UNIVERSIDADE FEDERAL DE MINAS GERAIS
Engineering Department
Graduate Program in Electrical Engineering

Omid Orang

High-Order Fuzzy Cognitive Maps and Randomized Networks for Time Series and Nonlinear Dynamical Systems

Belo Horizonte

2023

Omid Orang

High-Order Fuzzy Cognitive Maps and Randomized Networks for Time Series and Nonlinear Dynamical Systems

Thesis presented to the Graduate Program in Electrical Engineering at the Universidade Federal de Minas Gerais, as a partial requirement for obtaining the title of Doctor in Electrical Engineering.

Supervisor: Prof. Dr. Frederico Gadelha Guimarães
Co-supervisor: Prof. Dr. Petrônio Cândido de Lima e Silva

Belo Horizonte
2023

Orang, Omid.
O63h High-Order Fuzzy Cognitive Maps and randomized networks for time series and nonlinear dynamical systems [recurso eletrônico] / Omid Orang. - 2023.
1 recurso online (191 f. : il., color.) : pdf.

Orientador: Frederico Gadelha Guimarães.
Coorientador: Petrônio Cândido de Lima e Silva.

Tese (doutorado) - Universidade Federal de Minas Gerais, Escola de Engenharia.

Anexos: f. 184-191.

Bibliografia: f. 168-183.
Exigências do sistema: Adobe Acrobat Reader.

1. Engenharia elétrica - Teses. 2. Séries temporais - Teses. 3. Mapas cognitivos (Psicologia) - Teses. 4. Algoritmos - Teses. 5. Algoritmos genéticos - Teses. 6. Redes neurais (Computação) - Teses. 7. Aprendizado do computador - Teses. 8. Mínimos quadrados - Teses. 9. Análise de componentes principais - Teses. 10. Sistemas de comunicação sem fio - Teses. 11. Kernel, Funções de. - Teses.
I. Guimarães, Frederico Gadelha. II. Silva, Petrônio Cândido de Lima e. III. Universidade Federal de Minas Gerais. Escola de Engenharia. IV. Título.

CDU: 621.3(043)

Ficha catalográfica elaborada pela bibliotecária Ângela Cristina Silva CRB/6 2361
Biblioteca Prof. Mário Werneck, Escola de Engenharia da UFMG

UNIVERSIDADE FEDERAL DE MINAS GERAIS
ESCOLA DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

FOLHA DE APROVAÇÃO

"HIGH-ORDER FUZZY COGNITIVE MAPS AND RANDOMIZED NETWORKS FOR TIME SERIES AND NONLINEAR DYNAMICAL SYSTEMS"

OMID ORANG

Tese de Doutorado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Doutor em Engenharia Elétrica. Aprovada em 30 de junho de 2023. Por:

Prof. Dr. Frederico Gadelha Guimarães - Orientador
DEE (UFMG)

Prof. Dr. Petrônio Cândido de Lima e Silva - Coorientador
IFNMG - Campus Januária

Prof. Dr. Guilherme de Alencar Barreto
DETI (UFC)

Prof. Dr. Tatiane Nogueira Rios
DCC (UFBA)

Prof. Dr. Hugo Valadares Siqueira
DAEE (UTFPR)

Prof. Dr. Rodrigo César Pedrosa Silva
DECOM (UFOP)

Documento assinado eletronicamente por **Frederico Gadelha Guimaraes**, **Professor do Magistério Superior**, em 02/07/2023, às 21:14, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Hugo Valadares Siqueira**, **Usuário Externo**, em 06/07/2023, às 10:15, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Guilherme de Alencar Barreto**, **Usuário Externo**, em 06/07/2023, às 17:01, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Rodrigo Cesar Pedrosa Silva**, **Usuário Externo**, em 07/07/2023, às 09:32, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Petrônio Cândido de Lima e Silva**, **Usuário Externo**, em 11/07/2023, às 14:41, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Tatiane Nogueira Rios**, **Usuária Externa**, em 14/07/2023, às 13:47, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site https://sei.ufmg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **2419016** e o código CRC **FE89C346**.

To God, my family, and everyone who participated in this journey.

ACKNOWLEDGMENTS

First and foremost, I am immensely grateful to my parents for their unwavering support, love, and sacrifices in raising me and giving me the chance to pursue my education. I also sincerely appreciate my siblings' encouragement and support during this long journey.

I would like to express my sincere and deepest gratitude to my talented and patient research supervisor, Prof. Dr. Frederico Gadelha Guimarães, for giving me the opportunity to be a researcher in the Machine Intelligence and Data Science (MINDS) laboratory and for providing invaluable guidance throughout this study. It was an enormous honor and privilege to study for my Ph.D. in MINDS under his supervision. I would also like to thank him for his empathy, compassion, intimate relationship, and sense of humor. Special thanks and regards also to his wife, Rúbia Pereira, and his adorable daughter for their patience during my discussion with him to prepare this research work.

Many heartfelt thanks to my supportive co-advisor, Prof. Dr. Petrônio Cândido de Lima e Silva, and his family as well. He continuously and kindly taught me plenty of topics. I feel immense pride and consider myself incredibly fortunate to have had the chance to work alongside him. I've been highly inspired by his talent, skills, dynamism, and motivation.

I would like to extend my gratitude to all professors and staff of the PPGEE – UFMG. I would like to extend my appreciation to my research colleagues and labmates in the MINDS laboratory for your collaboration, stimulating discussions, and friendship throughout this journey. Your support has made this experience enjoyable and enriching.

Thank you, UFMG, for providing the necessary resources, facilities, and academic environment for my research. I am also grateful for the financial support provided by CAPES. Last but not least, my thanks go to all the people who motivated and supported me, directly or indirectly, to complete this research study.

RESUMO

Os mapas cognitivos nebulosos (FCM, do inglês Fuzzy Cognitive Maps) surgiram como métodos interpretáveis das Séries Temporais Nebulosas (FTS, do inglês Fuzzy Time Series) para uma variedade de aplicações no campo de previsão. A construção da estrutura dos FCMs e a extração das conexões ponderadas entre os conceitos compõem a contribuição central das abordagens baseadas em FCMs na literatura. Apesar do sucesso das metodologias propostas, ainda existem algumas lacunas e limitações nesse domínio. Para cobrir alguns desses desafios, esta tese apresenta novas técnicas de previsão baseadas em FCMs para prever séries temporais univariadas e multivariadas, focando no design da nova arquitetura e na aceleração da fase de treinamento. Assim, a principal contribuição desta tese é introduzir novas técnicas de previsão pela fusão de FTS e FCMs para gerar FCMs aleatórios de alta ordem (R-HFCM, do inglês Randomized high-order FCM) como modelos de computação de reservatório pela primeira vez na literatura. O R-HFCM é um tipo de rede de estado de eco (ESN, do inglês Echo State Network), onde a camada do reservatório consiste em um grupo de sub-reservatórios de tal forma que os pesos dentro de cada sub-reservatório são escolhidos aleatoriamente de acordo com a inicialização de pesos do ESN. Os experimentos computacionais demonstram que o R-HFCM supera em termos de precisão e velocidade de treinamento quando comparado aos FCMs tradicionais treinados por algoritmos evolutivos como o algoritmo genético (GA, do inglês genethic algorithm). Para preencher a ausência de modelos de Entrada Múltipla e Saída Múltipla (MIMO, do inglês Multiple-Input Multiple-Output), extensões do método R-HFCM univariado são apresentadas para lidar com a previsão de séries temporais de baixa e alta dimensionalidade. Vale ressaltar que, em ambos os métodos MIMO, apenas a camada de saída é treinável utilizando o método dos mínimos quadrados por ser de baixo custo computacional. Os métodos propostos obtiveram resultados promissores e competitivos em comparação com uma variedade de métodos de aprendizado profundo e aprendizado de máquina em termos de precisão e parcimônia.

Palavras-chave: Previsão de Series Temporais; Séries Temporais Nebulosas; Mapas Cognitivos Nebulosos; Algoritmo Genético; Computação de Reservatório; Rede de Estado de Eco; Mínimos Quadrados; Entrada Múltipla Saída Múltipla; Análise de Componentes Principais por Kernel.

ABSTRACT

Fuzzy Cognitive Maps (FCMs) have emerged as interpretable Fuzzy Time Series (FTS) methods used in a variety of forecasting applications due to their interesting features. Constructing the structure of FCMs and extracting weighted connections among the concepts compose the crux contribution of the proposed FCM-based approaches in the literature. Despite the success of the proposed methodologies, there are still some gaps and limitations in this domain. To cover some of these challenges, this thesis introduces new forecasting techniques based on FCMs to predict univariate and multivariate time series focusing on both aspects including designing the new architecture and speeding up the training phase. Thus, the main contribution of this thesis is to introduce novel forecasting techniques by merging FTS and FCMs to generate randomized high-order FCM (R-HFCM) as reservoir computing models for the first time in the literature. R-HFCM is a kind of ESN where the reservoir layer consists of a group of sub-reservoirs such that the weights inside each sub-reservoir are randomly chosen according to the ESN weight initialization. The computational experiments demonstrate that R-HFCM outperforms in terms of both accuracy and training speed when compared to the traditional FCMs trained via evolutionary algorithms like genetic algorithm (GA). To fill the absence of Multiple-Input Multiple-Output(MIMO) models, extensions of the univariate R-HFCM method are presented to handle low-dimensional and high-dimensional time series forecasting. It is worth noting that in both MIMO methods, only the output layer is trainable using the time-effective least squares method. The proposed methods obtained promising and competitive results compared with a range variety of deep learning and machine learning methods in terms of accuracy and parsimony.

Keywords: Time Series Forecasting; Randomized Fuzzy Cognitive Maps; Reservoir Computing; Echo State Network; Multiple-Input Multiple-Output.

LIST OF FIGURES

Figure 1 – Proposed thesis methods and contributions. 23
Figure 2 – Simple FCM with 5 nodes (A) graphical structure (B) Weight matrix . 32
Figure 3 – Generic time series forecasting procedure using FCM 89
Figure 4 – Structure of FCM where the number of concepts and partitions are equal 90
Figure 5 – The structure of the proposed FCM-FTS method. 92
Figure 6 – The structure of the proposed HFCM-FTS method. 93
Figure 7 – Samples of solar radiance time series 95
Figure 8 – Structure of Traditional ESN 100
Figure 9 – Generic structure of the proposed R-HFCM method. 102
Figure 10 – The simple example of the proposed model mechanism considering $N_{SR} = 2$. 105
Figure 11 – Plotting of (A) 8,000 Samples of Brazilian solar radiance and (B) 8,760 samples of hourly electric load Malaysia dataset 108
Figure 12 – Plotting of 8,000 samples of hourly solar radiance time series including DHHL_1 (A), DHHL_2 (B), and DHHL_3 (C) 109
Figure 13 – Plotting of 8,000 Samples of hourly load time series from three zones of GEFCom2012 dataset zone 1 (A), zone 2 (B), and zone 3 (C) 110
Figure 14 – Plotting of 8,000 samples of PJM hourly energy consumption including AEP data (A) and DEOK data (B) 110
Figure 15 – Comparison of the model performance in terms of the average RMSE and U considering different map sizes and the different sizes of reservoirs (SONDA data) 113
Figure 16 – Comparison of the model performance in terms of the average RMSE and U using different map sizes, size of reservoirs, and $\Omega = \{2, 3\}$ for SONDA data 117
Figure 17 – Comparison of the model performance in terms of the average RMSE and U using different map sizes, size of reservoirs, and $\Omega = \{4, 5\}$ for SONDA data 118
Figure 18 – Comparison of the model accuracy in terms of the average RMSE, MAPE, and U considering the different number of concepts and sub-reservoir when $\Omega \in \{2, 3\}$(Malaysia load data) 123
Figure 19 – Comparison of the model accuracy in terms of the average RMSE, MAPE, and U considering the different number of concepts and sub-reservoir when $\Omega \in \{4, 5\}$(Malaysia load data) 123
Figure 20 – General structure of the proposed MO-RHFCM method. 133

Figure 21 – A simple example of the proposed MO-RHFCM mechanism considering $N = 2$, $N_{SR} = 2$,$\Omega = 2$ and $k = 3$. Firstly, the original time series is fed to the reservoir. Interestingly, each variable is presented by one FCM with $k = 3$ concepts. A, B, and C are used to model variable y_1 while variable y_2 is modeled by another FCM with D, E, and F. Since $\Omega = 2$, eight randomly weight matrices are set in each sub-reservoir. After that, the generated outputs are injected into the output layer. Finally, the least squares algorithm is applied to train the output layer and produce the final predicted values. 134
Figure 22 – General structure of the proposed MIMO-FCM method. Firstly, the original data set is transmitted through KPCA to create M principal components. Then each component is predicted by one univariate cascade randomized HFCM unit. Finally, inverse KPCA is applied to reconstruct the original time series. 148
Figure 23 – Internal structure of each univariate CR-HFCM unit. Each CR-HFCM is composed of three layers: the input layer, the reservoir, and the output layer. The reservoir consists of L sub-reservoirs in which weights are initialized randomly so that the Echo State Property (ESP) condition is satisfied in the ESN reservoir computation. Each extracted principal component is only fed into the first block to generate the input for both the second block and the output layer and so on. Accordingly, the generated output from each randomized HFCM block is injected as input to the next successive block as well as to the output layer. Finally, the least squares algorithm is employed to train the output layer and generate the predicted value for each principal component. 149
Figure 24 – A simple example of the proposed MIMO-FCM mechanism considering $\Omega = 2$, $M = 1$, $N_{SR} = 2$. 152

LIST OF TABLES

Table 1 – Most widely used FCM activation functions . . . 35
Table 2 – Review of fuzzification and defuzzification methods used in FCM-based forecasting techniques . . . 43
Table 3 – Summary of the most relevant time series forecasting methods using FCM in the literature . . . 67
Table 4 – Proposed FCM learning techniques in the literature . . . 81
Table 5 – The effect of the bias on the accuracy of the proposed HFCM-FTS method with Sigmoid activation function . . . 95
Table 6 – Performance of the proposed HFCM-FTS method using different activation functions when bias has been considered . . . 95
Table 7 – Comparison of the accuracy of the proposed HFCM-FTS by considering ReLU activation function and bias weights with other methods . . . 96
Table 8 – Hyper-parameters for GA learning algorithm . . . 96
Table 9 – Summary statistics of the ten-time series datasets used in the experiments 107
Table 10 – SONDA dataset variables . . . 107
Table 11 – Malaysia dataset variables . . . 108
Table 12 – Proposed activation functions to design R-HFCM model . . . 112
Table 13 – The summary of the model performance for SONDA dataset considering different k and N_{SR} when $\Omega = 2$ and activation function is Softplus . . . 113
Table 14 – The effect of using different activation functions on the model accuracy 114
Table 15 – The summary of the best performance of the model considering different activation function(f), $k = 5$ and $\Omega = 2$ . . . 114
Table 16 – The effect of the order on the accuracy of the proposed R-HFCM model for different k and N_{SR} . . . 116
Table 17 – The summary of the best performance of the model considering different k, N_{SR} and Ω . . . 117
Table 18 – The summary of the best model performance considering 60 sub-reservoirs and $k \in \{3, 4, 5\}$ . . . 117
Table 19 – Evaluation of the accuracy of the proposed method against other models in terms of RMSE and NRMSE . . . 120
Table 20 – The Summary of statistical ranking of models . . . 120
Table 21 – The summary of the best R-HFCM performance considering different map sizes $k \in \{3, 4, 5, 7, 10\}$ . . . 124
Table 22 – Comparison of the proposed R-HFCM method with other models in terms of average MAPE, average RMSE and NRMSE . . . 125

Table 23 – The summary of the best R-HFCM performance in terms of average RMSE . . . 127
Table 24 – The summary of the R-HFCM performance in terms of average RMSE (AEP and DEOK data) . . . 128
Table 25 – Evaluation of the accuracy of the proposed method against other models 129
Table 26 – The summary of the rank of models considering eight various datasets . 130
Table 27 – The summary of the best performance of the model for SONDA dataset considering different k, N_{SR} and Ω . . . 140
Table 28 – The summary of the best performance of the model for Malaysia dataset considering different k, N_{SR} and Ω . . . 142
Table 29 – Evaluation of the model's performance in terms of average NRMSE . . . 142
Table 30 – Optimal parameters of each dataset for baseline models . . . 143
Table 31 – Evaluation of the accuracy of the proposed method against other baseline models . . . 144
Table 32 – Evaluation of the models' accuracy in terms of average NRMS over all variables for each dataset . . . 144
Table 33 – Kruskal-Wallis mean comparison test results . . . 144
Table 34 – The summary of the ranking of the forecasting models . . . 144
Table 35 – Comparison among the number of parameters of our proposed model with deep learning methods . . . 145
Table 36 – Optimal parameters of our proposed model for each data set . . . 156
Table 37 – Optimal parameters of each data set for RF and SVR models . . . 157
Table 38 – Evaluation of the accuracy of the proposed method against other baseline models in terms of Average RMSE and average MAE (AE-DS data set) 158
Table 39 – Evaluation of the accuracy of the proposed method against other baseline models in terms of Average RMSE and average MAE (SH-DS data set) 159
Table 40 – PM2.5 and PM10 prediction accuracy of our proposed models and baseline methods over six stations data (AQB-6) in terms of RMSE and MAE . . . 159
Table 41 – PM2.5 and PM10 prediction accuracy of our proposed models and baseline methods over all stations data (AQB-12) in terms of RMSE and MAE . . . 160
Table 42 – Evaluation of the models' accuracy in terms of average NRMSE . . . 160
Table 43 – Kruskal-Wallis mean comparison test results . . . 161
Table 44 – The summary of the ranking of the forecasting models . . . 161
Table 45 – Comparison among the number of parameters of our proposed model with deep learning methods . . . 162

Table 46 – Complete results of the proposed R-HFCM for different values of Ω, k, and N_{SR} in terms of average RMSE, MAPE, and U metrics (Malaysia data). 184
Table 47 – Complete results of the proposed R-HFCM for different values of order, map size and reservoir size in terms of average RMSE (NREL and GEFCom2012 datasets). 186
Table 48 – Wilcoxon *post hoc* test results for SONDA dataset. 188
Table 49 – Kruskal-Wallis mean comparison test results for datasets NREL (DHHL_1, DHHL_2, DHHL_3), GEFCom2012 (zone1, zone2, zone3) and PJM (AEP and DEOK) 188
Table 50 – Wilcoxon *post hoc* test results for datasets NREL (DHHL_1, DHHL_2, DHHL_3), GEFCom2012 (zone1, zone2, zone3) and PJM (AEP and DEOK) 189

LIST OF ABBREVIATIONS AND ACRONYMS

ARIMA	Auto-Regressive Integrated Moving Average
ACO	Ant Colony Optimization
AHL	Active Hebbian Learning
ANFIS	Adaptive Neuro Fuzzy Inference System
AI	Artificial Intelligence
ARIMA	Autoregressive Integrated Moving Average
ARMA	Auto Regressive Moving Average
ANN	Artificial Neural Networks
ABC	Artificial Bee Colony
BSTS	Bayesian Structural Time Series
BB-BC	Big Bang-Big Crunch
BPNN	Back Propagation Neural Network
BCM	binary cognitive maps
BDD-FCM	Belief-Degree-Distributed Fuzzy Cognitive Maps
BDA	Balanced Differential Algorithm
CM	Cognitive Map
CNN	Convolutional Neural Networks
DD-NHL	Data-Driven Nonlinear Hebbian Learning
DHL	Differential Hebbian Learning
DE	Differential Evolutionary
DCN	Dynamical Cognitive Networks
DRFCM	Dynamic Random Fuzzy Cognitive Maps
ESN	Echo State Network

ESP	Echo State Property
E-FCM	Evolutionary Fuzzy Cognitive Maps
EWT	Empirical Wavelet Transformation
EGDA	Extended Great Deluge Algorithm
EMD	Empirical Mode Decomposition
FTS	Fuzzy Time Series
FCMs	Fuzzy Cognitive Maps
FBN	Fuzzy Boolean Nets
FTCM	Fuzzy Time Cognitive Maps
FGCM	Fuzzy Grey Cognitive Maps
FRI-FCM	Fuzzy Rules incorporated in Fuzzy Cognitive Maps
GFCM	Generalised FCM
GA	Genetic Algorithm
GRU	Gated Recurrent Unit
HOFTS	High Order FTS
HFCM	High-Order FCM
HFCM-FTS	High Order FCM-High Order FTS
IoT	Internet of Things
iFCM	Intuitionistic Fuzzy Cognitive Maps
IVI-FS	Interval-Valued Intuitionistic Fuzzy Sets
ITS	Interval-Valued Time Series
KPCA	Kernel Principal Component Analysis
LS	Least Squares
LSTM	Long Short-Term Memory
MLP	Multi-Layer Perceptron
MISO	Multiple Input Single Output

MIMO	Multiple Input Multiple Output
MAPE	Mean Absolute Percentage Error
MVFTS	Multivariate FTS
NHL	Non-Linear Hebbian Learning
NRMSE	Normalized Root Mean Square Error
PCA	Principal Component Analysis
PSNN	Pi-Sigma Neural Network
PSO	Particle Swarm Optimization
PWFTS	Probabilistic Weighted Fuzzy Time Series
QAR	Quantile Auto-Regression
R-HFCM	Randomized HFCM
RFCM	Rough Cognitive Maps
RC	Reservoir Computing
RMSE	Root Mean Square Error
RNN	Recurrent Neural Networks
RB-FCM	Rule-Based Fuzzy Cognitive Maps
RCGA	Real-Coded Genetic Algorithm
SARIMA	Seasonal Auto-Regressive Integrated Moving Average
SVM	Support Vector Machine
SA	Simulated Annealing
TSF	Time Series Forecasting
UoD	Universe of Discourse
WHOFTS	Weighted High Order Fuzzy Time Series

LIST OF SYMBOLS

Y	crisp time series
$y(t)$	crisp time series elements at time t
t	time index
U	Universe of Discourse of Y
F	fuzzy time series
$f(t)$	fuzzy time series elements at time t
Ω	model order
k	number of fuzzy sets plus the number of FCM concepts
μ	fuzzy membership function
$\alpha \in [0, 1]$	Alpha-Cut
$\tilde{A}$	the linguistic variable for univariate
$H \in \mathbb{N}^+$	The forecasting horizon
$\hat{y}(t+1)$	a point forecast for time $t+1$
β	Scaling Parameter
$a(t)$	State value of FCM concept
ρ	Eigenvalue of Matrix
ρ_{max}	Maximum Eigenvalue of Matrix
N_{SR}	Sub-Reservoir number
W	Weight Matrix
w^0	Bias weight
w_{ij}	Weight among concept c_i and c_j
f	Activation Function
λ	Least Squares Coefficient

$t \in T$	The time index
$T \in \mathbb{N}^+$	The total length of Y (or the number of samples)
N	The number of variables
M	The number of principal components
γ	The kernel coefficient of KPCA

CONTENTS

1 INTRODUCTION 22
1.1 Objectives 25
1.2 Thesis Structure 25
2 FUZZY COGNITIVE MAPS 27
2.1 Fuzzy Cognitive Map 30
2.1.1 Fundamentals 30
2.1.2 High Order Fuzzy Cognitive Map (HFCM) 35
2.1.3 Advanced Aspects 36
2.1.3.1 Extensions of FCM 36
2.1.3.2 Dynamic Properties of FCM 37
2.1.3.3 FCM against Artificial Neural Network (ANN) 37
2.1.4 Application Domains of FCM 38
2.2 Time Series Forecasting using Fuzzy Cognitive Map 39
2.2.1 Terminology of the problem 39
2.2.2 Design Issues 41
2.2.3 Overview of FCM-based models in the literature 50
2.3 Performance Evaluation Metrics 71
2.4 Learning Algorithms 71
2.4.1 Hebbian-Based Learning Methods 71
2.4.2 Population-Based Learning Techniques 73
2.4.3 Hybrid Methods 76
2.4.4 Other Methods 77
2.5 Discussion 84
2.6 Chapters' Highlights 86
3 PROPOSED HFCM-FTS TECHNIQUE 88
3.1 Proposed HFCM-FTS methodology 88
3.1.1 FCM-FTS Model 88
3.1.2 HFCM-FTS method 93
3.2 Computational Experiments 94
3.3 Chapter's Highlights 97
3.3.1 Model's limitation and challenges 97
4 PROPOSED R-HFCM TECHNIQUE 98
4.1 Echo State Network 99
4.2 Proposed R-HFCM method 101
4.2.1 Training Procedure 102

4.2.1.1 Randomized Model Initialization . . . 102
4.2.1.2 Partitioning . . . 103
4.2.1.3 Fuzzification . . . 104
4.2.1.4 Activation . . . 104
4.2.1.5 Defuzzification . . . 104
4.2.1.6 Least Squares coefficients determination . . . 104
4.2.2 Forecasting Procedure . . . 105
4.2.2.1 Fuzzification . . . 105
4.2.2.2 Activation . . . 105
4.2.2.3 Defuzzification . . . 105
4.3 Computational Experiments . . . 106
4.3.1 Datasets . . . 106
4.3.1.1 SONDA Dataset . . . 107
4.3.1.2 Malaysia Dataset . . . 107
4.3.1.3 NREL solar energy dataset . . . 108
4.3.1.4 GEFCom 2012 load dataset . . . 109
4.3.1.5 PJM Hourly Energy Consumption Data . . . 109
4.3.2 Experimental Methodology . . . 109
4.3.3 SONDA Case Study . . . 111
4.3.3.1 The influence of map and reservoir size . . . 112
4.3.3.2 The influence of activation function (f) . . . 114
4.3.3.3 The influence of order . . . 115
4.3.3.4 Comparison with other Fuzzy Time Series methods . . . 118
4.3.3.5 Statistical testing . . . 120
4.3.4 Malaysia Case Study . . . 121
4.3.4.1 Forecasting Hourly electric load dataset . . . 121
4.3.4.2 Comparison With Other Methods . . . 124
4.3.5 Other case studies . . . 125
4.3.5.1 NREL dataset(DHHL_1, DHHL_2, DHHL_3) . . . 126
4.3.5.2 GEFCom2012 dataset (zone1, zone2, zone3) . . . 126
4.3.5.3 PJM dataset (AEP and DEOK) . . . 127
4.3.5.4 Comparison with other methods . . . 127
4.3.5.5 Statistical testing . . . 128
4.4 Chapter's Highlights . . . 130
5 PROPOSED MO-RHFCM TECHNIQUE . . . 132
5.1 Proposed MO-RHFCM model . . . 132
5.1.1 Training Procedure . . . 134
5.1.1.1 Randomized Model Initialization . . . 134
5.1.1.2 Partitioning . . . 135

5.1.1.3 Fuzzification . . . 135
5.1.1.4 Activation . . . 135
5.1.1.5 Defuzzification . . . 136
5.1.1.6 Least Squares coefficients determination . . . 137
5.1.2 Forecasting Procedure . . . 137
5.1.2.1 Fuzzification . . . 137
5.1.2.2 Activation . . . 137
5.1.2.3 Defuzzification . . . 138
5.2 Computational Experiments . . . 138
5.2.1 Case studies . . . 139
5.2.2 Experimental Methodology . . . 139
5.3 Results . . . 140
5.3.1 Parameter setting for MO-RHFCM . . . 141
5.3.2 Baseline models . . . 141
5.3.3 Comparison against baselines . . . 142
5.3.4 Statistical testing . . . 143
5.4 Chapter's Highlights . . . 145
6 PROPOSED MIMO-FCM TECHNIQUE . . . 146
6.1 Preliminaries . . . 147
6.2 Proposed MIMO-FCM model . . . 148
6.2.1 Training Procedure . . . 150
6.2.1.1 Embedding . . . 150
6.2.1.2 Randomized model initialization . . . 150
6.2.1.3 Partitioning . . . 151
6.2.1.4 Fuzzification . . . 151
6.2.1.5 Activation . . . 151
6.2.1.6 Defuzzification . . . 151
6.2.1.7 Least Squares coefficients determination . . . 152
6.2.2 Forecasting Procedure . . . 152
6.2.2.1 Embedding: . . . 152
6.2.2.2 Fuzzification: . . . 153
6.2.2.3 Activation: . . . 153
6.2.2.4 Defuzzification . . . 153
6.3 Computational Experiments . . . 154
6.3.1 Case studies . . . 154
6.3.2 Experimental methodology . . . 155
6.4 Results . . . 155
6.4.1 Parameter setting for MIMO-FCM . . . 156
6.4.2 Baseline models . . . 156

6.4.3 Comparison against baselines . . . 157
6.4.4 Statistical testing . . . 161
6.5 Chapter's Highlights . . . 161
7 CONCLUSION AND FUTURE WORKS . . . 163
7.1 Summary of contributions . . . 165
7.2 Methods limitations and future investigations . . . 166
7.3 Publications . . . 167

REFERENCES . . . 168

Appendix A Long Tables- Complete R-HFCM results for Malaysia, NREL, and GEFCom2012 . . . 184
Appendix B Complete statistical results for SONDA, NREL, GEFCom2012, and PJM data . . . 188

1 INTRODUCTION

Time series forecasting (TSF) is a challenging problem that has been long attended by many researchers in various fields such as medicine, economy, finance, hydrology, energy, and so on (BOSE; MALI, 2019; TEALAB, 2018). Despite the existence of various forecasting procedures in the literature, the accurate prediction of future values remains an open challenge in various domains. Complicating factors are, for instance, the existence of non-linearity and uncertainty in some real phenomena (SHANCHAO; LIU, 2018; PANIGRAHI; BEHERA, 2020). Classic statistical time series forecasting methods, such as Auto Regressive Integrated Moving Average (ARIMA), Quantile Auto-Regression (QAR), and the Bayesian Structural Time Series (BSTS) tend to be time-consuming, lack scalability and explainability, and are not well-suited to deal with uncertainty and complex problems in the real world (SILVA, 2019).

Fuzzy time series (FTS), presented by Song e Chissom (1993), Song e Chissom (1994), Qiang, Leland e Chissom (1997) have shown the ability to handle the mentioned problems. They are known to be simple, explainable, flexible, easy to update and read, and highly accurate (BOSE; MALI, 2019). In Silva (2019), it has been shown that producing forecasts through a weighted sum of fuzzy rules can significantly improve the accuracy of the FTS models. Nevertheless, the right way to assign weights to each fuzzy rule remains an open question.

Fuzzy Cognitive Maps (FCMs), proposed by Kosko (1986), combine concepts of fuzzy logic and neural networks, and it can be seen as a weighted FTS approach in which the weights are learned from the data. This weighted model consists of nodes (representing concepts) and signed directed relations (causal relations) between pairs of concepts. It has been used to represent the dynamic behavior of various complex systems with a high ability to deal with uncertainties (SHANCHAO; LIU, 2018; PAPAGEORGIOU et al., 2019; VANHOENSHOVEN et al., 2018; VANHOENSHOVEN et al., 2020; YUAN et al., 2020).

Time series forecasting by FCMs, in general, consists of two stages (GAO; DU; YUEN, 2020). Firstly, one has to design the appropriate architecture using common techniques including granularity (STACH; KURGAN; PEDRYCZ, 2008b), membership values representation (SONG et al., 2010), Fuzzy c-means clustering (LU et al., 2014) and combination of wavelet transformation with empirical mode decomposition (EMD) (LIU; LIU, 2020; SHANCHAO; LIU, 2018). Secondly, the model has to learn the weight matrices which is one of the main concerns of the method. Salmeron et al. (2019), Felix et al. (2019), Orang et al. (2020) applied population-based methods in the learning process. However, as the number of parameters grows, the use of these methods becomes prohibitive. Therefore, evolutionary learning has been replaced with other techniques in some references. For

instance, the least squares in Feng, Lu e Yang (2021a), the ridge regression in Shanchao e Liu (2018), Yuan et al. (2020), Wu et al. (2019), the Bayesian ridge regression in Liu e Liu (2020), and the Moore-Penrose inverse in Vanhoenshoven et al. (2020).

In spite of the advances in training procedures, they can still be very time-consuming due to the high number of parameters that must be adjusted. In this context, one of the main objectives of this work is to reduce this cost by introducing univariate and multivariate FCM-based time series forecasting models inspired by the Echo State Network (ESN) reservoir computing. In the ESN, the weights in the hidden layer are assigned randomly and are not trainable. Thus, the training cost can be significantly decreased.

