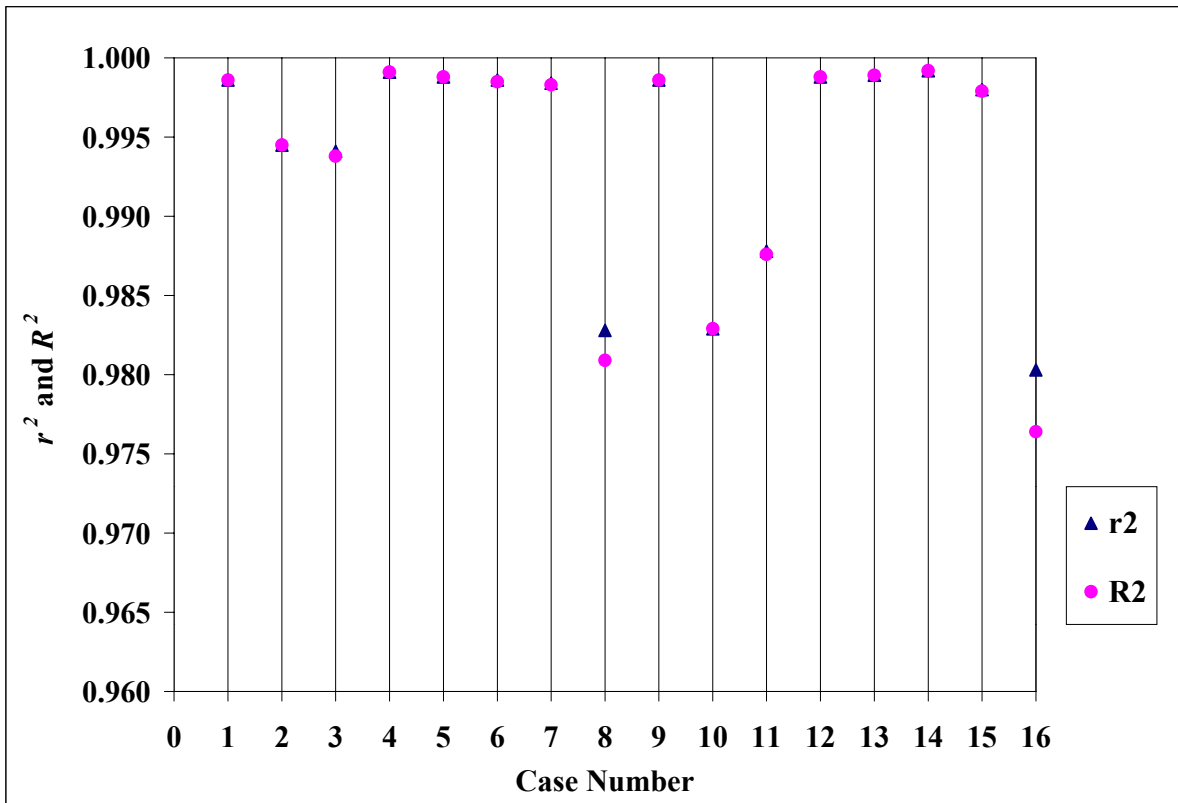


Figure 9.50. Variation of r^2 and R^2 with all cases. The ANN was trained on a FTP predicting CO₂ emissions of the same test schedule. Cases are described in Table 6.40.

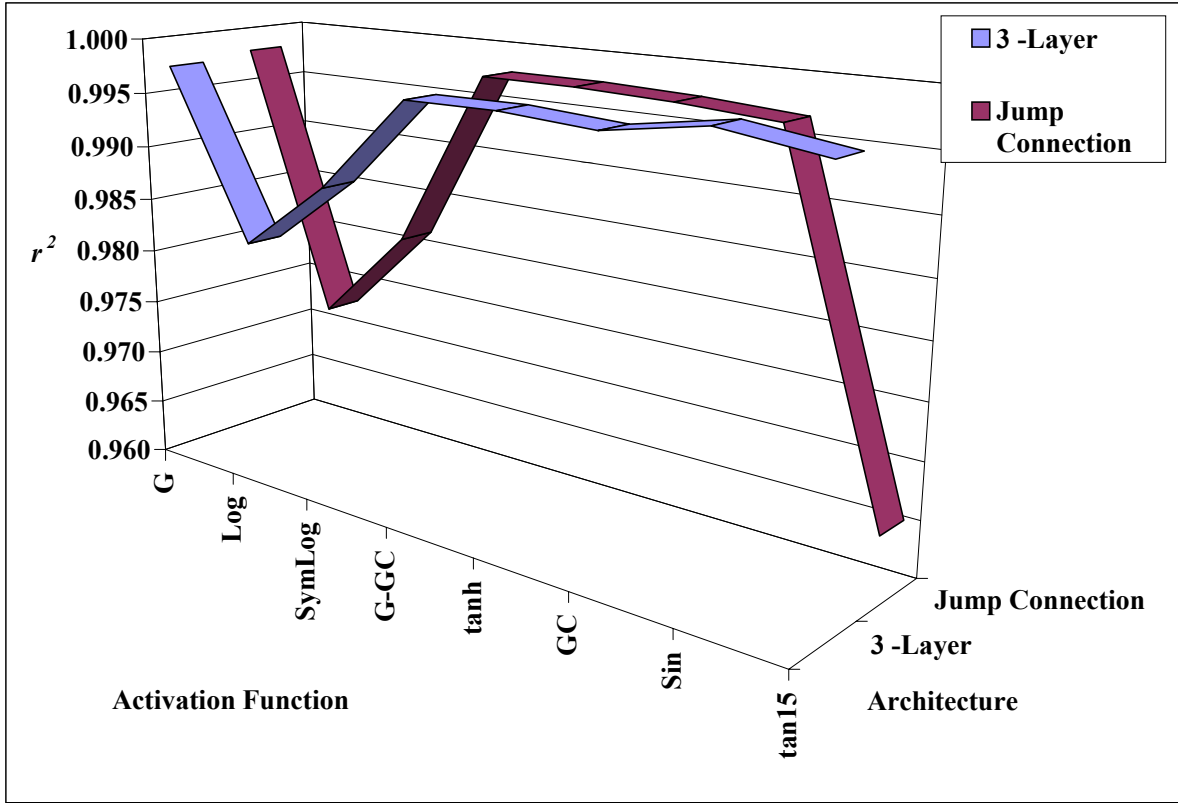


Figure 9.51. Variation of r^2 in the plane of activation function and architecture. The ANN was trained on an E-Highway predicting CO₂ emissions of the same test schedule. Cases are described in Table 6.41.

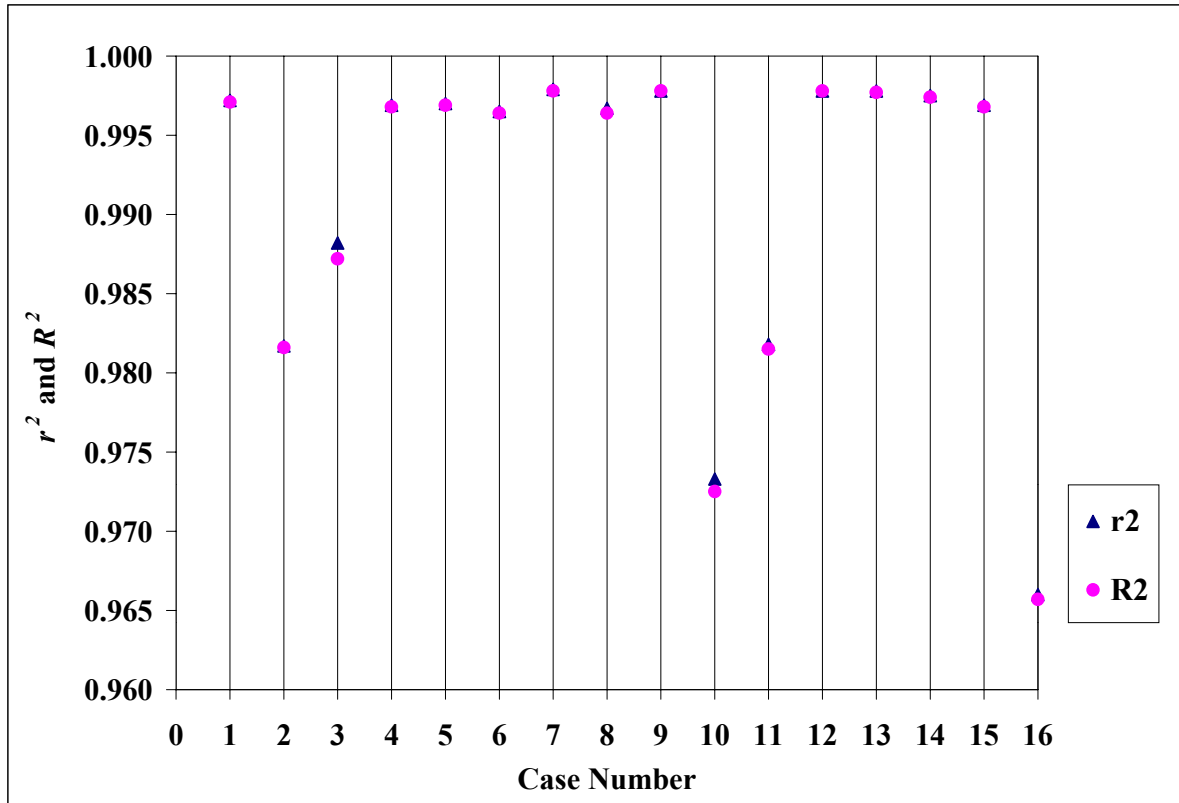


Figure 9.52. Variation of r^2 and R^2 with all cases. The ANN was trained on an E-Highway predicting CO₂ emissions of the same test schedule. Cases are described in Table 6.41.

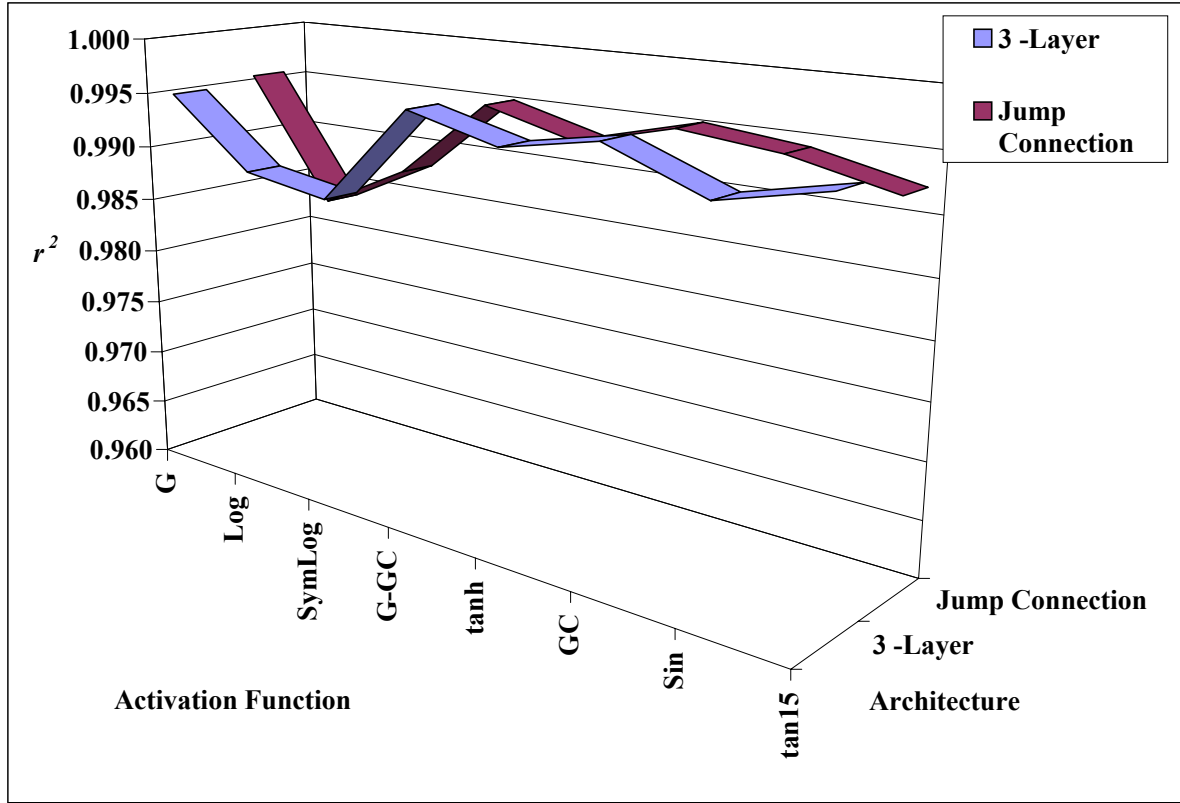


Figure 9.53. Variation of r^2 in the plane of activation function and architecture. The ANN was trained on an E-WVU-5 Peak predicting CO₂ emissions of the same test schedule. Cases are described in Table 6.42.

