

LB/TH/33/2025

TH5909

**HYPERPARAMETER PREDICTION IN ANN AND RNN:  
A CASE STUDY IN HOTEL DOMAIN**

Yasantha Subash Samarasekara

(199484N)

Department of Computational Mathematics

University of Moratuwa

Sri Lanka

April 2024

**HYPERPARAMETER PREDICTION IN ANN AND RNN:  
A CASE STUDY IN HOTEL DOMAIN**

Wijesooriya Ralalage Yasantha Subash Samarasekara

(199484N)

Thesis/Dissertation submitted in partial fulfilment of the requirements for the degree  
Master of Science

Department of Computational Mathematics

University of Moratuwa

Sri Lanka

April 2024

## Declaration

I declare that this is my work, and this thesis/dissertation does not incorporate without acknowledgement any material previously submitted for a degree or Diploma in any other University or institute of higher learning. To the best of my knowledge and belief, it does not contain any material previously published or written by another person except where acknowledgement is made in the text.

Also, I hereby grant the University of Moratuwa the non-exclusive right to reproduce and distribute my thesis/dissertation, in whole or in part, in print, electronic, or other mediums. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature:

Date:

.....  
Mr. W.R.Y.S. Samarasekara

.....09/05/2024.....

The above candidate has carried out research for the Master thesis/dissertation under my supervision.

Signature of the supervisor:

Date:

.....  
Dr. A.T.P. Silva

19/08/2024  
.....

## Abstract

Deep Neural Networks are a subfield of the subsymbolic paradigm of Artificial Intelligence. Usually, when one wants to use artificial neural networks for a specific task, the neural network should be configured with optimal hyperparameters. To conduct such a task, the user should know about neural networks, as finding the optimal set of hyperparameters is time-consuming when manually configuring. This matter is prominent in data science because the field's requirements are directed toward deep learning and LLMs with autoML platforms. Automated hyperparameter prediction and optimization results in knowing about configuration and turning neural networks entirely or partially eliminated.

This research starts its journey to achieve hyperparameter prediction in a naval way by considering the selection of independent and dependent features. Hyperparameter generation happens by using neural networks and training 34 predictors and classifiers. Mapping between different feature sets and the existing model configurations, achieved by a naval general frame, has been introduced, using natural language processing to arrange feature names in similar columns. This nominal feature set is so diverse in such a way that it has 2-3 100% identical feature names per 40 feature name rows. So, naval natural language processing encoding was introduced as a solution. When the complete feature set is inserted into the model, the set of classifiers and the predictors achieve the configuration in RNN and ANN—the solution given the name HPPGeneral model. Then, the resultant configurations are applied to select problems in the hotel domain to evaluate the approach. This research has delivered several naval research outputs, such as general frame and natural language encoding, besides a model that can be used for both ANN and RNN. Finally, the hyperparameter prediction achieved by this approach also gives results almost similar to those of manual hyperparameter prediction.

Keywords: hyperparameter prediction, ANN, RNN, general frame, NLP encoding, general model

## **Dedication**

To my lovely family

## **Acknowledgement**

First of all, I'm so grateful to my supervisor, Dr. Thushari, for her kind guidance and demonstration of support. Without her, this research would never have been possible. Also, thank you for the kind guidance provided by the professor. Karunananda, through the comments on the evaluations. Because of it, I got the opportunity to excel my research experience in Artificial General Intelligence and hyperparameter prediction. Because of them, I found this opportunity to do such beautiful research. Also, I am grateful for the comments the course coordinator and evaluators provided to make this research and the thesis a success.

Additionally, I would be grateful to my friend Dr. Haishchandra, an engineering faculty member at the University of Peradeniya, for his guidance. This type of research topic is not easy to do, and research has been selected due to the leave granted by the Sabaragamuwa University of Sri Lanka. It is a great pleasure to work with them.

The support given by my family, especially my wife and her mother, in tackling and bearing difficult situations greatly supported me in this research. Also, my two children, even at their more minor ages, who least disturbed me, allowed me to focus on the research while working at home. Last but not least, I sincerely thank anyone who helped me enormously to make this research successful.