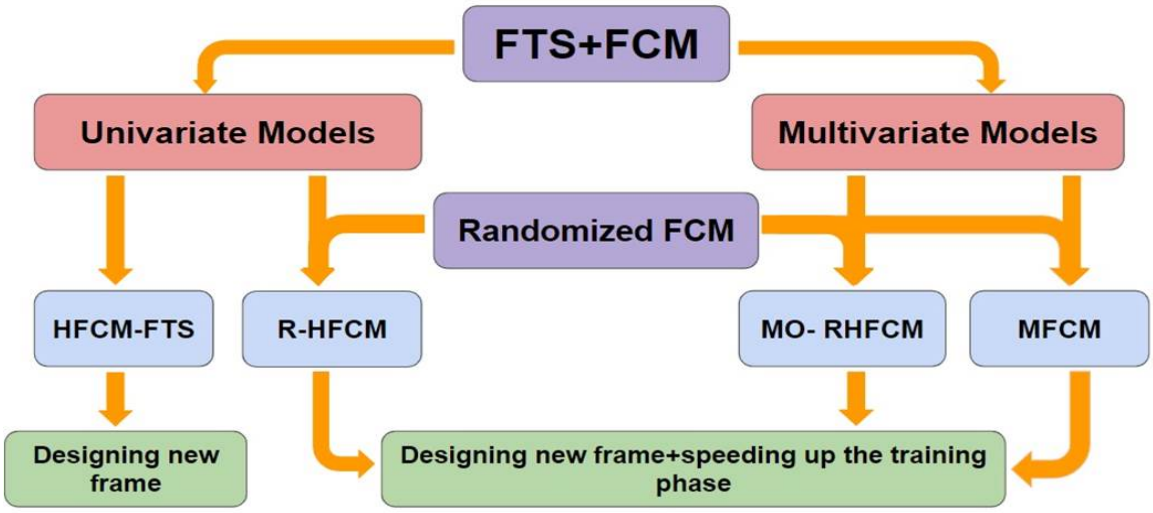


Figure 1 – Proposed thesis methods and contributions.

Figure 1 shows the main contributions and proposed techniques in this thesis. As can be seen from this figure, the main contribution is to combine FCM and FTS to generate univariate forecasting models (HFCM-FTS and R-HFCM) and multivariate models (MO-RHFCM and MFCM). Therefore, this thesis is mainly focused on both challenging aspects of FCM-based time series forecasting models: designing a new FCM structure as well as speeding up the training phase.

It is worth observing that HFCM-FTS is a hybrid univariate FCM-based method namely combines high order FCM (HFCM) and high order FTS (HOFTS) trained via genetic algorithm (GA) (ORANG et al., 2020). This method is only focused on designing a new frame of FCM. Since the proposed training methodology will get very time-consuming as the number of concepts increases, the concepts of HFCM and ESN are merged to build a new forecasting technique, called Randomized HFCM (R-HFCM) (ORANG et al., 2022) trained via least squares (LS) algorithm. The architecture of R-HFCM is similar to ESN and composed of three layers including an input layer, reservoir, and output layer such that only the output layer is trainable. It is worth noting that the reservoir consists

of a group of randomized HFCM-FTS models (as sub-reservoirs) in which weights are initialized randomly according to weight initialization in ESNs.Thus, the training cost can be significantly decreased compared to the HFCM-FTS method. As a result, the R-HFCM technique is focused on both aspects including designing a new forecasting structure as well as speeding up the training phase.

To fill the absence of Multiple-Input Multiple Output (MIMO) FCM-based methodologies, two randomized-based methods called MO-RHFCM and MFCM are introduced. MO-RHFCM is an extended frame of the univariate R-HFCM technique and is designed to predict low-dimensional multivariate time series. MO-RHFCM is a hybrid method merging the concepts of multivariate FTS (MVFTS), HFCM, and ESN. Although the structure of MO-RHFCM is inspired by the R-HFCM method, each univariate HFCM-FTS is replaced by a combination of MVFTS and HFCM termed MVFCM-FTS. In this case, each MVFCM-FTS receives all the variables as input and generates outputs fed to the output layer. Then LS algorithm is applied to train the output layer and generate the final predicted values. Therefore, MO-RHFCM is also focused on both aspects: designing a new frame and speeding up the training phase.

On the other side, the study explores the application of MIMO-FCM (MFCM) to assess the potential benefits of using randomized FCMs as a MIMO forecasting model for high-dimensional time series datasets in IoT(Internet of Things) applications. MFCM is a hybrid method combining data embedding transformation, HFCM, and ESN. Firstly, the original data set is transmitted through kernel principal component analysis (KPCA) to create a user-defined number of principal components. Then each component is predicted by one univariate cascade randomized HFCM unit. Finally, inverse KPCA is applied to reconstruct the original time series. The underlying model within MFCM is a cascade of R-HFCMs (or CR-HFCM) models. Each CR-HFCM consists of three layers, including the input layer, reservoir (internal layer), and output layer, such that only the output layer is trainable using the least squares minimization algorithm. The weights inside each sub-reservoir are selected randomly and remain fixed during the training process. Accordingly, MFCM also considers both aspects in terms of new structure and fast training.

The results indicate the efficiency and efficacy of our proposed models in handling a range of univariate and multivariate time series datasets, including both low-dimensional and high-dimensional time series. Our proposed methods outperform a variety of baseline and state-of-the-art machine learning and deep learning forecasting methods with competitive results.

1.1 Objectives

The focal purpose of this thesis is to design a new scalable forecasting technique based on the FCM concept. The specific objectives are pointed out as follows:

- Combine FTS and FCM into new hybrid models for time series modeling and forecasting;
- Propose randomized FCM-based forecasting techniques as reservoir computing models for the first time in the literature to predict multivariate and univariate time series;
- Investigate the elements affecting the accuracy of the proposed models;
- Introduce MIMO forecasting models based on FCMs for the first time in the literature;
- Investigate the potential benefits of the proposed method for dealing with high-dimensional time series data;
- Compare the efficacy and efficiency of the proposed method with other baseline and state-of-the-art machine learning and deep learning techniques.

1.2 Thesis Structure

The thesis is organized as follows:

- Chapter 2: provides an overview of recent developments on time series forecasting methods using FCMs and explores potential future research opportunities. Also, it covers an introduction and revision on some corresponding properties/fundamentals of FCM (including the structure of FCMs and reasoning rules, high order FCMs, extensions of FCMs, and dynamic properties of FCMs) and FCMs learning methodologies.
- Chapter 3: introduces a hybrid method, named HFCM-FTS, which combines HFCM and HOFTS, where the weight matrices associated with the state transitions are learned via the genetic algorithm from the data.
- Chapter 4: introduces a novel univariate time series forecasting technique, which is composed of a group of randomized high-order FCM models labeled R-HFCM. The novelty of this model is relevant to merging the concepts of FCM and ESN where the least squares algorithm is applied to train the model.
- Chapter 5: proposes an FCM-based methodology to predict low-dimensional multivariate time series for MIMO systems, called MO-RHFCM. This hybrid model

combines MVFTS, ESN, and randomized HFCM trained with time-effective learning inspired by reservoir computing.

- Chapter 6: introduces an FCM-based methodology to predict high-dimensional multivariate time series for MIMO systems named MIMO-FCM. MIMO-FCM is a hybrid method combining data embedding transformation (KPCA), HFCM, and ESN, which is trained using the least squares algorithm inspired by reservoir computing.
- Chapter 7: summarizes the obtained results by highlighting the limitations as well as suggestions for future work directions.

2 FUZZY COGNITIVE MAPS

Tackling [1] complex problems in a highly reliable way has always been a major challenge for researchers. Increasing complexity arises due to some characteristics, such as uncertainty, ambiguity, inconsistency, and high dimensionality. Some of these features are common among most real-world problems, which are considered complex and dynamic. In other words, since the data and relations in real-world applications are usually highly complex and inaccurate, modeling complex systems based on observed data is a challenging task, especially for large-scale and non-stationary datasets. Therefore, to cover and address these difficulties, the existence of a computational system with the capability of extracting knowledge from the complex system with the ability to simulate its behavior is essential (YE et al., 2015).

Soft computing methods, such as fuzzy logic, neural networks, genetic algorithms, or a combination of these, have the potential to make complex problems tractable, to deal with issues of non-linearity, uncertainty, and impreciseness, and to obtain more practical solutions. Two types of methods are used for analyzing and modeling dynamic systems, namely quantitative and qualitative approaches. In some cases, modeling complex and nonlinear systems through quantitative techniques is difficult and costly (AGUILAR, 2005). In contrast, qualitative methods do not suffer from the mentioned restrictions. Fuzzy Cognitive Map (FCM) is a kind of important qualitative soft computing technique proposed by Kosko (1986), as an extension of cognitive maps. FCM has attracted great attention among researchers, with a high capability of modeling dynamic and complex problems.

FCM is a graph knowledge-based method and, in the same way as any traditional cognitive map, is composed of concepts and causal relationships among them. The difference is that in FCM the concepts are modeled by fuzzy sets and the relationships among them are defined by fuzzy connections. In other words, the existence of fuzzy feedback in the FCM structure creates an option to extract and model the causal knowledge. According to Papageorgiou (2014), FCM can be better described by two defining characteristics. The first one is the type of relationship among the concepts with different intensities, as represented by uncertain fuzzy numbers. The second one is the system dynamics, i.e., activation of the concepts evolves with time. In FCM structures composed of feedback, changes in one concept may influence all the others. These models have the ability to incorporate human knowledge and adapt it through learning procedures (PARSOPOULOS; PAPAGEORGIOU; VRAHATIS, 2002). The fundamental features of FCM are briefly

[1] This chapter is derived from a survey paper titled: Time series forecasting using fuzzy cognitive maps: a survey, published in Artificial Intelligence Review journal (ORANG; SILVA; GUIMARAES, 2022)

presented in Section 2.1.

FCM seems to have similarities to recurrent neural networks (RNNs) and can be classified as a neuro-fuzzy method. In other words, FCM is a mixture of aspects from fuzzy logic, neural networks, and expert systems, making it a powerful tool for simulating and studying the dynamic behavior of complex systems. Therefore, FCM can learn due to its dynamic structure from an Artificial Intelligence (AI) point of view. This learning capability makes the FCM able to adapt its structure and hence its computational behavior (PARSOPOULOS; PAPAGEORGIOU; VRAHATIS, 2002; PAPAGEORGIOU, 2011b; SALMERON, 2009). By considering this feature, diverse methods of learning algorithms have been proposed in the literature to improve the performance of FCM, such as Differential Hebbian Learning (DHL) (DICKERSON; KOSKO, 1993), Genetic Algorithm (GA) (FROELICH; JUSZCZUK, 2009; STACH et al., 2005), Ant Colony Optimization (ACO) (DING; LI, 2011) and many others (see Section 2.4).

Qualitative modeling, ease of perception, high ability to deal with uncertainties, capability to represent nonlinear and causal behaviors, flexibility, and explainability can be accounted as interesting attributes of FCM (VLIET; KOK; VELDKAMP, 2010). According to these properties, plenty of papers were published in the field of FCM that cover diverse domains involving decision-making, control systems, time series forecasting, classification, electrical and software engineering, and medicine (STACH; KURGAN; PEDRYCZ, 2005; FELIX et al., 2019; PAPAGEORGIOU, 2014).

FCMs and their extensions have been exploited widely in the area of time series forecasting (FELIX et al., 2019). These weighted fuzzy time series forecasting models are employed to predict univariate and multivariate time series considering different architectures as well as various learning algorithms. Basically, time series prediction using FCM includes two stages. Firstly, constructing the proper architecture of FCM using some strategies that may involve granularity (STACH; KURGAN; PEDRYCZ, 2008b), fuzzy c-means clustering (LU et al., 2014), grid partitioning (ORANG et al., 2020), wavelet transformation and empirical mode decomposition (EMD) (GAO; DU; YUEN, 2020). Secondly, learning the optimal or near-optimal values of the weight matrix – the learning methods can be classified into three groups consisting of Hebbian-based methods, population-based methods, and hybrid methods, which are a combination of Hebbian-based and evolution-based types of learning algorithms. Based on the references (GAO; DU; YUEN, 2020; ZOU; LIU, 2018), the major proportion of the references has focused on population-based algorithms to optimize the weight matrix, like GA (STACH; KURGAN; PEDRYCZ, 2008b; FROELICH; SALMERON, 2014; FROELICH et al., 2012; LU; YANG; LIU, 2013) and Particle Swarm Optimization (PSO) (LU et al., 2014; HOMENDA; JASTRZEBSKA; PEDRYCZ, 2014a; HOMENDA; JASTRZEBSKA; PEDRYCZ, 2014c; LU; YANG; LIU, 2014). However, some drawbacks including time

consuming and convergence issues limit their functionality particularly when raising the number of parameters. These limitations imply that there is a genuine need to develop and design FCM-based forecasting models with more time-efficient and robust learning strategies. Therefore, some researchers replaced evolutionary learning methods with other regression-based techniques such as ridge regression in Shanchao e Liu (2018), Yuan et al. (2020), Wu et al. (2019), Bayesian ridge regression in Liu e Liu (2020), Moore-Penrose inverse in Vanhoenshoven et al. (2020) and least squares in Feng, Lu e Yang (2021a).

The crux purpose of this study is to present an overview of the progress in the studies of time series forecasting utilizing FCM as a modeling technique. Although there are some review papers on FCM in the literature, they often have focused on a partial and specific subject. For instance, two surveys on FCM were presented in 2005. One is a survey of the several applications of FCM in different domains, which was presented by Aguilar (2005), and the other is a review of FCM learning methods, which was proposed by Stach, Kurgan e Pedrycz (2005). In 2011, another survey on FCM was published by Papageorgiou (2011a) focusing on learning methodologies and algorithms for FCM. In 2013, Papageorgiou e Salmeron (2012) published a survey on FCM applications and trends in diverse scientific fields over a decade (2000-2010). In 2018, a review study of FCM in medicine was reported by Amirkhani et al. (2017). To the best of our knowledge, the latest review paper on FCM was published by Felix et al. (2019), considering several aspects containing some fundamentals of FCM, applications of FCM to time series forecasting and time series classification, as well as providing an overview of software tools for FCM until 2017. In the context of fuzzy-based models for time series forecasting, a brief review of modeling approaches based on fuzzy time series (FTS) is given by Singh (2017) in 2017 and a comprehensive review of FTS models is presented by Bose e Mali (2019) in 2019. However, these reviews on FTS do not approach FCM models for time series forecasting.

In our investigation, the main effort is concentrated on providing a comprehensive review of up-to-date FCM-based time series forecasting techniques until mid-2022. Also, this thesis covers a comprehensive review of the proposed learning algorithms for FCM (highlighting those that have been applied for time series forecasting models), as well as a review of employed fuzzification and defuzzification techniques in the design of FCM-based forecasting models. Furthermore, some major characteristics and advanced aspects of FCM are discussed, including high-order FCM (HFCM), extensions of FCM, dynamic properties of FCM, and similarities and differences between FCM and Artificial Neural Networks (ANNs). It is worth noting that more than 200 papers have been analyzed and referenced or commented on in this investigation.

This chapter is structured as follows:

- In Section 2.1, some fundamental features of FCM are reviewed

- Section 2.2 focuses on time series forecasting using FCM.
- Section 2.3 lists performance evaluation metrics as well as datasets employed in the reviewed papers.
- Section 2.4 gives a summary of the most relevant learning strategies for FCM learning. Table 3 summarizes a list of the FCM-based time series forecasting models reviewed in Section 3, taking into account some important parameters such as learning methods, order number, presence of bias, the number of nodes, activation function, and prediction horizons. Table 4 summarizes the learning methods for FCM in general.
- Section 2.5 provides a discussion and insights over the proposed methodologies covered in the literature.

2.1 Fuzzy Cognitive Map

2.1.1 Fundamentals

Fuzzy Cognitive Map (FCM), proposed by Kosko (1986), is a particular family of Cognitive Map (CM) theory that was introduced by Axelrod a decade before (AXELROD, 1976). The term CM was introduced to investigate the cognitive activities of rats using some learning experiments on their choice of an appropriate path to food. Later Axelrod, a political scientist, used the knowledge of people to form the CM in the form of causal relationships between concepts to formulate decision-making in political science.

CM is a causal and qualitative model composed of concepts (nodes) and the causal connection between any two concepts (edges). These directed signed arrows between the concepts are represented as weights that reflect the effect of one node on another one. These assigned directions between concepts of the CM can be positive or negative. In Eden, Ackermann e Cropper (1992), Eden (2004) several methodologies have been suggested to represent the CM and analyze its complexity. If the node influences others, we say it is a cause and if influenced by others, it is an effect (cause and effect causality [2]). In binary cognitive maps (BCM) the concept labels are mapped to binary states denoted as $a_i \in \{0, 1\}$, where the value 1 means that the concept is activated. Also, the weights belong to the crisp set $w_{ij} \in \{-1, 0, 1\}$ in BCM (AXELROD, 1976). The value 1 represents positive causality, thus the activation (change from 0 to 1) of concept c_i occurs concurrently with the same activation of concept c_j or that deactivation (change from 1 to 0) of c_i occurs concurrently with the same deactivation of concept c_j. The value -1 represents

[2] Notic that causality in FCMs refers to the relationships between the concepts or variables represented in the map. These causal connections determine how the values of the concepts propagate and interact within the FCM.

the opposite situation, in which the activation of c_i deactivates the concepts c_j or vice versa. Those edges with $w_{ij} = 0$ mean that there are no concurrently occurring changes in the states of the concepts. CM was used in various areas such as decision-making (EDEN, 2004; EDEN; ACKERMANN, 2004; KLEIN; COOPER, 1982), financial (KIM, 2004), and other areas.

Accordingly, FCM is a hybrid soft computing technique based on the CM concept as a combination of fuzzy logic and CM to present uncertainties and complex characteristics of the systems. FCM has also emerged as a powerful paradigm for knowledge representation by providing a flexible mechanism for knowledge representation of intelligent systems (MIAO et al., 2001a; PAPAKOSTAS et al., 2011; SALMERON, 2009; TABER, 1991; TABER; YAGER; HELGASON, 2007). In FCM there exist partial causal relations among the concepts. Thus these connections can be represented by using fuzzy subsets which indicate vague values in the range $[-1, 1]$. It means that the elements of the weight matrix in FCM can be full positive $(+1)$, full negative (-1), or any value positioned in the interval $[-1, 1]$. Thus, this feature discriminates FCM from CM and enables FCM to deal with imprecise and uncertain data. For each FCM with k number of concepts, the causal relations among concepts are represented by one $k \times k$ square weight matrix (also called connection matrix) W.

$$W = \begin{pmatrix} w_{11} & \dots & w_{1k} \\ \vdots & \ddots & \vdots \\ w_{k1} & \dots & w_{kk} \end{pmatrix} \tag{2.1}$$

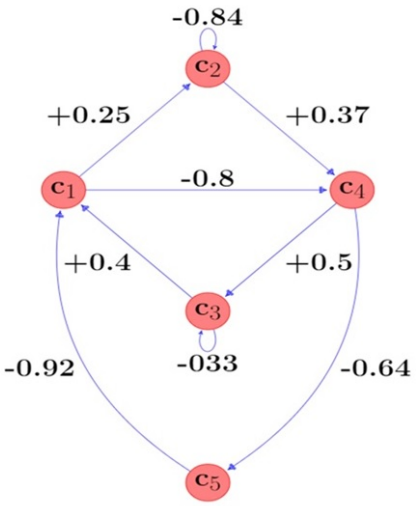
Each member w_{ij} of the matrix denotes the directed connection between concepts c_i and c_j which governs the intensity of the relationships between a pair of concepts to measure their influence degree on each other. On the other hand, the strength of causal relations among concepts is represented by absolute fuzzy values. The higher absolute value of weight among concepts leads to a stronger (closer) relationship among a couple of concepts. In detail, there exist three possible options for each individual of the weight matrix as follows:

1. $w_{ij} < 0$: means negative connection among concepts c_i and c_j. It means that a decrease/increase in the value of node c_i makes an increase/decrease in the value of node c_j.

2. $w_{ij} > 0$: means positive connection among concept c_i and c_j. It means that a decrease/increase in the node c_i makes a decrease/increase in the node c_j.

3. $w_{ij} = 0$: confirms null relation among nodes c_i and c_j.

Figure 2-A shows a simple example of FCM with $k = 5$ concepts while Figure 2-B presents the connections among the concepts, aggregated as a weight matrix, with the dimension of 5×5.



$$\mathbf{W} = \begin{pmatrix} 0 & +0.25 & 0 & -0.8 & 0 \\ 0 & -0.84 & 0 & +0.37 & 0 \\ +0.4 & 0 & -0.33 & 0 & 0 \\ 0 & 0 & +0.5 & 0 & -0.64 \\ -0.92 & 0 & 0 & 0 & 0 \end{pmatrix}$$

(A) (B)

Figure 2 – Simple FCM with 5 nodes (A) graphical structure (B) Weight matrix

From the graphical perspective, FCM is composed of concepts and connections among them that are represented by signed, weighted, and directed arcs. Thereupon, FCM is a signed, weighted digraph. In other words, the degree, sign, and direction of the influence of one concept on the others are required, apart from the number of concepts. $C = \{c_1, \ldots, c_k\}$ is the set of k concepts which are variables (nodes of the graph) that compose the system and W is the connection matrix among the concepts. FCMs are defined by the 4-tuple (C, W, a, f) where the third element, $a = (a_1, \ldots, a_k)$ is the state vector containing the activation degree of all concepts. At any time t, each concept c_i has an activation or fuzzy value restricted into the interval $[0, 1]$. On the other hand, FCM operation is based on the application of a specific inference rule, which calculates the next activation value of each concept by considering the total impact of the concepts directly connected to this one in each iteration. Generally, based on the literature Papakostas e Koulouriotis (2010), Felix et al. (2019), there exist different kinds of inference rules with respect to the past activation's values and self-connection relation for each concept. Equation (2.2) indicates Kosko's transition rule for concept c_i from time t to time $t + 1$, with no self-connection and without memory, considering the influence of all other concepts c_j. This is widely used in many FCM-based applications.

$$a_i(t+1) = f\left(\sum_{j=1, j\neq i}^{k} a_j(t) w_{ji}\right) \tag{2.2}$$

where k is the number of concepts and w_{ji} highlights the causal connection[3] between concepts c_j and c_i, whereas $a_i(t)$ denotes the sate value of concept c_i at time step t. It can be said that $a_i(t+1)$ and $a_i(t)$ are the response vector and state vector of FCM respectively. Once the values of all weights of the FCM (i.e. the relationship matrix of FCM) are determined, the model starts from a given initial state vector to carry out reasoning through the consecutive iterations computation according to (2.2). At each time step, the FCM generates a state vector that contains all concept activations. The state vector $a(t) = (a_1(t), \ldots, a_k(t))$ refers to the activation level of all concepts at time t. This updating rule can be iterated until the termination condition is satisfied. It means that in any iteration a new state value of the concepts is determined and after a certain number of repetitions, one of the three possible states can occur for FCM (KOSKO, 1986): converging to a fixed equilibrium point, to limited cycle or presenting a chaotic behavior. In both cases of chaotic or cyclic, the output may be partially unreliable because of the lack of stability. When the FCM reaches a fixed equilibrium point, it can be concluded that the FCM has converged.

Although the mentioned updating rule has been used in many FCM-based applications, other modifications have been developed in the literature (PAPAKOSTAS; KOULOURIOTIS, 2010; FELIX et al., 2019). The investigations proposed by Stylios e Groumpos (2004), Parsopoulos, Papageorgiou e Vrahatis (2002) introduce the modified version of the updating rule as the formula below indicates. In this way, in addition to the corresponding weights and activation values from other concepts, concepts consider their own past activation value known as memory factors for the nodes. Therefore, Equation (2.3) shows Kosko's updating rule with memory and without self-connection.

$$a_i(t+1) = f\left(a_i(t) + \sum_{j=1, j\neq i}^{k} a_j(t) w_{ji}\right) \tag{2.3}$$

Although the above equations ignore the effect of self-connection, equation (2.4) highlights the inference rule by adding self-connection.

$$a_i(t+1) = f\left(a_i(t) + \sum_{j=1}^{k} a_j(t) w_{ji}\right) \tag{2.4}$$

The other type of rule updating is described by equation (2.5) where only the effect of self-connection is taken into account and ignores the memory element.

$$a_i(t+1) = f\left(\sum_{j=1}^{k} a_j(t) w_{ji}\right) \tag{2.5}$$

[3] Notice that in order to compute the effect of the other concepts $c_j \neq c_i$ into c_i, we need to perform the summation of terms $w_{ji}a_j(t)$, given that w_{ji} reflects the causal connection between c_j and c_i.

Papageorgiou e Froelich (2012) have reported another version of updating the inference rule presented by equation (2.6), without considering memory and self-connection, according to the following equation. They found that this updating rule outperforms (2.2) when used for prediction tasks implemented on real-world data.

$$a_i(t+1) = f\left(\sum_{j=1, j\neq i}^{k} (2a_j(t) - 1)w_{ji}\right) \tag{2.6}$$

The other alternative was proposed in Papageorgiou (2011b) to avoid the conflicts emerging in the case of inactive concepts. From this perspective, it can be said that the element $(2a_i(t) - 1)$ acts as a bias weight. More clearly, it prevents saturation problems where there is no information about the initial concepts state (FELIX et al., 2019).

$$a_i(t+1) = f\left(\sum_{j=1, j\neq i}^{k} (2a_j(t) - 1)w_{ji} + (2a_i(t) - 1)\right) \tag{2.7}$$

Choosing the convenient updating rule to model the dynamic behavior of the system depends on the problem and a strong understanding of the target simulated system.

The function $f(\cdot)$ in updating rule equations specifies the transfer/activation/limiting function used to maintain the activation state of each concept to the predefined range. Table 12 highlights the most commonly used activation functions in the literature (FELIX et al., 2019). Both bivalent and trivalent are discrete and generate a finite number of states. The hyperbolic tangent and the sigmoid function are categorized in the continuous group and produce infinite states. Unlike generating finite states by using the first two discrete functions, hyperbolic tangent and sigmoid functions, as continuous functions, generate an infinite number of states used for simulating qualitative and quantitative cases. In fact, some studies verified the higher performance of FCM by applying continuous activation functions (TSADIRAS, 2008). A benchmarking study was organized in Bueno e Salmeron (2009) where the best performance is obtained using the sigmoid activation function in comparison to the other ones.

In the sigmoid activation function, γ is the steepness parameter (function slope). The higher the value of steepness, the more sensitive the function is to the changes in its input. Depending on the characteristics of the problem, different values of γ can be tuned. Generally, the values of γ are no greater than five in most of the references in the literature. For instance, Aguilar (2005) defined the value of γ equal to 5. Furthermore, a dynamical optimization of γ has been suggested by Salmeron e Froelich (2016) and Oikonomou e Papageorgiou (2013) to choose the optimal value according to the problem.

Table 1 – Most widely used FCM activation functions

Activation Function	Mathematical Representation
bivalent	$f(x) = \begin{cases} 0 & if \quad x < 0 \\ 1 & if \quad x \geq 0 \end{cases}$
trivalent	$f(x) = \begin{cases} 1 & if \quad x \geq 0.5 \\ 0 & if \quad -0.5 \leq x \leq 0.5 \\ -1 & if \quad x \leq -0.5 \end{cases}$
hyperbolic tangent	$f(x) = \frac{\exp^{2x} - 1}{\exp^{2x} + 1}$
sigmoid	$f(x) = \frac{1}{\exp^{-\gamma x} + 1}$

2.1.2 High Order Fuzzy Cognitive Map (HFCM)

Dynamically speaking, equations (2.2) to (2.6) express the first-order FCM. It means that through this way only the first-order dynamics of FCM will be obtained. For instance, by rewriting (2.5) in matrix form, it becomes clear to see that the activation value of each concept, at a particular time step $(t+1)$, only relies on the activation level of all concepts at t, as displayed via the equation (2.8).

$$a(t+1) = f\left(a(t) \cdot W\right) = f\left(W^t \cdot a^t(t)\right) \tag{2.8}$$

where W is the weight matrix as specified in (2.1).

For the purpose of tackling the aforementioned restriction and improving performance, especially for describing the dynamic behavior of the complex systems more accurately, high-order FCM was introduced in which the reasoning rule equation is modified and described as follows (LU et al., 2014; SHANCHAO; LIU, 2018; LIU; LIU, 2020):

$$a_i(t+1) = f_i\left(w_i^0 + \sum_{j=1}^{k} w_{ji}^1 a_j(t) + w_{ji}^2 a_j(t-1) + \ldots + w_{ji}^{\Omega} a_j(t-\Omega+1)\right) \tag{2.9}$$

where Ω represents the order, $w_{ij}^n \in [-1, 1]$ stands for the casual relation originating from c_i and pointing to c_j at time step $t-n+1$, while $w_i^0 \in [-1, 1]$ is the bias weight related to the i-th node.

As the above equation exhibits, the activation level of the i-th concept at $t+1$ depends on the activation values of all concepts at $\{t, t-1, \ldots, t-\Omega+1\}$ moments, not only the activation values of the concepts at time t, during the consecutive iterations.

f_i is defined as the associated activation function for each concept. For instance, if we consider the sigmoid activation function, the activation function for each concept can be defined based on the steepness parameters γ_i associated with the i-th concept as follows:

$$f_i(x) = \frac{1}{e^{-\gamma_i x} + 1} \tag{2.10}$$

In matrix form, the high-order FCM is replaced by the equation (2.11).

$$a(t+1) = f\left(w^0 + a(t) \cdot W^1 + \ldots + a(t-\Omega+1) \cdot W^{\Omega}\right) \tag{2.11}$$

where $f = (f_1(\cdot), f_2(\cdot), \ldots, f_k(\cdot))$ is the activation function vector where each member denotes the activation function related with the i-th concept. $W^1, W^2 \ldots, W^{\Omega}$ are the weight matrices storing all weights of the HFCM. $w^0 = (w_1^0, w_2^0, \ldots, w_k^0)$ is the bias vector of the HFCM. It is necessary to mention that, according to the above formula, increasing the order leads to increasing the number of parameters in the FCM model, regarding the size of the weight matrices.

2.1.3 Advanced Aspects

2.1.3.1 Extensions of FCM

According to the literature, extensions of FCM have been designed to enhance the performance of the traditional FCM proposed by Kosko. Fuzzy Grey Cognitive Maps (FGCM) (SALMERON, 2010), Intuitionistic Fuzzy Cognitive Maps (iFCM) (IAKOVIDIS; PAPAGEORGIOU, 2011; PAPAGEORGIOU; SALMERON, 2012), Belief-Degree-Distributed Fuzzy Cognitive Maps (BDD-FCM) (RUAN; HARDEMAN; MKRTCHYAN, 2011), Rough Cognitive Maps (RCM) (CHUNYING et al., 2011), Dynamical Cognitive Networks (DCN) (MIAO et al., 2001b), Evolutionary Fuzzy Cognitive Maps (E-FCM) (CAI et al., 2009), Fuzzy Time Cognitive Maps (FTCM) (PARK; KIM, 1995), Dynamic Random Fuzzy Cognitive Maps (DRFCM) (AGUILAR, 2003), Rule-Based Fuzzy Cognitive Maps (RB-FCM) (CARVALHO; TOMé, 2001), Fuzzy Rules Incorporated in Fuzzy Cognitive Maps (FRI-FCM) (SONG et al., 2011) and Generalized FCM (GFCM) (NAIR; RECKIEN; MAARSEVEEN, 2019) are some of these extensions of the traditional FCM.

A proposal of various FCM models was raised to cover FCM limitations including dealing with uncertainty or dynamic problems. For instance, FGCM, iFCM, BDD-FCM, and RCM have been presented to deal with uncertainty problems. On the other side, DCN, E-FCM, FTCM, and DRFCM have been developed to deal with dynamic problems. Nair, Reckien e Maarseveen (2019) proposed a new form of FCM called GFCM to deal with both uncertainty and dynamic problems. Also, it provides a literature review of the conventional FCM approaches and figures out the strengths and weaknesses of some important advances in FCM to understand their potential in modeling complex qualitative systems. Since most of these models are employed in decision-making domains, they are not elaborated on in detail in this review paper.

2.1.3.2 Dynamic Properties of FCM

As discussed, FCM produces a new state vector at each discrete time step in an iterated process until the stop condition is met such as a maximum number of iterations or the system stabilizes. More precisely, the relational structure and inference mechanism of FCM play critical roles in its dynamic behavior to determine the state value at time $t+1$ by considering the previous state at time t. Based on this perspective, three possible scenarios are available to present the dynamic behavior of FCMs (GREGOR; GROUMPOS, 2013b; FELIX et al., 2019). If the activation degree of the concepts remains stable after a finite number of time steps, then a fixed point is obtained with the convergence ability. Mathematically speaking, $\exists t_\alpha \in \{1, 2, \ldots, T-1\} : a(t+1) = a(t), \forall t \geq t_\alpha$. In the limit cycle scenario, FCM generates dissimilar responses with the exception of a few states that are periodically produced. Mathematically, $\exists t_\alpha, P \in \{1, 2, \ldots, T-1\} : a(t+P) = a(t), \forall t \geq t_\alpha$ Finally, in the chaos scenario, a different state vector is generated at each iteration. Thus, the system is neither fixed nor cyclic.