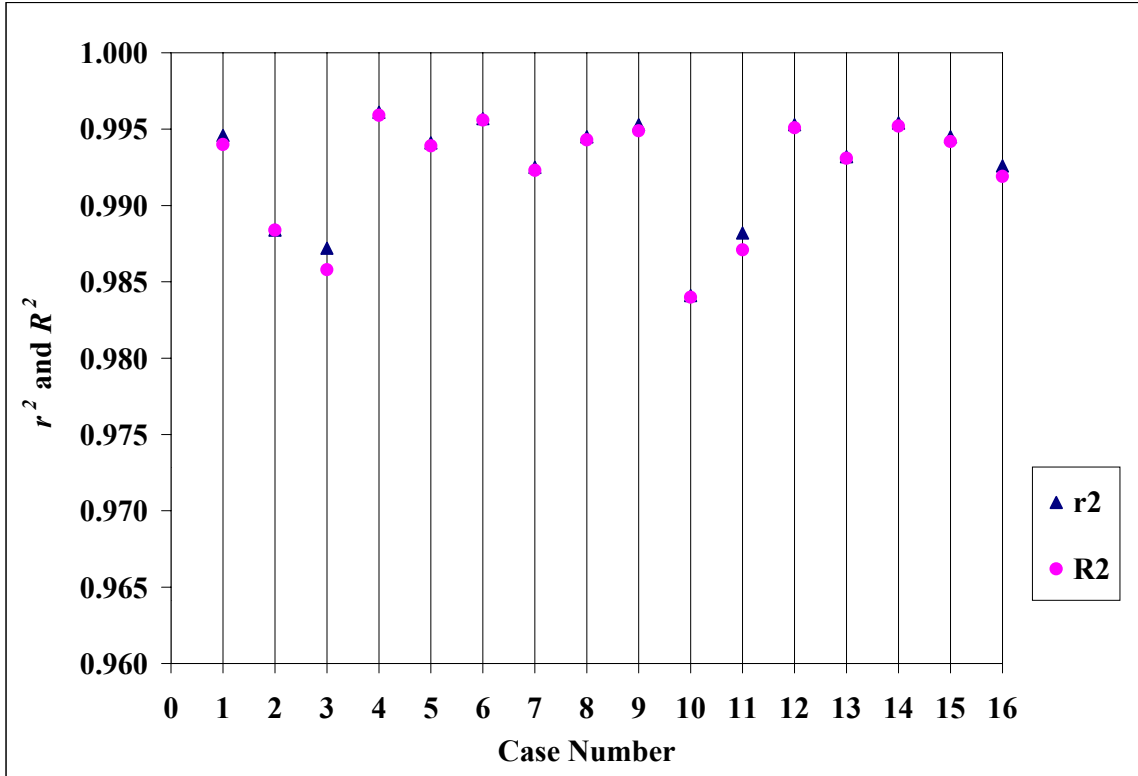


Figure 9.54. Variation of r^2 and R^2 with all cases. The ANN was trained on an E-WVU-5 Peak predicting CO₂ emissions of the same test schedule. Cases are described in Table 6.42.

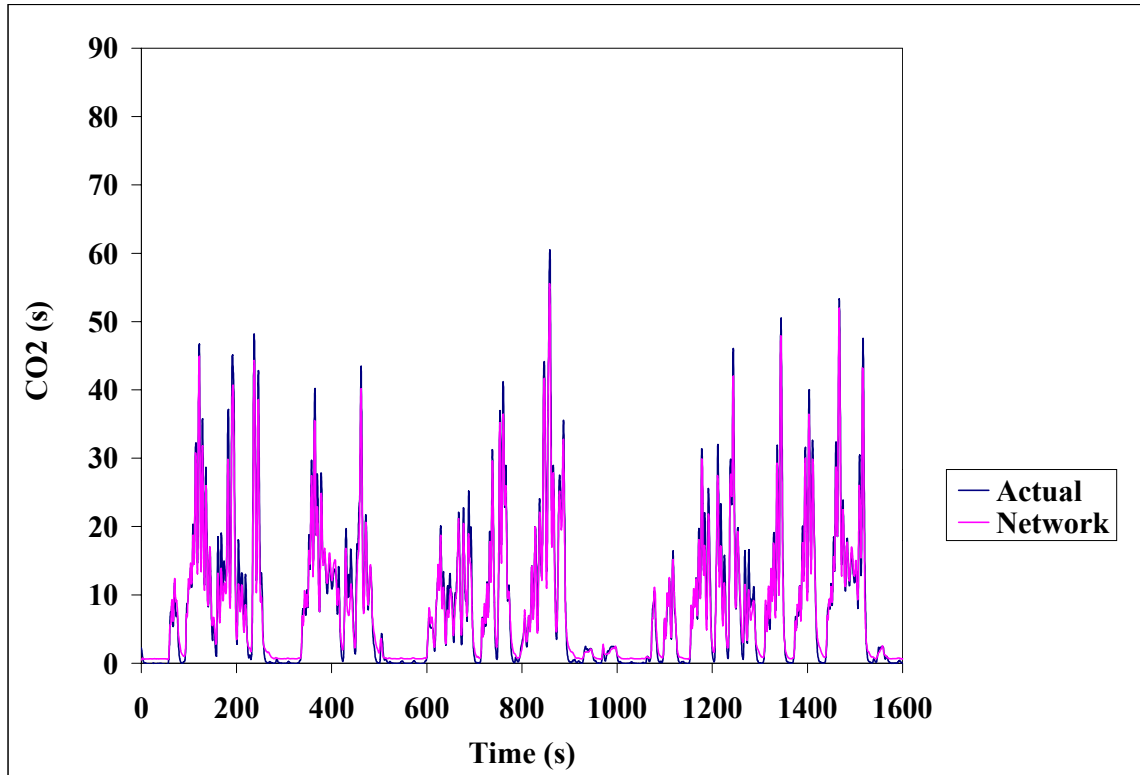


Figure 9.55. Second by second prediction of CO₂ in an E-CSHVR test schedule by a 3-Layer network of both Gaussian and Gaussian Complement activation functions. The ANN was trained on all the transient test schedules combined.

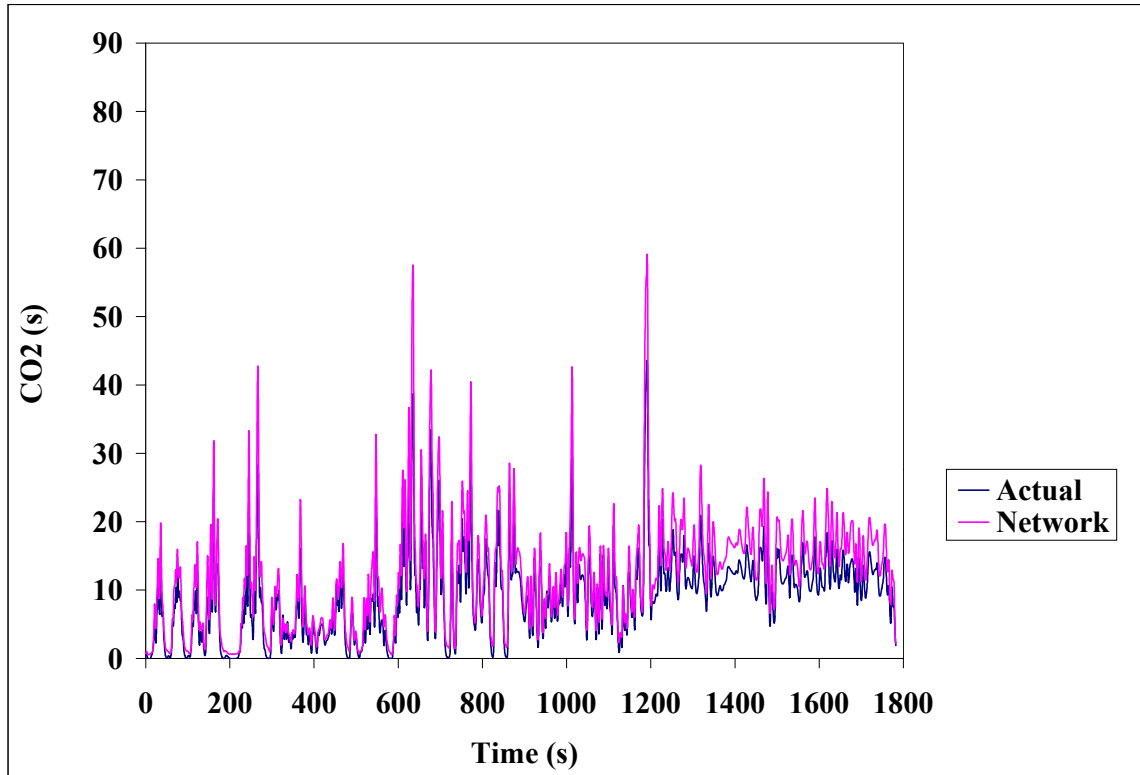


Figure 9.56. Second by second prediction of CO₂ in an ETC test schedule by a 3-Layernetwork of both Gaussian and Gaussian Complement activation functions. The ANN was trained on all the transient test schedules combined.

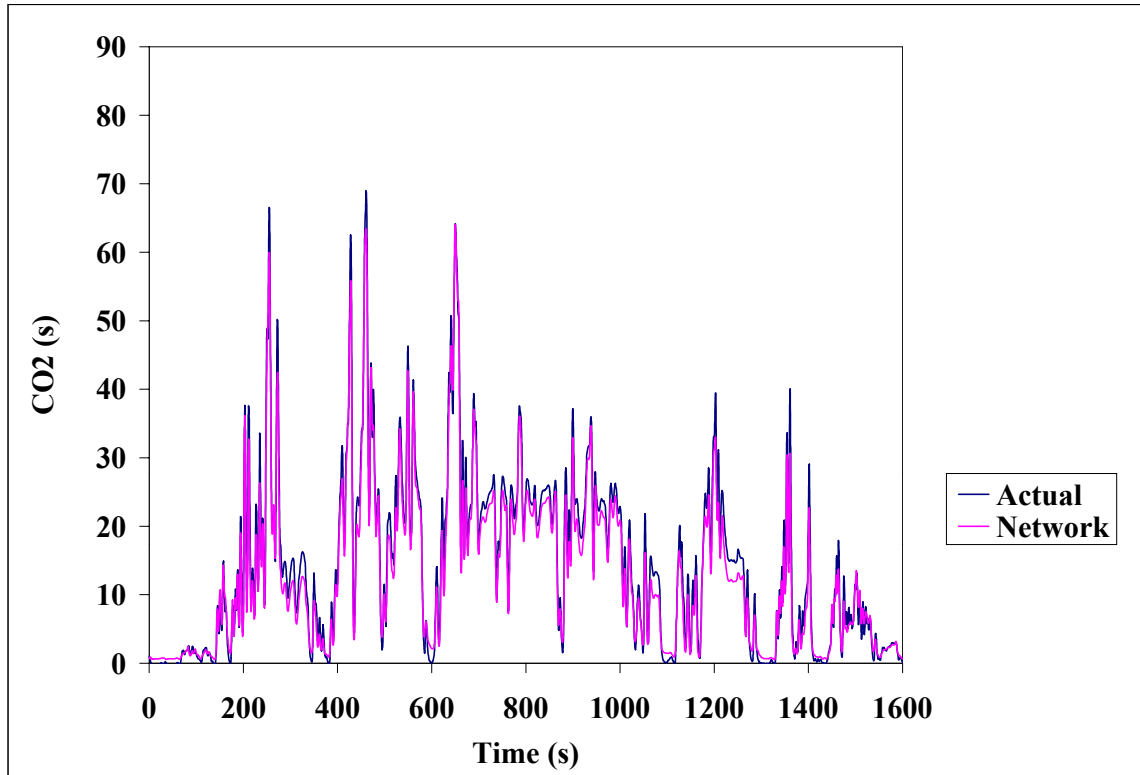


Figure 9.57. Second by second prediction of CO₂ in an E-Highway test schedule by a 3-Layer network of both Gaussian and Gaussian Complement activation functions. The ANN was trained on all the transient test schedules combined.