## **Table of contents**

Declaration	iii
Abstract	iv
Dedication	v
Acknowledgement	vi
Table of contents	vii
List of Figures	xiii
List of tables	xxii
List of abbreviations	xxiii
<b>CHAPTER 1 INTRODUCTION</b>	<b>1</b>
1.1 Prolegomena	1
1.2 Aim and Objectives	2
1.3 Background and Motivation	3
1.4 Problem definition	5
1.5 General model	6
1.5.1 Input	6
1.5.2 Output	7
1.5.3 Process	7
1.5.4 Users	7
1.5.5 Features	8
1.6 Resource requirements	8
1.7 Structure of the Thesis	8
1.8 Summary	8
<b>CHAPTER 2 DEVELOPMENT AND ISSUES</b>	<b>9</b>
2.1 Introduction	9

2.2	Early developments in ANN, RNN, and hyperparameter Research with hotel classification and review prediction	9
2.3	hyperparameter prediction, optimization in neural networks	13
2.4	New Trends in hyperparameter prediction and optimization	23
2.5	Analysis of Literature	26
2.6	Problem definition	38
2.7	Summary	39
CHAPTER 3 TECHNOLOGIES ADOPTED		40
3.1	Introduction	40
3.2	Panda data frame	40
3.3	TensorFlow	41
3.4	Mathematical Model of ANN	43
3.4.1	Mathematical model of multilayer ANN	44
3.4.1.1	Backpropagation algorithm	47
3.4.1.2	Other learning rules	49
3.5	Long Short-Term Memory	49
3.6	Dataset building with existing problems.	52
3.7	Application programming interface	54
3.8	Summary	54
CHAPTER 4 APPROACH		55
4.1	Introduction	55
4.2	Hypothesis	56
4.3	Input	56
4.4	Output	56
4.5	Process	56
4.6	Users	57

4.7	Features	58
4.8	Summary	58
CHAPTER 5 DESIGN		59
5.1	Introduction	59
5.2	Module 01 & Module 02: Preprocessing	60
5.3	Module 03: ANN-RNN Classifier	60
5.3.1	General frame	60
5.3.2	Encoding general data frame	64
5.3.3	NLP Encoding	65
5.3.3.1	NLP encoding method 01	65
5.3.3.2	NLP encoding method 02	65
5.3.4	Label encoding with random fit	65
5.3.5	The neural network for ANN-RNN classifier	66
5.4	Module 04: Hyperparameter predictor module	66
5.4.1	Layer Number Predictor	67
5.4.2	Optimizer classifier	68
5.4.3	Lost function classifier	68
5.4.4	Lost metrix classifier	68
5.4.5	Number of epochs predictor	68
5.4.6	Batch size predictor	68
5.4.7	Layer type classifier	69
5.4.8	Number of neurons predictor	69
5.4.9	Dropout predictor	69
5.4.10	Activation function classifier	70
5.4.11	Embeding Layer-Input dimension predictor	70
5.4.12	Embeding Layer -Output dimention predictor	70

5.5	Module 05: General module	70
5.5.1	Create a general model	71
5.5.2	ANN & RNN Layer Predictors	71
5.5.3	ANN & RNN layer restrictors	72
5.5.4	ANN & RNN layer creators	72
5.5.5	Configuration loader	72
5.5.6	Model runner	72
5.5.7	Evaluator and Predictor	72
5.5.8	User process for HPPGeneral model	73
5.6	Summary	74
CHAPTER 6 IMPLEMENTATION		75
6.1	Introduction	75
6.2	Module 01 & Module 02: Preprocessing	75
6.3	Module 03: ANN-RNN Classifier	76
6.3.1	General frame	76
6.3.1.1	Sequence Row	77
6.3.1.2	Transfer dependent variables into the dependent section(dependent variables) 82	
6.3.2	Encoding general data frame	85
6.3.2.1	NLP encoding method 01	86
6.3.2.2	NLP encoding method 02	90
6.3.2.3	Label encoding with random fit	92
6.3.3	The neural network	95
6.4	Module 05: Hyperparameter prediction module	97
6.4.1	Layer Number Predictor	97
6.4.2	Optimizer classifier	98

6.4.3	Lost function classifier	100
6.4.4	Lost matrix classifier	102
6.4.5	Number of epochs predictor	104
6.4.6	Batch size predictor	106
6.4.7	Layer type classifier	107
6.4.7.1	Layer type classifier for layer 01	108
6.4.7.2	Layer type classifier for layer 02	109
6.4.7.3	Layer type classifier for layer 03	112
6.4.7.4	Layer type classifier for layer 04	113
6.4.7.5	Layer type classifier for layer 05	115
6.4.7.6	Layer type classifier for layer 06	117
6.4.8	Number of neurons predictor	119
6.4.9	Dropout predictor	126
6.4.10	Activation function classifier	134
6.4.10.1	Activation function classifier for layer 01	135
6.4.10.2	Activation function classifier for layer 02	137
6.4.10.3	Activation function classifier for layer 03	139
6.4.10.4	Activation function classifier for layer 04	141
6.4.10.5	Activation function classifier for layer 05	142
6.4.10.6	Activation function classifier for layer 06	144
6.4.11	Embedding Layer - Input dimension predictor	146
6.4.12	Embedding Layer - Output dimension predictor	147
6.4.13	Final Hyperparameter Predictor Module	148
6.5	Module 05: General module	149
6.5.1	Create a general model	149
6.5.2	ANN & RNN Layer Predictors	150