Although updating the reasoning rule stops once a maximum number of iterations is reached, the output can be unreliable due to the lack of stability in chaotic or cyclic situations. It means that in order to better understand the system's behavior, stable responses are needed. Hence, convergence is often preferable in some scenarios including decision-making and pattern classification. According to Felix et al. (2019), some factors including the method for updating activation values, the pattern encoded in the weight matrix, and non-decreasing activation functions are the most important issues regarding the convergence of FCM-based systems. Accordingly, there exist some studies around this subject such as (BOUTALIS; KOTTAS; CHRISTODOULOU, 2009; KOTTAS; BOUTALIS; CHRISTODOULOU, 2007; KOTTAS; BOUTALIS; CHRISTODOULOU, 2010; KYRIAKARAKOS et al., 2012; NAPOLES; BELLO; VANHOOF, 2014).

2.1.3.3 FCM against Artificial Neural Network (ANN)

Gregor e Groumpos (2013b) proposed the relation between FCM and ANN. Based on the close resemblance between FCM and ANN, it can be said that an FCM can be expressed as a single-layer recurrent neural network with sigmoid squashing functions and synchronous activation of units. Besides, FCM can be regarded as an extension of Hopfield RNN with a different structure to which $w_{ij} = w_{ji}$ is needed to converge to a fixed point. Also, the structure of the Hopfield network has been constructed for binary or bipolar activations using the sign activation function, and its units are activated asynchronously. Based on these differences, the modified ANN learning methods are used as FCM learning strategies and it is impossible to use them directly.

Generally speaking, however, there are similarities between FCM and ANN, but

there are some limitations and differences. From a structural point of view, an FCM is taken into account as a special type of RNN. In other words, FCM can be viewed as interpretable RNNs based on their interpretability as the significant feature. From this perspective, FCM consists of fuzzy logic elements during the knowledge engineering phase. In fact, the concepts of FCM can be supposed as neural processing entities indicating that the activation degree of each neuron is defined through the value of the transformed weighted sum unit received from the connected neurons in the network.

Apart from the topological similarity between FCM and RNN, they function differently. It is related to the knowledge interpretability of FCM as its super property. This feature enables FCM to be used successfully in modeling complex real-world problems. This means that, unlike traditional ANNs/RNNs, FCM neurons and the causal relations among them provide an accurate interpretation of the system for humans. However, in ANNs/RNNs, the neurons are considered as a computational unit without reflecting any clear meaning. In other words, an ANN/RNN performs its duty in a perfect manner but with limited interpretability because the presence of hidden neurons is neither interpretable nor explains why/how the solution is appropriate for the desired problem. On the other hand, in an FCM, each node has its meaning with a clear relation to other nodes. Therefore, as discussed, it is often said that there is a trade-off between the interpretability of FCM and the approximation ability of ANN. A few studies interpret FCM-based methods as ANNs (TSADIRAS; MARGARITIS, 1999; TSADIRAS, 2008; PAPAKOSTAS et al., 2012; NAPOLES et al., 2016; NAPOLES et al., 2017).

To encapsulate, like other fuzzy logic systems, learning has a basic role in designing FCM in practical applications. Although a variety of methods are introduced in the literature to adapt FCM, the close relation between FCM and ANN theories can provide a promising FCM learning strategy. Hence, the next section examines some important learning methods proposed in the literature for FCM.

2.1.4 Application Domains of FCM

Studies show remarkable advances in FCM based on some specific characteristics such as simplicity, qualitative modeling, modeling complex dynamic systems, flexibility, interpretability, and high ability to deal with uncertainties. Due to the mentioned features, FCMs are applied in various areas. Social and politics in behavioral sciences (ANDREOU; MATEOU; ZOMBANAKIS, 2005; ANDREOU; MATEOU; ZOMBANAKIS, 2003), business (JETTER, 2006; WEI; LU; YANCHUN, 2008; XIROGIANNIS; GLYKAS, 2004), software engineering (VARGAS; SALMERON, 2012), control engineering (STYLIOS; GROUMPOS, 2004; LU; YANG; LI, 2010), medicine (AMIRKHANI et al., 2017; PAPAKOSTAS et al., 2011; SALMERON; PAPAGEORGIOU, 2012), environmental and agriculture (TAN; ÖZESMI, 2005; RAJARAM; DAS, 2010; KOK, 2009), education

(LAUREANO-CRUCES; RAMIREZ; TERAN, 2004) and telecommunications (LI et al., 2009) are some examples of FCM applications to solve classification (FELIX et al., 2019; SZWED, 2021; RAMIREZ-BAUTISTA et al., 2020; YU et al., 2022) prediction (FELIX et al., 2019), modeling (BAGDATLI; DOKUZ, 2021; GROUMPOS, 2015) and decision making (LUO; WEI; ZHANG, 2009; BEENA; GANGULI, 2011; GLYKAS, 2010; PAPAGEORGIOU; GROUMPOS, 2005; PAPAGEORGIOU; STYLIOS; GROUMPOS, 2006; KETIPI et al., 2020) problems. Since the main goal of this study is focused on time series forecasting using FCM, the rest of this chapter covers a review and summary of the presented methods in this field until mid-2022.

2.2 Time Series Forecasting using Fuzzy Cognitive Map

2.2.1 Terminology of the problem

Time series forecasting includes predicting future observations based on the trained model on historical observations. Thus, defining an accurate prediction model plays an essential role in making better decisions in several fields for instance engineering, medicine, economy, meteorology, and so on. A univariate time series is represented by a collection of values (samples, observations) $y(t) \in Y \subset \mathbb{R}$ at time $t = 1, \ldots, N$, where N is the sample size. A multivariate time series is a collection of historical observations of n variables under the same time index $t = 1, \ldots, N$. The goal of time series forecasting models is to predict the next values of time series given the past values of the series and considering the prediction horizon ($H \in \mathbb{Z}_+$), which can be divided into four categories including very short, short, medium and long term forecasting. Also, models for time series forecasting can be classified as univariate and multivariate models. In the case of univariate models, only information up to time t from a single variable is used to form predictions along the time. More formally, we have

$$\hat{y}(t+H) = M_u(y(t), y(t-1), y(t-2), \ldots, y(t-L)) \tag{2.12}$$

where $\hat{y}$ is the forecast (estimated, predicted) value and $M_u : \mathbb{R}^{L+1} \mapsto \mathbb{R}$ is the univariate forecasting model, which is used to calculate the next values of time series with regards to the H value. More generally, M_u can be regarded as a nonlinear auto-regressive mapping.

In the case of multivariate models, information up to time t from multiple variables is used to forecast the value of either a target variable or all the variables at $t + H$.

Mathematically, we have

$$\begin{pmatrix} \hat{y}_1(t+H) \\ \hat{y}_2(t+H) \\ \vdots \\ \hat{y}_n(t+H) \end{pmatrix} = M_m \begin{pmatrix} y_1(t) & \dots & y_1(t-L) \\ y_2(t) & \dots & y_2(t-L) \\ \vdots & & \vdots \\ y_n(t) & \dots & y_n(t-L) \end{pmatrix} \tag{2.13}$$

where $M_m : \mathbb{R}^{n\times(L+1)} \mapsto \mathbb{R}^n$ is the multivariate forecasting model.

Multivariate models can be grouped into Multiple Input Single Output (MISO) and Multiple Input Multiple Output (MIMO) models, depending on whether the model returns the forecast for a single target variable or for all the variables. For instance, (SILVA, 2019) provides both MISO-FTS and MIMO-FTS forecasting models. In MISO approaches, one of the variables is chosen as the endogenous (or target) variable while the others are considered as explanatory or exogenous variables. In contrast, in the MIMO approach, all variables have a chance to be forecast, that is, all variables become target and explanatory variables.

However, there is no perfect model to predict exact future values due to the existence of uncertainty and non-linearity associated with most real-world phenomena. Accordingly, numerous time series forecasting methods have been presented in the literature to perform forecasting operations and estimate forecasted value $\hat{y}(t+H)$, from statistical methods, such as Auto Regressive Moving Average ARMA), Auto Regressive Integrated Moving Average (ARIMA), Seasonal Auto-Regressive Integrated Moving Average (SARIMA), to some new intelligent techniques, such as Long Short-Term Memory (LSTM), Temporal Convolutional Neural Network (TCNN), Gated Recurrent Unit (GRU). Thereby, constructing the proper model is vital to minimize the prediction error between forecasted and real values.

Song e Chissom (1993) proposed the concept of Fuzzy Time Series (FTS) in 1993 to handle vague and imprecise knowledge in time series data which has been widely used during recent decades. Although plenty of FTS forecasting methods were introduced in the literature, they follow the following steps (BOSE; MALI, 2019; SINGH, 2017). Firstly, the Universe of Discourse (UoD) is defined for the given time series. Then, UoD is partitioned into some overlapping intervals using fuzzy sets. Then the membership degree of each value $y(t) \in Y$ to each fuzzy set $C_i \in C$ is calculated using membership function $\mu_{C_i} : \mathbb{R} \to [0, 1]$. The most common membership functions are triangular, trapezoidal, sigmoidal, and Gaussian functions. The fuzzification process is employed to convert the original crisp time series $y(t)$ into a fuzzy time series $f(t)$, which is a sequence of membership values of the sample $y(t)$ at time t to the fuzzy sets $C_i \in C$. In the context of FCM, the first key step in designing an FCM model for time series is the definition of concepts, see Section 2.2.2. The fuzzification of the time series into concepts leads to an FTS considering these

concepts. In other words, the state vector $a(t)$ with the activation degree of all concepts can be viewed as an FTS representation of the original time series.

Studies show remarkable advances in time series forecasting using FTS models based on some basic characteristics such as simplicity, readability, scalability, and high forecasting accuracy (LEE; JAVEDANI et al., 2011; SADAEI et al., 2017; SILVA et al., 2020a). A comprehensive review of FTS forecasting models was proposed by Bose e Mali (2019), covering some recent univariate and multivariate techniques focusing on different aspects and forecasting stages. It is worth observing that some extensions of FTS were introduced in the literature to embrace other extensions of time series forecasting. For instance, Alves et al. (2018) proposed an approach to handle non-stationary time series. Silva, Sadaei e Guimarães (2016) introduced an interval forecasting technique. The authors in Silva et al. (2020b) and Silva et al. (2017) developed probabilistic FTS forecasting models. In addition, some researchers introduced weighted rule-based FTS models to improve the accuracy of forecasting (YU, 2005; SILVA et al., 2020b). Also, FCMs as weighted knowledge-based FTS techniques have found excellent applicability in the field of time series forecasting as a significant part of FCM's research in many areas. Therefore, the following subsections provide the general structure of FCM-based forecasting models as well as a review of some proposed FCM forecasting models in the literature.

2.2.2 Design Issues

FCMs have been introduced as convenient models to represent time series in an easy and meaningful way. Knowledge-based representation and realizing inference processes enable FCM with the ability to capture the dynamic behavior of a given system. The main objective of this section is how to design a forecasting model based on FCM to achieve modeling and forecasting time series. Fundamentally, there are three substantial problems that must be taken into account when an FCM is applied to model time series. Therefore, designing proper FCM structure under a given time series, calculating the FCM parameters (weight matrices), and reconstructing numerical values based on the FCM are essential considerations as they are applied for time series modeling, time series dynamic behavior explanation and forecasting (HOMENDA; JASTRZEBSKA, 2016; LU et al., 2014).

One of the key stages in designing FCM is concept definition and extraction. The aim of FCM-based models is to select abstract concepts that represent the information in an aggregated fashion and cover the dataset. For instance, granular knowledge representation schemes and fuzzy sets are used commonly to represent concepts mathematically.

The number of concepts defined by the users have meaningful relations with the specificity and generality of the model. In this perspective, there is a trade-off between the

specificity and generality of the model. It means that the more the number of concepts, the more specific the model. Conversely, the model must be designed with a small number of concepts to reach more generalization. Therefore, there must be a balance between specificity and generality. In more particular, although increasing the number of concepts can improve the accuracy of problem modeling in the real world, the interpretability of the model is degraded (HOMENDA; JASTRZEBSKA, 2016). Although some papers (PAPAGEORGIOU; POCZETA, 2016; PAPAGEORGIOU; POCZETA; LASPIDOU, 2016) focused on reducing the complexity of FCM by selecting the most important nodes and their connections, extracting the optimal number of concepts remains a challenging question, without a precise solution in the literature (HOMENDA; JASTRZEBSKA, 2016).

Discovering the appropriate structure of FCMs has been mainly done using common successful strategies comprising granularity (STACH; KURGAN; PEDRYCZ, 2008b; FROELICH; PEDRYCZ, 2016), membership values representation (SONG et al., 2010), fuzzy c-means clustering (LU et al., 2014; LU; YANG; LIU, 2013; HOMENDA; JASTRZEBSKA; PEDRYCZ, 2014a; HOMENDA; JASTRZEBSKA; PEDRYCZ, 2014c; LU; YANG; LIU, 2014; LU et al., 2014) to extract concepts and construct the architecture of FCM in time series forecasting domains. Furthermore, cluster validity indexes (HOMENDA; JASTRZEBSKA, 2016), moving window (HOMENDA; JASTRZEBSKA; PEDRYCZ, 2014b; HOMENDA; JASTRZEBSKA; PEDRYCZ, 2014a), wavelet transformation and empirical mode decomposition (EMD) are also proposed to identify FCM's structure and enhance the forecasting performance, see for instance Liu e Liu (2020), Shanchao e Liu (2018), Gao, Du e Yuen (2020).

After mapping FCM nodes to time series variables or intervals of these variables, learning FCM is considered the second phase. This involves searching for the optimal or near-optimal weight matrices to describe the dynamic behavior of the system by testing within the prediction horizon. Without high-quality weight matrices, the system will lose its interpretability, even with crystal clear FCM reasoning (VANHOENSHOVEN et al., 2018).

In conclusion, it can be said that input fuzzification, FCM learning, modeling, and output defuzzification can be considered significant stages of designing FCM-based forecasting models (FENG et al., 2021; STACH; KURGAN; PEDRYCZ, 2008b; SHANCHAO; LIU, 2018). Therefore, Table 2 summarizes a brief review of different fuzzification and defuzzification methods adopted in this area. In addition, it provides some information relating to the main focus of each paper as well as the hyper-parameters affecting the performance of the associated approaches. Regarding the column for hyper-parameters, blank cell means that the authors of that paper did not evaluate the effect of hyper-parameters in their study.

Table 2 – Review of fuzzification and defuzzification methods used in FCM-based forecasting techniques

Ref.	Fuzzification	Defuzzification	Main focus	HPs
Stach, Kurgan e Pedrycz (2008b)	Triangular MF	Center of area method	FCM structure	Number of linguistic labels, window size
Wojciech e Juszczuk (2009)	Min-Max normalization	Without	FCM Learning	
(SONG et al., 2010)	Gaussian MF via linguistic layer	standard volume based centroid via mapping layer	learning mechanism + causality estimation	–
Papageorgiou e Froelich (2012)	Min-Max normalization	Not detailed	Interpretability + multi-step ahead prediction + learning FCM	–
Froelich e Papageorgiou (2014)	Equation (2.14)	Equation (2.15)	FCM learning + fuzzification and activation function	Steepness parameter, fuzzification coefficients
Froelich et al. (2012)	Min-Max Normalization	Not detailed	multi-step ahead prediction + learning FCM	–
Lu, Yang e Liu (2013)	Fuzzy C-means clustering	Not needed (linguistic prediction)	FCM learning + structure	–
Homenda, Jastrzebska e Pedrycz (2014a)	Fuzzy C-means clustering	without	FCM structure	Window size, NC
Homenda, Jastrzebska e Pedrycz (2014c)	Fuzzy C-means clustering	Not detailed	FCM structure + complexity reduction	NC, weights
Lu, Yang e Liu (2014)	Fuzzy C-means clustering	Equation (2.16)	FCM structure + large-scale time series prediction	–
Lu et al. (2014)	Fuzzy C-means clustering	Via output HFCM: (1) By adding a new node to the FLFCM (2) by adding edges from FLFCM to the newly added node	FCM structure + large-scale time series prediction	NC, order
Froelich e Salmeron (2014)	Equation (2.17)	Equation (2.18)	FCM structure + Learning FCM + forecasting IVTS	Prediction horizon
Homenda, Jastrzebska e Pedrycz (2014b)	Without	Without	FCM structure	Map size, bias
Poczketa e Yastrebov (2015)	Not detailed	Not detailed	FCM Learning	Learning parameters

Continued on next page

Table 2 – continued from previous page

Author	Fuzzification	Defuzzification	Main focus	Parameters
Papageorgiou, Poczeta e Laspidou (2015)	Normalization	Without	FCM Learning	Learning parameters
Papageorgiou e Poczeta (2015)	Not detailed	Not detailed	FCM Learning	Learning parameters
Papageorgiou e Poczeta (2016)	Normalization	Without	FCM structure + complexity reduction	Learning parameters
Papageorgiou, Poczeta e Laspidou (2016)	Normalization	Without	FCM structure + complexity reduction	Learning parameters
Homenda e Jastrzebska (2016)	Fuzzy C-means clustering	Without	FCM structure (node selection)	NC
Salmeron e Froelich (2016)	Min-Max normalization	Denormalization	FCM structure	Learning parameters, experimental parameters
Vanhoenshoven et al. (2018)	Normalization	Not detailed	Predicting oscillating time series + multiple steps ahead prediction + Maintaining both predictive accuracy and transparency of the weight matrix	
Poczeta e Papageorgiou (2018)	Normalization	Without	FCM structure+ Increase flexibility + improve simplicity + reduce training time	–
Shan, Lu e Yang (2018)	Improved fuzzy C-means clustering	Equation (2.16)	FCM structure	NC, steepness index, fuzzification coefficients
Poczeta, Papageorgiou e Yastrebov (2018)	Min-Max normalization	Without	FCM learning + complexity reduction + multi-step ahead prediction	Learning parameters
Shanchao e Liu (2018)	Min-Max normalization	Not detailed	FCM structure + time efficient learning + handling large-scale non-stationary	NC, order, regularization factor
Papageorgiou et al. (2019)	Min-Max normalization	Without	FCM structure+ FCM learning	–

Continued on next page

Table 2 – continued from previous page

Author	Fuzzification	Defuzzification	Main focus	Parameters
Wu et al. (2019)	Min-Max normalization	Not detailed	FCM structure + FCM learning	NC, order, sparsity, SPTc, WDIC
Wang et al. (2019)	Fuzzy C-Means	Equation (2.19)	FCM structure + concept-drift handling	–
Hajek, Froelich e Prochazka (2020)	Equations (2.20)-(2.23)	Equations (2.25)-(2.26)	FCM structure + dealing with high uncertainty time series	–
Liu e Liu (2020)	Normalizing	Renormalizing	handling non-stationary and large-scale time series + FCM learning + FCM structure	Order, learning parameters
Hajek, Prochazka e Froelich (2018)	IVI-fuzzy sets + triangular MF	Equations (2.25)-(2.26)	FCM structure	–
Gao, Du e Yuen (2020)	Min-Max normalization	Not detailed	FCM structure + FCM learning	NC, order
Yuan et al. (2020)	Kernel mapping to generate KFTS instead of FTS	Reverse kernel mapping	FCM structure + FCM learning	NC, order, regularization factor
Vanhoenshoven et al. (2020)	Not detailed	Not detailed	Learning FCM	–
Wang et al. (2020)	Sigmoid MF	Equation (2.27)	FCM structure	–
Orang et al. (2020)	Grid partitioning + triangular MF	Equation (2.28)	FCM structure	Bias, activation function, NC
Poczeta, Papageorgiou e Gerogiannis (2020)	Min-Max normalization	without	FCM structure + Complexity reduction	–
Feng et al. (2021)	Modified fuzzy C-means clustering	Equation (2.29)	FCM structure + FCM learning + multi-step prediction	prediction horizon
Wang et al. (2021)	Fuzzy C-means algorithm	maximum membership rule	FCM structure + FCM learning + prediction large-scale data + flexibility improvement	Number of clusters, cooperation coefficient

Continued on next page

Table 2 – continued from previous page

Author	Fuzzification	Defuzzification	Main focus	Parameters
Feng, Lu e Yang (2021a)	Fuzzy C-means clustering	Equation (2.30)	FCM Learning + FCM structure	NC, steepness (shape) parameter in sigmoid function
Feng, Lu e Yang (2021b)	Fuzzy C-means + triangular MF	Equation (2.31)	FCM structure	NC, number of partitions
Ding e Luo (2022)	Fan-shape fuzzy information granule + Fermi function as MF	Not detailed	FCM structure + learning large-scale FCM + interpretability + generalization improvement	NC, order, sliding window size, sliding step size
Xixi, Ding e Luo (2022)	IFM + IFS	Not detailed	FCM structure + enhance uncertainty representation + dealing with non-stationary data	NC, order
Wang et al. (2022)	Gaussian LFIG + fuzzy C-means algorithm	Equation (2.34)	FCM structure + interpretability + trend capturing + flexibility improvement + error reduction	Cluster number, l_1-trend filtering, min length of granules, min slope difference of evaluating key points, prediction horizon

MF → membership function; **NC → Number of concepts**
SPTc → sparsity penalty term controller; **FLFCM → Fully learned FCM**
WDIC → weight decay importance controller; **KFTS → key features time series**
IVTS → Interval-valued time series; **IVI → Interval-valued intuitionistic**
IMF → Intrinsic mode functions; **IFS → Intuitionistic fuzzy sets**
LFIG → Linear fuzzy information granules

A modified normalization function was proposed by Froelich e Papageorgiou (2014) as a fuzzification function described in the equation (2.14).

$$a_i(t) = \left(\frac{v_i(t) - mean(v_i) + \lambda_i.stdDEV(v_i)}{2.\lambda_i.stdDEV(v_i)} \right) \tag{2.14}$$

where $a_i(t) = \mu(v_i(t))$ calculates the activation state of concept a_i at time t or the membership degree in which v_i belongs to fuzzy set c_i. λ_i depends on the distribution of source data. Through this way, the directional coefficient of the fuzzification function is not affected by the outliers that are far away from the mean of the variable. Note that the

defuzzified value can be realized via Equation (2.15).

$$\hat{x}(t) = \left(\mu^{-1}(a(t)\right) \tag{2.15}$$

in Lu, Yang e Liu (2014) the defuzzification is performed based on the equation (2.16)

$$\hat{x}(t) = \left(\frac{\sum_{j=1}^{k} a_j(t)^m(p_j)}{\sum_{j=1}^{k} a_j(t)^m}\right) \tag{2.16}$$

where k is the number of clusters, m is the fuzzification coefficient, p_j is prototype, $a_j(t)$ is the activation value of jth node of FCM model at time t, and $\hat{x}(t)$ is the predicted numerical value at t.

Equations (2.17) and (2.18) stand for the proposed fuzzification and defuzzification approaches in (FROELICH; SALMERON, 2014), respectively.

$$a_i(t) = \left(\frac{v_i(t) - \min(v_i(t))}{\max(v_i(t)) - \min(v_i(t))}\right) \tag{2.17}$$

where the value of $a_i(t) = \mu(v_i(t))$ denotes the degree to which real valued variable $(v_i(t))$ belongs to the fuzzy set c_i at time t.

$$\mu^{-1}(a_i(t)) = v_i(t) = a_i(t)(\max(v_i(t)) - \min(v_i(t))) + \min(v_i(t)) \tag{2.18}$$

Wang et al. (2019) exploited the equation (2.19) to compute the final output.

$$\hat{x}_i(t) = \left(\sum_{j=1}^{k} w_i(j)\hat{c}_i(j) \times (M - m) + m\right) \tag{2.19}$$

where k is the number of clusters, w_i is the membership of premise to the cluster k, m and M represent the minimum and maximum values of the original time series.

The universe of discourse is divided into n equal intervals such that for each interval, an initial triangular membership function is constructed (HAJEK; FROELICH; PROCHAZKA, 2020). Thus, the fuzzy set $a_i = [\mu^L(\otimes(a_i), \mu^U(\otimes(a_i)]$ is obtained, where $\mu^L(\otimes(a_i)$ and $\mu^U(\otimes(a_i)$ denote the membership degree for the lower bound and upper band, respectively and $\nu^L(\otimes(a_i)$ and $\mu^U(\otimes(a_i)$ denote the non-membership degree for the lower bound and upper band of interval-valued time series (ITS). The fuzzification of the lower and upper bounds of the ITS has been given using the equations (2.20) to (2.23).

$$\mu^L(\otimes(a_i)(t) = \left(\mu^L(\otimes(a_i)(t) \times (1 - \delta D)\right) \tag{2.20}$$

$$\mu^U(\otimes(a_i)(t) = \left(\mu^U(\otimes(a_i)(t) \times (1 - \delta D)\right) \tag{2.21}$$

$$\nu^L(\otimes(a_i)(t) = \left(1 - \mu^U(\otimes(a_i)(t) \times (1 - \delta D) - \delta D\right) \tag{2.22}$$

$$\nu^U(\otimes(a_i)(t) = \big(1 - \mu^L(\otimes(a_i)(t) \times (1 - \delta D) - \delta D\big) \tag{2.23}$$

where D=(max($\mu^U(\otimes(a_i)(t)$,$\mu^U(\otimes(a_i)(t-1)$,...,$\mu^U(\otimes(a_i)(t-4))$−min(($\mu^L(\otimes(a_i)(t)$, $\mu^L(\otimes(a_i)(t-1)$,...,$\mu^L(\otimes(a_i)(t-4)))$ and $\delta = 1$. Thus, interval-valued intuitionistic fuzzy sets are attained as

$$\otimes(a_i)(t) = [\mu^L(\otimes(a_i)(t), \mu^U(\otimes(a_i)(t)], [\nu^L(\otimes(a_i)(t), \nu^U(\otimes(a_i)(t)] \tag{2.24}$$

To calculate the defuzzified values, equations (2.25) and (2.26) are employed for the upper and lower bounds of ITS:

$$\hat{x}^L(t+1) = \left(\frac{\sum_{i=1}^{n}(\mu^L(\otimes\hat{a}_i(t+1)) - \nu^L(\otimes\hat{a}_i(t+1))).p_i}{\sum_{i=1}^{n}(\mu^L(\otimes\hat{a}_i(t+1)) - \nu^L(\otimes\hat{a}_i(t+1)))}\right) \tag{2.25}$$

$$\hat{x}^U(t+1) = \left(\frac{\sum_{i=1}^{n}(\mu^U(\otimes\hat{a}_i(t+1)) - \nu^U(\otimes\hat{a}_i(t+1))).p_i}{\sum_{i=1}^{n}(\mu^U(\otimes\hat{a}_i(t+1)) - \nu^U(\otimes\hat{a}_i(t+1)))}\right) \tag{2.26}$$

where p_i denotes the modal values of the triangular membership functions $\mu^L(\otimes(a_i(t))$ and $\mu^U(\otimes(a_i(t))$. Note that only those concepts for which the membership degrees are greater than the non-membership degrees were considered that is $\mu^L(\otimes\hat{a}_i) - \nu^L(\otimes\hat{a}_i) = 0$ and $\mu^U(\otimes\hat{a}_i) - \nu^U(\otimes\hat{a}_i) = 0$ if $\nu^L(\otimes\hat{a}_i) > \nu^L(\otimes\hat{a}_i)$ and $\nu^U(\otimes\hat{a}_i) > \nu^U(\otimes\hat{a}_i)$.

Wang et al. (2020) utilized the equation (2.27) to defuzzify the fuzzy activation state of each concept.

$$\hat{x_j}(t+1) = \big(f^{-1}(a_j(t)).\sigma_j + \mu_j\big) \tag{2.27}$$

where σ_j and μ_j are the mean and standard deviation of x_j.

The final defuzzified value in Orang et al. (2020) is obtained using the equation (2.28).

$$\hat{x}(t+1) = \left(\frac{\sum_{j=1}^{k} a_j(t+1).mp_j}{\sum_{j=1}^{k} a_j(t+1)}\right) \tag{2.28}$$

where k is the number of concepts, $a_j(t+1)$ is the activation state of each concept at time step $t+1$ and mp_j is the center of each concept c_j.

In Feng et al. (2021), Equation (2.29) has been utilized as defuzzification technique:

$$\hat{x}(t+1) = \left(\frac{\sum_{j=1}^{k} \hat{a}_j^m(t+1).(V_j)}{\sum_{j=1}^{k} \hat{a}_j^m(t+1)}\right) \tag{2.29}$$

where $a(t) = (a_1(t), a_2(t), ..., a_k(t))$ is the state vector at time t and $\hat{a}(t) = (\hat{a}_1(t), \hat{a}_2(t), ..., \hat{a}_k(t))$ is the next predicted state vector which is calculated by $\hat{a}(t+1) = f(a(t)W)$, m is the fuzzification coefficient, V_j denotes j-th cluster center and finally $\hat{s}(t+1)$ presents the prediction segment.

The numerical predicted values in Feng, Lu e Yang (2021a) are reconstructed using the following equation (2.30).

$$\hat{x}(t+1) = \left(\frac{\sum_{j=1}^{k} \hat{a}_j^m(t+1).(v_j)}{\sum_{j=1}^{k} \hat{a}_j^m(t+1)} \right) \tag{2.30}$$

where

m is the fuzzification coefficient, and v_j is the FCM concept calculated by the fuzzy C-means clustering algorithm.

The final predicted value in Feng, Lu e Yang (2021b) is given by aggregating the produced results by an ensemble of sub-FCM models based on the equation (2.31).

$$\hat{y}(t+1) = \left(\frac{\sum_{j=1}^{P} {w_j}^t.y_j}{\sum_{j=1}^{P} {w_j}^t} \right) \tag{2.31}$$

where y_j indicates the obtained output from each sub-model at time t and $w_j^t \in [0,1]$ is the corresponding weight to the outputs at time t.

In Ding e Luo (2022), no details on defuzzification were provided however the outputs of the attention module are calculated according to the equations (2.32) and (2.33).

$$c = \alpha \odot U \tag{2.32}$$

$$Z = \sigma(W_c[c; U]) \tag{2.33}$$

where $\odot$ represents the element-wise product, α represents the vector of the attention coefficients, c denotes the vector of the attention values, W_c denotes the learnable parameters of the fully connected layer, and Z denotes the outputs of the attention module.

In Wang et al. (2022) the equation (2.34) is exploited to output the final prediction.

$$\hat{x}(t) = \left(\hat{k}'_{N'+1} \times (t-N) + \hat{b}'_{N'+1} \right) \tag{2.34}$$

where $\hat{k}'_{N'+1}$ and $\hat{b}'_{N'+1}$ are obtained using the inverse function of normalization, N' is the number of segments, N is the number of data and $t \in [N+1, N+\hat{T}_{N'+1}]$.

As recorded in Table 2, various fuzzification and defuzzification techniques have been developed. For univariate time series, concepts usually represent fuzzy partitions of the range of that variable; these partitions can be defined by grid partition or found with fuzzy C-means clustering. For multivariate time series, concepts can represent clusters in n-dimensional space, fuzzy granules, or in some studies, each variable becomes a concept and the membership is found by normalization of the variable to $[0,1]$. Clearly, the fuzzy C-means technique has been widely used as a common fuzzification strategy in FCM-based prediction models. According to the literature (AL-GUNAID et al., 2017) and as can be seen in Table 2, normalization can be considered as a usual fuzzification

procedure in multivariate FCM. It means that, instead of the complex construction of fuzzy concepts, we simply normalize the value sets of the attributes to a specific range (WOJCIECH; JUSZCZUK, 2009). However, in some references, no details of fuzzification and defuzzification have been provided. The following section gives more information about the proposed methods.

2.2.3 Overview of FCM-based models in the literature

With the support of the Web of Science (WoS) database, the search for potential papers was carried out by cross-searching a comprehensive set of keywords (time series AND fuzzy AND fuzzy cognitive map AND (forecast* OR predict* OR analysis OR modeling)) appearing in the title or keywords of an article. In addition, the abstracts of all identified papers were checked to verify that they were directly related to the topic addressed here.

The main purpose of this section is to provide a review study of some of the relevant FCM-based time series forecasting methods in the literature which are listed by year of publication. According to Table 3, after filtering for relevant papers, 49 FCM-based time series forecasting articles retrieved from Elsevier journals, IEEE Xplore (conferences and journals), Springer (conferences and journals), and MDPI online journals for the time period 2008 to 2022 have been reviewed in our investigation. We could not find any paper using FCM in time series forecasting prior to 2008.

An extension of the FCM-based time series forecasting method has been designed in Stach, Kurgan e Pedrycz (2008b). This two-level technique enables the model to carry out modeling and forecasting in both numerical and linguistic terms as the crux advantage by combining FCM and granules. The outstanding characteristic of this candidate FCM corresponded to the benefits of the proposed RCGA learning technique. It must be considered that the number of granular time series from the output of the fuzzification module represents the number of nodes in FCM-based modeling. Further, equal width of fuzzy sets and equal data frequency are used in this paper. The results indicate linguistic accuracy of the proposed method decreases as the number of fuzzy sets becomes higher, while at the same time, the numerical prediction accuracy increases. This shows that some trade-off exists between the quality of the numerical and linguistic predictions. Also, the tests show that the statistical characteristics of the input time series influence the quality of the results. A higher standard deviation of the input time series results in a slightly worse accuracy of prediction.