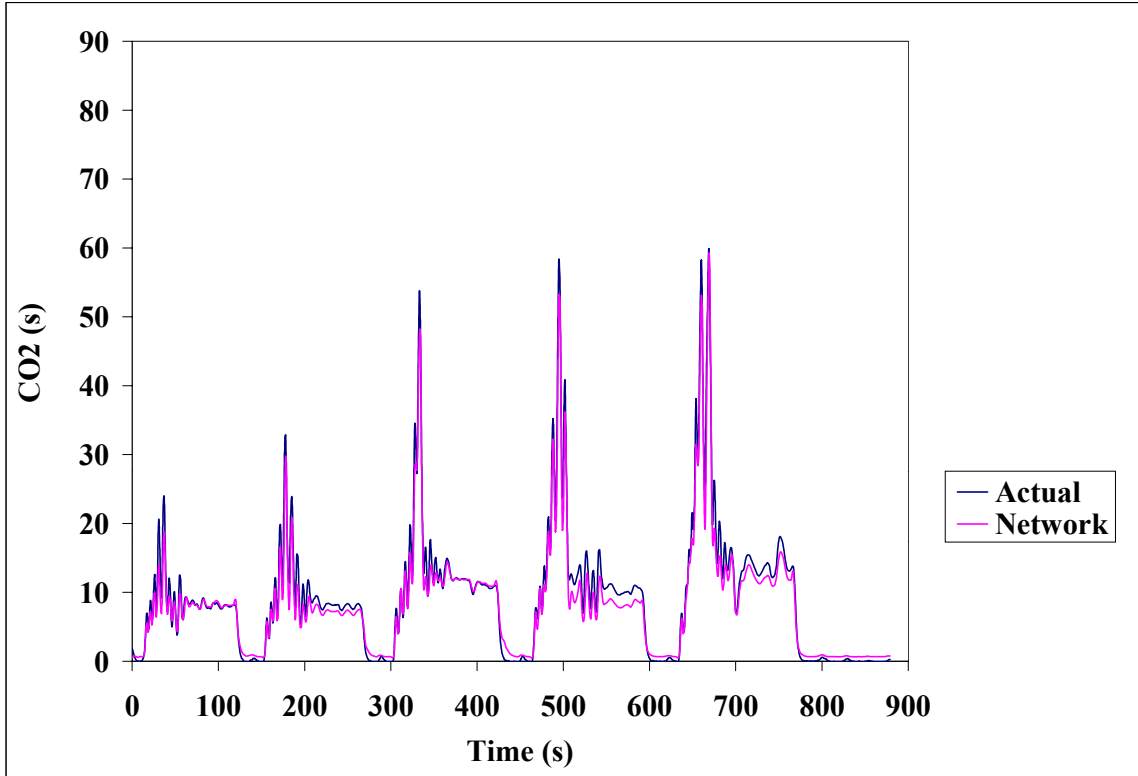


Figure 9.58. Second by second prediction of CO₂ in an E-WVU-5 Peak test schedule by a 3-Layer network of both Gaussian and Gaussian Complement activation functions. The ANN was trained on all the transient test schedules combined.

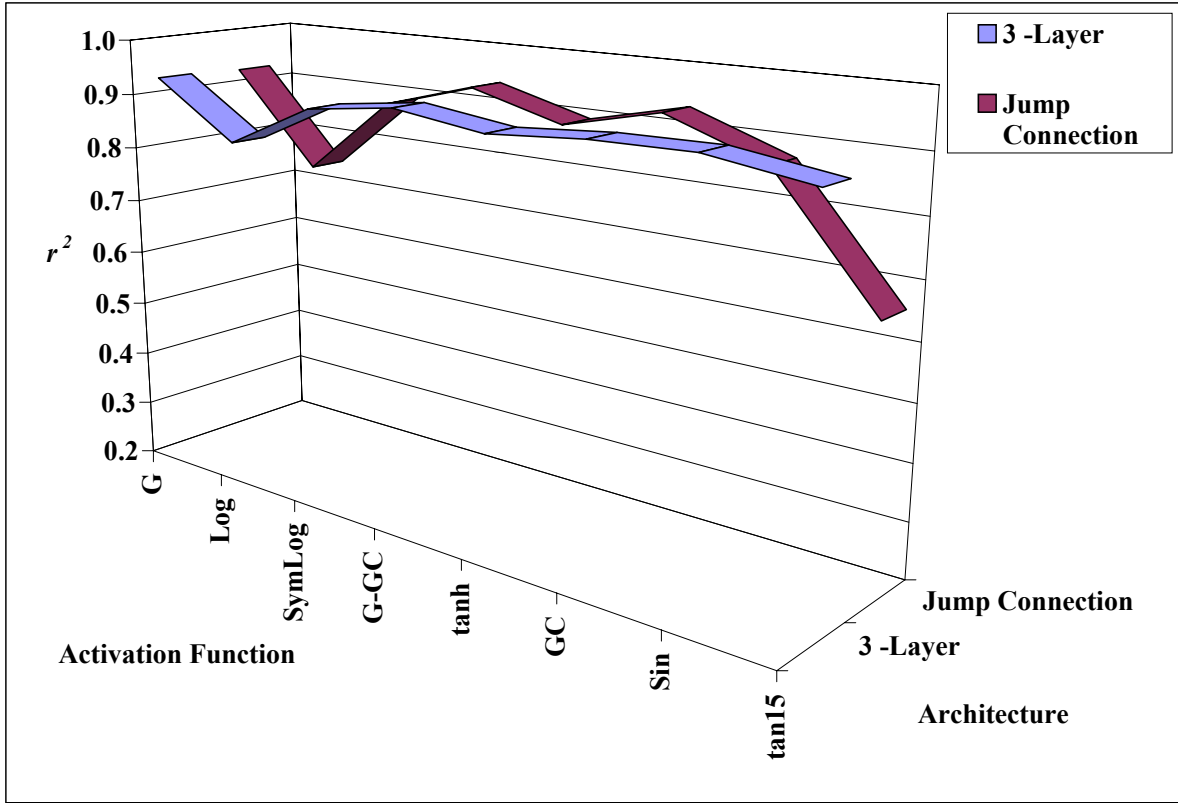


Figure 9.59. Variation of r^2 in the plane of activation function and architecture. The ANN was trained on an E-CSHVR predicting HC emissions of the same test schedule. Cases are described in Table 6.45.

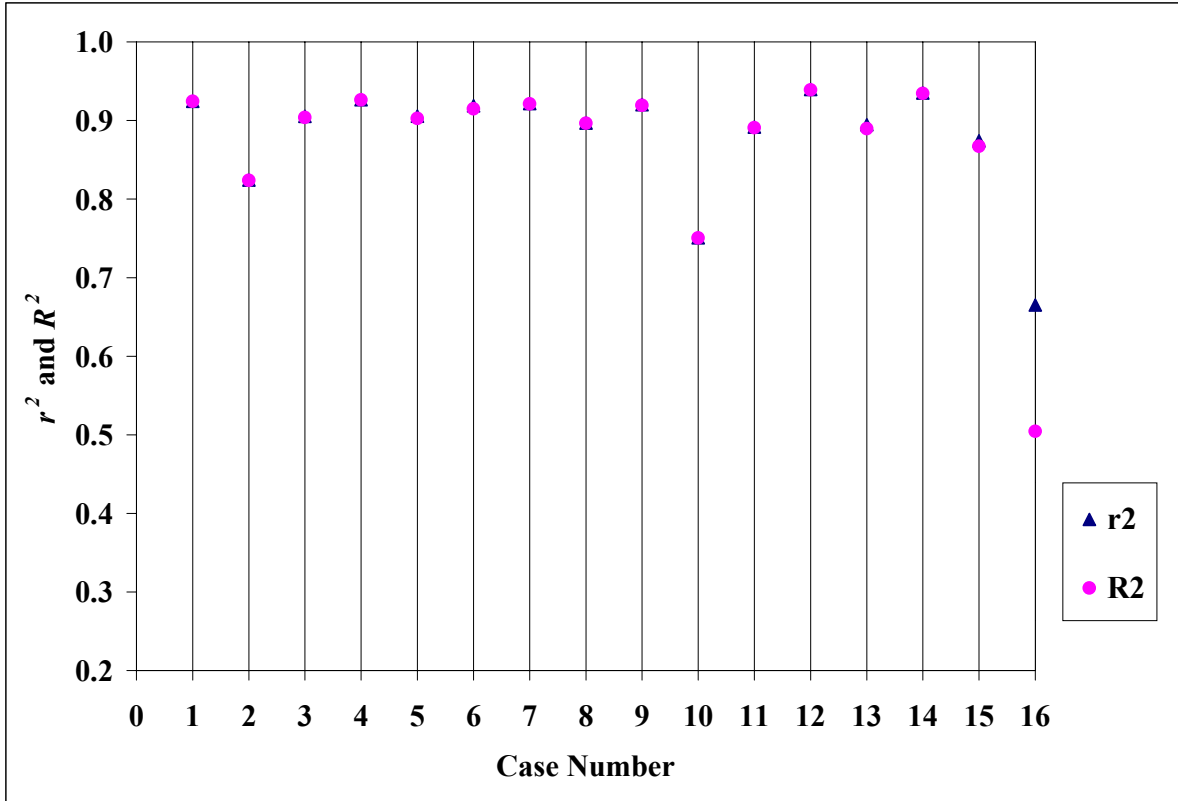


Figure 9.60. Variation of r^2 and R^2 with all cases. The ANN was trained on an E-CSHVR predicting HC emissions of the same test schedule. Cases are described in Table 6.45.

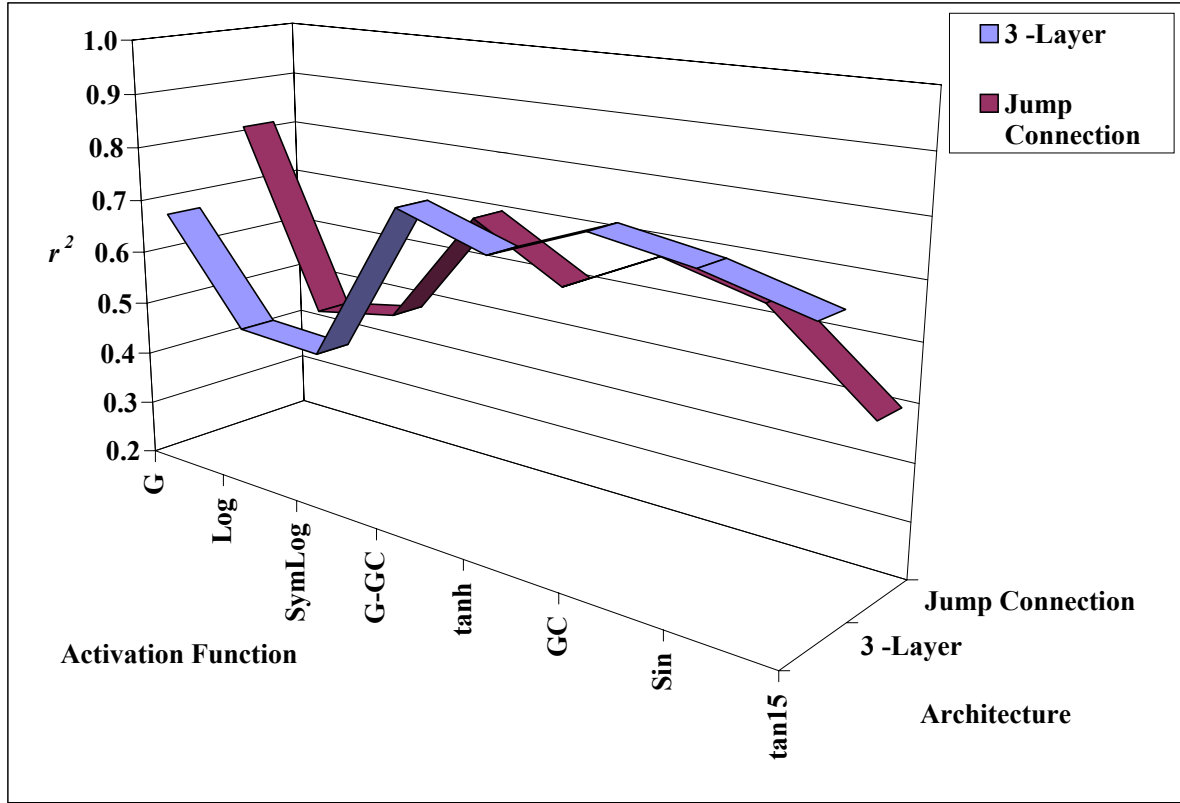


Figure 9.61. Variation of r^2 in the plane of activation function and architecture. The ANN was trained on an ETC predicting HC emissions of the same test schedule. Cases are described in Table 6.46.