6.5.2.1	ANN Layer Predictor	150
6.5.2.2	RNN Layer Predictor	151
6.5.3	ANN & RNN layer restrictors	153
6.5.3.1	ANN layer restrictors	153
6.5.3.2	RNN layer restrictors	154
6.5.4	ANN & RNN layer creators	155
6.5.5	Configuration loader	157
6.5.6	Model runner	160
6.5.7	Evaluator and Predictor	163
6.5.8	User process for HPPGeneral model	163
6.6	Summary	168
CHAPTER 7 EVALUATION		169
7.1	Introduction	169
7.2	Training results comparison table	170
7.3	Summary	175
CHAPTER 8 CONCLUSIONS AND RECOMMENDATIONS		176
8.1	Introduction	176
8.2	Other Important Research Outputs	176
8.3	Recommendations	177
8.4	Summary	178
References		179
Appendices		188

## List of Figures

Figure 2.1	ANN network architectures	10
Figure 2.2	Quad—arrangement of nodes as a basic representational unit	22
Figure 3.1	Architecture of a typical vanilla LSTM block	50
Figure 5.1	Overview of the Design diagram of the general model	59
Figure 5.2	Row adding process to the general frame.	61
Figure 5.3	Process of adding feature row to the general frame.	62
Figure 5.4	Transfer dependent variable from independent side to dependent side	63
Figure 5.5	Encoder class	64
Figure 5.6	General Frame	65
Figure 5.7	Training and prediction process of the classifier	66
Figure 5.8	Structure of the hyperparameter predictor	67
Figure 5.9	General module class	71
Figure 5.10	General Model Creation	73
Figure 6.1	Converting camelCase feature with spaces with lowercase	75
Figure 6.2	Code to introduce lowercase with spaces	76
Figure 6.3	Code for creating general frame structure.	76
Figure 6.4	Code for adding the first row directly otherwise sequence the items.	77
Figure 6.5	Code to add to the general frame.	77
Figure 6.6	Code to compare the maximum comparison value	78
Figure 6.7	The output created by the code.	78
Figure 6.8	Code for the action when the value has already existed	79
Figure 6.9	Output when a higher comparison value is found	79
Figure 6.10	Code for checking candidate values to consider for a shift	80
Figure 6.11	Visual output when shifting occurs.	80
Figure 6.12	Code for adding the item to the sequence list by available list.	81

Figure 6.13 Visual output from removing an item and adding to the sequence list.	81
Figure 6.14 Code for rearranging the feature items in the order	81
Figure 6.15 Output of the rearranged list according to the order	82
Figure 6.16 Appearance of the frame after adding 15 rows.	82
Figure 6.17 General frame with empty cells in dependent variables side	82
Figure 6.18 Code to find the first empty row in the dependent side	83
Figure 6.19 Set of codes for transferring the dependent variables	84
Figure 6.20 Output after transferring dependent variables to the dependent side.	84
Figure 6.21 Code for saving the output as CSV or loading it if it already existed.	84
Figure 6.22 Add column method to add a column to the general frame.	85
Figure 6.23 Add one or more columns to the general frame at once.	85
Figure 6.24 Final general frame after generating all the columns.	85
Figure 6.25 Code to check whether a frame is saved as a CSV.	86
Figure 6.26 Output after encoding for the independent side by NLP encoding 01	86
Figure 6.27 Code sees whether the general frame is already saved as a CSV	87
Figure 6.28 The code compares the values with an empty string	87
Figure 6.29 Output after encoding the dependent side.	88
Figure 6.30 After encoding the Nature of the neural network column	88
Figure 6.31 Correlated matrix for the NLP encoding method 01	89
Figure 6.32 Output of the neural network trained: encoded values(the lost)	89
Figure 6.33 Output of the neural network trained: encoded values(the accuracy)	90
Figure 6.34 Output for the second time training of the neural network.	90
Figure 6.35 This is the code for NLP encoding method 02.	90
Figure 6.36 Encoding output for NLP method 02.	91
Figure 6.37 This is the correlation matrix for the NLP encoding method 02.	91
Figure 6.38 The output for the training by NLP method 02.	92