The authors in Wojciech e Juszczuk (2009) published a comparative analysis of the evolutionary and adaptive learning method of FCM to evaluate the proper method for a special prediction problem. In other words, the performance of the proposed method

was evaluated by RCGA and DE as the evolutionary methods in comparison to DHL and balanced differential learning algorithm (BDA) as adaptive methods. The results demonstrate that the predictive capabilities of adaptive methods are not competitive with evolutionary algorithms. The predictive capabilities of the proposed method were examined by forecasting weather conditions in this study.

An unprecedented model for predicting chaotic time series was designed in Song et al. (2010) and Song et al. (2010) by implementing FCM based on a novel fuzzy neural network. On contrary to the principal drawback of the conventional FCM models, where determining the states of the investigated system and quantifying causalities mainly depends on the expert's knowledge, the proposed model equips the inference mechanism of original FCM with the automatic identification of membership functions and quantification of causalities. Thereby, the construction of FCM can be modeled automatically from the data independently with less intervention from experts. In this manner, the proposed fuzzy neural network is composed of four layers. The first layer consists of input variables and each node represents a concept in the concerned system. The second layer performs fuzzification while each node represents the linguistic-term set of inputs. In the third layer, the nodes represent the output variables' linguistic terms, and two tasks comprising the causalities among concepts in FCM and defuzzification are carried out. Finally, in the last layer, the nodes represent non-fuzzy variables. The application of the proposed fuzzy neural network provides a crystal mathematical representation of the causalities and makes the inference process more understandable. Note that the Backpropagation-based learning algorithm is employed to adjust the relevant parameters by minimizing the RMSE as the objective function. Simple architecture and better forecasting accuracy confirmed the superior performance of the proposed method.

The aim of the proposed research in Froelich et al. (2012) is to create the FCM model for long-term prediction of prostate cancer based on an improved learning evolutionary learning method enabling FCM to better optimize the fitness function for long-term prediction of multivariate time series. In other words, in this investigation, two types of FCM are designed for both short and long-term prediction by adding a horizon parameter (H) to the prediction error. Therefore, two different FCM structures are developed for $H = 1$ and $1 < H \leq 7$. The experimental results confirmed that obtained in-sample and out-of-sample prediction errors are much better for the second type by considering the propagation of errors that occurred given $H > 1$. Noteworthy, the model complexity has been reduced by removing those weights for which $|w_{ij}| < 0.3$.

The model elaborated in Papageorgiou e Froelich (2012) is another application of evolutionary learning FCM in which the main target is focused on the multi-step prediction of pulmonary infection based on the real clinical dataset. The experimental results confirm the good performance of the model for long-term forecasting and the capability of the

proposed multi-step learning method to simulate fully both the system's dynamic nature and internal changes.

A feasible and effective time series forecasting framework based on FCM was investigated in Lu, Yang e Liu (2013) to predict the enrollments of the University of Alberta on the linguistic level. The proposed quantitative technique is a composition of the C-means clustering algorithm, FCM, and RCGA. The FCM structure is formulated by using C-means clustering, which transforms the original time series into an FTS and extracts linguistic labels. Then, the constructed FCM is trained through the RCGA algorithm to represent the converted FTS and realize linguistic forecasting.

Apart from various partitioning methods proposed in the literature, the fuzzy c-means clustering technique has been used widely to generate fuzzy sets (concepts) by dividing the Universe of Discourse (U) into intervals with some overlaps. The research reported by Homenda, Jastrzebska e Pedrycz (2014a), Homenda, Jastrzebska e Pedrycz (2014c) are examples of univariate time series forecasting using FCM in which fuzzy C-means clustering technique is applied to cluster all learning values of time series such that the number of clusters (denoted as concepts) is defined by the user and each cluster acts as a fuzzy set. Through this method, at any time, the membership degree indicates the value of the time series that belongs to the generated cluster. Meanwhile, the degree of membership of the current value of time series $y(t)$ to the concept c_i determines the activation value a_i. This procedure is done for all concepts to reach the state vector. PSO is used as a learning method to obtain a weight matrix to minimize the Mean Absolute Error (MAE) error between the FCM responses and the targets.

Homenda, Jastrzebska e Pedrycz (2014a) proposed a method based on the moving window technique, influenced by window size and widow step size, to extract the concepts. Then, a map composed of several layers is formed such that the obtained concepts in the previous step are located in each layer. Although it is an interpretable and clear model, extracting the appropriate number of concepts is a hard task. Thus, there is a trade-off between window size and accuracy in this model. In other words, the accuracy of this model depends significantly on the size of the window. The larger the window size, the bigger the map which creates some problems in relation to training, visualizing, and interpreting. On the other hand, the researchers in Homenda, Jastrzebska e Pedrycz (2014c) suggested a posteriori FCM simplification strategies, and complexity reduction, by removing weak nodes and weights after evaluating their effectiveness in the map. Hereby it turned out that around $\frac{1}{6}$ of the edges and $\frac{1}{3}$ of the nodes could be dropped without any substantial increase in the prediction error.

In (LU; YANG; LIU, 2014), a univariate time series forecasting method was introduced based on the coupling of FCM and information granules employing fuzzy c-means clustering technique to translate the original time series into granular time series.

Thus, in the following, FCM is employed to describe obtained granular time series and accomplish forecasting tasks. PSO has been used as the learning algorithm in this method to extract the weight matrix among the concepts (granules). The proposed model is suitable for large-scale time series prediction because of its automatic framework. Noticeable that the greater the number of clusters, the higher the accuracy, and the less interpretable the model is.

A new time series forecasting method was outlined in Lu et al. (2014) by combining HFCM and fuzzy c-means clustering. The proposed model consists of two stages. In the first stage, the HFCM model is constructed and the second stage exploits the generated HFCM prediction model to perform inference and prediction. In other words, firstly, a fuzzy c-means clustering algorithm is exploited to create information granules by translating the original time series into granular time series and generating a structure of the HFCM model automatically. Next, the PSO algorithm is implemented as a learning method to extract weight matrices and complete the HFCM prediction model. Then, the designed HFCM is used to generate numeric prediction by doing inference in the granular space. To be noted that the adopted method is driven by a couple of adaptable parameters: the number of clusters and the order Ω. The best outcomes are obtained when the number of clusters and order are 9 and 4 respectively.

Researchers in Homenda, Jastrzebska e Pedrycz (2014b) implemented a new method to predict time series where the lagged time series is mapped to the concepts. The method is a mixture of FCM reconstruction procedures with a moving (sliding) window approach considered for both training and testing samples. In this method, the size of the map corresponds to the moving window size and it informs about the length of historical data, which produces a time series model. It is a note of worth that, unlike other existing forecasting methods, fuzzification and defuzzification steps are not performed in this study and fuzzification has been replaced by min-max normalization in this paper. More precisely, time series modeling and forecasting depend on the moving window technique. PSO learning method is used to optimize objective function (MSE) for three real-time series data. Moreover, the performance of the model was evaluated in two cases: with and without the presence of the bias weights. Thus, this paper investigated the roles of map size and bias on the accuracy performance of the proposed method. Additionally, it is not equally good for all kinds of data and does not perform well for trend and seasonal data.

One extended evolutionary learning of FCM has been introduced in Froelich e Papageorgiou (2014) to predict multivariate time series. The basic core of this research study is concentrated on three extensions. Firstly, transformation function optimization is such that the steepness values of logistic functions are not the same for all the concepts and they are optimized separately. Secondly, optimization of the fuzzification function. Finally, implementing modified quality metrics due to the integration of the fuzzification

function into the process of FCM training. The effectiveness of the proposed model has been tested over a number of real medical data gathered from the intensive care unit (ICU).

In another investigation, a new method was proposed in Froelich e Salmeron (2014) to alleviate the problem of forecasting multivariate interval-valued time series for the first time. For this purpose, FGCMs as an easy-to-interpret knowledge representation tool was designed as a non-linear predictive model. A genetic algorithm has been developed for training FGCM based on historical data. In this model, the approximation of the time series is subjected to a forecast instead of accurate numerical time series. As the experimental results indicated, better prediction performance is obtained when the prediction horizons are limited or for short-term prediction. For instance, it outperformed compared to ARIMA, VAR, or naïve models for just up to three days prediction horizons. Note that this investigation has used a sliding window concerning both the size of training and testing windows and their effect on the final results. The model meets the better performance with the short prediction horizon, fewer data variability, and absence of trends in data.

A combination of FCM with a multi-step supervised learning algorithm for time series prediction and monitoring was developed in Poczketa e Yastrebov (2015). The application of the gradient model and Markov model of the gradient are used as a multi-step learning strategy for FCM in this investigation. On the other hand one-step gradient, multi-step gradient method, and Markov model of the gradient were employed to learn FCM and tune the relevant parameters by minimizing the forecasting error. Using the mentioned strategy provides the possibility of forecasting the next values according to the currently monitored values of the system. High flexibility, especially in small datasets, ease of usage, and cheap cost of implementation are the bold characteristics of the model. Most importantly though, less complex FCM is produced through the SOGA training algorithm with semantic meaning in the provided relationships among concepts. The concepts of the obtained FCM are used as input for ANN with high accuracy.

Papageorgiou, Poczeta e Laspidou (2015) investigated a new FCM learning algorithm for the water demand problem. Thereby, the focal objective of this research is to present the new SOGA learning method compared with other learning procedures including the multi-step gradient model (MGM) and RCGA to optimize an objective function. In addition to investigating learning algorithms of FCM for multivariate time series modeling and prediction in this research, introducing a new FCM learning algorithm that finds the most important connections is considered as the other objective of this paper. However MSE, RMSE, MAPE, and MAE were used to evaluate the model's performance, it seems that the SOGA performs more efficiently regarding the MSE as the accuracy metric of the model. Further, based on the outcomes, the accuracy of summer demand prediction is better than winter demand.

A multivariate FCM-based forecasting model was implemented to predict electricity consumption by Papageorgiou e Poczeta (2015). The key attention of this paper has been focused on handling the complexity of the model using three learning algorithms consisting of MGM, RCGA, and SOGA. It is worth mentioning that the main aim is to forecast the electricity demand for three targets (output) concepts. Based on the results, the most accurate model is obtained when MGM is applied for training the model. Also, the obtained results confirmed the outperformance of the model against ARIMA, ANFIS, and ANN.

The other approximate time series forecasting framework was proposed by Froelich e Pedrycz (2016), which is a kind of new double-phased approach. In the first phase, the original time series is transformed into a sequence of granules to organize granular time series (GTS). In the second phase, the obtained granules from the first phase are clustered using fuzzy c-means clustering, and the centers of obtained clusters are regarded as the FCM concepts. Through this procedure, FCM is applied to forecast the concepts' activation vectors. Conceptually, the maximum activation level of the FCM concept is depicted in the forecasted granule. Numerically, the forecasted granule is considered a fuzzy set that is described in terms of its bounds and modal value. It should be reminded that in this research GA has been defined as the FCM learning method. The model is not suitable for random time series; however, it is the most proper for ones with a stable cyclic component as well as forecasting the changes of the amplitude.

An approach proposed for time series modeling and forecasting is developed by Homenda e Jastrzebska (2016), using clustering technique to design FCM in which the main focus is on the structure or map node selection. In the proposed scheme, firstly, the time series is transformed into a two-dimensional space of amplitude–change of amplitude as a basic and popular method. In other words, the set of concepts in FCM is positioned in the two-dimensional space of amplitude and change of amplitude (STACH; KURGAN; PEDRYCZ, 2008b). Then, the FCM clustering technique is used to create concepts ($k = 3, 4, 5, ..., 24$). Cluster validity indexes are applied in the next step to evaluate the generated concepts. 40 different validity indexes are exploited to evaluate the properties of the obtained concepts such as similarity of the points located in the same clusters, dissimilarity of the points belonging to various clusters, both of them, or other features. Among all, just five cluster validity indexes performed well as the top five indexes. Step four includes the application of PSO as a learning algorithm for proposed FCM with $k = 3, 4, 5, \ldots, 24$ concepts, for minimizing the MSE as the fitness function and evaluating the model performance. In other words, the evaluation is centralized on selecting the best structure of FCM using cluster validity indexes with the promising and minimum MSE value.

Furthermore, Papageorgiou e Poczeta (2016) explored a novel ensemble time

series forecasting technique that combines FCM and artificial neural network (ANN). This multivariate method is based on the efficient capabilities of evolutionary FCM enhanced by structure optimization algorithms and ANNs including two stages. In the first stage, an evolutionary FCM is made based on the SOGA in (PAPAGEORGIOU; POCZETA; LASPIDOU, 2015; ZAMORA-MARTINEZ et al., 2014). The use of SOGA allows the construction of the FCM model automatically from the historical data in which the most effective nodes (attributes) and weights are chosen to provide a less complex and more efficient FCM-based model. In the next stage, the created FCM determines the inputs of ANN where the BP method with momentum or Levenberg-Marquardt (LM) algorithm is exploited as ANN training strategies. Various kinds of accuracy metrics were employed to assess the model's performance using four datasets. It is worth noting that the developed software tool ISEMK (intelligent expert system based on cognitive maps) is used to evaluate the proposed method in all the experiments.

Despite several advantages of FCM in solving decision-making problems due to their transparent and comprehensive nature, some FCM shortcomings limit its performance in the domain of time series forecasting and analysis. Vanhoenshoven et al. (2018) introduced a novel practical FCM method employing ARIMA to enable FCM to predict multi-steps ahead of a fluctuating time series which keeps prediction accuracy, as well as keeping transparency of the weight matrix. In other words, to overcome the convergence issue, the updating rule adopted utilizing moving average and weight amplification function. Also, it has applied a modified version of the sigmoid activation function to confine the activation values in the allowable range. The results confirm the advantage of the model in predicting multiple steps of an oscillating time series, some parameters like lower prediction accuracy as well as less transparency affect these achievements.

Dynamic optimization of the FCM (DFCM) method was suggested by Salmeron e Froelich (2016) to predict time series. The main goal of the DFCM method is to optimize not only the weights but also all elements of FCM as well as the whole learning process to reach more accuracy. Accordingly, the structure of the FCM together with the transformation function and its parameters as well as the length of the learning period are optimized. In this case, five different population-based algorithms such as GA, PSO, SA, ABC, and DE were applied to optimize all of the mentioned parameters. The proposed method is competitive compared with other methods in the literature and is useful only for linear and stationary time series.

A novel two-stage forecasting technique is introduced by Poczeta e Papageorgiou (2018), which combines FCM and ANN to predict day-ahead gas consumption in Greece. In the first step, the SOGA learning algorithm is employed to construct the FCM from the original time series. In other words, SOGA is applied to extract the weight matrices as well as the model simplification by discarding the less effective concepts on the prediction

accuracy. In the second stage, the selected concepts of the simplified FCM are considered as inputs for ANN trained with the application of the back-propagation method with momentum and the Levenberg-Marquardt algorithm.

In addition to the aforementioned options, to determine the FCM's architecture and elevate the forecasting performance of FCM, wavelet transformation and empirical mode decomposition (EMD) were recommended by some authors. A time series forecasting method has been adopted by Shanchao e Liu (2018) based on the synergy of high order FCM and wavelet transform (redundant Haar wavelet) which is termed as Wavelet-HFCM. The proposed method is useful to handle large-scale non-stationary time series while the original signal is decomposed into the multivariate time series by wavelet transform. Then, HFCM is exploited for modeling and forecasting the multivariate time series. Making more generalization, the proposed model was trained via ridge regression to solve optimization problems unlike the lasso regression procedure adopted for FCM learning in Tsaih, Hsu e Lai (1998), Martens (2002) . Based on the outcomes, the regularization factor and the number of nodes have a considerable effect on the accuracy, unlike the lower impact of the order.

Another combination of fuzzy C-means clustering with FCM was proposed by Shan, Lu e Yang (2018) to predict univariate time series. In this approach, the architecture of FCM is constructed using an improved fuzzy C-means clustering which involves optimizing the membership matrix with a membership optimization function to reduce the sensitivity of clustering against outliers. The clustering method has been exploited to extract knowledge from data and generate the concepts of FCM, while PSO has been employed to learn FCM's weights automatically. This research evaluated the influence of the steepness index, clustering number as well as fuzzification coefficient on the accuracy of the model.

In the case of multivariate time series forecasting based on FCM, Poczeta, Papageorgiou e Yastrebov (2018) developed a new method to predict power electric consumption. In contrast to the proposed model in Papageorgiou e Poczeta (2016), Papageorgiou e Poczeta (2015), the main contribution of this investigation has been focused on multi-step ahead prediction with the exploitation of FCM and SOGA. In this approach, SOGA was used to automatically select the most important concepts and weights similar to the first stage in Papageorgiou e Poczeta (2016). Thereafter, SOGA was executed for three-step ahead prediction. Since SOGA is an extension of RCGA, the proposed model was compared with the model trained with RCGA. The best performance was reached with 5 concepts and 9 weight interconnections using SOGA with respect to the effect of learning parameters.

Interval-valued intuitionistic FCM (IVI-FCM) was implemented by Hajek, Prochazka e Froelich (2018) to predict univariate interval-valued time series (ITS). In spite of traditional FCM, interval-valued intuitionistic fuzzy sets were employed to represent

fuzzy sets in this approach. In this way, firstly universe of discourse is partitioned into some even-length intervals using the triangular membership function. Thus, IVI-fuzzy sets are built through the fuzzification of both bounds of the ITS. After that IVI-FCM is constructed such that every concept of the proposed model is an IVI-fuzzy set considering the value of the membership function for both lower and upper bounds of the ITS as well as the value of the non-membership function for the lower and upper bounds of the ITS. Furthermore, the value of the membership function and non-membership function for the upper and lower bounds of the weights are considered. Then, the state of each concept is updated via the reasoning rule. Finally, the final predicted output is obtained by exploiting the defuzzification step. It is noticeable that daily minima and maxima of the Nasdaq-100 index were used to represent the lower and upper bounds of the ITS.

A new time series forecasting strategy is proposed by Wu et al. (2019) based on a sparse Autoencoder (SAE) and a high-order FCM (HFCM), named SAE-FCM to handle some time series forecasting pitfalls such as the inability to extract good features of original time series, trapping in local minima and low prediction accuracy. SAE has been exploited to deal with the first limitation by extracting features from the original time series through an unsupervised training method. Then the second limitation is solved by a combination of outputs of SAE and HFCM to calculate the forecasted value. Finally, the batch gradient descent method (limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS)), as a fine-tuning algorithm inspired by deep learning, is used to update the weights of SAE-FCM and to improve the performance of SAE-FCM by removing the third limitation. The obtained results demonstrate the high accuracy performance of the proposed SAE-FCM.

Moreover, another novel ensemble forecasting methodology developed by Papageorgiou et al. (2019) is based on evolutionary FCM, artificial neural networks (ANNs), and their hybrid structure (FCM-ANN). In other words, the ensemble learning technique in this method combines various learning algorithms including SOGA-based FCM, RCGA-based FCM, efficient and adaptive ANNs architectures, and a hybrid SOGA-FCM-ANN recently proposed for time series forecasting structure proposed by Papageorgiou e Poczeta (2016) to solve the time series prediction problem of gas consumption in Greece. In this research, two of the most popular ensemble methodologies including the simple average and error-based (LEMKE; GABRYS, 2010; MAKRIDAKIS; WINKLER, 1983) are employed to evaluate the performance of individual predictors, the ensemble predictors, and their combination. The obtained results indicate the efficacy and efficiency of the defined method when compared with other autonomous methods like ANN, FCM, and LSTM.

An adaptive online time series forecasting based on a novel dynamic FCM (DFCM) was presented by Nannan e Chao (2019). In this investigation, FCM is dynamically constructed by means of the set of information granules in which PSO was exploited as

a training method. Also, a dynamic fuzzy C-means clustering algorithm is employed to online regulate the cluster center and weight based on the effect of the incoming data. More explicitly, both the map and the weights of the proposed FCM model can be dynamically adjusted along with the pattern change of the updated data which has a direct effect on the forecasting accuracy improvement.

A new adaptive FCM-based forecasting technique was introduced by Wang et al. (2019), composed of a collection of FCM. Thus, the architecture of the proposed model consists of multiple FCMs to enhance the adaptability of FCM as well as deal with concept drift in time series. In particular, constructing the premises at each time step; employing of fuzzy C-means algorithm to cluster the given time series into the predefined clusters; building individual FCM on each cluster, and calculating the final prediction using the weighted sum of the predictions of all the generated FCM according to the clusters of the given premise are the main steps of the proposed forecasting technique. The novelty of this adaptive technique is related to the clustering stage. After building both input and target matrices, fuzzy C-means is employed to cluster the premises of training and test data into 6 clusters which leads to generating a partition matrix. It is noticeable that each FCM is constructed using the PSO algorithm to adjust the weight matrices with the same dimension for each FCM. The obtained results for real and synthetic time series confirmed the superiority of the model compared with other fuzzy-based models.

Although Shanchao e Liu (2018) developed a wavelet-HFCM method to handle large-scale non-stationary times series successfully, it suffers from some weaknesses relating to the wavelet transformation. Regarding this issue, a novel and robust forecasting technique is proposed by Gao, Du e Yuen (2020) based on the synergy of HFCM and empirical wavelet transformation (EWT) to boost the performance of conventional FCM dealing with non-stationary and outliers. EWT is used as a novel adaptive signal decomposition method with a significant effect on the model in analyzing non-stationary time series data. EWT is employed to decompose the original time series into different levels in the Fourier domain which captures information of different frequencies. Afterward, HFCM will be trained through a novel method based on ϵ-support vector regression (ϵ-SVR), which elevates its robustness against outliers. The experimental results on eight publicly available time series show the superiority of the proposed model by comparing it with other models.

A new model has been adopted by Hajek, Froelich e Prochazka (2020) for forecasting interval-valued time series mixing interval-valued intuitionistic fuzzy sets (IVI-FS) with an FCM trained via the DE learning method. Also, this study has been focused on forecasting the approximation of time series to create interval-valued time series (ITS). That is, the exact values are replaced by minimal and maximal values in the predefined periods. For this reason, an FCM-based model has been designed to predict ITS which

is termed the Intuitionistic Fuzzy Grey Cognitive Map (IFGCM) using interval-valued intuitionistic fuzzy sets. The obtained results verify the satisfactory effect of the method compared with FCM and FGCM.

An accurate and robust method to deal with non-stationary and large-scale time series was proposed by Liu e Liu (2020), which is based on the combination of empirical mode decomposition (EMD) as a self-adaptive feature extraction technique and high-order FCM (HFCM), known as EMD-HFCM. Therefore, the EMD-HFCM is useful to handle some of the limitations of existing FCM methods including low precision and sensitivity to hyper-parameters. This novel method exploits EMD to create the set of stationary nodes of HFCM by extracting features from the original sequence. Then, a precise and efficient learning model based on Bayesian ridge regression, which is more robust than ridge regression, was employed to provide regular parameters from data. The experimental results verify the excellent performance of the proposed EMD-HFCM on eight public time series datasets dealing with large-scale and non-stationary time series compared to other available models.

Although FCMs have a strong ability to apply for time series forecasting, their performance is limited due to some deficiencies in using available feature extraction frameworks which influence the FCM's prediction accuracy. In other words, in these methods, some features of the original time series will be lost when it is mapped to FTS. For instance, Shanchao e Liu (2018) applied Harr wavelet to extract the features time series. To deal with some limitations of proposed feature time series models, a novel and generalized feature time series extraction method was suggested by Yuan et al. (2020) merging kernel mapping and HFCM that has been inspired by the kernel methods and the support vector regression (SVR), referred to as kernel HFCM. Kernel mapping is defined to transfer the original one-dimensional time series into the multidimensional feature time series, and then key feature time series (KFTS) from the multidimensional feature time series are selected through the proposed feature selection algorithm to develop HFCM. In the next step, a fast HFCM learning algorithm based on ridge regression is applied to adopt the fuzzy relationship of the HFCM. Lastly, the predicted one-dimensional time series is generated from the feature time series by exploiting reverse kernel mapping. However employing kernel mapping can help to capture the implicit patterns in the data, a suitable method for KFTS evaluation is still required to reduce the regression problem of the non-stationary time series and to improve the accuracy of the prediction.

Another FCM-based time series forecasting strategy was presented by Vanhoenshoven et al. (2020) using a pseudoinverse learning model namely FCM-MP. Therefore, this study mainly has focused on developing a new time-efficient learning algorithm based on the Moore-Penrose inverse to deal with some FCM learning limitations including time-consuming and poor accuracy of evolutionary and Hebbian-based methods respectively.

High forecasting accuracy, cheap cost of computation, multiple-step-ahead multivariate forecasting, and lack of laborious adjustment of parameters are counted as the significant strengths of the proposed learning strategy. The experiment results on 41 different time series highlight the superiority of the mentioned technique considering the value of the slope parameter of activation function λ in the set $\{1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0\}$.

Wang et al. (2020) formulated a new extension of FCM for multivariate time series forecasting termed DEEP FCM (DFCM). A deep neural network-based FCM model was introduced to reach interpretable multivariate prediction. The proposed model combines the strength of interpretability of FCM with the strength of deep neural networks by introducing the deep neural network models into the FCM knowledge-based models as the main solution for building an interpretable predictor strategy. DFCM as an extension of conventional FCM modeled the nonlinear and non-monotonic effects among the concepts and unknown exogenous factors that have a latent influence on system dynamics via a fully connected neural network and a recurrent neural network (LSTM-based u-function) respectively. Furthermore, to calculate the strength of the connection between a pair of concepts, the model is equipped with a partial derivative-based approach to guarantee interpretability. An Alternate Function Gradient Descent (AFGD) approach based on Backpropagation (BP) also was exploited for parameter inference which enhanced its prediction ability compared to other standard FCMs.

Feng, Lu e Yang (2021a) published a new time series modeling based on least square FCM termed as LSFCM. In this developed method, Fuzzy c-means clustering is employed to construct concepts of FCM while the least square method is exploited to adopt the weight matrices from the given historical observation of time series. In contrast to other traditional FCMs, the LSFCM model is a direct and one-time solution of matrix equation without repetitious stochastic searching. The LSFCM model is a straightforward, robust, and rapid learning method, owing to its reliability and efficiency. In the first stage of this two-stage model, the least square method is exploited to form an FCM model. In the second step, the LSFCM is optimized through concept refinement by relocating the concept's positions to obtain the optimal concepts through PSO to improve the prediction accuracy. In this study, the slope parameter of sigmoid activation function λ plays a significant role in keeping the values of weights elements in the interval $[-1, 1]$. It means that the proper weight matrix is obtained when $\lambda \geq \lambda_1$ and $\lambda_1 = \max\{|w_{ij}|\}$, $i, j = 1, 2, \ldots, k$. The results confirm the efficiency and accuracy of the model compared with other conventional models considering various time series for the different values of concepts ranging from $\{2, 3, ..., 20\}$ and λ belongs to $\{1, 5, 10, 50\}$.

In order to promote the reasoning capability in time series forecasting, a novel technique termed CNN-FCM was introduced by Liu, Liu e Wu (2020) embedding FCM into deep neural networks. The ability of FCM in system modeling enhances the stability

of deep learning in time series prediction. Since FCM-based methods have the ability to learn complex systems, the proposed CNN-FCM model can fill the gap in deep learning methods. For this purpose, CNN-FCM is based on a fuzzy cognitive block (FCB) which embeds the learning of HFCM into the deep learning architecture. Inside FCB, a Temporal Convolutional Network (TCN) is exploited to decompose the original time series into some components and extract the time series data of latent variables. Then, the state of each latent variable is updated based on the updating rule in HFCM and the fully connected layers within FCB learn the correlation of the latent variables (weights) in a different order. Finally, a regression model predicts the next observation according to the output of FCB. Structurally speaking, decomposition, dimensionality reduction, FCM, and linear regression construct the architecture of CNN-FCM. The decomposition unit consists of both residual blocks and average pooling. Afterward, a dimensionality reduction unit is employed. In this step, the output of TCN is inferred and compressed by two convolution layers to obtain the time series of latent variables. Then FCB and fully connected layer are utilized to produce the final output. Moreover, the sensitivity of the proposed model to the number of latent variables as well as the length of input has been analyzed in this research.

Poczeta, Papageorgiou e Gerogiannis (2020) introduced an FCM-based forecasting model with nested structure so that each concept at a higher map level is decomposed into another FCM to present a precise representation of complex time series. Then evolutionary learning algorithm is employed to optimize this nested structure. In order to extract significant links among the concepts as well as to assign the weights of these relations, the entire nested architecture is restructured using dynamic optimization. More clearly, a nested FCM structure is constructed through the following steps. Firstly, data is clustered by applying the K-means clustering algorithm. Secondly, the first level of the nested structure is constructed such that for each cluster the average values for concepts are calculated. The calculated values are normalized into the interval $[0, 1]$ using the standard min–max normalization. The averaged normalized data are divided into training and testing records. Then, the general FCM with k concepts is initialized and trained via RCGA and SOGA to determine the relationships between concepts at the first level of the nested structure. After that, in the next stage, the FCM models for the second level of the nested structure are constructed. Subsequently, the forecasted values for the second level of the nested structure are calculated. Finally, the testing data is used to evaluate the performance accuracy of the obtained models.

Shen, Liu e Wu (2021) implemented a fast prediction hybrid model combining elastic net and HFCM to deal with multivariate long non-stationary time series to predict human actions through the Electroencephalogram (EEG) data. In this technique, each node in an FCM represents a variable in the multivariate time series, thus an N-node FCM should be constructed as the prediction model. HFCM is responsible for capturing the

patterns of trends. In the first step, EEG signals are predicted using the historical data, and then a 1d-convolutional neural network (1d-CNN) is exploited to classify the predicted time series. It is important to highlight that the penalized least squares approach was exploited to solve a convex optimization problem and obtain the weight matrix that works better than LASSO. In other words, the optimization problem is a convex combination of LASSO and ridge penalty known as an elastic net penalty (or Naive elastic net) which has the characteristics of both the LASSO and ridge regression.

A novel multi-step ahead FCM-based long-term forecasting model was introduced by Feng et al. (2021). The proposed model integrates FCM, time series segmentation, and fuzzy clustering. In the first step, the normalized time series is divided into a set of non-overlapping segments in chronological order. Then, the obtained segments are transformed into fuzzy time series (FTS) utilizing a modified fuzzy c-means based on dynamic time warping (DTW). Subsequently, the long-term prediction model with FCM is constructed from the fuzzy time series. Next, a convex optimization-based method is applied to learn FCM by solving constrained least squares problems. Finally, the well-learned FCM model is exploited to produce predictive outputs and then defuzzify and compute the forecasted output values. In this study, 5, 10, 15, and 20 have been chosen as lengths of horizons for small-scale time series whereas for large-scale time series 10, 20, 40, and 60 have been predefined as the length of horizons.

A novel adaptive FCM-based (AFCM-based) forecasting method trained via Knowledge-Guidance Learning Mechanism was presented by Wang et al. (2021) to predict large-scale time series. In traditional FCM-based causal links among the concepts are determined based on all the training data and remain fixed later on. This issue restricts the model's flexibility, particularly when the goal is to predict large-scale drifting data. Therefore, one of the added values of this paper is to alleviate this problem. Furthermore, this article can tackle another problem evident in the original FCM-based forecasting procedure, in which the map remains unaltered. Given these two aspects, the authors developed a knowledge-guidance learning mechanism to train the AFCM-based forecasting model. Thus, dynamic and adaptive characteristics are exhibited in two ways so first, AFCM is trained through different causal relationships from the old data to build an AFCM-based forecasting model. Secondly, the already-constructed model is retrained on the new data with the guidance of knowledge mined from the old data.

The impact of the number of clusters and cooperation coefficient (the ratio of the new data to the old data) on the learning method was assessed in this study. The pseudo-F-statistics (PFS) index has been used to decide about the number of clusters for each case study. The larger the value of F, the more reasonable the clustering result. Therefore, a different number of clusters have been selected for each case study. But experimentally the value of the cooperation coefficient was set to 0.9 as the default value.

The obtained results indicate that the proposed model is much faster and more accurate than other competitors.

A new FCM-based prediction model has been proposed by Feng, Lu e Yang (2021b) based on partitioning strategies. Firstly, fuzzy c-means clustering is employed to partition time series into several sub-sequences. Consequently, each partition has its corresponding sequences. Subsequently, FCM is constructed in terms of these sub-sequences respectively. Finally, the FCM models are merged by fuzzy rules. Therefore, this model has not been designed by modeling the whole data directly similar to other existing models in the literature. The constructed model performs well in numerical prediction and also has good interpretability. As the results indicate, the forecasting precision can not continuously improve by increasing the number of nodes. In reverse, the model is more accurate as the number of partitions increases.