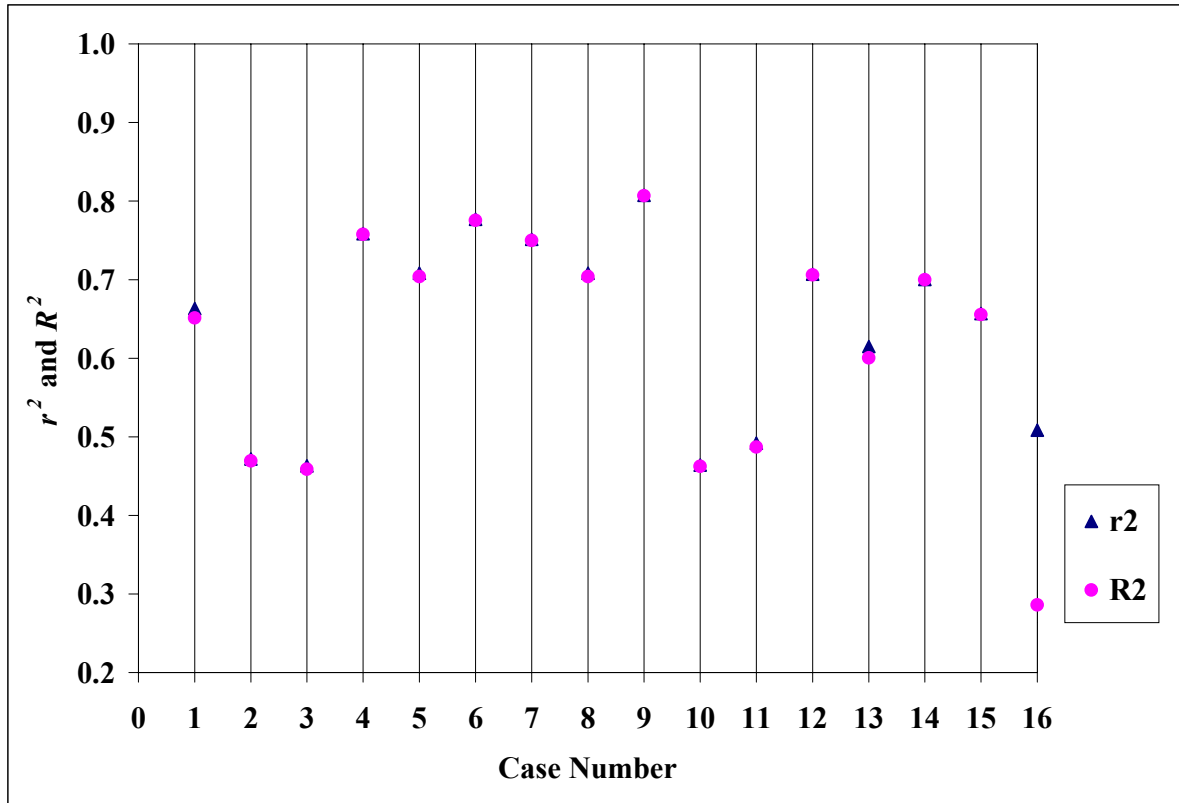


Figure 9.62. Variation of r^2 and R^2 with all cases. The ANN was trained on an ETC predicting HC emissions of the same test schedule. Cases are described in Table 6.46.

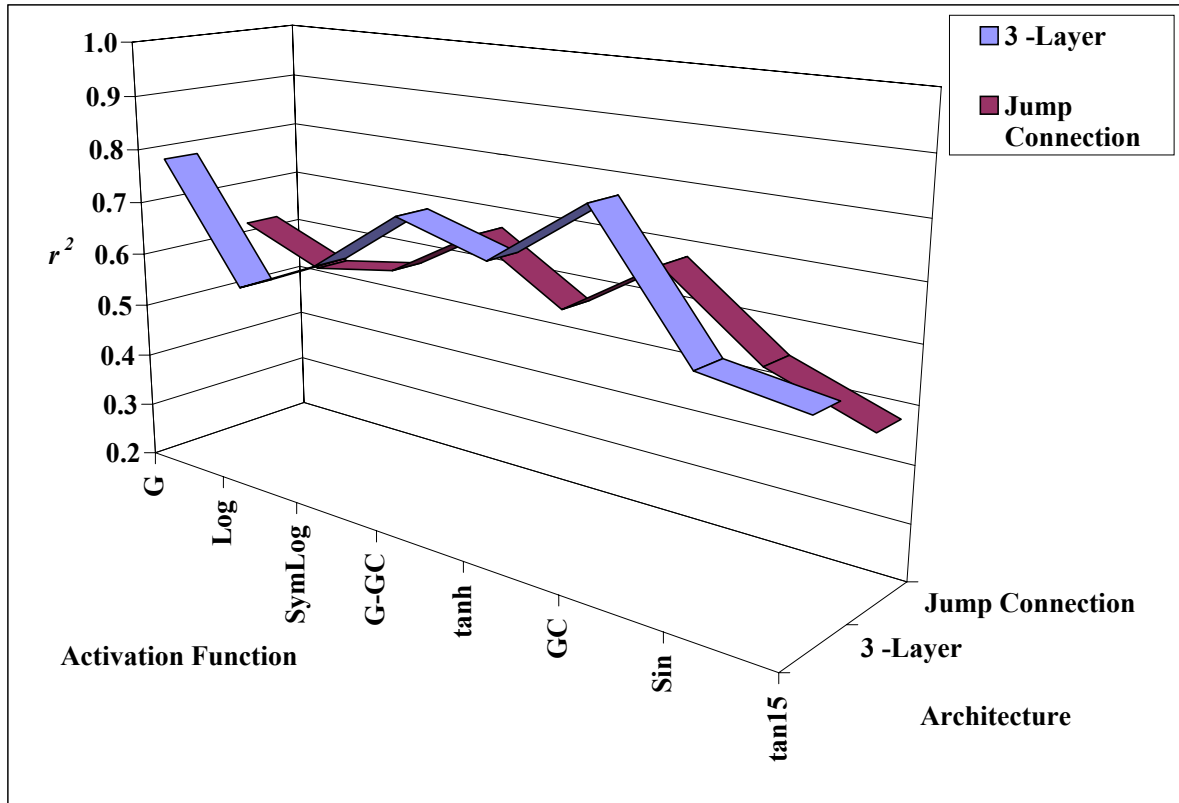


Figure 9.63. Variation of r^2 in the plane of activation function and architecture. The ANN was trained on a FTP predicting HC emissions of the same test schedule. Cases are described in Table 6.47.

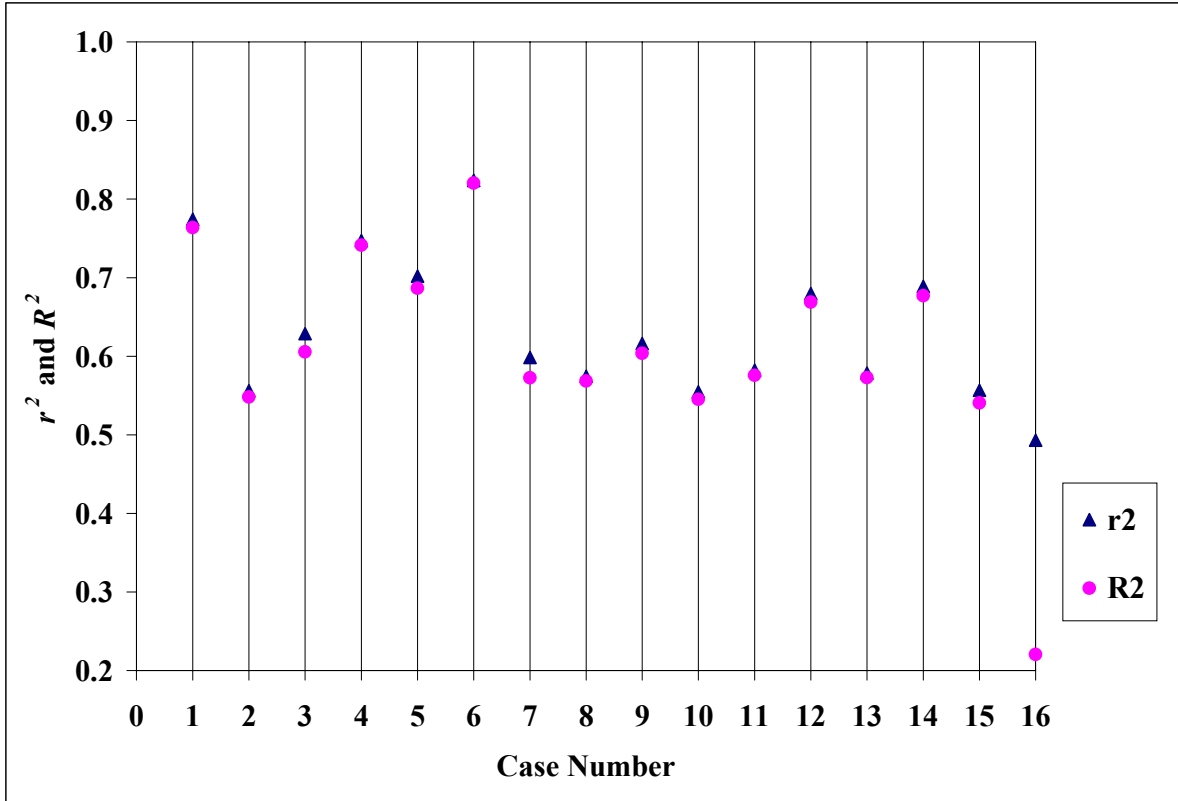


Figure 9.64. Variation of r^2 and R^2 with all cases. The ANN was trained on a FTP predicting HC emissions of the same test schedule. Cases are described in Table 6.47.

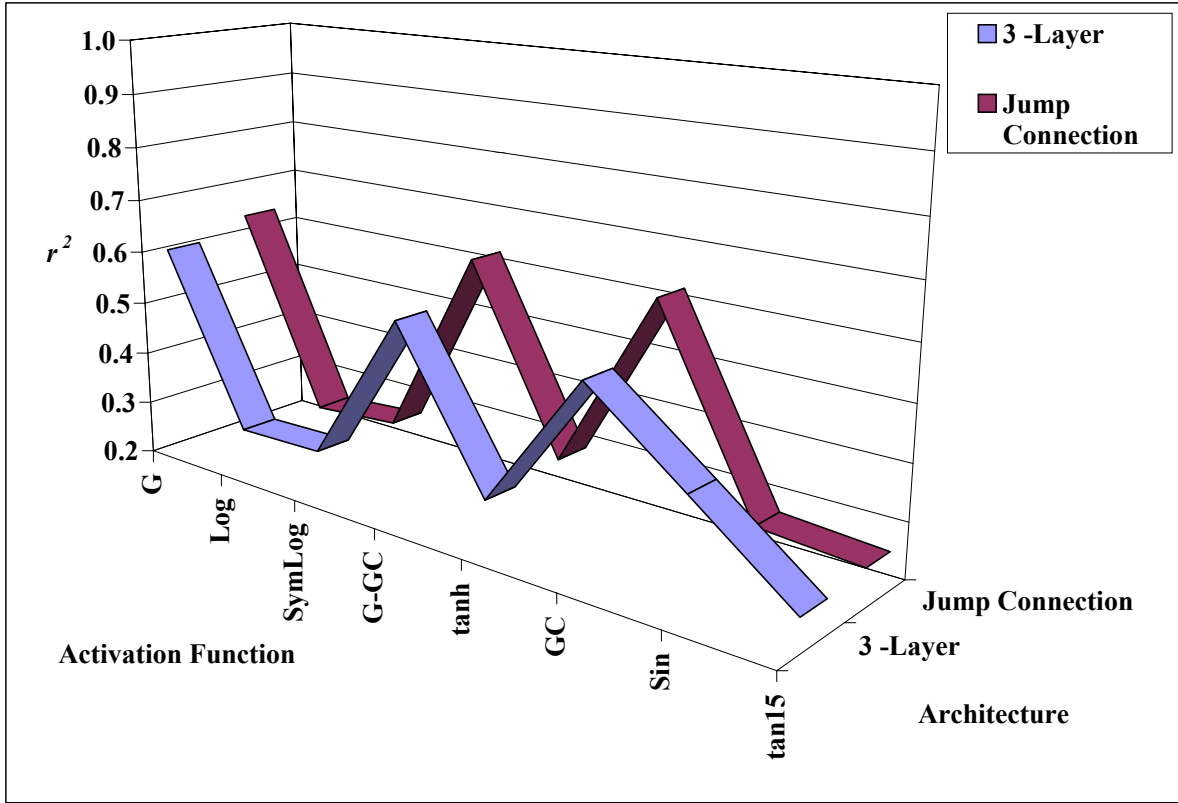


Figure 9.65. Variation of r^2 in the plane of activation function and architecture. The ANN was trained on an E-Highway predicting HC emissions of the same test schedule. Cases are described in Table 6.48.