Figure 6.39	Accuracy of the neural network after training: encoding method 02.	92
Figure 6.40	This is the code for the label encoding for the entire table.	93
Figure 6.41	Output after applying label encoding with the random fit.	93
Figure 6.42	This is the correlation matrix after label encoding.	94
Figure 6.43	Output: neural network after applying encoded values for the training	94
Figure 6.44	This is the accuracy after the training by label encoding.	95
Figure 6.45	Prediction accuracy for the NLP encoding method01.	95
Figure 6.46	Predicting accuracy for the label encoding.	95
Figure 6.47	Value transfer through NumPy and min-max Scaler.	96
Figure 6.48	Traning testing split by sklearn.	96
Figure 6.49	Neural network configuration for the binary classifier.	96
Figure 6.50	Neuralnetwork configuration number of layers predictor	97
Figure 6.51	Loss function number of layers predictor	97
Figure 6.52	Actual value and predicted value of the number of layers	98
Figure 6.53	Neural network for optimizer classifier	98
Figure 6.54	Lost function for optimizer classifier	99
Figure 6.55	Accuracy of optimizer classifier	99
Figure 6.56	Evaluation and prediction of optimizer classifier	100
Figure 6.57	Predicted values of optimizer classifier	100
Figure 6.58	Neural network for lost function classifier	101
Figure 6.59	Lost function for lost function classifier	101
Figure 6.60	Accuracy of lost function classifier	101
Figure 6.61	Evaluation and Prediction of Lost Function Classifier	102
Figure 6.62	Predicted values of lost function classifier	102
Figure 6.63	Neural network for matrix classifier	102
Figure 6.64	Lost function for matrix classifier	103

Figure 6.65 Accuracy of matrix Classifier	103
Figure 6.66 Evaluation and prediction of matrix classifier	103
Figure 6.67 Predicted values of matrix classifier	104
Figure 6.68 Neural network for epochs predictor	104
Figure 6.69 Lost function for epochs predictor	105
Figure 6.70 Evaluation and prediction of epochs Predictor	105
Figure 6.71 Predicted values of epochs predictor	105
Figure 6.72 Neural network for batch size Predictor	106
Figure 6.73 Lost function for batch size predictor	106
Figure 6.74 Evaluation and prediction of batch size Predictor	107
Figure 6.75 Predicted values of batch size predictor	107
Figure 6.76 Neural network for layer type classifier for layer one	108
Figure 6.77 Lost function for layer type classifier for layer one	108
Figure 6.78 Accuracy of layer type classifier for layer one	108
Figure 6.79 Evaluation of layer type classifier for layer one	109
Figure 6.80 Predicted values of layer type classifier for layer one	109
Figure 6.81 Neural network for layer type classifier for layer two	109
Figure 6.82 Lost function for layer type classifier for layer two	110
Figure 6.83 Accuracy of layer type classifier for layer two	110
Figure 6.84 Evaluation and prediction of layer type classifier for layer two	111
Figure 6.85 Predicted values of layer type classifier for layer two	111
Figure 6.86 Neural network for layer type classifier for layer three	112
Figure 6.87 Lost function for layer type classifier for layer three	112
Figure 6.88 Accuracy of layer type classifier for layer three	112
Figure 6.89 Evaluation and prediction of layer type classifier for layer three	113
Figure 6.90 Predicted values of layer type classifier for layer three	113