Another univariate FCM-based forecasting technique was proposed by Xixi, Ding e Luo (2022) combining high-order intuitionistic FCM (HIFCM) with variational mode decomposition (VMD). In this study, an intuitionistic fuzzy set (IFS) is introduced into FCM with a temporal high-order structure. In more detail, IFS and VMD are responsible for improving the capability of the model to deal with uncertainty and capturing fluctuation features of series data, respectively. Therefore, VMD is applied to decompose the original time series into several intrinsic mode functions (IMFs). In the next step, FCM is constructed by using the obtained IMFs as concepts. For the purpose of learning temporal dependence, FCM is extended to construct HFCM. Subsequently, an intuitionistic fuzzy set (IFS) is introduced into HFCM to enhance the model ability of uncertain data. Finally, PSO is utilized to train the model and search for the weight parameters.

A novel long-term interpretable univariate HFCM-based forecasting model with spatial attention for high-volatility time series was presented by Ding e Luo (2022). In this model, a kind of extended polar fuzzy information granules (FIGs) has been employed to capture fluctuation features of time series and to generate granule sequences from the original time series. As such, an FCM with a different structure is formulated in which each node in the FCM is defined through the obtained FIG, representing a specific short-term fluctuation feature. Equipping the model with polar-FIGs enables it to handle uncertainties as well as eliminate noise, outliers and disturbance from the original data. Noteworthy that this model is based on the attention mechanism and HFCM. Firstly, the generalization ability of the model is improved by eliminating the noisy data through the attention mechanism. Further, HFCM is exploited to upgrade the capability of the model to cope with the long-term dependence relationship. Finally, the proposed model is trained via a two-stage training process. In the first part, HFCM is trained using the ridge regression algorithm to generate its outputs for the training of the spatial attention

mechanism module. To be more precise, HFCM with spatial attention mechanism includes the following steps. Firstly, the attention module is fed through the obtained measured weights and state values (as inputs) of each order in HFCM. Then, the fully connected layer is applied to get the feature vector transforming the inputs. In the following, the second fully connected layer is used to guarantee that the output is the same length as the input layer. Subsequently, the attention coefficients are produced employing the softmax layer. Then attention values are obtained via weighted inputs considering attention coefficients. Finally, the attention values are concatenated and injected into a fully connected layer used to get the final outputs concatenated with the inputs, and the concatenated result is output through a fully connected layer. The obtained results indicate the outperformance of this technique compared to some recent FCM models like Wavelet-HFCM (SHANCHAO; LIU, 2018) and EMD-HFCM (LIU; LIU, 2020) as well as other baseline models such as LSTM, SVR, ANN, ARIMA, and Holt-Winters.

The Trend-Fuzzy-Granulation-Based Adaptive FCM for long-term time series forecasting was proposed by Wang et al. (2022). Different from original FCM-based forecasting models with fixed causal relations, this adaptive FCM model is based on non-constant causal relationships because constant causal relations are unreasonable and good prediction accuracy cannot be achieved as discussed earlier in Wang et al. (2021). Interpretability improvement, trend capturing and a higher level of flexibility are considered as the main added values of this investigation. Also, since in this approach, long-term prediction is not obtained based on the successively iterated one-step predictions, cumulative forecasting errors are reduced as the main advantage of this method. Granulating, normalizing, clustering, building the input matrix and the target matrix, constructing sub-FCMs, and forecasting constitute the main stages of the proposed model. Therefore, firstly the original time series is transformed into a granular time series $LG = \{LG_1, ..., LG_N\}$ consisting of Gaussian linear fuzzy information granules (LFIGs). LG_i is expressed as a vector $G_i = (k_i, b_i, \sigma_i, T_i)$. Thus each FCM with four concepts can be built corresponding to the slope k, the intercept b, the standard deviation σ, and the length of granules T respectively. After normalizing each parameter according to the Min-Max normalization, the fuzzy C-means algorithm is applied to generate some number of clusters such that the intercept of a granule is not significant in comparison to the slope, length and fluctuation amplitude of each granule. The obtained partitioning matrix and clustering centers from the clustering step are respectively employed for training sub-FCMs and calculating the membership degrees of the input vectors to clusters. In the next step, the input matrix and target matrix are constructed. Also, in this step, the sub-FCMs are built while each sub-FCM is trained by applying PSO to realize the optimum weight and bias matrices. Finally, in the forecasting step, before calculating the final predictions, the inverse function of normalization is utilized to predict the trend information. The minimum slope difference of evaluating key points ($slop_{min}$); minimum length of granules; the parameter of l_1-trend filtering and the

number of clusters are the most effective parameters on the model performance.

It is worth mentioning that Table 3 summarizes a list of these proposed FCM-based time series forecasting strategies arranged according to the date of publication taking into account some important parameters such as learning methods, order number, presence of bias, the number of nodes, activation function and prediction horizons.

Table 3 – Summary of the most relevant time series forecasting methods using FCM in the literature

Author	**Method**	**U/M**	**Learning method**	**Order**	**Bias**	**Concepts**	f	H
Stach, Kurgan e Pedrycz (2008b)	FCM+ig	M	RCGA	1	Y	9	bivalent	1
Wojciech e Juszczuk (2009)	adaptive+ E-FCM	M	DHL, BDA, DE, RCGA	1	N	$\{4, 5, 8\}$	sigmoid	1
Song et al. (2010)	FCM+ANN	M	gradient descent(BP)	1	N	-	-	1
Papageorgiou e Froelich (2012)	E-FCM	M	Single-step RCGA+ New Multi-step learning	1	N	15	logistic	$\{2, 3, ..., 7\}$
Froelich e Papageorgiou (2014)	FCM+EEL	M	EEL based on GA	1	N	15	logistic	1
Froelich et al. (2012)	E-FCM	M	RCGA	1	N	6	logistic	$\{1, 2, ..., 7\}$
Lu, Yang e Liu (2013)	FCMc+FCM	U	RCGA	1	N	5	sigmoid	1
Homenda, Jastrzebska e Pedrycz (2014a)	FCMc+FCM+ MW	U	PSO	1	N	Node=3, map size=$\{3,6,..,27\}$	sigmoid	1
Homenda, Jastrzebska e Pedrycz (2014c)	FCMc+FCM	U	PSO	1	N	27	sigmoid	1
Lu, Yang e Liu (2014)	FCMc+FCM+ig	U	PSO	1	Y	$\{3, 4, ..., 8\}$	sigmoid	1
(LU et al., 2014)	FCMc+HFCM	U	PSO	$\{1, 2, ..., 5\}$	Y	$\{3, 5, 7, 9, 10\}$	sigmoid	1
Froelich e Salmeron (2014)	FGCM,FGCM/ARIMA, FGCM/naive,FGCM/ES, FGCM/VAR	M	Interval-based EA	1	N	5	sigmoid	$\{1, 2, ..., 7\}$
Homenda, Jastrzebska e Pedrycz (2014b)	FCM+MW	U	PSO	1	N+Y	$\{3, 4, ..., 12\}$	sigmoid	1
Poczketa e Yastrebov (2015)	FCM+M-S-LA	M	Gradient method and Markov Model of Gradient	1	N	22	sigmoid	1

Papageorgiou, Poczeta e Laspidou (2015)	FCM+MGM+ population-based methods	M	MGM,RCGA,SOGA	1	N	6	sigmoid	1
Papageorgiou e Poczeta (2015)	FCM-RCGA,FCM-SOGA, FCM-MGM	M	RCGA,SOGA,MGM	1	N	8	sigmoid	1
Papageorgiou e Poczeta (2016)	FCM+ANN	M	RGGA,SOGA BP with momentum and LM	1	N	{8,9,14,26}	sigmoid	1
Papageorgiou, Poczeta e Laspidou (2016)	FCM+ANN	M	SOGA +BP SOGA+ LM	1	N	9	sigmoid	1
Froelich e Pedrycz (2016)	FCMc+granular FCM	U	GA	1	N	{2,3,4,5,6}	sigmoid	1
Homenda e Jastrzebska (2016)	FCM+VIC	M	PSO	1	N	$\{3, 4, \ldots, 24\}$	sigmoid	1
Salmeron e Froelich (2016)	DFCM	U	PSO,RCGA,ABC,SA,DE	self-adaptive	N	self-adaptive	Trivalent sigmoid tanh	1
Pedrycz, Jastrzebska e Homenda (2016)	FCM+ig+FCMc	U	PSO	1	N	$\{5, 6, ..., 10\}$	sigmoid	1
Vanhoenshoven et al. (2018)	FCM+ARIMA	M	Real-valued Genetic Algorithm	1	N	7	Modified sigmoid	≥ 1
Poczeta e Papageorgiou (2018)	FCM+ANN	M	RCGA,SOGA+BP,LM	1	N	6	sigmoid	1
Shan, Lu e Yang (2018)	FCM+improved FCMc	U	PSO	2	Y	$\{3, 4, ..., 10\}$	sigmoid	1
Poczeta, Papageorgiou e Yastrebov (2018)	FCM+SOGA	M	SOGA	1	N	{5,8}	sigmoid	$\{1, 2, 3\}$
Shanchao e Liu (2018)	Wavelet+HFCM	M	Ridge regression	$\{2, 3, 4, 6, 23\}$	Y	$\{4, 5, ..., 8\}$	tanh	1

Papageorgiou et al. (2019)	FCM+ANN	M	SOGA,RCGA,BP	1	N	-	sigmoid	1
Nannan e Chao (2019)	DFCM	U	PSO	1	Y	$\{2, 3, ..., 10\}$	sigmoid	1
Wang et al. (2019)	Adaptive FCM	U	PSO	1	Y	3	sigmoid	1
Wu et al. (2019)	SAE+HFCM	U	Ridge regression+(L-BFGS)	$\{4, 5, 8, 13\}$	Y	{25,30,35,45}	sigmoid	1
Hajek, Froelich e Prochazka (2020)	IFGCM	U+M	DE	1	N	5	logistic	$\{1, 5\}$
Liu e Liu (2020)	EMD+HFCM	U	Bayesian ridge regression	{2,6,9,11,22}	N	self-adaptive	tanh	1
Hajek, Prochazka e Froelich (2018)	IVI-FCM	U	DE	1	N	Not detailed	Logistic	1
Gao, Du e Yuen (2020)	EWT+HFCM	U	ϵ-SVR	$\{12, 48\}$	N	3	tanh	1
Yuan et al. (2020)	Kernel mapping+ HFCM	U	Ridge regression	$\{2, 3, ..., 7\}$	Y	$\{2, 3, ..., 7\}$	sigmoid	1
Vanhoenshoven et al. (2020)	FCM-MP	M	Moore-Penrose inverse	1	Y	{2,3,4,5,6,10,14,50}	sigmoid	$\{1, 2, ..., 60\}$
Wang et al. (2020)	Deep FCM	M	AFGD based on BP	1	N	$\{6, 9\}$	tanh	1
Orang et al. (2020)	HFCM-FTS	U	GA	2	N+Y	$\{5, 10, 20\}$	sigmoid tanh ReLU	1
Liu, Liu e Wu (2020)	CNN+HFCM	U	linear regression	$\{2, 3, , 4, 5\}$	Y	$\{2, 3, \ldots, 14\}$	sigmoid	1
Poczeta, Papageorgiou e Gerogiannis (2020)	Nested FCM	M	RCGA,SOGA	1	N	26	sigmoid	1
Shen, Liu e Wu (2021)	$ElasticNet_{HFCM}$	M	Naive elastic net	$\{1, 2, ..., 6\}$	Y	32	sigmoid tanh	1
Feng et al. (2021)	FCM+TSS+FCMc	U	convex optimization	1	N	$\{6, 7, 8, 9\}$	sigmoid	{5,10,15, 20,40,60}

Wang et al. (2021)	AFCM+KGLM	U	PSO	1	Y	3	sigmoid	1
Feng, Lu e Yang (2021a)	LSFCM	U	least squares	1	N	$\{2, 3, ..., 20\}$	sigmoid	1
Feng, Lu e Yang (2021b)	Partitioning strategies	U	constrained least squares	1	N	$\{3, 5, 7\}$	sigmoid	1
Ding e Luo (2022)	HFCM+Polar FIG+spatial attention mechanism	U	Ridge regression+BP	$\{2, 3, \ldots, 14\}$	Y	n^2 for n $\in$ $\{3, 4, \ldots, 12\}$	sigmoid	1
Xixi, Ding e Luo (2022)	HIFCM+VMD	U	PSO	$\{2, 3, 4, 5\}$	N	$\{3, 4, ..., 8\}$	tanh	1
Wang et al. (2022)	TFGBA-FCM	U	PSO	1	Y	4	sigmoid	{PH1+PH2}

ig → information granules ; E-FCM → Evolutionary-based FCM ; MW → Moving Window ; FCMc → Fuzzy C-means clustering
M-S-LA → Multi-step learning algorithm ; TSS → time series segmentation ; MGM → multi-step gradient method
U/M → Univariate/ Multivariate ; Y/N → Yes(with bias)/ No(without bias) ; f → Activation function; H → Prediction horizon
DFCM → Dynamic optimization of FCM ; BP → back-propagation algorithm ; EEL → Extended evolutionary learning
VIC → Validity Index Clustering ; KGLM → Knowledge-Guidance learning mechanism
PH1 $= \{\mathbf{1}, \mathbf{14}, \mathbf{28}, \mathbf{42}, \mathbf{52}, \mathbf{90}, \mathbf{100}, \mathbf{104}, \mathbf{116}, \mathbf{155}, \mathbf{180}, \mathbf{200}, \mathbf{208}, \mathbf{232}, \mathbf{235}, \mathbf{270}, \mathbf{300}, \mathbf{304}, \mathbf{348}, \mathbf{398}, \mathbf{416}, \mathbf{470}, \mathbf{608}, \mathbf{624}, \mathbf{678}, \mathbf{705}, \mathbf{749}, \mathbf{796}, \mathbf{912}\}$
PH2 $= \{\mathbf{1195}, \mathbf{1235}, \mathbf{1498}, \mathbf{1835}, \mathbf{2248}, \mathbf{4382}, \mathbf{8764}, \mathbf{13146}\}$ **; TFGBA-FCM → Trend Fuzzy Granulation-based FCM**

2.3 Performance Evaluation Metrics

As shown in Table 3, various time series forecasting based on FCMs were introduced in the literature. With the goal of estimating the accuracy of these proposed models, different error measurement techniques have been used that can be found via a table presented in our review paper (ORANG; SILVA; GUIMARAES, 2022). Since these experiments were performed using different real and synthetic datasets, this table also highlights datasets applied to each forecasting method.

2.4 Learning Algorithms

The core contribution of FCM learning is to extract the weight matrix according to either expert intervention and/or the available historical data. In fact, learning algorithms aim to fine-tune FCMs. However, diverse FCM learning methodologies are available in the literature, they are mostly classified in a triple class based on their fundamental learning pattern comprising Hebbian-based, population-based and hybrid techniques (PAPAGEORGIOU, 2011a). Therefore, this section concentrates specifically on examining and explaining the available methods in the literature.

2.4.1 Hebbian-Based Learning Methods

The particular aim of such unsupervised learning methods is to find weight matrices based on the domain experts' knowledge. Various Hebbian-based learning algorithms exist in the literature and some of these relevant methods are introduced in the following.

Differential Hebbian Learning (DHL) was suggested by Dickerson e Kosko (1993) based on the Hebbian theory (MORRIS, 1999); it is the first Hebbian-Based algorithm proposed by Dickerson and Kosko. In DHL, the weight matrices are modified when the value of corresponding concepts changes. Thus, the value of weights is updated repeatedly such that if there is no change in the value of concepts, the weight values remain the same in the next iteration. In other words, the learning process modifies the weight values until it reaches the desired state. The main drawback of this learning methodology is that the weight between a couple of concepts is updated by considering only the corresponding concepts and the effect from other concepts is ignored. Moreover, the order of data presentation plays a considerable role in the DHL method (SALMERON et al., 2019). The weight is updated according to the equation (2.35).

$$w_{ij}(t+1) = \begin{cases} w_{ij}(t) + \eta_i[\Delta a_i(t)\Delta a_j(t) - w_{ij}(t)] & if \quad \Delta a_i(t) \neq 0 \\ w_{ij}(t) & if \quad \Delta a_i(t) = 0 \end{cases} \tag{2.35}$$

where η_i is the learning factor, $a_i(t)$ is the activation value of concept c_i and $\Delta a_j(t) = a_i(t) - a_i(t-1)$ computes discrete changes along the time.

To handle the aforementioned DHL issue, an improved version of DHL known as the Balanced Differential Algorithm (BDA), was suggested by Huerga (2002). Through this way, the change of all concepts at the same time step and with the same direction is considered in the weight updating process. This method improves on the limitation of DHL by considering the values of all the concepts that change at the same time as the weights are updated. Although DBA alleviates the limitation of the DHL method, its application is restricted to just binary FCMs. Noteworthy that the Equation (2.36) is employed to update the weights.

$$w_{ij}(t+1) = \begin{cases} w_{ij}(t) + \eta_i \left[\left(\frac{\frac{\Delta a_i(t)}{\Delta a_j(t)}}{\sum_{k=1}^{n} \frac{\Delta a_i(t)}{\Delta a_k(t)}} \right) - w_{ij}(t) \right] & if \quad \Delta a_i(t)\Delta a_j(t) > 0, i \neq j \\ w_{ij}(t) + \eta_i \left[\left(\frac{\frac{-\Delta a_i(t)}{\Delta a_j(t)}}{\sum_{k=1}^{n} \frac{\Delta a_i(t)}{\Delta a_k(t)}} \right) - w_{ij}(t) \right] & if \quad \Delta a_i(t)\Delta a_j(t) < 0, i \neq j \end{cases} \tag{2.36}$$

In 2004, other two unsupervised Hebbian-based learning algorithms called Active Hebbian Learning (AHL) in Papageorgiou, Stylios e Groumpos (2004) and Nonlinear Hebbian Learning (NHL) in Papageorgiou, Stylios e Groumpos (2003) were adopted by Papageorgiou et al., although the learning of FCMs was still dependent on the expert's intervention (BEENA; GANGULI, 2011; PAPAGEORGIOU, 2005; PAPAKOSTAS et al., 2011). In the NHL method the range of values for the concepts, as well as the sign of edges, are specified by expert intervention while zero edges are not updated. Hence the main drawback of NHL is the construction of the initial graph which is recommended by experts. The initial graph structure elicited by the experts has remained during the learning process, thus, its physical interpretation is preserved. The stopping criterion is formed based on constraints imposed on nodes. The weights are adjusted when the stopping criteria are satisfied which include: (1) a close-enough solution to the desired response has been reached or (2) a fixed-point attractor has been identified. In this method, the weights are updated using the equation (2.37).

$$w_{ij}(t+1) = w_{ij}(t) + \eta a_j(t)(a_i(t) - a_j(t).w_{ij}(t)) \tag{2.37}$$

In AHL the desired set of concepts, an initial structure, and interconnections of concepts as well as the sequence of activation concepts are determined by expert knowledge. Unlike other methods, all weights are updated not only nonzero ones. Thus, the weights in AHL are adjusted in a seven-step iterative learning process to satisfy the predefined criteria. Furthermore, to prevent getting stuck in a local minimum, Li e Shen (2004) designed the Improved Nonlinear Hebbian Learning (INHL) technique in which a new

term named impulse was added to the update rule. In this case, the rule to compute the weights can be summarized via the equation (2.38).

$$w_{ij}(t+1) = (1-\gamma)w_{ij}(t) + \eta a_i(t)[a_j(t) - a_j(t).w_{ij}(t)] \tag{2.38}$$

Data-driven nonlinear Hebbian Learning (DD-NHL) was introduced by Stach, Kurgan e Pedrycz (2008a) as an improved NHL method. DD-NHL method relies on a similar learning principle as NHL, but the learning quality of DD-NHL is higher because of the use of historical data and decision concepts. Unlike other Hebbian methods, the initial weight matrix can be produced randomly in DD-NHL. Although the experimental results support the superior performance of the proposed method compared to NHL methods, its performance is poor in classification problems.

2.4.2 Population-Based Learning Techniques

In these supervised learning methods, the historical data is used in the shape of input-output pairs to train the model. In other words, we design the algorithm with the correct answer for each member of a dataset. In such problems, the algorithm tries to calculate the output for each new input by considering datasets into account. Therefore, the main goal is to derive the weight matrix to reflect the impact among the concepts by utilization of optimization algorithms to minimize error among the target and predicted responses. For this reason, these approaches can also be called Error-driven approaches Felix et al. (2019) and the experts are replaced by historical data. Such learning methods aim to search the parameter space to optimize a given objective function, which is computationally expensive. Thus, developing automatic learning methodologies based on historical observations becomes an important challenge. Accordingly, various types of evolutionary algorithms were introduced in the literature for training FCM and searching for the (near) optimal weight matrix.

Parsopoulos, Papageorgiou e Vrahatis (2002) introduced PSO as a learning method of FCMs using historical data to determine and formulate the sub-optimal weight matrix of FCMs. The authors used the proposed method to minimize the fitness function to reach a desired final value for the FCMs with fixed architecture, whereas determining the constraints depends on human knowledge in this way.

Real-coded GA (RCGA) was exploited by Stach et al. (2005) as an FCM learning method to create an FCM structure using the historical data in the frame of time series, which includes a single sequence of state vector values and without human intervention. The proposed RCGA is employed to minimize three various fitness functions to select the best one. Then the selected function is used to perform experiments by considering a different number of concepts and FCM densities. The bigger the size of the input data, the

more precise the learning is. On the other hand, the accuracy depends on the size of input data in the proposed RCGA and the learning performance degrades when increasing the size of the maps.

Multi-objective evolutionary FCMs are the other evolutionary learning algorithm proposed by Mateou, Moiseos e Andreou (2005) based on GA to support multi-objective decision-making problems. This method is applied to search the optimal weight matrix, which satisfies predefined activation levels among the participant nodes by the selection of the initial weights randomly.

Evolution strategy (ES) has been introduced by Koulouriotis, Diakoulakis e Emiris (2001) as an efficient way to design and construct FCMs. To ignore external intervention for fine-tuning FCM parameters, more specifically in complex systems, the paper focused on ES as a robust and flexible training procedure. The proposed method, which is a composition of FCM and ES, is examined for potential implementation in FCM-based systems. In this algorithm, the learning process will stop when the optimal FCM structure is obtained.

A Memetic PSO (MPSO) algorithm has been suggested by Petalas et al. (2005) as another FCM learning method to extract the weight matrix by minimizing the fitness function to construct the desired system. The proposed MPSO is a combination of PSO as a global search algorithm and the Hooke and Jeeves (HJ) algorithm as a local search component. The final results have proved the outperformance of MPSO in comparison to PSO.

The simulated Annealing (SA) algorithm was introduced by Ghazanfari et al. (2007) as another metaheuristic FCM learning method to extract weight matrices from the input historical data without expert intervention. The comparison of the performance of SA and GA in this paper illustrates that with more nodes (complex FCMs), the GA algorithm will deteriorate the FCM performance meanwhile SA improves the learning quality by covering the GA limitation, as well as improving the speed of training for each node's number. Thereby, the proposed method performs effectively for every map size. That is, the GA learning method is used for small FCM sizes while SA in large FCM sizes in this experiment.

Later an improved SA learning algorithm called Chaotic Simulated Annealing (CSA) was considered as an FCM learning method by Alizadeh e Ghazanfari (2009). According to the results, with more nodes, CSA outperforms SA with smaller learning errors. Although the CSA performs well for every map size, the execution time is longer in comparison to SA. Thus, Alizadeh et al. (2007) introduced another new learning strategy based on Tabu Search (TS) in which the quality of learning improved in comparison to GA. This algorithm generates smaller errors in comparison to GA as well as using fewer nodes. Besides, in some cases, the computational time of TS is less than GA, especially

for FCMs with small map sizes. To sum up, in terms of efficiency, the performance of CSA and TS is greater than conventional SA.

In addition to the above learning strategies, there exist other various types of population-based learning algorithms in the literature. For instance, Salmeron et al. (2019) has presented a novel learning technique namely asexual reproduction optimization (ARO) and Modified ARO (MARO). Also, it provides a comprehensive review of the available population-based learning methods. Game-based learning model was reported by Luo, Wei e Zhang (2009) by mixing FCM with game-based learning. Immune algorithm (IA) was developed by Lin (2009), which can be utilized for designing and modeling complex systems, Big Bang-Big Crunch (BB-BC) by Yesil e Urbas (2010), Ant Colony Optimization (ACO) as FCM learning algorithm by Ding e Li (2011), Extended Great Deluge Algorithm (EGDA) in Baykasoğlu, Durmusoglu e Kaplanoglu (2011), Artificial Bee Colony (ABC) algorithm in Yesil et al. (2013), Cultural Algorithm (CA) in Ahmadi et al. (2014), Imperialist Competitive Learning Algorithm (ICLA) Ahmadi et al. (2015) as a new robust, fast and accurate FCM learning method, Multi-objective optimization algorithm so-called MOEA-FCM in Chi e Liu (2015), dynamic multi-agent genetic algorithm (DMAGA) proposed in Liu, Chi e Zhu (2015), evolutionary multi-tasking multi-objective memetic FCMs (MMMA-FCMs) learning algorithm adopted in Shen, Liu e Wu (2019), Inactivation-based batch many-task evolutionary algorithm (IBMTEA-FCM) proposed in Wang et al. (2021) are some other examples of population-based FCM learning methodologies which have been introduced in the literature.

As discussed, learning algorithms are employed in FCMs to extract/tune/adjust the weight matrix. With regard to the problem, based on the expert's knowledge and historical data, the proper learning method is used to construct an accurate FCM model. There are some strengths and limitations for both population-based and Hebbian-based methods.

Various learning methodologies are applied in different domains for FCM modeling, FCM time series prediction, FCM classification, FCM decision making, and FCM optimization as detailed in Papageorgiou (2014). Based on the literature, population-based methods are used widely in a major proportion of the applications due to their lower simulation error, higher functionality, robustness, and generalization abilities (PAPAGEORGIOU, 2011a). Unlike the mentioned advantages, some generic limitations such as time-consuming, a large number of learning parameters, availability of historical data, a large number of learning processes, and convergence issues may hinder the application of these learning approaches in some cases in this category. Hebbian-based methods, on the opposite side, are useful because of some features such as cheap cost of computing, ease of use, keeping signs of connection, and causal meaning of adjusted weights. Poor generalization, reliance on knowledge extraction from the experts, dependence on initial states and connections,

and low deviation from the initial weights can be taken into account as the main drawbacks of the Hebbian-based methods. Due to the mentioned analysis, it can be deduced that Hebbian-based algorithms are suitable for control problems Salmeron e Papageorgiou (2013), while the population-based methods are applied more widely in areas such as time series forecasting, classification, simulating the chaotic behavior, and virtual system (STACH et al., 2005).

2.4.3 Hybrid Methods

As noted earlier, both Hebbian and population-based methods are confined in some cases due to their limitation. To alleviate the existing drawbacks and improve efficiency, hybrid techniques that incorporate Hebbian and population methods were proposed as an alternative. In the sense that hybrid learning methods consider both features including the effectiveness of Hebbian learning and the global search ability of population-based methods to train FCMs. Indeed, the focus of hybrid methods is to update/modify the connection matrices extracted from the historical data and experts' knowledge. Although hybrid algorithms can cope well with complex systems, according to the literature, few studies have been devoted to this approach.

A hybrid method was introduced in Papageorgiou e Groumpos (2005) which is a mixture of unsupervised learning methods, NHL algorithm, and differential evolution (DE) strategy to handle some FCMs drawbacks, improve the behavior of the FCMs dynamically and enhance the flexibility of FCMs. Ease of implementation, inexpensive computation, lower number of control parameters, comparable DE convergence properties as well as efficient handling of nonlinear, non-differentiable, and multimodal fitness functions are the motivation factors of the investigated technique in this paper. The proposed NHL-DE algorithm is divided into two steps. Primarily the NHL method is employed to learn FCM then DE is used for FCM retraining. The goal of the first stage is to search for the appropriate weights while the responsibility of the second stage is recomputing the weights and minimizing the fitness function. Note that in this method the initial DE population directly depends on the performance of the first stage. Good solutions in the first stage are incorporated in the evolutionary computation stage thus affecting its performance. Two termination conditions are considered in this method and the minimization process will be finished as the optimization criterion is satisfied in the second stage. The experimental results illustrate the high speed and efficiency of the model examined in three various FCM models.

Besides, Yan-chun e Wei (2008) introduced a new hybrid learning FCM method coupling NHL and RCGA algorithms. Exploiting RCGA and NHL as the key components of the algorithm improves the FCM's ability to extract data from historical data and search for the optimal weight matrices based on expert knowledge. This method benefits from both

the effectiveness of the NHL learning method and the global search ability of the RCGA method. In the proposal method, RCGA is applied as an input data mining algorithm while the NHL is used for refining the weights. Although the results are promising, the model is time-consuming as the model's size increases.

A hybrid method has been proposed by Ren (2012) using NHL and extended great deluge algorithm (EGDA). EGDA approach is similar to the SA training method with global search ability however it requires a lower number of parameters. Simplicity, high speed of convergence, and dealing with continuous values for the concepts are the main outstanding properties of the NHL algorithm. After training FCM using the NHL algorithm, the outcome is fed to EGDA. In other words, the method has been constructed in two steps. In the first step, NHL is applied for FCM training to extract a local optimal weight matrix. Afterward, the candidate objective function is optimized using the EGDA algorithm in the second step.

2.4.4 Other Methods

There exist other methods proposed by investigators to construct or optimize FCM-based systems that do not belong to the triple above categories.

Konar e Chakraborty (2005) developed a novel unsupervised learning method. The weights of the directed edges are adapted from the transition to the Petri Net via the learning process. To analyze the dynamic behavior of the algorithm, the Hebbian-based learning algorithm was exploited considering natural decay in weights. After the convergence of the learning algorithm, the network can be used to compute the beliefs of the desired propositions from the supplied beliefs of the axioms (places with no input arcs). The conditional nature of the algorithm in terms of stability allows the model to be used in complex decision-making and learning. The other model was developed for knowledge refinement by adaptation of weights in a fuzzy Petri net using a different form of Hebbian learning. Also, combined fuzzy cognitive maps (FCMs)–Petri nets (PN) approach has been developed by Kyriakarakos et al. (2012) for the energy management of autonomous polygeneration microgrids. The PN is used as an activator in the fuzzy cognitive map structure to enable different FCMs to be activated depending on the state of the microgrids.

Another technique based on Fuzzy Boolean Nets (FBNs) as a hybrid fuzzy neural technique adopted by Carvalho e Tomé (2007) for training rule-based FCMs. In this method, fuzzy Boolean nets are employed to extract qualitative fuzzy rules from crisp/quantitative data. Even though the main focus of this work is on the optimization and completion of Fuzzy Causal Rule Bases (FCRb), it can be generalized to all fuzzy rule-based. In other words, FBNs act like qualitative interpolators with perfect generalization ability.

However, as the process generalizes, the performance of the method deteriorates due to the exponential increase in FBN size as the number of predecessors increases, even if it can be compensated by granularity employment in FBN's internal memories.

Zhang, Zhang e Sun (2017) proposed a new FCM learning framework (e-FCM) for recognizing speech emotion. In this approach, the pleasure-arousal-dominance emotion scale is employed to measure the causal relations among emotions where the structure of the network is determined via certain mathematical derivations. The structure of e-FCM includes an input layer that collects data from speech features and an output layer composed of emotion classes. The proposed method is much faster and more accurate than traditional and population-based methods such as GA, particularly for large-scale FCMs.

Moreover, Mls et al. (2017) outlined a learning method based on partial expert estimation and evolutionary algorithms to handle the lack of certainties in expert estimations of FCM weight matrices. This article hired a modification of Interactive Evolutionary Computing (IEC) for training and optimizing FCM termed Interactive Evolutionary Optimization of Fuzzy Cognitive Maps (IEO-FCM).

Besides the aforementioned methods, the other FCM adaptation methods include gradient-based techniques. However these methods are used to minimize the cost function as a kind of error-driven approach, they do not use metaheuristics. Therefore, they are classified into this learning group. A new FCM gradient-based learning method was suggested by Madeiro e Zuben (2012) to improve the performance of population-based methods through their combination with local search approaches (MADEIRO; ZUBEN, 2012). Hence, this study evaluates the performance of the proposed automatic learning method which combines RCGA and DE with a gradient-based local search method. In fact, the mentioned method considers both abilities of exploitation and exploration in gradient procedures and evolutionary ones respectively.