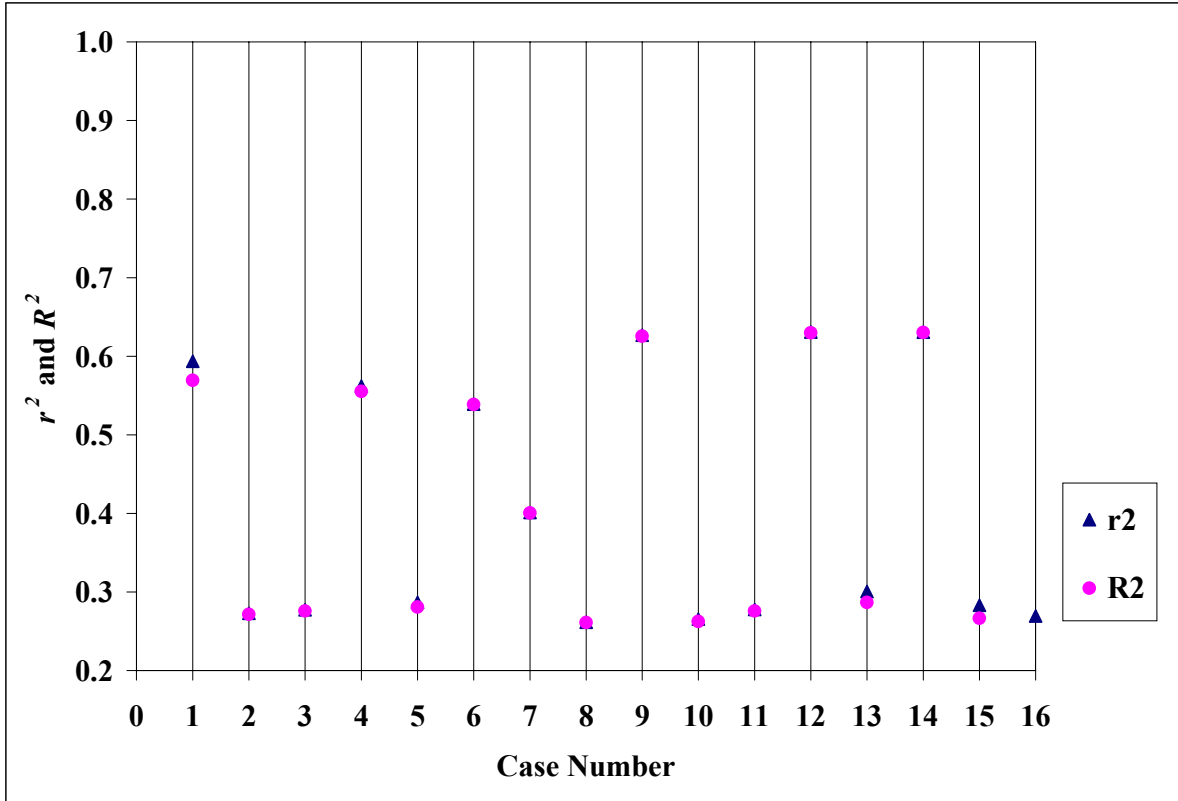


Figure 9.66. Variation of r^2 and R^2 with all cases. The ANN was trained on an E-Highway predicting HC emissions of the same test schedule. Cases are described in Table 6.48.

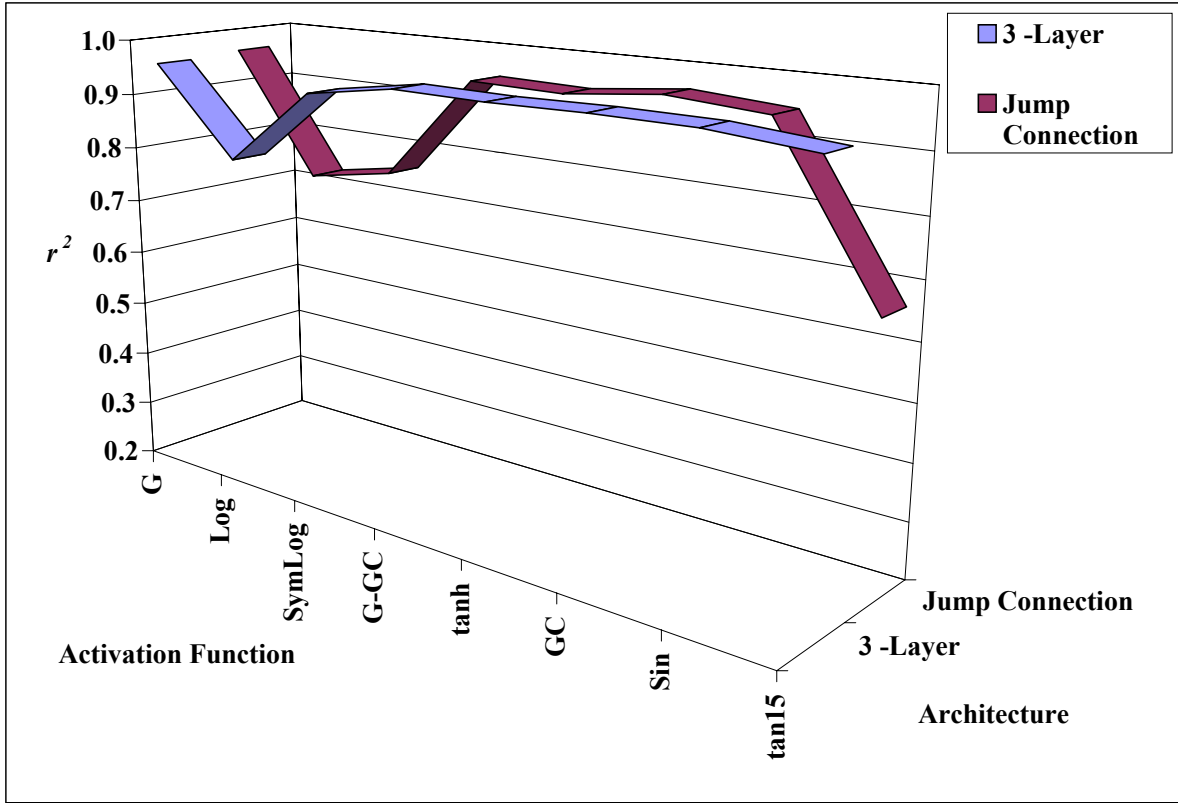


Figure 9.67. Variation of r^2 in the plane of activation function and architecture. The ANN was trained on an E-WVU-5 Peak predicting HC emissions of the same test schedule. Cases are described in Table 6.49.

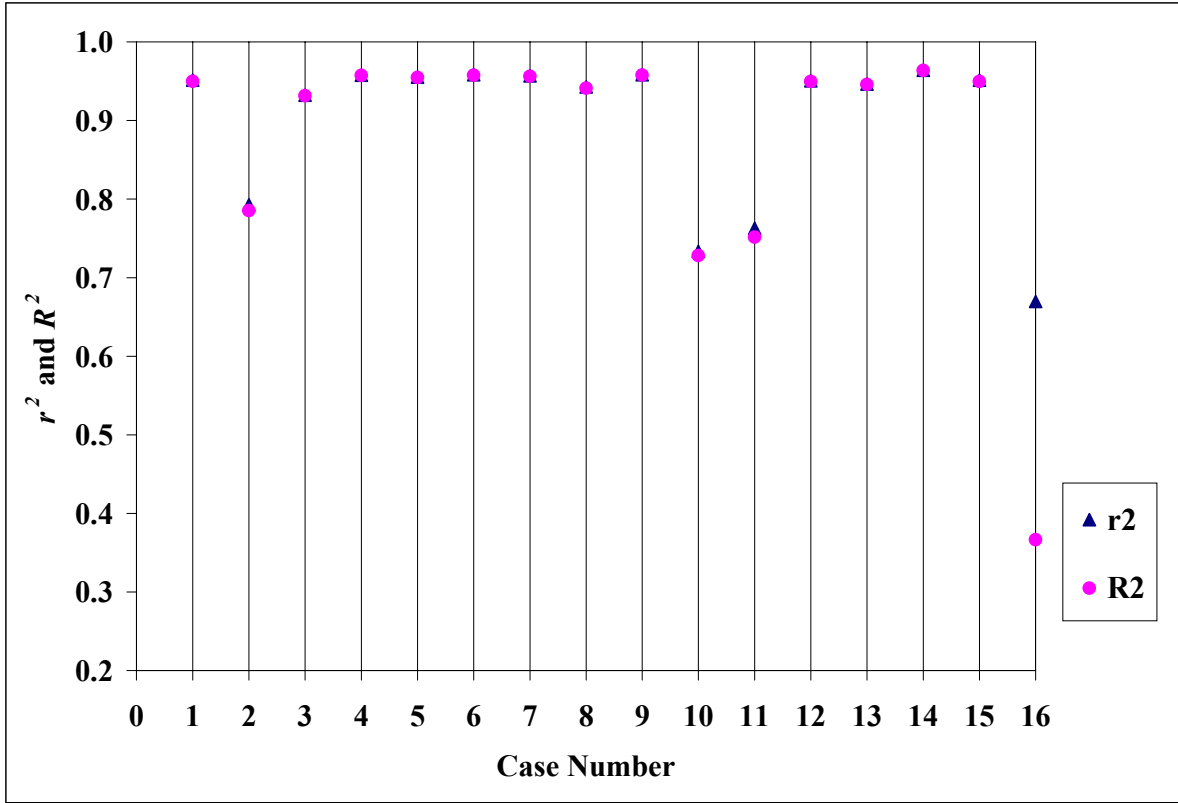


Figure 9.68. Variation of r^2 and R^2 with all cases. The ANN was trained on an E-WVU-5 Peak predicting HC emissions of the same test schedule. Cases are described in Table 6.49.

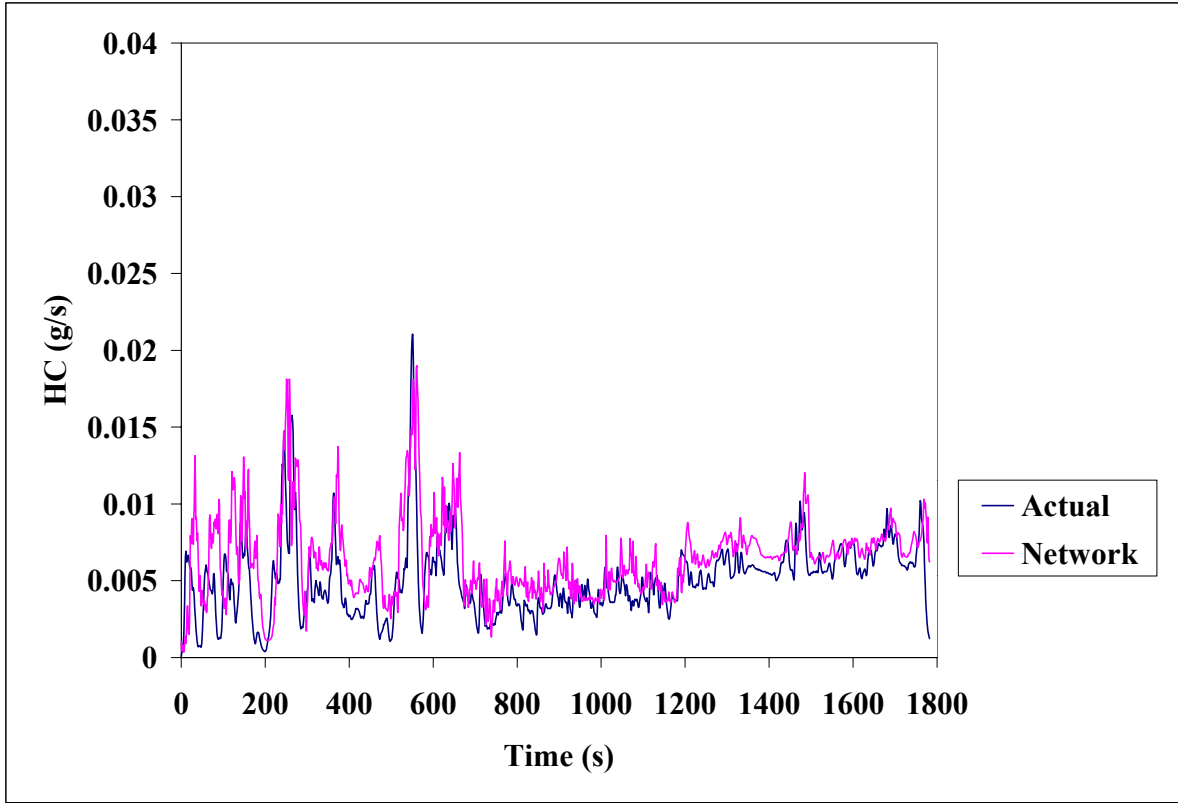


Figure 9.69. Second by second prediction of HC in an ETC test schedule by a Jump Connection network of both Gaussian and Gaussian Complement activation functions. The ANN was trained on all the transient test schedules combined.