Figure 6.91 Neural network for layer type classifier for layer four	113
Figure 6.92 Lost function for layer type classifier for layer four	114
Figure 6.93 Accuracy of layer type classifier for layer four	114
Figure 6.94 Evaluation and prediction of layer type classifier for layer four	115
Figure 6.95 Predicted values of layer type classifier for layer four	115
Figure 6.96 Neural network for layer type classifier for layer five	115
Figure 6.97 Lost function for layer type classifier for layer five	116
Figure 6.98 Accuracy of layer type classifier for layer five	116
Figure 6.99 Evaluation and prediction of layer type classifier for layer five	116
Figure 6.100 Predicted values of layer type classifier for layer five	117
Figure 6.101 Neural network for layer type classifier for layer six	117
Figure 6.102 Lost function for layer type classifier for layer six	117
Figure 6.103 Accuracy of layer type classifier for layer six	118
Figure 6.104 Evaluation and prediction of layer type classifier for layer six	118
Figure 6.105 Predicted values of layer type classifier for layer six	118
Figure 6.106 Neural network for number of neurons predictor for layer one	119
Figure 6.107 Lost function for the number of neurons predictor for layer one	119
Figure 6.108 Evaluation and prediction, prediction: No of neuron predictor L one	120
Figure 6.109 Neural network for number of neurons predictor for layer two	120
Figure 6.110 Lost function for number of neurons predictor for layer two	120
Figure 6.111 Evaluation & prediction, prediction: No of neuron predictors L-two	121
Figure 6.112 Neural network for number of neurons predictor for layer three	121
Figure 6.113 Lost function for number of neurons predictor for layer three	122
Figure 6.114 Evaluation & prediction, prediction: No of neuron predictor L-three	122
Figure 6.115 Neural network for number of neurons predictor for layer four	123
Figure 6.116 Lost function for number of neurons predictor for layer four	123

Figure 6.117 Evaluation & prediction, prediction: No of neuron predictors L-four	123
Figure 6.118 Neural network for number of neurons predictor for layer five	124
Figure 6.119 Lost function for number of neurons predictor for layer five	124
Figure 6.120 Evaluation & prediction, prediction: No of neuron predictors L-five	125
Figure 6.121 Neural network for number of neurons predictor for layer six	125
Figure 6.122 Lost function for the number of neurons predictor for layer six	125
Figure 6.123 Evaluation & prediction, prediction: No of neuron predictors L-six	126
Figure 6.124 Neural network for dropout predictor for layer one	126
Figure 6.125 Lost function for dropout predictor for layer one	127
Figure 6.126 Evaluation & prediction, prediction: No of dropout predictor L-one	127
Figure 6.127 Neural network for dropout predictor for layer two	128
Figure 6.128 Lost function for dropout predictor for layer two	128
Figure 6.129 Evaluation & prediction, prediction: No of dropout predictor L-two	129
Figure 6.130 Neural network for dropout predictor for layer three	129
Figure 6.131 Lost function for dropout predictor for layer three	130
Figure 6.132 Evaluation & prediction, prediction: No of dropout predic. L-three	130
Figure 6.133 Neural network for dropout predictor for layer four	131
Figure 6.134 Lost function for dropout predictor for layer four	131
Figure 6.135 Evaluation & prediction, prediction: No of dropout predictor L-four	132
Figure 6.136 Neural network for dropout predictor for layer five	132
Figure 6.137 Lost function for dropout predictor for layer five	132
Figure 6.138 Evaluation & prediction, prediction: No of dropout predictor L-five	133
Figure 6.139 Neural network for dropout predictor for layer six	133
Figure 6.140 Lost function for dropout predictor for layer six	134
Figure 6.141 Evaluation & prediction, prediction: dropout predictor for layer six	134
Figure 6.142 Neural network for activation function classifier for layer one	135

Figure 6.143	Lost function for activation function classifier for layer one	135
Figure 6.144	Accuracy of activation function classifier for layer one	135
Figure 6.145	Evaluation & prediction: activation function classifier for layer one	136
Figure 6.146	Predicted values of activation function classifier for layer one	136
Figure 6.147	Neural network for activation function classifier for layer two	137
Figure 6.148	Lost function for activation function classifier for layer two	137
Figure 6.149	Accuracy of activation function classifier for layer two	138
Figure 6.150	Evaluation & prediction of activation function classifier L-two	138
Figure 6.151	Predicted values of activation function classifier for layer two	138
Figure 6.152	Neural network for activation function classifier for layer three	139
Figure 6.153	Lost function for activation function classifier for layer three	139
Figure 6.154	Accuracy of activation function classifier for layer three	140
Figure 6.155	Evaluation & prediction of activation function classifier L-three	140
Figure 6.156	Predicted values of activation function classifier for layer three	140
Figure 6.157	Neural network for activation function classifier for layer four	141
Figure 6.158	Lost function for activation function classifier for layer four	141
Figure 6.159	Accuracy of activation function classifier for layer four	141
Figure 6.160	Evaluation & prediction of activation function classifier L-four	142
Figure 6.161	Predicted values of activation function classifier for layer four	142
Figure 6.162	Neural network for activation function classifier for layer five	142
Figure 6.163	Lost function for activation function classifier for layer five	143
Figure 6.164	Accuracy of activation function classifier for layer five	143
Figure 6.165	Evaluation & prediction of activation function classifier L-five	143
Figure 6.166	Predicted values of activation function classifier for layer five	144
Figure 6.167	Neural network for activation function classifier for layer six	144
Figure 6.168	Lost function for activation function classifier for layer six	144