Chi e Liu (2014) implemented a hybrid method combining the memetic algorithm (MA) and artificial neural network (ANN) has been adopted to learn large-scale FCMs, referred to as MA-NN-FCM. However MA is considered as a fast evolutionary-based method to find a set of regulated nodes, it suffers from the slow rating of weight matrix exploration. Therefore, a neural network is employed to extract the weight matrix using the gradient descent approach. Experimental results prove the efficiency of the proposed method of learning large-scale FCMs up to 100 nodes.

Based on the network topology of FCM, several FCM learning methods are inherited from ANN learning approaches to computing the weight matrix. Gregor e Groumpos (2013a) proposed a supervised gradient-based methodology based on the delta rule and Backpropagation principle which was originally developed for multi-layered networks. The extended version of the Backpropagation algorithm termed Backpropagation through time (BPTT) is applied in this investigation. Thus, three approaches including

One-step Delta Rule, Every-step Delta Rule with Windowed BPTT, and One-step Delta Rule with Windowed BPTT have been employed to measure the regression problem of FCM training.

Chunmei e Yue (2012) introduced an automated FCM learning method that exploits the evolutionary structure of cellular automata to train the weight matrix of FCM based on historical data consisting of one sequence of state vectors. Encoding the weight parameters is adopted via one-dimension cellular automata and a cell space is made by selecting the states of cellular in the range [0,1]. In this method, the learning algorithm is composed of 7 elements including C-Cellular Automata code, E-fitness function, P-initialize Cellular Automata Configuration, M-initialize Cellular Automata Configuration size, S-selection, R-rule, T-stopping condition which can be defined as a 7-tuple (C, E, P, M, S, R, T). It is necessary to highlight that the employment of a mutation operator in the algorithm will improve the speed of convergence.

An optimization technique was introduced by Papageorgiou et al. (2016) to model complex systems containing a large number of concepts in the domain of decision-making and management. The focus of this paper is to present a new FCM concept reduction approach and its application to develop a less complex FCM that is easy to use. Accordingly, a clustering algorithm based on the fuzzy tolerance method was introduced as an FCM reduction procedure by reducing the number of nodes and connections among them. The concepts with the same behavior are classified into the same cluster. Afterward, the weight values are recalculated. The constructed FCM model with less complexity is capable of dealing with the uncertainty in different scientific fields.

Deterministic learning of hybrid Fuzzy Cognitive Maps and network reduction approach was proposed by Napoles et al. (2020). In this technique, FCMs are exploited to model dynamic systems, which are conditioned by several input variables that influence output ones. Thus, in this study expert knowledge combines with population-based knowledge mining as a hybrid method to create such a system. It means that the relation among input variables is determined by experts, while the extracted model from data provides the simulation of the output variable states. So, the scheme of the proposed model depends on both expert knowledge and historical data. As the other contribution, the authors introduced a very fast, deterministic (inverse method), and accurate learning rule to determine the weight matrix to define the system based on the Moore-Penrose inverse, thus it does not require any parameter to be specified. Also, to guarantee that the learned weights are within the allowable range, a weight normalization approach has been considered. At last, a model was proposed to distinguish irrelevant weights in the learned FCM network. As the model motivation, the absolute weight values have been taken into account as well as the concept activation values. Additionally, offset correction and weight correction methods were employed to tune FCM-based after removing a weight, however,

the results verify that sigmoid slop calibration is more convenient than weight calibration in terms of efficacy. The reported MSE and processing time over 35 datasets indicate the superiority of the model compared with PSO, RCGA, and DE.

Feng et al. (2021) introduced a new straightforward, rapid, and robust learning method referred to as LEFCM proposed to tackle some limitations of convenient learning methods such as time-consuming dealing with large-scale FCMs, lack of robustness when the experimental data contain noise as well as rarely weight distribution which affects the FCM performance. The crux of the investigated method is that the learning problem of FCM can be considered as a convex optimization problem with constraints that can be solved in polynomial time complexity by applying the gradient-based method. In the LEFCM method, the introduction of the least squares term into the convex optimization problem ensures the robustness of the well-learned FCM. Furthermore, the performance of the generated FCM is improved because of the existing more reasonable weight distribution by considering entropy constraints in the convex optimization problem. Overall the proposed method is rapid and robust to learning large-scale FCMs up to 200 neurons, specifically, for learning FCMs using noisy data.

Also, Gradient Residual Algorithm (ZHANG; SHEN; MIAO, 2011), Extreme Learning Machine (HUANG; SHEN, 2013), Gradient-based search (GREGOR; GROUMPOS, 2013c), Multi-step Gradient (PAPAGEORGIOU et al., 2015) and $LASSO_{FCM}$ by Wu e Liu (2016) belong to the last group of FCM learning methods, as can be seen from the Table 4.

Table 4 summarizes the proposed FCM learning algorithms into different groups while population-based methods make up the major proportion of learning methods. A more detailed look at the table reveals that some of the population-based algorithms have been introduced to deal with large-scale problems. In other words, based on Wang et al. (2021), the large-scale FCM learning problems can be solved by exploiting three types of evolutionary-based algorithms including high-dimensional optimization technique (LIU; CHI; ZHU, 2015; YANG; LIU, 2020), decomposition strategy (CHEN; MAZLACK; LU, 2012; YANG; LIU, 2019) and search space reduction (ZOU; LIU, 2018; LIU; LIU, 2018).

However, all of these methods are either ineffective or time-consuming to cope with the large-scale FCM learning problems due to the large search space (WANG et al., 2021). In order to enhance the speed and performance of the existing methods, a random inactivation-based batch multitasking evolutionary algorithm, IBMTEA-FCM was proposed by Wang et al. (2021) to handle large-scale FCM learning problems. In this method, the learning of local connections of nodes in a single FCM is modeled as a many-task optimization (MaTO) problem.

Table 4 – Proposed FCM learning techniques in the literature

Category	Learning Technique	Author	Number of Nodes
Hebbian-based	Differential Hebbian Learning (DHL)	(DICKERSON; KOSKO, 1993, 1993)	10
	Balance Differential Algorithm (BDA)	(HUERGA, 2002)	{5,7,10}
	Nonlinear Hebbian Learning (NHL)	(PAPAGEORGIOU; STYLIOS; GROUMPOS, 2003)	5
	Active Hebbian Learning (AHL)	(PAPAGEORGIOU; STYLIOS; GROUMPOS, 2004)	8
	Data-Driven NHL	Stach, Kurgan e Pedrycz (2008a)	{5,10,20}
Population-based	Particle Swarm Optimization(PSO)	Parsopoulos, Papageorgiou e Vrahatis (2002)	5
	Genetic Algorithm(GA)	Mateou, Moiseos e Andreou (2005)	16
	Real Coded Genetic Algorithm (RCGA)	Stach et al. (2005)	{4,6,8,10}
	Parallel RCGA	Stach, Kurgan e Pedrycz (2007)	{10,20,40,80}
	Evolutionary Strategy (ES)	Koulouriotis, Diakoulakis e Emiris (2001)	6
	Memetic Particle Swarm Optimisation (MPSO)	Petalas et al. (2005)	18
	Simulated Annealing (SA)	Ghazanfari et al. (2007)	{2,3,...,15}
	Chaotic Simulated Annealing (CSA)	Alizadeh e Ghazanfari (2009)	14
	Tabu Search (TS)	Alizadeh et al. (2007)	{3,4,...,15}
	Game based learning	Luo, Wei e Zhang (2009)	33

Continued on next page

Table 4 – continued from previous page

Category	Learning Technique	Author	Number of Nodes
Population-based	Immune Algorithm (IA)	Lin (2009)	26
	Differential Evolutionary (DE)	Juszczuk e Froelich (2009)	5
	Big Bang-Big Crunch (BB-BC)	Yesil e Urbas (2010)	{5,6,10}
	divide and conquer RCGA	Stach, Kurgan e Pedrycz (2010)	40
	Ant Colony Optimization (ACO)	Ding e Li (2011)	40
	Extended Great Deluge Algorithm (EGDA)	Baykasoğlu, Durmusoglu e Kaplanoglu (2011)	6
	Sparse RCGA	Stach, Pedrycz e Kurgan (2012)	40
	Decomposed ACOR based on ACO	Chen, Mazlack e Lu (2012)	100
	Artificial Bee Colony (ABC)	Yesil et al. (2013)	13
	Cultural Algorithm (CA)	Ahmadi et al. (2014)	13
	Imperialist Competitive Learning Algorithm (ICLA)	Ahmadi et al. (2015)	{5,7,8,10,13,20,24,30,40,50}
	Structure Optimisation Genetic Algorithm (SOGA)	Poczeta, Yastrebov e Papageorgiou (2015)	8
	Dynamic Multi-Agent Genetic Algorithm (DMAGA)	Liu, Chi e Zhu (2015)	{5,10,20,40,100,200}
	Decomposed RCGA with tournament selection	Chen et al. (2015)	{10,50,100,200,300}
	Multi-objective Evolutionary Algorithm (MOEA-FCM)	Chi e Liu (2015)	{5,10,20,40}
	Decomposition-based DMAGA	Wang et al. (2021)	{300,500}
	Mutual Information based Two-phase Memetic Algorithm(MIMA-FCM)	Zou e Liu (2018)	{7,8,10,13,20,24,40 ,100,200,300,500}
	Niching-based Multi-Modal Multi-Agent GA ($NMM_{MAGA} - FCM$)	Yang e Liu (2019)	{5,10,20,40,100}
	Asexual Reproduction Optimisation (ARO) and Modified ARO (MARO)	Salmeron et al. (2019)	{5,6,7,10,24,100}

Continued on next page

Table 4 – continued from previous page

Category	Learning Technique	Author	Number of Nodes
Population-based	RCGA combined boosting strategy(RCGA-BFCMs)	Yang, Liu e Wu (2019)	{10,20,40,100}
	Multi-tasking Multi-objective Memetic FCMs (MMMA-FCMs)	Shen, Liu e Wu (2019)	{10,20,40,100,200,400,600}
	MAGA based on the convergence error (MAGA-Convergence)	Yang e Liu (2020)	{5,10,20,40,100,200}
	Inactivation-based batch many-task evolutionary algorithm(IBMTEA-FCM)	Wang et al. (2021)	{20,40,100,200,330,500}
Hybrid	NHL+DE	(PAPAGEORGIOU; GROUMPOS, 2005)	8
	RCGA +SA	Ghazanfari et al. (2007)	15
	NHL+RCGA	Yan-chun e Wei (2008)	3
	NHL+EGDA	Ren (2012)	3
	PSO+ACO	Napoles et al. (2014)	25
Other	Gradient Residual Algorithm	Zhang, Shen e Miao (2011)	7
	RCGA+ DE+a gradient-based method	Madeiro e Zuben (2012)	{4,6,8,10,12,24,38}
	Extreme learning machine	Huang e Shen (2013)	6
	Gradient-based search	Gregor e Groumpos (2013c)	20
	MA+Gradient descent	Chi e Liu (2014)	{20,40,100}
	Multi-step Gradient	Papageorgiou et al. (2015)	8
	Extended Delta Rule	Rezaee, Yousefi e Babaei (2016)	34
	$LASSO_{FCM}$	Wu e Liu (2016)	{10,20,40,100,200}
	e-FCM	Zhang, Zhang e Sun (2017)	5
	IEO-FCM	Mls et al. (2017)	6
	Entropy-based method (LEFCM)	(FENG et al., 2021)	{20,40,100,200}

2.5 Discussion

As discussed in Sections 2.2.2 and 2.2.3, building the structure of the FCM and learning the weights of FCM based on historical data are critical issues in designing FCM-based forecasting models. We can summarize the observations from the above study as follows:

1. Building structure: In relation to multivariate models, the structures are built in such a way that each variable represents a concept. Equivalently, an FCM is constructed so that the number of concepts and the number of variables in the time series are equal. Despite the success of the existing methods in the literature, in our opinion, these methods will not be very effective in dealing with high-dimensional time series. The more the number of variables, the more the number of concepts. This fact leads to an increment in the dimension of the weight matrices, particularly in high-order models, and consequently, model training will be more time-consuming. Although some researchers provided solutions in the literature to reduce the complexity of the models (HOMENDA; JASTRZEBSKA; PEDRYCZ, 2014c; PAPAGEORGIOU; POCZETA, 2016; PAPAGEORGIOU; POCZETA; LASPIDOU, 2016; POCZETA; PAPAGEORGIOU; GEROGIANNIS, 2020), it is still an open challenge in this area. This issue becomes more acute when dealing with big data, for example when the goal is to predict the IoT time series with more than 100 variables.

   To the best of our knowledge, univariate FCM-based forecasting methods followed the same pattern. More specifically, the given univariate time series is transformed into a multivariate one applying prevalent strategies comprising clustering, moving window, wavelet transform, empirical mode decomposition, and other techniques as explained in detail in section 2.2.3. However, Wang et al. (2021) implemented a new adaptive FCM-based model using different techniques focusing on map size and causal alterations.

2. Learning methods: Another valuable problem that needs to be discussed is how to train FCM effectively and efficiently. Hebbian-based and population-based models have been successfully used for small-scale FCM learning problems (FENG; LU; YANG, 2021a), with population-based techniques showing superior performance in comparison to the Hebian-based methods in the domain of time series forecasting (FROELICH; JUSZCZUK, 2009). Also, Table 3 evidently confirms that a major proportion of the research has concentrated on a population-based algorithm to train weights of FCM, where GA and PSO have been widely used.

   It is worth highlighting that various fitness functions were employed to assess the performance of the proposed FCM-based forecasting models trained via population-based techniques. To be more precise, each chromosome's measurement of the fitness

function is done via one-step modeling of the relevant FCM. The output error among the actual responses of the system against the obtained state vectors, as FCM responses, is exploited to measure the fitness value. Therefore, the objective of fitness function applications is the assessment of the corresponding FCM by calculating the accumulated prediction error among the response values and real ones.

Based on Wang et al. (2021), the construction of FCM using Hebbian-based methods is conducted through a very fast iterative optimization scheme, but the generalization ability of the model is limited. On the other side, population-based (or evolutionary algorithm-based (EA-based)) methods are the most popular, because of improve the forecasting accuracy, robustness, generalization abilities as well as simplicity of application in comparison to the accuracy of the Hebbian-based method. In spite of these achievements, as explained in Shen, Liu e Wu (2021), the application of EA-based algorithms is limited to modeling short stationary time series and they are unable to deal with long non-stationary time series. In addition, EA-based models are more time-consuming and this issue becomes more apparent with the increase in the size of the time series. In other words, they lose their ability and efficiency to handle large-scale high-dimensional time series because evolutionary algorithms must evaluate the fitness functions in each iteration when optimizing targets. This leads to a very slow and time-consuming training process. Therefore, there is a necessity to replace EA-based models with other rapid learning algorithms. Accordingly, regression-based prediction models have recently been introduced to address the problems of EA-based models. For example, Shen, Liu e Wu (2021) proposed a new fast multivariate long-time series forecasting framework based on the naïve elastic net integrating both LASSO and ridge regression. Feng, Lu e Yang (2021a) developed an FCM-based forecasting model utilizing a least-squares algorithm to eliminate the strenuous iterative computation. Ding e Luo (2022) implemented a new FCM-based forecasting method trained with ridge regression to predict large-scale time series. Yuan et al. (2020) proposed a kernel-HFCM technique learned by ridge regression and the comparison between ridge regression-based HFCM learning algorithm and other classical learning algorithms like RCGA, ICLA, and MARO shows the outperformance of ridge regression. In addition, ridge regression by Shanchao e Liu (2018), Bayesian ridge regression by Liu e Liu (2020), support vector regression-based (ϵ-SVR) learning by Gao, Du e Yuen (2020) demonstrated the fact that recently the main focus is on regression-based learning techniques.

Nevertheless, Wang et al. (2021) introduced a new knowledge-guidance learning mechanism based on PSO for large-scale time series forecasting which is much faster and more accurate than the original and other adaptive FCM-based models. The existing consensus reveals that equipping FCM with efficient learning methods still remains the main concern in this area.

3. Order: According to Tables 2 and 3, among the parameters affecting the performance accuracy of the models, the first rank belongs to the number of concepts followed by the order. However, the majority of proposed forecasting models have been designed based on first-order FCM. Also, the effect of bias has been considered in a few numbers of references.

4. Activation function: as listed in Table 3, sigmoid has been widely used as the most common activation function with different steepness values. From this point of view, the effect of using different activation functions on the performance of the models has not been evaluated in the literature. However, very few studies considered this subject in their investigations (SALMERON; FROELICH, 2016; ORANG et al., 2020; SHEN; LIU; WU, 2021).

5. Forecasting horizon: more than 80% of the papers have been focused on one-step ahead forecasting. Interestingly, an adaptive FCM-based forecasting model was implemented by Wang et al. (2022) considering different forecasting horizons ($Min_H = 1$ and $Max_H = 13146$) as mentioned in Table 3.

2.6 Chapters' Highlights

This chapter has been organized to provide an overview of recent developments in time series forecasting methods using FCMs and to explore potential future research opportunities. Also, this chapter covers an introduction and revision on some corresponding properties/fundamentals of FCM (including the structure of FCMs and reasoning rules, high-order FCMs, extensions of FCMs, and dynamic properties of FCMs) and FCMs learning methodologies.

With respect to the FCM structure, the core attention of FCM-based approaches is basically focused on building FCM's structure and extracting weight connections among the concepts. It means that common methods such as C-means clustering or information granules are employed to formulate the FCM structure and then learning algorithms are introduced to capture causal relations among a couple of nodes. Hebbian-based methods, population-based methods, and their combinations, as well as ANN-based methods, are the main categories of FCM learning approaches, which, as mentioned, have advantages and disadvantages. In addition, a variety of activation functions and reasoning rules were explored by researchers to generate exact and meaningful results by choosing the appropriate ones.

FCMs as an efficient member of soft computing society, have been widely used in the time series forecasting domain to improve the accuracy as well as interpretability as a nonlinear predictive model. Time series forecasting applying FCMs includes two stages:

designing a proper FCM structure and appropriate weight training techniques. Hence, different FCM models have been implemented to predict time series taking into account different numbers of concepts and orders as well as various learning strategies. Since most FCMs have used evolutionary learning algorithms, the recent remarkable challenge of FCMs is to find a fast training method such as least squares because evolutionary-based learning methods are more time-consuming. Although various univariate and multivariate methods have been presented to boost the conventional FCM, particularly dealing with uncertainty in data, still exist some open challenges and future research possibilities which will be discussed in chapter 7.

3 PROPOSED HFCM-FTS TECHNIQUE

The main contribution of this chapter is to introduce a new FCM-based forecasting model to predict univariate time series. FCMs as weighted knowledge-based models are used to do the task of forecasting by extracting knowledge in FTS models as Figure 3 represents. Thus, this chapter introduces a hybrid method, named HFCM-FTS, which combines HFCM and High Order Fuzzy Time Series (HOFTS), where the weight matrices associated with the state transitions are learned via the GA from the data. The objective of FCM is to find the weight matrices that model the causal relations among the concepts defined in the Universe of Discourse. Generating the rules in FTS by employing the FCM learning model is considered as the novelty of the approach in the literature.

The proposed model is compared with some recent methodologies in the literature Silva (2019) such as HOFTS, Weighted High Order FTS (WHOFTS), and Probabilistic Weighted FTS (PWFTS). The obtained results are competitive with regard to the influence of the basic elements on the forecasting accuracy including the number of concepts, activation function, and bias. Thus, we organized the rest of this chapter as follows: Section 3.1 introduces the proposed method in detail; Section 3.2 presents the experimental results and discussion; finally, the paper conclusion and some possibilities of future works are drawn in Section 3.3.

3.1 Proposed HFCM-FTS methodology

The main contribution of this section is to introduce our proposed prediction HFCM-FTS model. First of all, to fully comprehend the concept of the proposed methods, the FCM-FTS method is introduced as the first order of the HFCM-FTS method which is a combination of FCM and HOFTS (SILVA, 2019). Then, the proposed HFCM-FTS method is detailed as a mixture of HFCM and HOFTS methods.

3.1.1 FCM-FTS Model

FCM-FTS model is obtained considering $\Omega = 1$. What this really means is that the structure of the both FCM-FTS and HFCM-FTS models are the same and composed of some stages including partitioning, fuzzification, FCM learning method, defuzzification, and measurement of the accuracy. Hereby, firstly we focus on the FCM-FTS model subsequently the HFCM-FTS method will be explained by highlighting the differences.

As discussed earlier, apart from learning strategy, defining the appropriate structure of FCM is the basic step in time series modeling and forecasting. Figure 4 illustrates

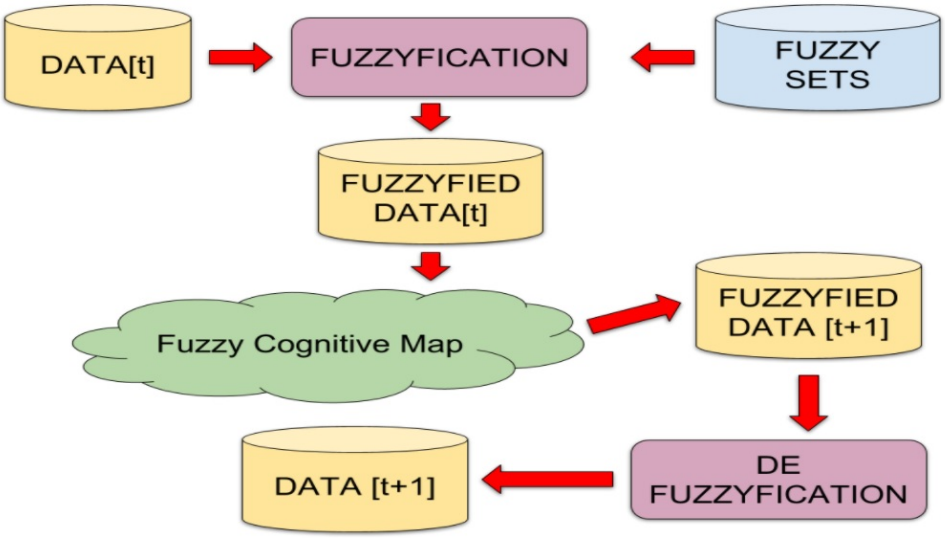


Figure 3 – Generic time series forecasting procedure using FCM

the structure of the FCM which is used in the FCM-FTS model. Figure 4-A displays a given partition of the universe of discourse and the definition of seven concepts, which is also the number of fuzzy sets because in our model the number of concepts and fuzzy sets are considered the same. As shown in Figure 4-C, FCM is a collection of concepts and causal interactions among these concepts as described in 2. According to Figure 4-B, this weight matrix is a square connection matrix. For an FCM with k concepts, the weight matrix is of size $k \times k$. Each weight saves the corresponding relationship between the source and the target concept, that is, for example, w_{05} presents the relation between source concept c_0 and target concept c_5.

It is clear from Figure 4 that the objective of FCM is to find the weight matrix for the concepts defined in the Universe of Discourse. These weights should be learned from the data. Regardless of the number of orders, the general procedure for the proposed method is divided into two stages: the Training Procedure and the Forecasting Procedure. Since GA is exploited to train the model, before diving into the FTS-FCM procedure, it is worth having a brief introduction to the GA algorithm and the steps involved.

Genetic Algorithm (GA)

Genetic Algorithm(GA) is a member of the evolutionary algorithm(EA) family on the basis of biological evolution (BE). The main aim of GA is to reproduce offsprings that are better than their parents. Thus, GA consists of some steps including parents selection, reproduction, mutation, and offsprings as detailed in the following:

1. **Initializing population (P)**: In GA, a population of candidate solutions for an optimization problem is evolved to a better solution. Each candidate of individuals(

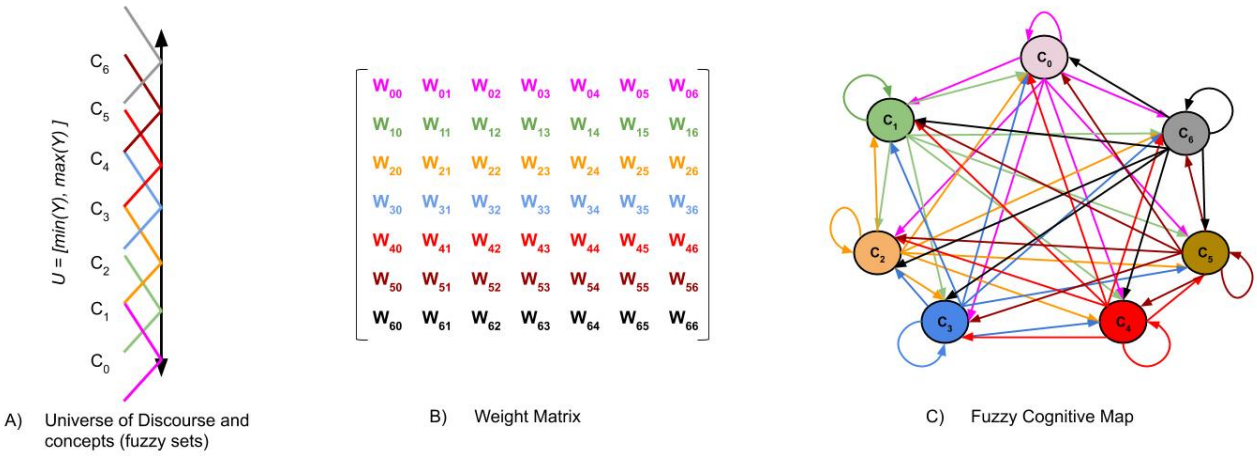


Figure 4 – Structure of FCM where the number of concepts and partitions are equal

phenotypes) contains a set of features that can be muted and changed. Solutions can be presented in binary or real-valued to the optimization problem. In this step, the population is generated randomly while its size depends on the nature of the problem.

2. **Selection of parents**: To breed a new generation, a portion of the randomly generated population over a consecutive generation. A fitness-based procedure is employed to select the population solutions. Random selection, tournament selection, and roulette wheel selection are the main techniques to select the best individuals for breeding.

3. **Crossover(*Pcrus*)**: In this step, new offspring is generated through overlapping, mixing, or swapping certain genes from both selected parents. It inherits the characteristics of both parents using common methods such as single-point crossover,two-point crossover, uniform crossover, etc.

4. **Mutation(*Pmut*)**: Mutation is employed to keep the diversity of genetics from one generation of a population of chromosomes to the next. In other words, since the parent chromosomes reproduced during the crossover step, there is no guarantee that the whole genes of the parents are copied. Thus, there exists an error. The value of *Pmut* defines the percentage of the child chromosome that undergoes mutation and it helps to solve the problem more efficiently.

5. **Evaluating the offsprings**: After mutation, the offsprings are evaluated with cost function to determine their fitness values. In each iteration, the best solution is replaced.

6. **Integrating offspring with the initial population**: To reproduce parents in the next generation, it is necessary to merge offspring. Then the new population is sorted based on the fitness score. Since the size of the population remains stable in

each some individuals with the worst fitness values are eliminated from the selection process to breed new offspring and the process continues.

FCM-FTS Procedures

In this section, the training and forecasting steps related to the proposed FCM-FTS model are described.

Training Procedure

The aim of the Training Procedure is to create the linguistic variable C with k concepts and find the set of $\Omega = 1$ weight matrices $\mathbf{W}^t$, given a crisp training set Y and the activation function f informed by the user. The steps of the method are listed below:

1. **Partitioning**: Split the Universe of Discourse, $U = [\min(Y), \max(Y)]$ into k even length and overlapped intervals, and for each interval, a fuzzy set C_i (i.e. concept) is defined with a membership function μ_{C_i}. The group of the k concepts forms the linguistic variable C, such that $C_i \in C$, $\forall i = 1..k$, as shown in Figure 4.

2. **Weight Matrix Learning**: Each matrix $\mathbf{W}^t$, for $t = 1..\Omega$, is a $k \times k$ matrix where $w_{ij}^t \in \mathbb{R}$ is the weight between the concepts C_i and C_j at the time lag t, and Ω is the order of the model. These weight matrices are trained using a Genetic Algorithm (GA) where the genotype is real encoded and simply contains the list of all w_{ij}^t matrix values.

The fitness of each genotype is the forecasting error, using the Root Mean Squared Error (RMSE) metric, using the training sample Y and the weight matrices represented by the genotype in the Forecasting step. Initially a population P with random individuals is created, such that $w_{ij}^t \sim \mathcal{N}(0, 0.1)$, $\forall i, j = 0..k, t = 0..\Omega$.

After the evaluation of the individuals, a percentage P_{sel} of the individuals are selected using tournament selection. Then a percentage P_{cross} of the population undergoes crossover and a percentage P_{mut} is chosen for random mutation. This process is repeated for G generations. This iterative process aims to minimize the fitness function (the RMSE) for the given training set Y.

It is worth noting that in the FCM-FTS model, only the first-order FCM ($\Omega = 1$) was considered. Thus there is only one weight matrix for times t as shown in Figure5.

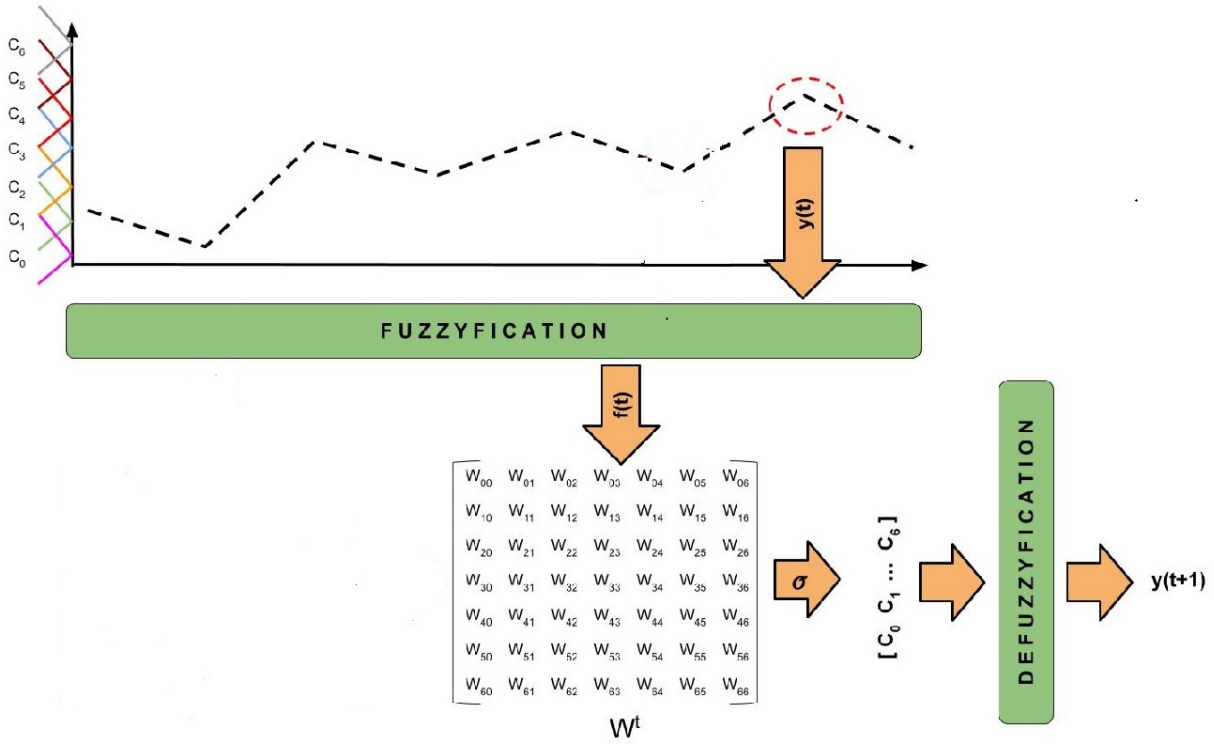


Figure 5 – The structure of the proposed FCM-FTS method.

Forecasting Procedure

The aim of the Forecasting Procedure is to estimate the crisp value $y(t+1)$ given the linguistic variable C, the weight matrices $\mathbf{W}^t$, the activation function f and a crisp input Y. The steps of the method are listed below:

1. **Fuzzification**: Given the crisp input sample Y with size T, each instance $y(t) \in Y, t = 1..T$, is transformed into an activation vector $\mathbf{a}(t)$ such that $a_i(t) = \mu_{C_i}(y(t))$, $\forall C_i \in C$, that is, each value $a_i(t) \in \mathbf{a}(t)$ corresponds to the membership degree of $y(t)$ to the concept C_i.