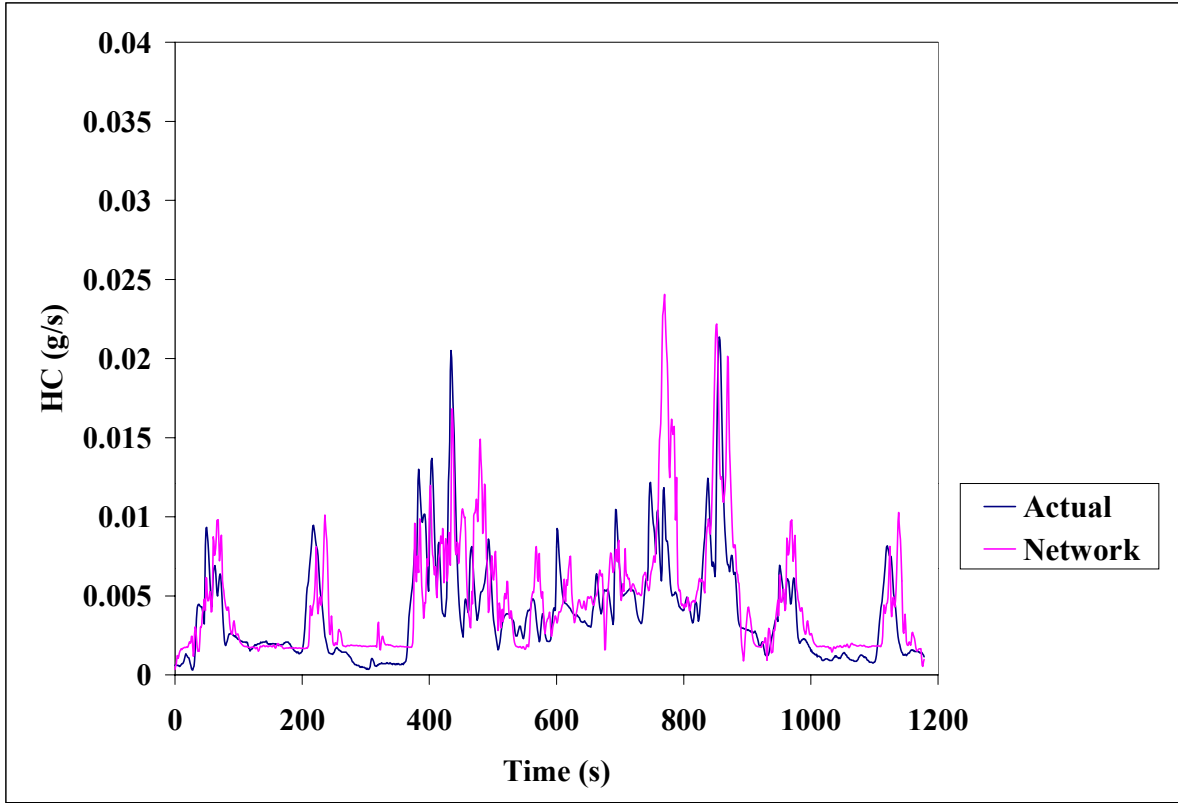


Figure 9.70. Second by second prediction of HC in a FTP test schedule by a Jump Connection network of both Gaussian and Gaussian Complement activation functions. The ANN was trained on all the transient test schedules combined.

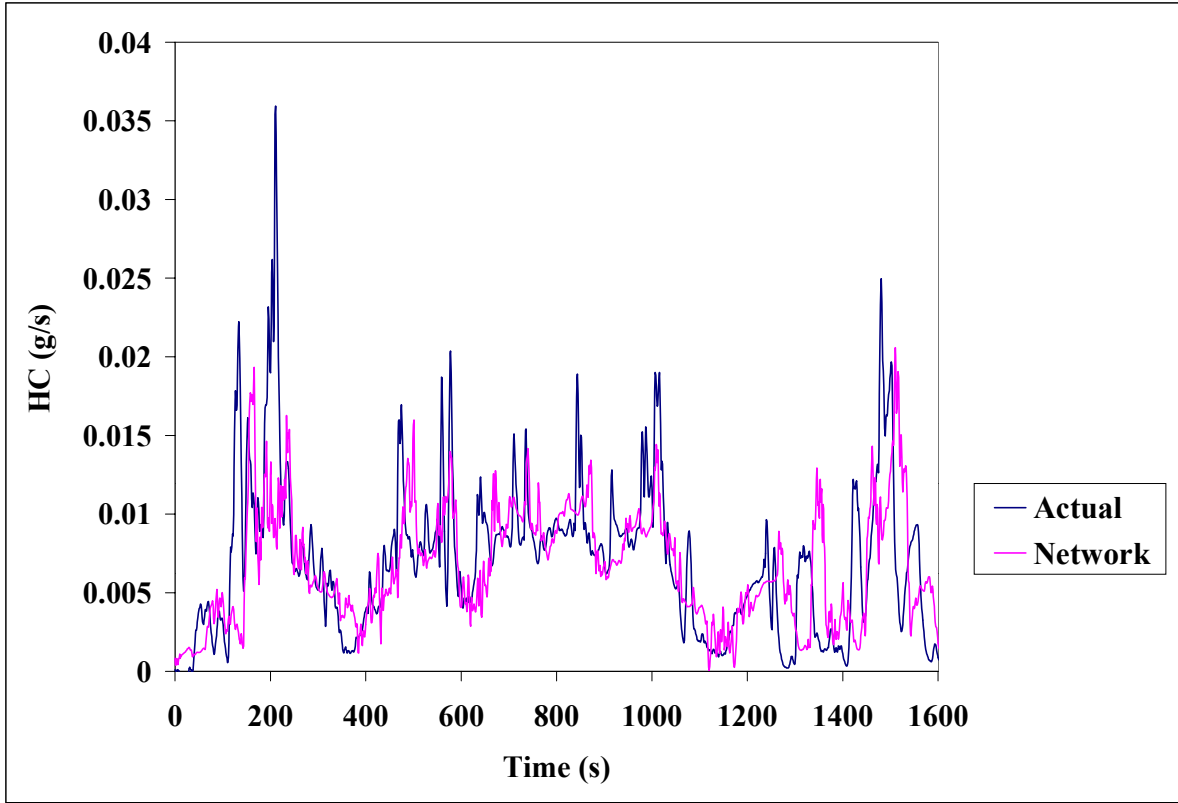


Figure 9.71. Second by second prediction of HC in an E-Highway test schedule by a Jump Connection network of both Gaussian and Gaussian Complement activation functions. The ANN was trained on all the transient test schedules combined.

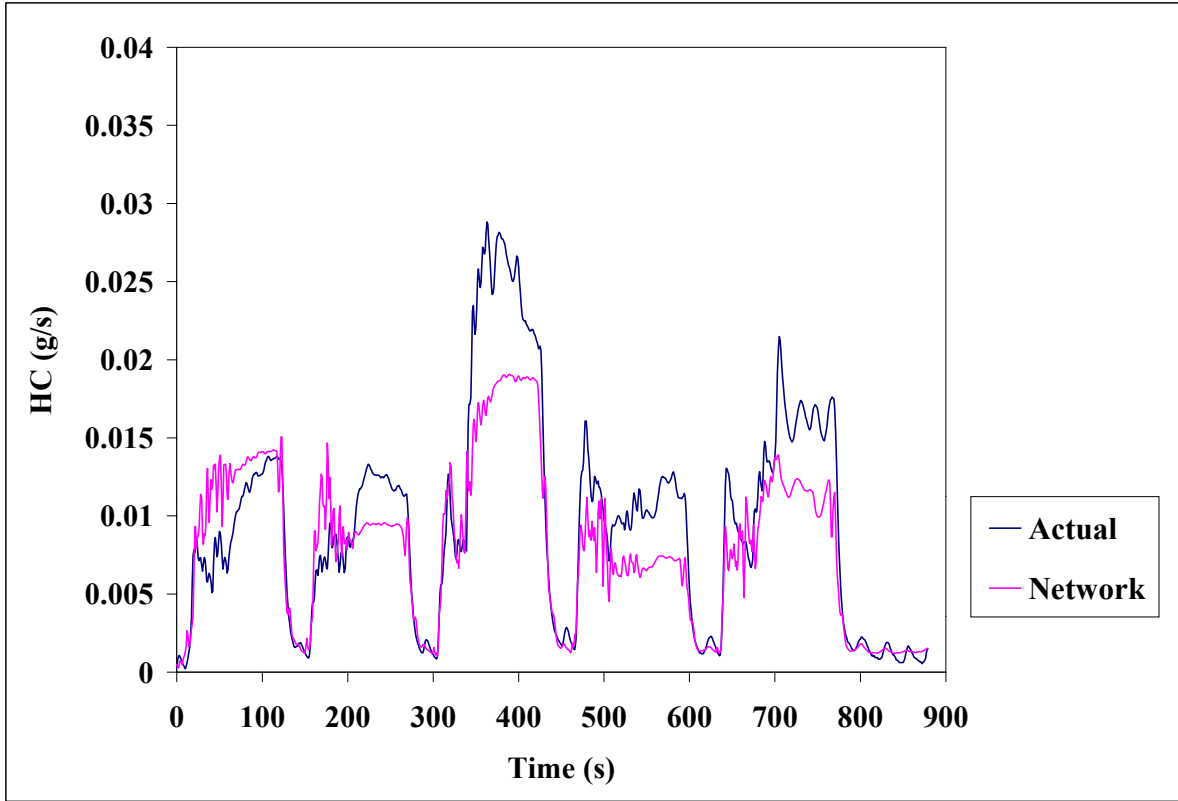


Figure 9.72. Second by second prediction of HC in an E-WVU-5 Peak test schedule by a Jump Connection network of both Gaussian and Gaussian Complement activation functions. The ANN was trained on all the transient test schedules combined.

10. APPENDIX B: GMDH FORMULAS

In the following subtitles, the relationship between torque, speed and their derivatives over time, with the emissions are presented in terms of the GMDH outputs. Outputs are NO_x, TEOM, CO, CO₂ and HC.

10.1. GMDH NO_x Prediction Formula

$$\begin{aligned}
 Y = & 0.17X_{10} + 0.46X_2 - 0.11X_{13} - 0.61X_9 - 8.4 \times 10^{-2} X_{12} + 0.37X_{11} - 0.44 - 6 \times 10^{-3} X_1 \\
 & - 0.11X_{14} + 0.9X_8 + 7 \times 10^{-2} X_5 - 0.21X_3 + 0.13X_2^2 + 0.39X_8^2 + 0.31X_9^2 + 0.41X_2X_8 \\
 & - 0.3X_8X_9 + 5.3 \times 10^{-2} X_1^2 + 4.1 \times 10^{-2} X_3^2 + 0.2X_{11}^2 - 7.6 \times 10^{-2} X_{11}X_{12} - 0.11X_1X_8 \\
 & - 0.19X_3X_{10} - 0.16X_5X_{12} + 0.12X_{14}^2 - 0.19X_{11}X_{14} - 1 \times 10^{-2} X_{10}^2 + 0.16X_{13}^2 + 0.16X_{10}X_{13} \\
 & + 2.4 \times 10^{-2} X_6^2 - 0.24X_8X_{13} - 9 \times 10^{-2} X_{10}X_{11} - 0.22X_9X_{14} - 7.5 \times 10^{-2} X_6X_{10} - 0.16X_5X_{13} \\
 & + 0.14X_1X_{11} + 0.13X_3X_9 + 9.3 \times 10^{-2} X_5X_{11} + 6 \times 10^{-2} X_8X_{12} - 6 \times 10^{-2} X_2X_9 - 0.11X_3X_{13}
 \end{aligned}$$