Figure 6.169 Accuracy of activation function classifier for layer six	145
Figure 6.170 Evaluation & prediction of activation function classifier L-six	145
Figure 6.171 Predicted values of activation function classifier for layer six	146
Figure 6.172 Neural network for input dimension of the embedding layer	146
Figure 6.173 Lost function for input dimension of the embedding layer	146
Figure 6.174 Evaluation & prediction of the input dimension of the embed. layer	147
Figure 6.175 Neural network for output dimension of the embedding layer	147
Figure 6.176 Lost function for output dimension of the embedding layer	148
Figure 6.177 Evaluation & prediction, prediction val. of output dim. of embed.	148
Figure 6.178 The hyperparameter prediction class	149
Figure 6.179 Create General Model	149
Figure 6.180 RNN part of the create general model	150
Figure 6.181 Starting part of the ANN layer predictor	150
Figure 6.182 Code for loading diff. predi. when anomalies happen	151
Figure 6.183 Starting code of the RNN layer predictor	152
Figure 6.184 RNN loads other predi. when anomalies in the before-fix value	153
Figure 6.185 ANN model restrictor, layer restrictor	154
Figure 6.186 RNN model restrictor, layer restrictor	155
Figure 6.187 ANN layer creator method	156
Figure 6.188 RNN layer creator method	156
Figure 6.189 Starting of configuration loader method	157
Figure 6.190 ANN code part of the configuration loader method	158
Figure 6.191 RNN code part of the configuration loader	159
Figure 6.192 RNN code part of the config. Loader including diff. input shapes	160
Figure 6.193 ANN part of the model runner	161
Figure 6.194 Print is used to give the code output for the user in the ANN part	161

Figure 6.195 Similar RNN restrictors usage in the RNN part of the model loader	162
Figure 6.196 Simler data conversion and print usage in the RNN: model runner	163
Figure 6.197 Implementation of evaluator and predictor methods	163
Figure 6.198 Generate general frames and apply encoding by the user	164
Figure 6.199 Creating the general model: run the hyperparameter prediction	164
Figure 6.200 Generated ANN general frame output	165
Figure 6.201 Generated RNN general frame output	165
Figure 6.202 Configuration loader usage by the user for a regression problem	166
Figure 6.203 Configuration loader usage for text classification	166
Figure 6.204 Model runner method usage by the user for a regression problem	166
Figure 6.205 Model runner method usage for a text classification problem	167
Figure 6.206 Usage of evaluate and predict method by the user	167
Figure 7.1 Evaluating a regression problem for the HPPGeneral Model	171
Figure 7.2 Result of the HPPGeneral model for a regression problem	171
Figure 7.3 Evaluating the AutoML for a regression problem in deep learning	171
Figure 7.4 Result of the AutoML for a regression problem	172
Figure 7.5 Evaluating HPPGegeral model for a classification problem	172
Figure 7.6 Result of the HPPGeneral model for a classification problem	172
Figure 7.7 Evaluating AutoML for a classification problem in deep learning	173
Figure 7.8 Result of AutoML for the classification problem in deep learning	173
Figure 7.9 Comparion the HPPGeneral model with AutoML	175

## **List of tables**

Table 2.1	Table of analysis of the literature	26
Table 3.1	Datasets considered for adding features to the general frame	52
Table 5.1	General Frame Structure	61
Table 7.1	Problem table for testing the solution	169
Table 7.3	Comparison between Expert, AutoML and HPPGeneral Model	174

## List of abbreviations

AI	artificial intelligence
ANN	artificial neural network
CNN	Convolutional neural network
DAN	Deep averaging network
DBN	Deep belief networks
GNN	Graph neural networks
HR	High resolution
HTM	Hierarchical temporal memory, xliii; Hierarchical Temporal Memory
LLMs	Large Language Models
LSTM	Long Short-Term Memory
LSTM	long short-term memory
MLP	Multiplayer perceptron
NeSyL	Neuro-symbolic learning
NLP	Natural language processing
RAAM	Recursive auto-associative memory
RAN	Residual attention network
RBM	Restricted Boltzmann machine
RNN	Recurrent neural network
RvNN	Recursive neural networks
SOM	Self-organizing map