2. **Activation**: The state value for each concept in time $t+1$ can be defined by the following formula:

$$\mathbf{a}(t+1) = f\left(\mathbf{w}^0 + \mathbf{W}^t \cdot \mathbf{a}(t)\right) \tag{3.1}$$

where $\mathbf{w}^0$ is bias, $\mathbf{W}^t$ is the weight matrices at time t and $\mathbf{a}(t)$ represents the state value of each concept at time t. In addition to the sigmoid and tanh, as described in table 12, ReLU is also employed in this study which can be defined by the following formula:

$$ReLU(x) = \max(0, x) \tag{3.2}$$

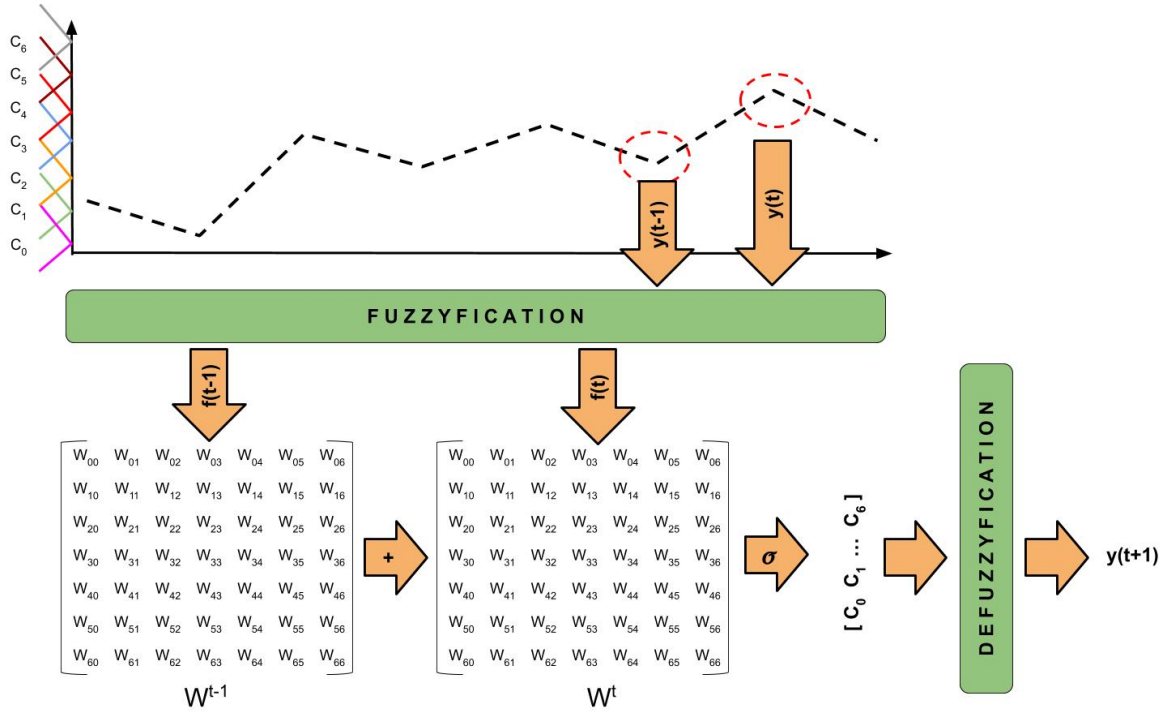


Figure 6 – The structure of the proposed HFCM-FTS method.

3. **Defuzzification**: After calculating the activation degree of each concept, in this step, the defuzzification is carried out to produce the forecasts. Equation (3.3) defines the forecast produced by FCM-FTS:

$$\hat{y}(t+1) = \frac{\sum_{i=1}^{k} a_i(t+1) \cdot mp_i}{\sum_{i=1}^{k} a_i(t+1)} \tag{3.3}$$

where $a_i(t+1)$ is the activation calculated from the previous step for each concept at time $t+1$ and mp_i is the center of each concept C_i. With this equation, the forecast value at time $t+1$ is obtained in numeric terms.

3.1.2 HFCM-FTS method

The HFCM-FTS method is a combination of HFCM and HOFTS as a high-order version of the FCM-FTS model using similar structures and procedures. Since the order is selected as $\Omega = 2$ in our model, there are two weight matrices in time t and $t-1$ as depicted in Figure 6. It means that, GA must train $\mathbf{W^t}$ and $\mathbf{W^{t-1}}$ not only the weight matrices at time step t. Accordingly, the activation step in the forecasting procedure alters to compute the activation state of each concept at time step $t+1$ according to the modified equation as follows.

$$\mathbf{a}(t+1) = f\left(\mathbf{w}^0 + \sum_{j=1}^{\Omega} \mathbf{W}^{t-j+1} \cdot \mathbf{a}(t-j+1)\right) \tag{3.4}$$

For $\Omega = 2$, the equation is rewritten as follows:

$$\mathbf{a}(t+1) = f\left(\mathbf{w}^0 + \mathbf{W}^t \cdot \mathbf{a}(t) + \mathbf{W}^{t-1} \cdot \mathbf{a}(t-1)\right) \tag{3.5}$$

The predicted value is obtained by replacing the Equation 3.5 in Equation 3.3. Noticeable that the grid partitioning is used to generate versions of the FCM with $k = \{5, 10, 20\}$ as well as triangular membership function μ_{C_i}. Root Mean Square Error (RMSE), described in 3.6, is employed as the accuracy metric to evaluate the performance of the HFCM-FTS model.

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_i(t) - \hat{y}_i(t)\right)^2} \tag{3.6}$$

3.2 Computational Experiments

(A) **SONDA Dataset**:

SONDA - *Sistema de Organização Nacional de Dados Ambientais* (Brazilian National System of Environmental Data Organization), is a governmental project that groups environmental data (solar radiance, wind speed, precipitation, etc.) from INPE - Instituto Nacional de Pesquisas Espaciais (Brazilian Institute of Space Research). This dataset was retrieved directly from the SONDA Project page at ⟨http://sonda.ccst.inpe.br/⟩[1].

Since the HFCM-FTS model is designed to predict univariate time series, to test the utility of the proposed method, in this section, we apply it to solar radiation time series data [2]. Noteworthy that the minimum, maximum, average and standard deviation of the proposed solar radiance time series are $\{-6.0667, 1228.65, 223.261, 311.239\}$, respectively.

In this experiment, as shown in Figure 7, 8,000 samples have been used, with a sliding window of 2,000 samples in the cross-validation method. 80% of the window for training and 20% for test.

The number of concepts (partitions), the type of activation function, and the presence of bias terms are the main parameters affecting the accuracy of the proposed model.

[1] Access in 19/05/2019

[2] Available at ⟨https://query.data.world/s/2bgegjggydd3venttp3zlosh3wpjqj⟩ accessed on April 4, 2020

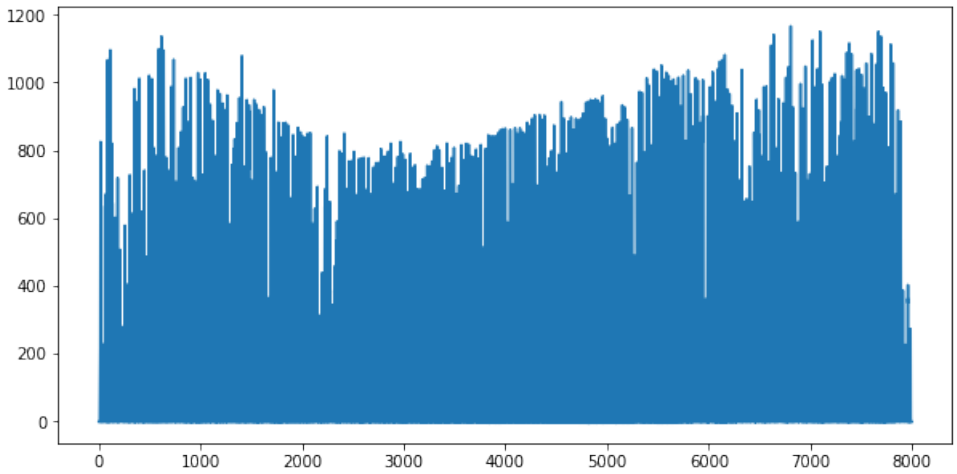


Figure 7 – Samples of solar radiance time series

Thus, in sections(B) and(C), we assess the importance of the bias term and test different types of activation functions, respectively. Finally, in Section(D), we compare the proposed method with state-of-the-art fuzzy time series methods.

(B) **The influence of the bias term**: To assess the effect of the bias term, we execute the proposed methodology with $5, 10$, and 20 concepts, with and without the bias terms. In this experiment, the sigmoid was used as the activation function and since the learning method is stochastic (a GA), each configuration was tested 15 times.

Table 5shows the number of parameters for each configuration, the best, the average, and the worst RMSE of the 15 independent runs. It can be seen that adding bias improves the accuracy of the proposed model. Without the bias term, a null activation vector at time $t-1$ can produce activations greater than 0 at time t which, in turn, should not make sense. It must be noticed, however, that the number of parameters increases when compared to the models without bias.

k	*Parameters* $(-bias)$	*Min.RMSE* $(-bias)$	*Ave.RMSE* $(-bias)$	*Max.RMSE* $(-bias)$	*Parameters* $(+bias)$	*Min.RMSE* $(+bias)$	*Ave.RMSE* $(+bias)$	*Max.RMSE* $(+bias)$
5	50	146.332	152.939	158.968	60	**111.207**	**119.565**	**132.957**
10	200	183.894	193.658	198.827	220	**141.576**	**162.576**	**184.484**
20	800	223.537	231.972	240.436	840	**172.900**	**183.355**	**196.890**

Table 5 – The effect of the bias on the accuracy of the proposed HFCM-FTS method with Sigmoid activation function

k	**Min. RMSE** *ReLU*	**Ave. RMSE** *ReLU*	**Max. RMSE** *ReLU*	**Min RMSE** *sigmoid*	**Ave. RMSE** *sigmoid*	**Max. RMSE** *sigmoid*	**Min. RMSE** *tanh*	**Ave. RMSE** *tanh*	**Max. RMSE** *tanh*
5	**105.886**	**119.191**	**131.393**	111.207	119.565	132.957	110.802	128.726	139.687
10	**136.60**	**157.401**	**171.147**	141.567	162.576	184.484	153.032	164.789	188.700
20	**166.922**	**182.044**	**189.306**	172.900	183.355	196.890	173.340	183.828	199.511

Table 6 – Performance of the proposed HFCM-FTS method using different activation functions when bias has been considered

k	HFCM-FTS Parameters	HFCM-FTS Min.RMSE	HFCM-FTS Ave.RMSE	HFCM-FTS Max.RMSE	HOFTS	WHOFTS	PWFTS
5	60	105.886	119.191	131.393	368.42	182.98	149.79
10	220	136.60	157.401	171.147	288.51	146.15	134.91
20	840	166.922	182.044	189.306	204.18	134.43	134.22

Table 7 – Comparison of the accuracy of the proposed HFCM-FTS by considering ReLU activation function and bias weights with other methods

GA Hyperparameters	*Pcrus*	*Pmut*	*Population*	*Generation*
	0.5	0.3	50	30

Table 8 – Hyper-parameters for GA learning algorithm

(C) **The influence of the activation function**:

In Table 6, we compare the accuracy of the model using different activation functions. In addition to varying the activation function, we vary the number of concepts, running the model with 5, 10, and 20 sets. In this experiment, all the models include a bias term. The results indicate that when the model uses ReLU, on average, the accuracy is slightly better.

(D) **Comparison with other Fuzzy Time Series methods**

In this section, we compare the proposed method with other fuzzy time series forecasting methods available in the pyFTS library (SILVA et al., 2019). In particular, the proposed method is tested against HOFTS (High Order Fuzzy Time Series), WHOFTS (Weighted High Order Fuzzy Time Series), and PWFTS (Probabilistic Weighted Fuzzy Time Series). PWFTS showed the best-performing method in a multitude of experiments against many statistic forecasting methods, other FTS, and machine learning methods (SILVA et al., 2020b). Since GA was used as a learning method, and it is a stochastic optimization algorithm, the HFCM-FTS method was executed 15 times, with the GA hyper-parameters defined according to Table8.

Table 7 presents the results considering bias terms, the ReLU activation function, and sliding window cross-validation for various numbers of concepts (partitions) in comparison with the aforementioned methods. The results show the superiority of HFCM-FTS, which is a mixture of HOFTS and FCM, over HOFTS. This indicates that adding FCM to HOFTS indeed improves the methodology. The results also show that the proposed method is superior to all the other methods when the number of sets was 5. On the other hand, as the number of concepts increases the HFCM-FTS performance deteriorates. Meanwhile, the performance of the other methods improves as the number of concepts increases.

We hypothesize that the decrease in the performance of HFCM-FTS is due to the lack of convergence of the learning algorithm. We suspect that the GA settings were

not adequate to train the model for such a high number of parameters.

Nevertheless, the results of HFCM-FTS were quite competitive with the other methods. It is noticeable that the best results achieved by WHOFTS and PWFTS required a partition with 20 concepts, with a higher number of rules, while the average results of HFCM-FTS with 5 concepts were significantly better.

3.3 Chapter's Highlights

In this chapter, we developed a hybrid univariate FCM-based forecasting model termed HFCM-FTS combining HFCM and HOFTS trained via GA. It also The paper also covers the influence of three elements on the accuracy of the model including the number of concepts, activation function, and bias. The results are still limited and more experiments are required to assess the efficacy of the method. Nevertheless, in the given scenario the proposed method presented competitive results when compared to other fuzzy time series methods in particular when the number of concepts is limited. It is important to highlight that keeping the number of concepts small may positively affect the readability and interpretability of the model.

3.3.1 Model's limitation and challenges

Despite the success of the HFCM-FTS model when the number of concepts is low, the proposed training methodology will get very time-consuming as this number increases. For instance, when the number of concepts is 30 the GA will have to solve an 1860 variable problem. Increasing this number by only 10, from 30 to 40 concepts, the GA will have to solve a 3280 variable problem. Besides, this issue will certainly be more acute if the model is used to predict a multivariate FTS. Thus, to tackle the aforementioned problem, finding an alternative to GA, which is not very efficient, is highly recommended.

4 PROPOSED R-HFCM TECHNIQUE

As we discussed, the HFCM-FTS model is very time-consuming due to the high number of parameters that must be adjusted. In this context, the main objective of this chapter is to reduce this cost by introducing a univariate FCM-based time series forecasting model inspired by the Echo State Networks (ESNs) reservoir computing. In the ESNs, the weights in the hidden layer are assigned randomly and are not trainable. Thus, the training cost can be significantly decreased.

More specifically, the concepts of High Order FCM (HFCM) and ESNs are merged to produce a new forecasting technique, called, Randomized HFCM (R-HFCM). The proposed model consists of three layers: the input layer, the reservoir, and the output layer. The reservoir layer consists of a group of HFCM-FTS (ORANG et al., 2020) but in this case, the weights are randomly initialized such that the Echo State Property (ESP) condition in ESN reservoir computing is satisfied. Then, each sub-reservoir generates its output independently after feeding input into each sub-reservoir separately. Finally, the least squares minimization technique is applied to train the output layer and generate the final predicted value. As the computational experiments reveal, in addition to improving the accuracy, the proposed model is much faster than the HFCM-FTS method.

It is important to highlight that ESNs and various ESN-based methods with different structures using multiple sub-reservoirs have been successful in many applications (YAO; WANG, 2019; LI; LI, 2019). Also, there are in the literature references that introduce a combination of ESNs and fuzzy logic systems (MAHMOUD et al., 2021; ZHANG et al., 2020; HAN; LEE, 2014). For instance, the authors in Mahmoud et al. (2021) introduced a new hybrid model referred to as Takagi-Sugeno-Kang (TSK) fuzzy ESN (TSKFESN) which incorporates the advantages of the TSK fuzzy inference system and the reservoir computing of ESN with multiple reservoirs in a unified network. In addition, in (ZHANG et al., 2020) a novel deep fuzzy ESN (DFESN) based on a Single-Layer Fuzzy ESN(SFESN) was presented. Although there are similarities between these models and the R-HFCM including the consideration of sub-reservoirs, integration of fuzzy logic, and fast training, the general architecture of these models is quite different from R-HFCM. Besides, ESNs are employed to generate sub-reservoirs in TSKFESN and DFESN. On the other hand, the randomized HFCMs are the basis of the structure of each sub-reservoir in R-HFCM. In addition, DFESN has a sequential structure in which there are links among successive sub-reservoirs whereas R-HFCM includes parallel frames with no links between sub-reservoirs. Accordingly, this study presents R-HFCM as a combination of reservoir computing (ESN) and HFCM for the first time in the literature.

The main contributions of this research can be summarized as follows:

1. The development of a forecasting model using HFCM in an ESN model. This leads to a hybrid fuzzy time series forecasting model with effective learning achieved using ESN;

2. A study of the effect of four hyperparameters including the number of concepts, the size of the reservoir, the model order, and the type of activation function on the performance of the proposed R-HFCM model; and

3. The validation of R-HFCM in ten univariate time series datasets. The experimental results demonstrate that R-HFCM is efficient and accurate in comparison to the other existing approaches.

The remainder of this chapter is organized as follows: Section 4.1 presents a brief description of Echo State Network; Section 4.2 introduces the proposed method in detail; Section 4.3 presents the experimental results and discussion; finally, some highlights and possibilities of future works are drawn in Section 4.4.

4.1 Echo State Network

The ESN was proposed by Jaeger (2001) as an improved Recurrent Neural Network (RNN). Standard RNNs have been considered as a powerful tool to simulate complex dynamic systems. However, training algorithms for RNNs involve some downsides including relatively high computational costs, potentially slow convergence, and vanishing gradients (ZHENG et al., 2020). With the goal of handling these issues, the approach known as Reservoir Computing (RC) was proposed to focus on the initialization conditions and stability instead of the training algorithm. The ESNs take advantage of this structure and consist of three basic components: (i) an input layer, (ii) a large recurrent hidden layer, the dynamical reservoir, with fixed sparse hidden-to-hidden connections, (iii) and an output layer.

Differently from traditional RNNs, the dynamical reservoir or hidden layer in ESNs is untrainable and only the output weights are trained. Figure 8 shows the structure of traditional ESN with M input units, N reservoir units and L output (readout) neurons, while $u(t) = (u_1(t), \ldots, u_M(t))^T, x(t) = (x_1(t), \ldots, x_N(t))^T$, and $y(t) = (y_1(t), \ldots, y_L(t))^T$ determine their activation at each time step. The output of dynamic reservoir neurons and the output of ESN at time step $t+1$ are updated according to the equations (4.1) and (4.2), respectively.

$$x(t+1) = f(\mathbf{W^{in}} \cdot u(t+1) + \mathbf{W^{res}} \cdot x(t) + \mathbf{W^{f_b}} \cdot y(t)) \tag{4.1}$$

$$y(t+1) = f^{out}(\mathbf{W^{out}}[x(t+1);\ u(t+1)]) \tag{4.2}$$

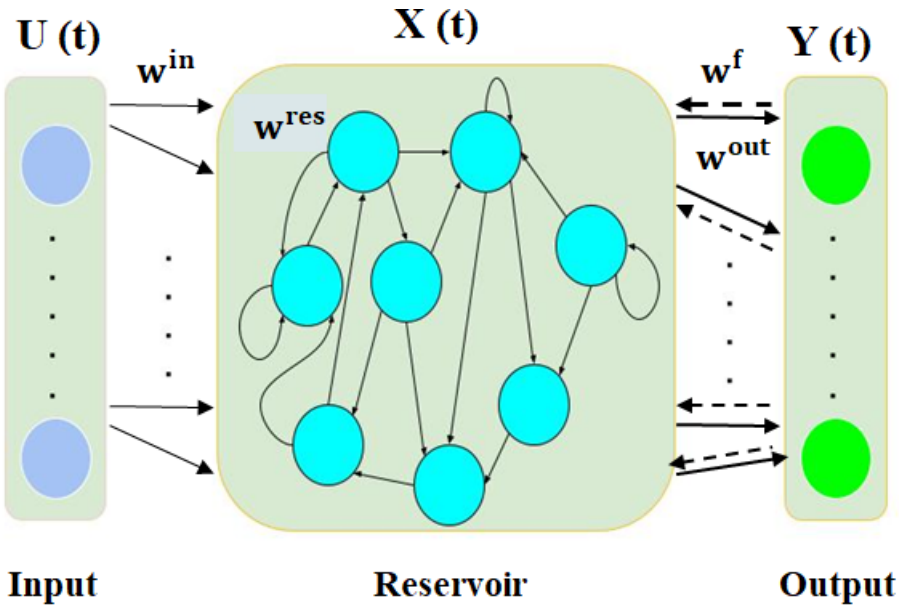


Figure 8 – Structure of Traditional ESN

where the $N \times M$ input weight matrix $\mathbf{W^{in}}$ represents the connections between the input units and the reservoir neurons; the internal connections among neurons inside the reservoir layer are represented by the $N \times N$ weight matrix $\mathbf{W^{res}}$; the output weight matrix is represented by $\mathbf{W^{out}}$ with $L \times (M + N)$ dimension. Finally, an $N \times L$ weight matrix represents the forward and backward connections between the readout neurons and the reservoir units. They are represented by the $\mathbf{W^{fb}}$ and the $\mathbf{W^{back}}$ matrices, respectively.

Interestingly, both internal and input weights are initialized randomly with no changes during the training and testing process. Thus, an ESN is trained through a supervised learning method in two stages. At the first level, the M dimensional inputs are mapped into a high dimensional reservoir state to reach the echo states x. Finally, the least squares method is used to fit the output weights.

Initialization and Hyper-parameters

An ESN has several hyper-parameters, such as the reservoir size, N, the input scale, IS, the spectral radius, ϵ, and the reservoir sparsity γ. The input scale, IS, is used to initialize the hidden weight matrix $\mathbf{W^{in}}$ such that each member of $\mathbf{W^{in}}$ obeys the uniform distribution in $[-IS, IS]$. γ determines the proportion of non-zero elements in $\mathbf{W^{res}}$ and ϵ is the spectral radius of $\mathbf{W^{res}}$, which must be set smaller than 1 (MA; SHEN; COTTRELL, 2017).

As mentioned before, just the readout weight matrices are trainable while the $\mathbf{W^{res}}$ is selected randomly with a uniform distribution before training execution to provide the requirements of echo state condition (JAEGER, 2001; MA; SHEN; COTTRELL, 2017).

The initialization of $\mathbf{W^{res}}$ is done with the following procedure to ensure that the maximum absolute eigenvalue or the spectral radius is less than one. Firstly an internal random weight matrix $\mathbf{W^{rand}}$ will be generated. Then it is scaled to meet the ESP condition described by the equation (4.3).

$$\mathbf{W^{res}} = \frac{\epsilon \cdot \mathbf{W^{rand}}}{\rho_{max}(\mathbf{W^{rand}})} \tag{4.3}$$

where $\rho_{max}(\mathbf{W^{rand}})$ denotes the maximum of the absolute value of the eigenvalues of the matrix $\mathbf{W^{rand}}$ when its elements are generated randomly. To guarantee the stability of the ESN, ϵ must be set smaller than 1 (JAEGER, 2001). The final step is to determine $\mathbf{W^{out}}$ which is computed with the equation (4.4).

$$\mathbf{W^{out}} = (X^T X)^{-1} X^T Y \tag{4.4}$$

where $(X^T X)^{-1}$ denotes the inversion of square matrix $X^T X$.

4.2 Proposed R-HFCM method

This section introduces the Randomized HFCM (or R-HFCM) which consists of a group of randomized HFCM-FTS models to predict univariate time series. The innovation of the proposed R-HFCM model is the integration of concepts from the FCMs and ESN reservoir computing method exploiting the least-squares regression algorithm as the learning strategy.

From a structural perspective, this model consists of three layers, as illustrated in Figure 9. The input layer, the reservoir layer, and the output layer. The reservoir is a group of random HFCM-FTS models fed only by the external input time series. Then, as Figure 9 shows, the obtained independent outputs from each sub-reservoir (represented by $\hat{y}_1(t+1)$,$\hat{y}_2(t+1)$,...,$\hat{y}_n(t+1)$) are fed to the output layer. As shown in Figure 9, the reservoir structure consists of a number of sub-reservoirs (N_{SR}) such that there are no links among them. This means that the proposed model does not have a hierarchical stacking structure similar to deep learning models.

The structure of each sub-reservoir in the proposed R-HFCM model is an HFCM-FTS model with random weights. The training of the R-HFCM model is mainly concentrated on finding the output layer coefficients using least squares regression. Thus, each randomized HFCM-FTS unit provides its output with different dynamics from the input. Afterward, the output layer is utilized to make a connection between the input and the outputs from each sub-reservoir.

From another perspective, it can be said that the proposed R-HFCM is a type of ESN in which only the output layer is trainable, while the reservoir parameters are

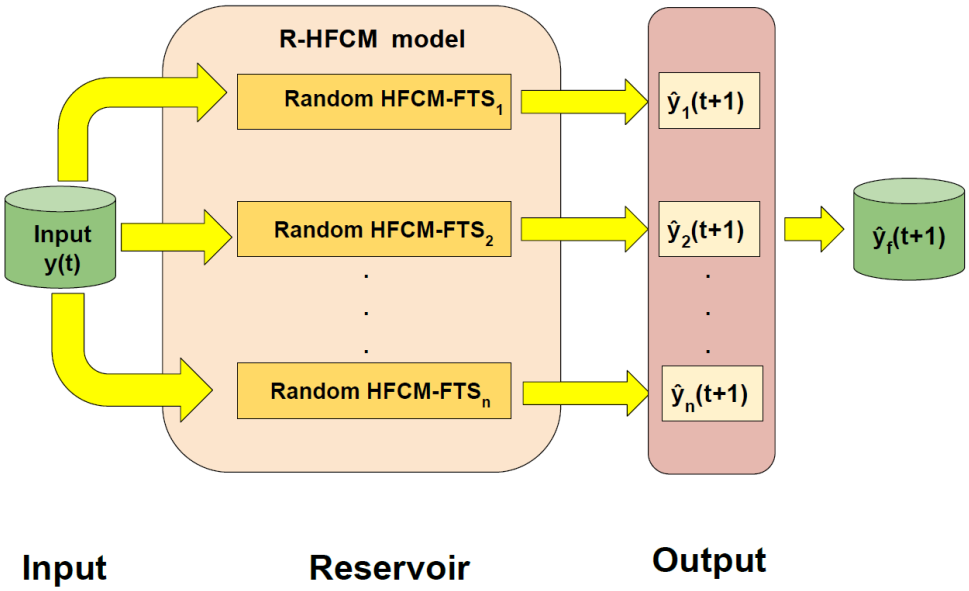


Figure 9 – Generic structure of the proposed R-HFCM method.

initialized randomly and remain unchanged in the training process. The weights of each sub-reservoir in the reservoir layer are chosen randomly to meet the ESP condition in ESN and then each sub-reservoir generates its output independently. It means that, in our model, the reservoir weights are initialized based on the ESP condition in the ESN reservoir computing, and then the least squares minimization algorithm is applied to the output units to generate the final prediction.

The proposed R-HFCM method is divided into two procedures: the Training procedure and the Forecasting procedure, detailed in sections 4.2.1 and 4.2.2.

4.2.1 Training Procedure

The main objective of the training procedure is to find appropriate weights for the output layer given a crisp training set Y. It means that the training dataset is fed to each sub-reservoir to generate the inputs for the output layer. Then, the least-squares algorithm is used to find the optimal least-squares coefficients. The training procedure is divided into two steps: constructing the structure of each sub-reservoir and determining the least squares coefficients.

4.2.1.1 Randomized Model Initialization

Each matrix $\mathbf{W}^t$, for $t = 1, \ldots, \Omega$, is a $k \times k$ matrix where, $w_{ij}^t \in \mathbb{R}$ is the weight between the concepts C_i and C_j at the time lag t, and Ω is the order of the model. As mentioned earlier, the innovation in the proposed model is that the weight matrices of each HFCM-FTS are randomly chosen from a uniform distribution over an interval [-1,1].

Then, they are scaled according to the ESN reservoir computing to preserve the ESP condition as described by the equation (4.5).

$$\mathbf{W}^t = \mathbf{W}^{\mathbf{rand}} \cdot \left(\frac{\epsilon}{\rho_{\mathbf{max}}(\mathbf{W}^{\mathbf{rand}})} \right) \tag{4.5}$$

where $\rho_{\mathbf{max}}(\mathbf{W}^{\mathbf{rand}})$ is the maximum of the absolute value of the eigenvalues of $\mathbf{W}^{\mathbf{rand}}$, $\mathbf{W}^{\mathbf{rand}}$ is selected randomly with a uniform distribution in the range [-1,1] and $\epsilon \in (0, 1)$ is the desired spectral value (scaling parameter). Also, since the members of the weight matrix in the FCM must be in the range [-1,1], the value of ϵ in our model is equal to 0.5 to satisfy the corresponding condition. Moreover, it is worth reminding that the condition $\rho_{\mathbf{max}}(\mathbf{W}^{\mathbf{rand}}) < 1$ is not necessary, but sufficient for the activation functions applied in the article (tanh, sigmoid, ReLU, softmax).

$\mathbf{w^0}$ is the bias weight vector in equation (**??**) which is initialized based on the equation (4.6).

$$\mathbf{w}^0 = \mathbf{w}^{\mathbf{0}}_{\mathbf{rand}} \cdot \left(\frac{\epsilon}{\mathbf{S}} \right) \tag{4.6}$$

where the value of ($\mathbf{S}$) represents the maximum singular value of $\mathbf{w}^{\mathbf{0}}_{\mathbf{rand}}$, $\mathbf{w}^{\mathbf{0}}_{\mathbf{rand}}$ is selected randomly with a uniform distribution in the range [-1,1] and $\epsilon = 0.5$.

4.2.1.2 Partitioning

Since R-HFCM is composed of a group of HFCM-FTS models, they must share the same linguistic variables, i.e., the group of fuzzy sets defined over the Universe of Discourse $U = [\min(Y), \max(Y)]$.

The partitioning is the process of splitting U into k even length, evenly spaced, overlapped intervals, where, mp_i, $i = 1..k$, are the midpoints of each interval. Then, a fuzzy set C_i (i.e. a concept) is defined with midpoint mp_i and a membership function μ_{C_i}. In each sub-reservoir, a group of the k concepts form the linguistic variable C, such that $C_i \in C$, $\forall i = 1, \ldots, k$. Here grid partitioning is used and the number of concepts is the same as the number of partitions.

Figure 4 shows a simple example of the FCM structure used in this model with triangular membership functions μ_{C_i}. In Figure 4-A the universe of discourse is partitioned to generate seven concepts, which is also the number of fuzzy sets. As shown in Figure 4-C, the FCM is a collection of nodes (concepts) and interactions among these concepts represented by the weight matrix. As can be seen in Figure 4-B, this weight matrix is a square connection matrix defined randomly.

4.2.1.3 Fuzzification

The fuzzification step aims to convert the crisp time series Y into a fuzzy series A which represents the activation state $a_i(t)$ of each concept $C_i \in C$ for each time series sample. Then, for each sample $y(t) \in Y$ the corresponding fuzzified sample $a(t) \in A$ is the set of activations $a_i(t) \in a(t)$ such that $a_i(t) = \mu_{C_i}(y(t))$, for $i = 1 \dots k$.

4.2.1.4 Activation

For each HFCM-FTS with index $j = 1 \dots N_{SR}$, the state value of each concept in time $t+1$ can be defined using the equation (4.7).

$$\mathbf{a}_j(t+1) = f\left(\mathbf{w}^0 + \sum_{l=1}^{\Omega} \mathbf{W}^l \cdot \mathbf{a}(t-l+1)\right) \tag{4.7}$$

4.2.1.5 Defuzzification

After calculating the activation level of each concept, in this step, the defuzzification is carried out to produce the output related to each sub-reservoir. Therefore, the forecast values for each sub-reservoir $j = 1 \dots N_{SR}$ at time $t+1$ in numeric terms can be calculated via the equation (4.8).

$$\hat{y}_j(t+1) = \frac{\sum_{i=1}^{k} a_{ji}(t+1) \cdot mp_i}{\sum_{i=1}^{k} a_{ji}(t+1)} \tag{4.8}$$

where $a_j(t+1)$ is the activation calculated from the previous step for each concept at time $t+1$ and mp_i is the center of each concept $C_i \in C$.

4.2.1.6 Least Squares coefficients determination

Given the outputs $y_j(t+1)$, $j = 1 \dots N_{SR}$ for each input sample $y(t) \in Y$, $t = 1 \dots T$, a design matrix $X \in \mathbb{R}^{N_{SR} \times T}$ is created to represent the linear system $Y = \lambda X$, where $\lambda = [\lambda_0, \dots, \lambda_j]$ is the coefficient vector. Then, the Least Squares method is employed to solve this linear system and find the λ vector of coefficients that minimizes the Mean Squared Error.

Figure 10 shows a simple structure of the R-HFCM model considering $N_{SR} = 2$ while in each sub-reservoir the order is $\Omega = 2$. Thus there are two weight matrices for times t and $t-1$ for each randomized HFCM-FTS as indicated in the figure. It is worth noting that the model contains the bias weights while they are discarded from Figure 10 just because of ease of notation.