$$X_1 = \frac{2(\text{Disp} \cdot \text{Spd} - 224.06)}{1943.85} - 1$$

$$X_2 = \frac{2\left(\frac{dS}{dt}(1s) + 279.74\right)}{518.46} - 1$$

$$X_3 = \frac{2\left(\frac{d^2S}{dt^2}(1s) + 100.55\right)}{195.19} - 1$$

$$X_4 = \frac{2\left(\frac{dS}{dt}(5s) + 231.78\right)}{411.36} - 1$$

$$X_5 = \frac{2\left(\frac{d^2S}{dt^2}(5s) + 53.82\right)}{121.01} - 1$$

$$X_6 = \frac{2\left(\frac{dS}{dt}(10s) + 130.55\right)}{257.58} - 1$$

$$X_7 = \frac{2\left(\frac{d^2S}{dt^2}(10s) + 40.17\right)}{74.19} - 1$$

$$X_8 = \frac{2(\text{Disp} \cdot T + 324.28)}{2227.6} - 1$$

$$X_9 = \frac{2\left(\frac{dT}{dt}(1s) + 332.28\right)}{671.2} - 1$$

$$X_{10} = \frac{2\left(\frac{d^2T}{dt^2}(1s) + 191.31\right)}{333.48} - 1$$

$$X_{11} = \frac{2\left(\frac{dT}{dt}(5s) + 268.13\right)}{541.79} - 1$$

$$X_{12} = \frac{2\left(\frac{d^2T}{dt^2}(5s) + 109.04\right)}{190.32} - 1$$

$$X_{13} = \frac{2\left(\frac{dT}{dt}(10s) + 160.69\right)}{343.23} - 1$$

$$X_{14} = \frac{2\left(\frac{d^2T}{dt^2}(10s) + 54.45\right)}{95.38} - 1$$

$$Y = \frac{2(NOx(g/s))}{.44} - 1$$

10.2. GMDH TEOM Prediction Formula

$$\begin{aligned}
 Y = & -9 \times 10^{-2} X_{11} - 0.17 - 2.6 \times 10^{-2} X_1 + 0.25 X_8 + 0.14 X_4 - 3.9 \times 10^{-2} X_{13} + 6.3 \times 10^{-2} X_3 \\
 & + 0.47 X_6 + 0.3 X_5 - 6.2 \times 10^{-2} X_{14} - 0.2 X_7 + 0.11 X_{10} + 2.2 \times 10^{-2} X_9 - 5.2 \times 10^{-2} X_{12} \\
 & + 8.4 \times 10^{-2} X_6^2 + 8.3 \times 10^{-2} X_8^2 - 0.11 X_{13}^2 + 0.55 X_6 X_8 - 0.94 X_6 X_{13} + 0.24 X_1^2 \\
 & + 4.3 \times 10^{-2} X_{14}^2 + 0.34 X_1 X_{13} - 0.49 X_1 X_{14} + 3.8 \times 10^{-3} X_1 X_8 + 0.26 X_8 X_{13} - 0.1 X_8 X_{14} \\
 & - 0.11 X_1^2 X_8 - 0.25 X_8 X_{13}^2 - 0.15 X_8 X_{14}^2 - 0.45 X_1 X_8 X_{13} - 0.5 X_1 X_8 X_{14} + 0.39 X_{11}^2 \\
 & - 0.29 X_6 X_{11} - 6.8 \times 10^{-2} X_8 X_{11} + 7.1 \times 10^{-2} X_1 X_8^2 + 0.74 X_1 X_{13}^2 + 0.26 X_1 X_{14}^2 \\
 & + 0.43 X_1 X_{13} X_{14} - 0.24 X_1 X_6 + 0.19 X_1 X_{11} - 0.33 X_1 X_{11}^2 - 0.19 X_1 X_6 X_8 + 0.4 X_1 X_6 X_{11} \\
 & + 8.1 \times 10^{-2} X_1 X_8 X_{11} + 0.42 X_5^2 + 0.11 X_3^2 - 1.1 \times 10^{-2} X_{10}^2 - 0.76 X_3 X_{10} + 0.28 X_4^2 \\
 & + 0.58 X_4 X_{13} - 0.24 X_3 X_4 - 0.21 X_4 X_{10} - 9.4 \times 10^{-2} X_4 X_5 - 8.2 \times 10^{-2} X_4 X_8 - 0.5 X_4 X_6 \\
 & - 2.7 \times 10^{-2} X_4 X_6^2 - 0.3 X_4 X_8^2 - 0.51 X_4 X_{13}^2 - 0.45 X_4 X_6 X_8 + 0.58 X_4 X_6 X_{13} - 0.11 X_1 X_4 \\
 & + 5.9 \times 10^{-2} X_4 X_{14} - 0.29 X_1^2 X_4 - 0.17 X_4 X_{14}^2 - 0.22 X_1 X_4 X_{13} + 0.52 X_1 X_4 X_{14} \\
 & - 0.13 X_1 X_4 X_8 + 9.8 \times 10^{-2} X_4 X_8 X_{13} - 5.5 \times 10^{-2} X_4 X_8 X_{14} + 0.11 X_1^2 X_4 X_8 \\
 & + 0.26 X_4 X_8 X_{13}^2 + 0.16 X_4 X_8 X_{14}^2 + 0.48 X_1 X_4 X_8 X_{13} + 0.53 X_1 X_4 X_8 X_{14} - 0.18 X_4 X_{11} \\
 & - 0.29 X_4 X_{11}^2 + 0.35 X_4 X_6 X_{11} + 7.1 \times 10^{-2} X_4 X_8 X_{11} - 7.5 \times 10^{-2} X_1 X_4 X_8^2 - 0.78 X_1 X_4 X_{13}^2 \\
 & - 0.28 X_1 X_4 X_{14}^2 - 0.45 X_1 X_4 X_{13} X_{14} + 0.25 X_1 X_4 X_6 - 4.2 \times 10^{-2} X_1 X_4 X_{11} + 0.34 X_1 X_4 X_{11}^2 \\
 & + 0.19 X_1 X_4 X_6 X_8 - 0.42 X_1 X_4 X_6 X_{11} - 8.4 \times 10^{-2} X_1 X_4 X_8 X_{11} - 0.61 X_4 X_5^2 - 0.18 X_3^2 X_4 \\
 & - 5.8 \times 10^{-2} X_4 X_{10}^2 + 1.2 X_3 X_4 X_{10} + 4.8 \times 10^{-2} X_{12}^2 - 0.62 X_5 X_{12} + 0.43 X_3 X_{11} + 0.29 X_7 X_{10} \\
 & + 0.27 X_7^2 - 0.27 X_5 X_7 + 0.15 X_3 X_{12} + 8.4 \times 10^{-2} X_1 X_{10} + 0.57 X_7 X_{13} - 0.31 X_5 X_{11} \\
 & + 9.8 \times 10^{-2} X_8 X_{12} - 0.52 X_7 X_{14} + 5.4 \times 10^{-2} X_9^2 + 0.21 X_5 X_9 + 0.1 X_{10} X_{14} - 0.21 X_7 X_8 \\
 & + 7.1 \times 10^{-2} X_{10} X_{11} - 0.17 X_9 X_{14} + 0.12 X_4 X_{12} + 0.2 X_7 X_{12} + 0.19 X_3 X_{14} + 9.8 \times 10^{-2} X_1 X_9 \\
 & - 9.7 \times 10^{-2} X_5 X_{14} + 9.5 \times 10^{-2} X_{10} X_{13} + 6.8 \times 10^{-2} X_6 X_{14} + 7.7 \times 10^{-2} X_5 X_{13} \\
 & - 5.4 \times 10^{-2} X_{11} X_{14} - 2.7 \times 10^{-2} X_{13} X_{14}
 \end{aligned}$$

Legend:

$$X_1 = \frac{2(\text{Disp} \cdot \text{Spd} - 224.06)}{1943.85} - 1$$

$$X_2 = \frac{2\left(\frac{dS}{dt}(1s) + 279.74\right)}{518.46} - 1$$

$$X_3 = \frac{2\left(\frac{d^2S}{dt^2}(1s) + 100.55\right)}{195.19} - 1$$

$$X_4 = \frac{2\left(\frac{dS}{dt}(5s) + 231.78\right)}{411.36} - 1$$

$$X_5 = \frac{2\left(\frac{d^2S}{dt^2}(5s) + 53.82\right)}{121.01} - 1$$

$$X_6 = \frac{2\left(\frac{dS}{dt}(10s) + 130.55\right)}{257.58} - 1$$

$$X_7 = \frac{2\left(\frac{d^2S}{dt^2}(10s) + 40.17\right)}{74.19} - 1$$

$$X_8 = \frac{2(\text{Disp } T + 324.28)}{2227.6} - 1$$

$$X_9 = \frac{2\left(\frac{dT}{dt}(1s) + 332.28\right)}{671.2} - 1$$

$$X_{10} = \frac{2\left(\frac{d^2T}{dt^2}(1s) + 191.31\right)}{333.48} - 1$$

$$X_{11} = \frac{2\left(\frac{dT}{dt}(5s) + 268.13\right)}{541.79} - 1$$

$$X_{12} = \frac{2\left(\frac{d^2T}{dt^2}(5s) + 109.04\right)}{190.32} - 1$$

$$X_{13} = \frac{2\left(\frac{dT}{dt}(10s) + 160.69\right)}{343.23} - 1$$

$$X_{14} = \frac{2\left(\frac{d^2T}{dt^2}(10s) + 54.45\right)}{95.38} - 1$$

$$Y = \frac{2(\text{TEOM}(g/s) + 0.02)}{0.04} - 1$$

10.3. GMDH CO Prediction Formula

$$\begin{aligned}
 Y = & 0.32X_8 - 0.23X_3 + 7.2 \times 10^{-2} X_4 + 9.1 \times 10^{-2} X_{11} - 2.8 \times 10^{-2} X_7 - 0.25X_{14} \\
 & + 6.3 \times 10^{-2} X_6 + 1.7 \times 10^{-2} X_2 - 0.18X_{12} + 0.3X_5 - 0.64 - 0.1X_{13} + 3.9 \times 10^{-2} X_1 + 0.22X_{10} \\
 & - 7.3 \times 10^{-2} X_9 + 0.35X_1^2 + 0.16X_8^2 + 0.28X_{12}^2 - 0.37X_1X_8 + 0.23X_8X_{12} + 0.4X_{11}^2 \\
 & + 0.3X_{14}^2 + 0.25X_1X_{11} + 6.6 \times 10^{-2} X_1X_{14} - 0.6X_{11}X_{14} + 0.15X_{10}^2 - 0.28X_1X_{10} \\
 & + 0.23X_8X_{10} - 0.64X_{10}X_{12} + 5.1 \times 10^{-2} X_1^2X_{10} + 5 \times 10^{-2} X_8^2X_{10} + 0.29X_{10}X_{12}^2 \\
 & - 0.2X_1X_8X_{10} + 0.27X_8X_{10}X_{12} - 0.38X_{10}X_{11} + 0.48X_{10}X_{14} - 0.98X_{10}X_{11}^2 - 0.56X_{10}X_{14}^2 \\
 & - 0.25X_1X_{10}X_{11} + 4.5 \times 10^{-2} X_1X_{10}X_{14} + 1.3X_{10}X_{11}X_{14} + 0.3X_5^2 - 0.35X_5X_{14} \\
 & + 3.8 \times 10^{-2} X_5X_{10} + 0.11X_1X_5 + 0.15X_5X_8 - 0.76X_5X_{12} + 0.15X_1^2X_5 + 2.5 \times 10^{-2} X_5X_8^2 \\
 & + 0.14X_5X_{12}^2 - 0.1X_1X_5X_8 + 0.14X_5X_8X_{12} + 0.4X_5X_{11} + 0.26X_5X_{11}^2 + 0.15X_5X_{14}^2 \\
 & + 6.5 \times 10^{-2} X_1X_5X_{11} - 1.2 \times 10^{-2} X_1X_5X_{14} - 0.35X_5X_{11}X_{14} + 0.16X_5X_{10}^2 - 0.13X_1X_5X_{10} \\
 & + 0.18X_5X_8X_{10} - 0.13X_5X_{10}X_{12} + 3 \times 10^{-2} X_1^2X_5X_{10} + 2.9 \times 10^{-2} X_5X_8^2X_{10} \\
 & + 0.17X_5X_{10}X_{12}^2 - 0.12X_1X_5X_8X_{10} + 0.16X_5X_8X_{10}X_{12} - 0.26X_5X_{10}X_{11} + 0.28X_5X_{10}X_{14} \\
 & - 0.57X_5X_{10}X_{11}^2 - 0.33X_5X_{10}X_{14}^2 - 0.14X_1X_5X_{10}X_{11} + 2.6 \times 10^{-2} X_1X_5X_{10}X_{14} \\
 & + 0.77X_5X_{10}X_{11}X_{14} + 0.26X_3^2 + 0.6X_3X_{11} + 0.15X_1X_{13} + 0.19X_7^2 - 0.35X_5X_7 + 0.12X_2^2 \\
 & - 0.42X_2X_{10} - 0.16X_{11}X_{12} + 0.19X_{13}^2 + 0.29X_9X_{13} + 0.95X_3X_{12} + 0.28X_9^2 - 0.74X_9X_{11} \\
 & + 0.2X_8X_{14} + 0.27X_4X_9 + 5.9 \times 10^{-2} X_6^2 + 6.7 \times 10^{-2} X_6X_{11} - 0.24X_6X_{14} - 0.26X_3X_{10} \\
 & - 0.39X_8X_{13} + 0.14X_4X_7 + 0.11X_4^2 + 0.15X_4X_8 - 0.44X_5X_{13} + 0.16X_4X_{10} + 0.13X_7X_{13} \\
 & + 0.13X_1X_3 + 0.19X_7X_9 - 0.17X_4X_6 - 6.6 \times 10^{-2} X_1X_6 - 5.4 \times 10^{-2} X_1X_2 + 0.14X_7X_{12} \\
 & + 4.6 \times 10^{-2} X_6X_8 - 0.1X_7X_{14} - 4.4 \times 10^{-2} X_3X_4 - 7.2 \times 10^{-2} X_2X_{11}
 \end{aligned}$$

Legend:

$$X_1 = \frac{2(\text{Disp} \cdot \text{Spd} - 224.06)}{1943.85} - 1 \qquad X_2 = \frac{2\left(\frac{dS}{dt}(1s) + 279.74\right)}{518.46} - 1$$

$$X_3 = \frac{2\left(\frac{d^2S}{dt^2}(1s) + 100.55\right)}{195.19} - 1$$

$$X_4 = \frac{2\left(\frac{dS}{dt}(5s) + 231.78\right)}{411.36} - 1$$

$$X_5 = \frac{2\left(\frac{d^2S}{dt^2}(5s) + 53.82\right)}{121.01} - 1$$

$$X_6 = \frac{2\left(\frac{dS}{dt}(10s) + 130.55\right)}{257.58} - 1$$

$$X_7 = \frac{2\left(\frac{d^2S}{dt^2}(10s) + 40.17\right)}{74.19} - 1$$

$$X_8 = \frac{2(\text{Disp } T + 324.28)}{2227.6} - 1$$

$$X_9 = \frac{2\left(\frac{dT}{dt}(1s) + 332.28\right)}{671.2} - 1$$

$$X_{10} = \frac{2\left(\frac{d^2T}{dt^2}(1s) + 191.31\right)}{333.48} - 1$$

$$X_{11} = \frac{2\left(\frac{dT}{dt}(5s) + 268.13\right)}{541.79} - 1$$

$$X_{12} = \frac{2\left(\frac{d^2T}{dt^2}(5s) + 109.04\right)}{190.32} - 1$$

$$X_{13} = \frac{2\left(\frac{dT}{dt}(10s) + 160.69\right)}{343.23} - 1$$

$$X_{14} = \frac{2\left(\frac{d^2T}{dt^2}(10s) + 54.45\right)}{95.38} - 1$$

$$Y = \frac{2(\text{CO}(g/s))}{0.2} - 1$$

10.4.GMDH CO₂ Prediction Formula

$$\begin{aligned}
 Y = & 0.48X_8 - 9.9 \times 10^{-2} X_{13} - 0.45X_9 + 0.13X_{10} + 0.18X_{11} - 0.71 + 4.9 \times 10^{-2} X_{12} \\
 & + 5 \times 10^{-2} X_5 + 2.2 \times 10^{-2} X_6 + 7.1 \times 10^{-2} X_4 + 8.9 \times 10^{-2} X_7 - 5.2 \times 10^{-2} X_{14} - 0.26X_3 \\
 & + 0.26X_1 + 6.8 \times 10^{-2} X_3^2 + 0.17X_8^2 + 7.8 \times 10^{-2} X_9^2 - 0.25X_3X_8 + 0.2X_3X_9 - 0.14X_8X_9 \\
 & + 0.4X_1^2 + 0.41X_1X_8 - 0.19X_1X_{12} + 0.12X_1^3 + 0.18X_1X_8^2 + 0.11X_1X_{12}^2 + 0.4X_1^2X_8 \\
 & - 0.27X_1^2X_{12} - 4.5 \times 10^{-2} X_1X_8X_{12} + 4.8 \times 10^{-2} X_4^2 + 0.11X_{11}^2 + 0.19X_4X_{11} \\
 & + 9.7 \times 10^{-2} X_{10}^2 + 0.22X_{10}X_{13} + 8.4 \times 10^{-2} X_1X_5 + 3.9 \times 10^{-2} X_{13}^2 + 0.2X_{11}X_{13} + 0.13X_8X_{12} \\
 & - 0.15X_3X_{10} + 6 \times 10^{-2} X_{14}^2 - 0.16X_{11}X_{14} + 4.4 \times 10^{-2} X_7^2 + 0.12X_7X_8 + 8.2 \times 10^{-2} X_{12}^2 \\
 & - 9.4 \times 10^{-2} X_9X_{12} - 0.14X_8X_{13} - 0.1X_7X_{10} + 4.1 \times 10^{-2} X_6X_8 - 6.4 \times 10^{-2} X_5X_{12} \\
 & + 5 \times 10^{-2} X_6^2 - 5.3 \times 10^{-2} X_6X_9 - 2.7 \times 10^{-2} X_1X_{10} - 4.6 \times 10^{-2} X_1X_{13} - 4.7 \times 10^{-2} X_1X_6
 \end{aligned}$$

Legend:

$$X_1 = \frac{2(\text{Disp} \cdot \text{Spd} - 224.06)}{1943.85} - 1$$

$$X_2 = \frac{2\left(\frac{dS}{dt}(1s) + 279.74\right)}{518.46} - 1$$

$$X_3 = \frac{2\left(\frac{d^2S}{dt^2}(1s) + 100.55\right)}{195.19} - 1$$

$$X_4 = \frac{2\left(\frac{dS}{dt}(5s) + 231.78\right)}{411.36} - 1$$

$$X_5 = \frac{2\left(\frac{d^2S}{dt^2}(5s) + 53.82\right)}{121.01} - 1$$

$$X_6 = \frac{2\left(\frac{dS}{dt}(10s) + 130.55\right)}{257.58} - 1$$

$$X_7 = \frac{2\left(\frac{d^2S}{dt^2}(10s) + 40.17\right)}{74.19} - 1$$

$$X_8 = \frac{2(\text{Disp } T + 324.28)}{2227.6} - 1$$

$$X_9 = \frac{2\left(\frac{dT}{dt}(1s) + 332.28\right)}{671.2} - 1$$

$$X_{10} = \frac{2\left(\frac{d^2T}{dt^2}(1s) + 191.31\right)}{333.48} - 1$$

$$X_{11} = \frac{2\left(\frac{dT}{dt}(5s) + 268.13\right)}{541.79} - 1$$

$$X_{12} = \frac{2\left(\frac{d^2T}{dt^2}(5s) + 109.04\right)}{190.32} - 1$$

$$X_{13} = \frac{2\left(\frac{dT}{dt}(10s) + 160.69\right)}{343.23} - 1$$

$$X_{14} = \frac{2\left(\frac{d^2T}{dt^2}(10s) + 54.45\right)}{95.38} - 1$$

$$Y = \frac{2(CO_2(g/s) + 0.23)}{83.81} - 1$$

10.5.GMDH HC Prediction Formula

$$\begin{aligned}
 Y = & -0.14X_{13} + 0.14X_{11} + 2.6 \times 10^{-2} X_8 + 5 \times 10^{-2} X_{10} - 0.11X_{12} + 0.14X_9 - 0.69 \\
 & -4.2 \times 10^{-2} X_1 + 0.35X_6 + 0.26X_3 - 0.33X_4 + 0.22X_5 - 0.1X_7 + 0.11X_2 - 0.13X_{14} \\
 & -4.7 \times 10^{-2} X_1^2 + 0.38X_6^2 + 0.1X_8^2 + 0.36X_1X_6 - 0.25X_1X_8 + 0.12X_6X_8 + 0.1X_{13}^2 \\
 & +0.15X_1X_{13} + 0.71X_1^3 + 0.78X_1X_6^2 + 0.14X_1X_8^2 - 0.61X_1^2X_6 - 0.7X_1^2X_8 + 0.65X_1X_6X_8 \\
 & -7.7 \times 10^{-2} X_8X_{13} + 0.48X_1^2X_{13} + 0.28X_6^2X_{13} + 9.4 \times 10^{-2} X_8^2X_{13} - 0.65X_1X_6X_{13} \\
 & -0.48X_1X_8X_{13} + 0.14X_6X_8X_{13} + 6.8 \times 10^{-2} X_3^2 + 7.6 \times 10^{-2} X_{10}^2 - 0.5X_3X_{10} + 0.56X_5^2 \\
 & +0.41X_7^2 - 0.84X_5X_7 - 3.6 \times 10^{-2} X_5X_6 + 0.25X_6X_7 - 0.25X_3X_6 + 2.4 \times 10^{-2} X_6X_{10} \\
 & -0.21X_6X_{13} - 0.26X_6^3 - 8.7 \times 10^{-2} X_6X_8^2 - 0.21X_6^2X_8 + 8.7 \times 10^{-2} X_6X_{13}^2 - 0.76X_1^3X_6 \\
 & -0.44X_1X_6^3 - 0.15X_1X_6X_8^2 + 0.62X_1^2X_6^2 + 0.76X_1^2X_6X_8 - 0.35X_1X_6^2X_8 \\
 & -0.52X_1^2X_6X_{13} - 0.3X_6^3X_{13} - 0.1X_6X_8^2X_{13} + 0.42X_1X_6^2X_{13} + 0.52X_1X_6X_8X_{13} \\
 & -0.24X_6^2X_8X_{13} - 0.18X_3^2X_6 - 0.11X_6X_{10}^2 + 0.54X_3X_6X_{10} - 0.43X_5^2X_6 - 0.39X_6X_7^2 \\
 & +0.57X_5X_6X_7 + 2.6 \times 10^{-2} X_{11}^2 + 0.11X_{14}^2 - 0.41X_5X_{14} + 0.15X_9X_{13} + 0.12X_{12}^2 \\
 & +0.42X_7X_{12} - 0.28X_{11}X_{14} + 0.9X_4^2 - 1.4X_4X_6 - 0.1X_{10}X_{12} + 0.13X_2^2 + 0.35X_2X_8 \\
 & +0.31X_6X_{11} + 0.14X_7X_{10} - 5.5 \times 10^{-2} X_9^2 - 9.1 \times 10^{-2} X_1X_9 + 0.1X_8X_{14} - 0.23X_2X_{13} \\
 & +0.24X_7X_8 - 0.3X_2X_9 + 0.14X_3X_{12} - 0.15X_7X_{13} - 8.4 \times 10^{-2} X_1X_{10} - 9.2 \times 10^{-2} X_5X_{12} \\
 & +5.5 \times 10^{-2} X_5X_{10} - 9.1 \times 10^{-2} X_{11}X_{13} + 7.6 \times 10^{-2} X_{13}X_{14} + 0.12X_3X_8 + 7.4 \times 10^{-2} X_1X_{11} \\
 & -9.3 \times 10^{-2} X_9X_{11}
 \end{aligned}$$

Legend:

$$X_1 = \frac{2(\text{Disp} \cdot \text{Spd} - 224.06)}{1943.85} - 1 \qquad X_2 = \frac{2\left(\frac{dS}{dt}(1s) + 279.74\right)}{518.46} - 1$$

$$X_3 = \frac{2\left(\frac{d^2S}{dt^2}(1s) + 100.55\right)}{195.19} - 1 \qquad X_4 = \frac{2\left(\frac{dS}{dt}(5s) + 231.78\right)}{411.36} - 1$$

$$X_5 = \frac{2\left(\frac{d^2S}{dt^2}(5s) + 53.82\right)}{121.01} - 1$$

$$X_6 = \frac{2\left(\frac{dS}{dt}(10s) + 130.55\right)}{257.58} - 1$$

$$X_7 = \frac{2\left(\frac{d^2S}{dt^2}(10s) + 40.17\right)}{74.19} - 1$$

$$X_8 = \frac{2(\text{Disp } T + 324.28)}{2227.6} - 1$$

$$X_9 = \frac{2\left(\frac{dT}{dt}(1s) + 332.28\right)}{671.2} - 1$$

$$X_{10} = \frac{2\left(\frac{d^2T}{dt^2}(1s) + 191.31\right)}{333.48} - 1$$

$$X_{11} = \frac{2\left(\frac{dT}{dt}(5s) + 268.13\right)}{541.79} - 1$$

$$X_{12} = \frac{2\left(\frac{d^2T}{dt^2}(5s) + 109.04\right)}{190.32} - 1$$

$$X_{13} = \frac{2\left(\frac{dT}{dt}(10s) + 160.69\right)}{343.23} - 1$$

$$X_{14} = \frac{2\left(\frac{d^2T}{dt^2}(10s) + 54.45\right)}{95.38} - 1$$

$$Y = \frac{2(\text{HC}(g/s))}{0.04} - 1$$