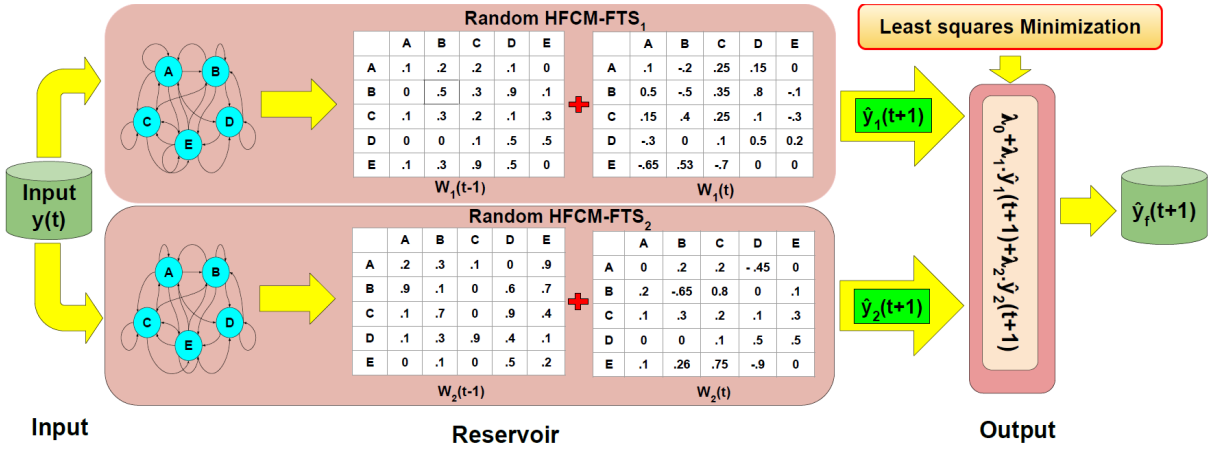


Figure 10 – The simple example of the proposed model mechanism considering $N_{SR} = 2$.

4.2.2 Forecasting Procedure

The main goal of this stage is to compute the predicted crisp values $\hat{y}(t+1)$ of each sub-reservoir as well as the final predicted value $\hat{y}_f(t+1)$, given the linguistic variable C, the weight matrices $\mathbf{W}^t$, the activation function f and a crisp input $Y(t)$. The forecasting process includes some stages which are fully explained in the following.

4.2.2.1 Fuzzification

The same as Step 3 in the Training procedure.

4.2.2.2 Activation

The same as Step 4 in the Training procedure.

4.2.2.3 Defuzzification

1. Sub-reservoir forecasts defuzzification: The same as Step 5 in the Training procedure.

2. Final forecast defuzzification:

   The predicted value at time $t+1$ is computed through the linear combination of the obtained outputs of all sub-reservoirs and the least squares coefficients. The final predicted value is described through the equation (4.9).

$$\hat{y}_f(t+1) = \lambda_0 + \sum_{j=1}^{N_{SR}} \lambda_j \cdot \hat{y}_j(t+1) \tag{4.9}$$

   For instance, as highlighted in Figure 10, the final output is estimated via the equation (4.10) when $N_{SR} = 2$:

$$\hat{y}_f(t+1) = \lambda_0 + \lambda_1 \cdot \hat{y}_1(t+1) + \lambda_2 \cdot \hat{y}_2(t+1) \tag{4.10}$$

where $\{\lambda_0, \lambda_1, \lambda_2\}$ denote the obtained least square coefficients from the training procedure and $\hat{y_1}(t+1)$ and $\hat{y_2}(t+1)$ are the generated outputs for the first and second sub-reservoirs respectively according with equation (4.8).

As shown in Figure 10, the number of least squares coefficients directly depends on the number of sub-reservoirs N_{SR}. To be more precise, the number of least squares coefficients in this model is equal to $N_{SR}+1$. Thereby, for $N_{SR}=n$, the final predicted value is described by the equation (4.11) using $n+1$ least-squares coefficients.

$$\hat{y_f}(t+1) = \lambda_0 + \lambda_1 \cdot \hat{y_1}(t+1) + \lambda_2 \cdot \hat{y_2}(t+1) + ... + \lambda_n \cdot \hat{y_n}(t+1) \tag{4.11}$$

The computational complexity of proposed methods can be analyzed together, once the most computationally expensive steps (the fuzzification and the activation) are the same in both. Then, given an input sample size T, the number of operations in each method scales linearly to the number of reservoirs N_{SR}, the order Ω of each reservoir, and the square of the number of concepts k, because the size of the squared matrices $W^t \in \mathbb{R}^{k \times k}$, such that the complexity cost is $O(N_{SR} \times \Omega \times k^2 \times T)$. It is worth noting that the training procedure accounts for an additional cost of $O((N_{SR} \times T)^3)$ due to the Least Squares Minimization.

4.3 Computational Experiments

In this section, the accuracy of the proposed model is evaluated and compared against other models available in the literature. Firstly, the dataset descriptions and experimental setup are briefly introduced. Then, the effectiveness of the proposed R-HFCM model in time series forecasting is assessed on hourly electrical and solar energy datasets.

All the computational experiments were implemented with Python 3.6.12 using open-source packages such as Scikit-learn, Pandas, Numpy, and pyFTS. The Python code of the model is publicly accessible for research replication via the link address given as follows: ⟨https://github.com/OMIDUFMG2019/RHFCM⟩.

4.3.1 Datasets

To evaluate the effectiveness of the proposed R-HFCM model, in this section, ten univariate time series from five datasets are used including the Brazilian SONDA

dataset, the Malaysian electrical load dataset, the solar energy dataset from the United States National Renewable Energy Laboratory (NREL), the electrical load dataset from Global Energy Forecasting Competition 2012 (GEFCom 2012) and PJM hourly energy consumption, which are shown below. Table 9 presents information about the minimum, maximum, average, and standard deviation of the chosen time series.

Table 9 – Summary statistics of the ten-time series datasets used in the experiments

Time series	Minimum	Maximum	Average	Standard deviation
SONDA (Brazil)	-5.355	1166.616	220.244	307.194
Malaysia (load)	19088.000	75447.000	3931.976	4346.684
NREL (DHHL_1)	-0.358	1126.881	223.768	295.962
NREL (DHHL_2)	-0.369	1129.417	226.168	299.341
NREL (DHHL_3)	-0.472	1164.598	229.575	306.113
GEFCom2012 (zone1)	8688.000	39584.000	17477.747	5189.690
GEFCom2012 (zone2)	82672.000	270013.000	163801.625	33478.453
GEFCom2012 (zone3)	89204.000	291344.000	176742.299	36123.321
PJM (AEP)	9823.000	24015.000	15657.673	2594.208
PJM (DEOK)	1870.000	5445.000	3056.217	640.022

4.3.1.1 SONDA Dataset

SONDA - *Sistema de Organização Nacional de Dados Ambientais* (Brazilian National System of Environmental Data Organization), is a governmental project that groups environmental data (solar radiance, wind speed, precipitation, etc) from INPE - Instituto Nacional de Pesquisas Espaciais (Brazilian Institute of Space Research). This dataset was retrieved directly from the SONDA Project [1]. Since the R-HFCM model has been designed to predict univariate time series, to test the utility of the proposed method, we apply the model to predict solar radiation time series data[2] (glo_avg variable in Table 10). In this experiment, 8,000 samples of the data have been used as shown in Figure 11.

Table 10 – SONDA dataset variables

Variable	Type	Description
DateTime	Time Stamp	yyyy-MM-dd HH:MM
glo_avg	Real	Global average solar radiation
ws_10m	Real	Wind speed in meters by second (m/s)

4.3.1.2 Malaysia Dataset

According to Table 11, the Malaysia dataset includes the hourly electric load of the power supply company of the city of Johor in Malaysia sampled between 2009 and

[1] ⟨http://sonda.ccst.inpe.br/⟩
[2] ⟨https://query.data.world/s/2bgegjggydd3venttp3zlosh3wpjqj⟩ accessed on April 4th, 2020

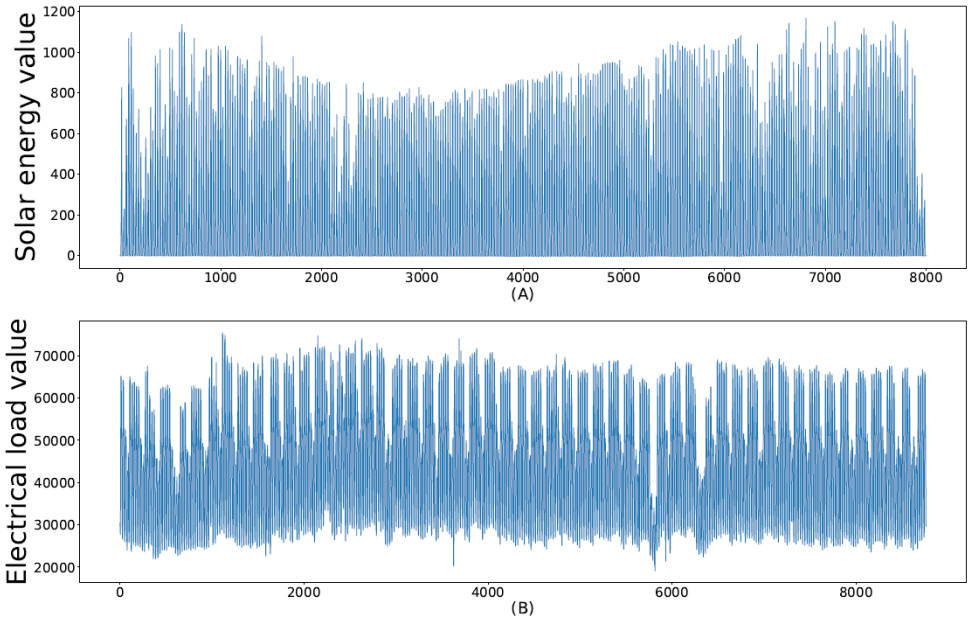


Figure 11 – Plotting of (A) 8,000 Samples of Brazilian solar radiance and (B) 8,760 samples of hourly electric load Malaysia dataset

2010, with 17,519 (ORANG et al., 2022). Figure 11 shows 8,760 samples (the year 2009) of the referred hourly load time series used in this chapter.

Table 11 – Malaysia dataset variables

Variable	**Type**	**Description**
DateTime	Time Stamp	yyyy-MM-dd HH:MM
temperature	Real	Temperature in Celsius degrees (^{o}C)
load	Integer	Electric load in Mega Watts by hour (MW/h)

4.3.1.3 NREL solar energy dataset

The solar energy dataset was obtained from the United States National Renewable Energy Laboratory (NREL)[3]. Among the solar stations, three of them are chosen as the case studies in this paper including DHHL_1 (Latitude: 21.31533, Longitude: -158.087), DHHL_2 (Latitude: 21.31451, Longitude: -158.08534) and DHHL_3 (Latitude: 21.31236, Longitude: -158.08463). There exist 38,766 samples per each site from 2010-06-01 to 2011-07-31. As Figure 12 shows, 8,000 samples of hourly solar energy (in w/m^2) from the selected time series are employed to evaluate the performance of the R-HFCM model. Accordingly, the original time series is resampled from 15 minutes to hourly before being injected into the model.

[3] ⟨https://midcdmz.nrel.gov/apps/sitehome.pl?site=OAHURS1M⟩

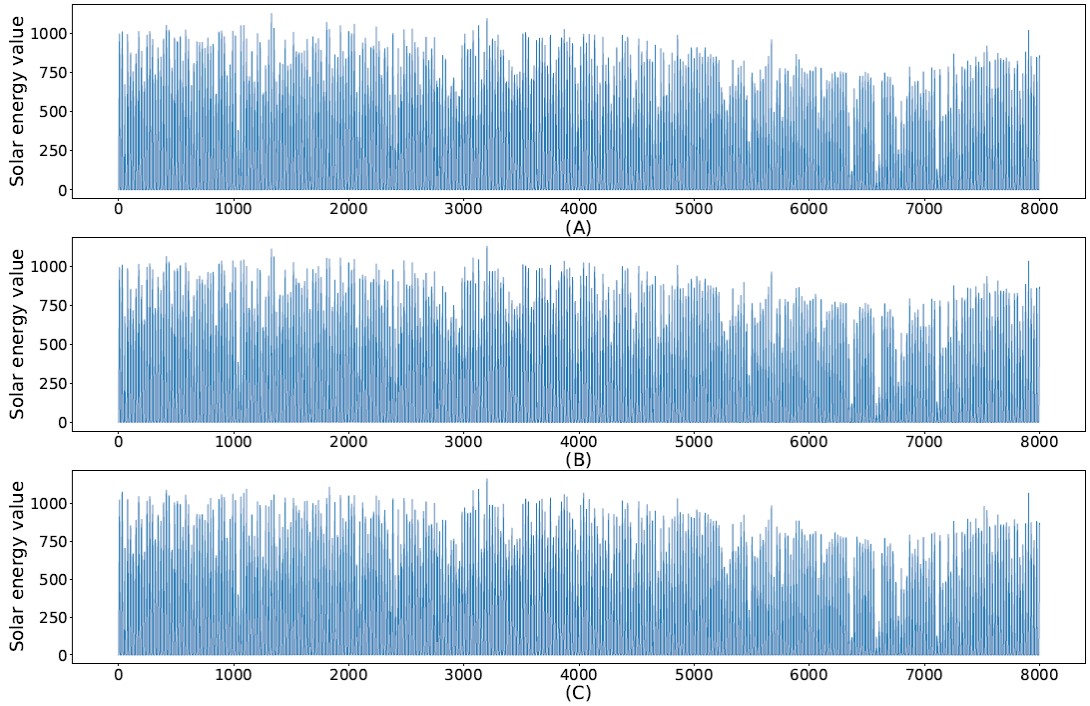


Figure 12 – Plotting of 8,000 samples of hourly solar radiance time series including DHHL_1 (A), DHHL_2 (B), and DHHL_3 (C)

4.3.1.4 GEFCom 2012 load dataset

Global Energy Forecasting Competition 2012 (GEFCom 2012) is a well-known dataset that is published on the Kaggle platform [4]. It is an hourly time series and consists of 20 zones starting from 2004-01-01 to 2008-07-07 and is in kilowatts. As shown in Figure 13, 8,000 samples of hourly load data of zones 1, 2, and 3 are applied to evaluate the performance of the R-HFCM model.

4.3.1.5 PJM Hourly Energy Consumption Data

PJM Interconnection LLC is a regional transmission organization (RTO) in the USA. The dataset is hourly energy consumption data (in megawatts (MW)), that is available at Kaggle and comes from PJM's website[5]. Figure 14 shows 8,000 samples of the hourly energy consumption of American Electric Power (AEP) and Ohio / Kentucky Duke Energy (DEOK) used to test the R-HFCM model in this study.

4.3.2 Experimental Methodology

For the quantitative evaluation of the proposed model, the performance metrics Root Mean Squared Error (RMSE), described in equation (3.6), Mean Absolute Percentage

[4] ⟨https://www.kaggle.com/c/global-energy-forecasting-competition-2012-load-forecasting/data⟩
[5] ⟨https://www.kaggle.com/robikscube/hourly-energy-consumption?select=NI_hourly.csv⟩

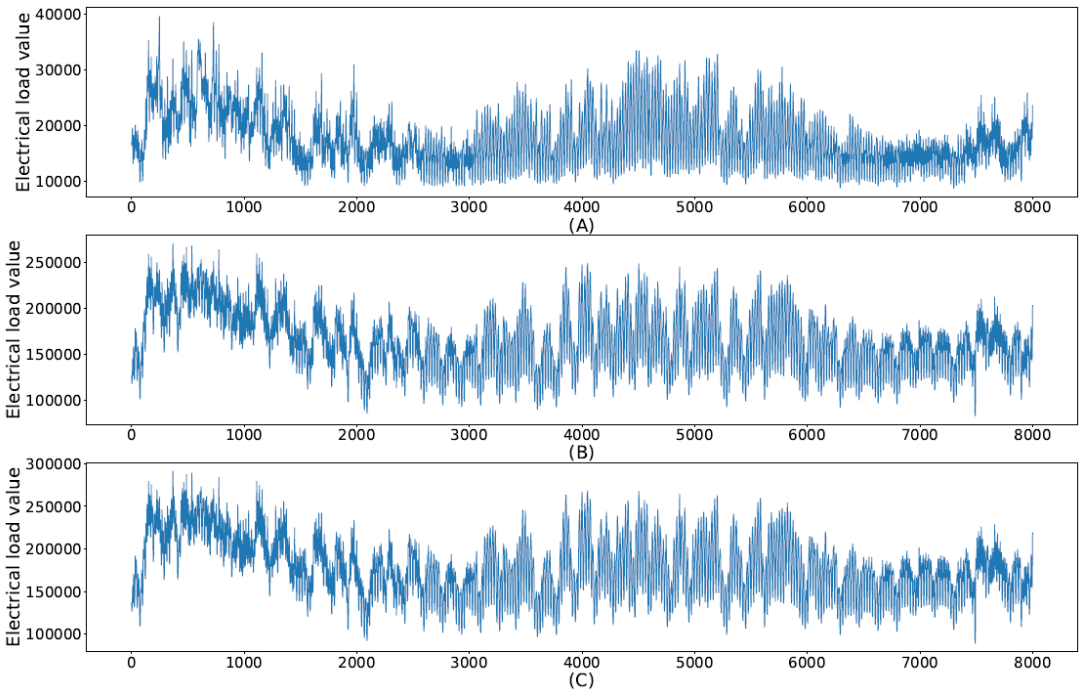


Figure 13 – Plotting of 8,000 Samples of hourly load time series from three zones of GEFCom2012 dataset zone 1 (A), zone 2 (B), and zone 3 (C)

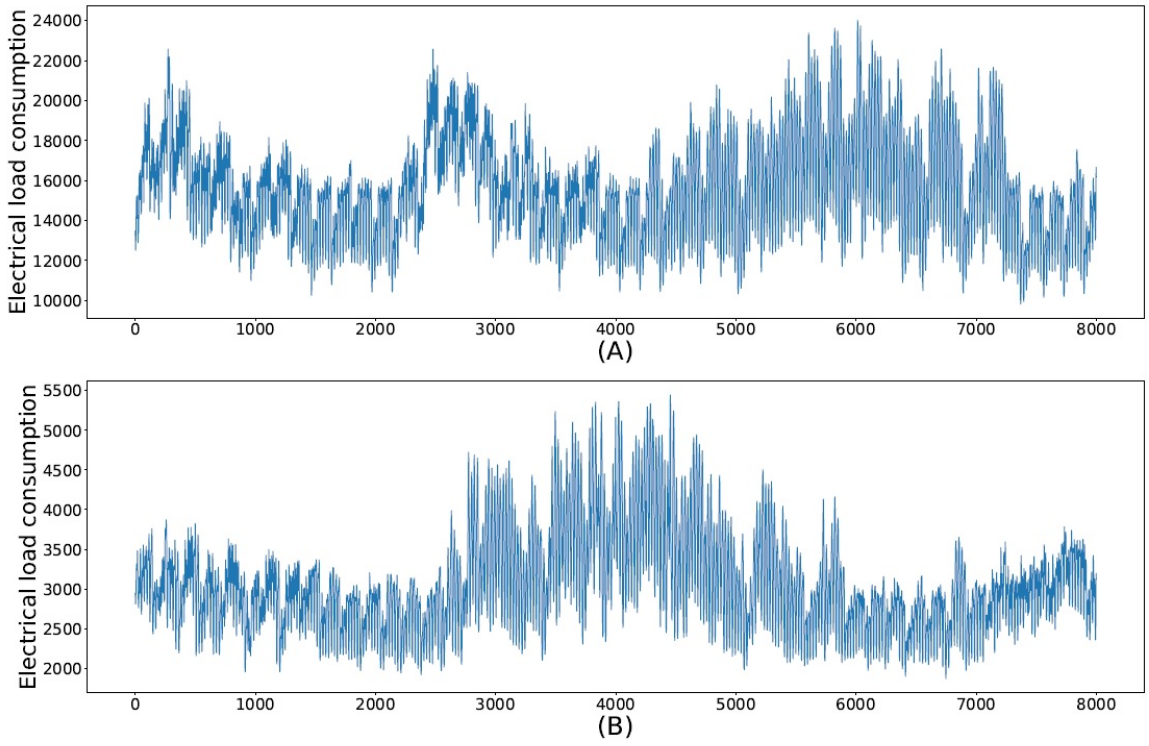


Figure 14 – Plotting of 8,000 samples of PJM hourly energy consumption including AEP data (A) and DEOK data (B)

Error (MAPE), described in equation (4.12) and Theil's U Statistic (U), described in equation (4.13) are used where $y(t)$ and $\hat{y}(t)$ stand for the actual and forecast values respectively.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{4.12}$$

$$U = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n} y_i^2} + \sqrt{\frac{1}{n}\sum_{i=1}^{n} \hat{y}_i^2}} \tag{4.13}$$

These performance metrics were computed with the sliding window cross-validation method, where the time series data is split into T overlapped data windows with size L, displaced by an increment, I. In each data window, a new model will be trained with 80% of the data and tested with 20% of the data, where the performance metrics are collected. In this research, the window size is $L = 2000$. Noteworthy, that the mentioned cross-validation method is used for measuring the influence of the R-HFCM parameters (in section 4.3.3) as well as for evaluating the performance of the proposed R-HFCM against the other baseline models in all case studies.

The Performance of the models was compared statistically using the Kruskal-Wallis approach for the equality of means. In the Kruskal-Wallis method, H_0 represents the null hypothesis that stands for equality of the average RMSE errors of all methods, and the H_1 represents the alternative hypothesis in which at least one of the averages is not equal to the others. The null hypothesis is rejected if the p-value is below 0.05. In case of H_0 rejection, it is necessary to apply *post hoc* tests that compare the equality of each pair of means (the H_0 hypothesis). In this research, the Wilcoxon method was chosen for *post hoc* tests.

4.3.3 SONDA Case Study

In this experiment, two aspects are covered: (i) the analysis of the effects of four significant hyperparameters including the number of concepts (k), the size of the reservoir (N_{SR}), the order of the model (Ω) and the type of activation function (f); and (ii) the validity of the proposed R-HFCM model. Thus, in sub-section 4.3.3.1 the influence of map and reservoir size is examined. In sub-sections 4.3.3.2, 4.3.3.3 the effect of different types of activation functions as well as the effect of the order of the model are evaluated, respectively. In sub-section 4.3.3.4 the proposed method is compared with the state-of-the-art fuzzy time series methods and, finally, in sub-section 4.3.3.5 the obtained results are compared statistically.

Table 12 – Proposed activation functions to design R-HFCM model

Activation Function	Mathematical Representation
sigmoid	$f(x) = \frac{1}{\exp^{-x}+1}$
hyperbolic tangent	$f(x) = \frac{\exp^{2x}-1}{\exp^{2x}+1}$
ReLU	$f(x) = \max(0, x)$
softplus	$f(x) = \ln(1 + \exp^{x})$

4.3.3.1 The influence of map and reservoir size

In the first scenario, 8000 samples from the SONDA dataset have been utilized as shown in Figure 11. The experiment was executed considering the presence of bias terms, $k = \{3, 4, 5, 7, 10, 20\}$, $N_{SR} = \{2, 5, 10, 20, 30, 40\}$, order ($\Omega = 2$), and Softplus as the activation function to assess the model performance.

Table 13 provides information about the average RMSE and U for 20 independent runs, given a different number of concepts and sub-reservoirs. As Table 13 shows, in all cases, the accuracy of the model is improved by increasing the number of sub-reservoirs. More specifically, raising it from 2 to 20 without considering the number of concepts. For instance, the model performs much better when $N_{SR} = 20$ compared with $N_{SR} = 5$. On the other hand, increasing the number of concepts has an inverse effect on the accuracy of the model, especially, for $N_{SR} \in \{2, 5, 10, 20\}$. It means that the higher the number of fuzzy sets, the worse the model accuracy.

A closer look at the results indicates some exceptions to this rule. For the models with 3 and 4 concepts, the accuracy is degraded by increasing the number of sub-reservoirs to 30 and 40 in comparison to the models with a higher number of concepts. For instance, the R-HFCM models with $k = \{3, 4\}$ and $N_{SR} = \{30, 40\}$ are less accurate than the models with $k = 20$ and $N_{SR} = \{30, 40\}$. Moreover, the performance of the models with $k = \{3, 4\}$ are much better than the proposed method with $k \in \{5, 7, 10, 20\}$ when $N_{SR} \in \{2, 5\}$. In contrast, for $N_{SR} \geq 10$ the minimum forecasting errors are related to the model with 5 concepts. Therefore, the best result is achieved for the model with 5 concepts and 20 sub-reservoirs.

Figure 15 shows the performance of the proposed model when the number of sub-reservoirs and concepts are varied. Each point represents the average value of RMSE or U after 20 experiments.

It can be seen in Figure 15 that when the number of concepts is greater than 4, the model accuracy improves by increasing the number of reservoirs. This figure also reveals that for $N_{SR} \in \{2, 5\}$ the minimum error is obtained when $k = 3$ while by raising the number of sub-reservoir($N_{SR} \geq 10$), regardless of the size of the reservoir, the best performance belongs to the model with 5 concepts. Moreover, for $k \leq 10$ the model

accuracy is decreased when $N_{SR} \in \{30, 40\}$. More specifically, there is a slight difference between the accuracy of the model when $k \in \{5, 7, 10\}$ and $N_{SR} \geq 20$. Conversely, model performance reduces significantly when $k \in \{3, 4\}$. Therefore, the model is highly sensitive to the selection of both k and N_{SR} parameters, and the optimal model in this setting was the one with $N_{SR} = 20$ and 5 concepts in each sub-reservoir.

Table 13 – The summary of the model performance for SONDA dataset considering different k and N_{SR} when $\Omega = 2$ and activation function is Softplus

k	**N_{SR}**	**RMSE**	**U**	**k**	**N_{SR}**	**RMSE**	**U**
3	2	184.796	1.440	7	2	231.143	1.803
	5	113.513	0.867		5	179.721	1.400
	10	106.094	0.819		10	126.137	0.964
	20	102.436	0.783		20	101.246	0.772
	30	214.189	1.770		30	101.329	0.772
	40	458.926	3.946		40	110.576	0.845
4	2	208.748	1.625	10	2	241.925	1.888
	5	134.830	1.039		5	198.386	1.544
	10	105.050	0.800		10	159.452	1.234
	20	102.843	0.784		20	106.474	0.812
	30	149.157	1.135		30	104.288	0.795
	40	347.024	2.640		40	110.104	0.843
5	2	223.973	1.748	20	2	256.276	2.003
	5	159.525	1.235		5	222.999	1.741
	10	104.022	0.794		10	194.612	1.516
	20	**98.548**	**0.750**		20	161.760	1.253
	30	100.675	0.766		30	133.573	1.020
	40	105.104	0.835		40	117.598	0.897

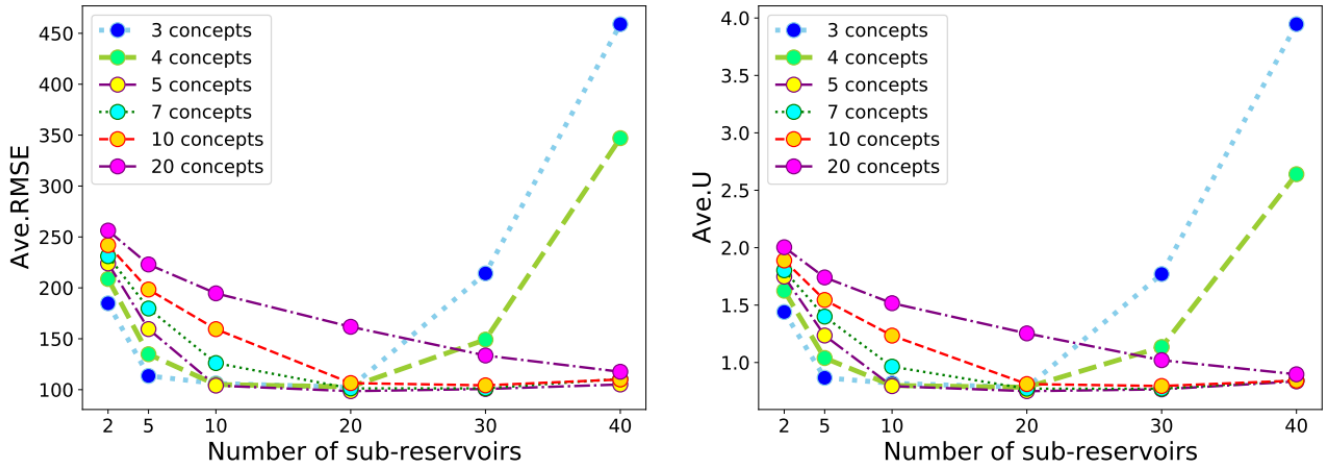


Figure 15 – Comparison of the model performance in terms of the average RMSE and U considering different map sizes and the different sizes of reservoirs (SONDA data)

4.3.3.2 The influence of activation function (f)

The goal of this section is to evaluate the effect of activation functions on model accuracy. According to the section 4.3.3.1, the best model performance is achieved with $k = 5$. Hence, in addition to varying the activation functions, the number of sub-reservoirs also changes. The number of concepts is fixed by $k = 5$.

Table 14 – The effect of using different activation functions on the model accuracy

f	N_{SR}	RMSE	U	f	N_{SR}	RMSE	U
sigmoid	2	218.796	1.710	softplus	2	223.973	1.748
	5	159.677	1.237		5	159.525	1.235
	10	104.005	0.794		10	104.022	0.794
	20	**98.882**	**0.753**		20	**98.548**	**0.750**
	30	101.484	0.772		30	100.675	0.766
	40	104.459	0.795		40	105.104	0.835
ReLU	2	226.013	1.766	tanh	2	221.770	1.732
	5	173.637	1.351		5	159.952	1.242
	10	131.307	1.013		10	109.322	0.836
	20	105.869	0.809		20	99.445	0.757
	30	100.411	0.766		30	**98.778**	**0.752**
	40	**99.539**	**0.759**		40	100.421	0.7643

Table 14 shows that the effect of the activation functions on the model accuracy is smaller than the effect caused by other factors. This means that there is no significant difference in the accuracy of the model when using different activation functions, particularly in the cases that have a high value of N_{SR}. In fact, the model performance is less sensitive to the choice of the activation function.

Table 15 shows the best-performing models from Table 14. Despite the slight difference in the accuracy of the model using different types of activation functions, the minimum error is achieved by applying Softplus. More precisely, the results indicate that when the model uses Softplus, the accuracy is slightly better. Accordingly, we employ Softplus as the model activation function in the next sections of this chapter.

Table 15 – The summary of the best performance of the model considering different activation function(f), $k = 5$ and $\Omega = 2$

f	N_{SR}	Parameters	RMSE	U
sigmoid	20	21	98.882	0.753
ReLU	40	41	99.539	0.759
softplus	**20**	**21**	**98.548**	**0.750**
tanh	30	31	98.778	0.752

4.3.3.3 The influence of order

In the previous sections, we assessed the effect of three parameters on our proposed R-HFCM model, including the size of the reservoir, the number of concepts, and the influence of the activation function. This section is mainly focused on the impact of the order on the proposed model performance. For this purpose, the proposed methodology is executed with $k \in \{3, 4, 5, 7, 10, 20\}$, $N_{SR} \in \{10, 20, 30, 40\}$, Softplus as the activation function and $\Omega \in \{3, 4, 5\}$. The summary of the experiments is presented via Table 16.

According to this table, it can be said that for the given order number, the relationship between the model accuracy, map size, and reservoir size is the same as the second-order R-HFCM model. It means that regardless of both the order value and the number of concepts, increasing the number of layers (sub-reservoirs) leads to better accuracy. Exceptions are the cases with $k \in \{3, 4\}$, $\Omega \in \{3, 4\}$. In these cases, the accuracy of the models does not improve by increasing N_{SR} from 30 to 40. The other highlight is that we can not define a specific rule for the effect of the order. In other words, it is not possible to say that in all cases increasing the order improves the accuracy. In more detail, increasing the order value amends the performance for the models including (a): $k \in \{3, 4, 5\}$, $N_{SR} \in \{30, 40\}$, (b): $k = 3$, $N_{SR} = 20$ and (c): $k = 7$, $N_{SR} = 40$. In all the other cases the better results are related to the models designed with $\Omega = 3$ except in the models including (a): $k = 4$, $N_{SR} = 20$, $\Omega = 4$ (b): $k = 7$, $N_{SR} = 30$, $\Omega = 4$ and (c): $k = 10$, $N_{SR} = 40$, $\Omega = 4$ which have the lower error in comparison to the model with order three. Moreover, it is interesting to mention that in comparison to the second-order randomized FCM, increasing the order improved the model accuracy significantly, particularly when $k \in \{3, 4\}$, $N_{SR} \in \{30, 40\}$. Also, it must be noted that for the model with 20 concepts, increasing the order reduces the accuracy and the model performs more accurately by adjusting $\Omega = 2$. In other words, we have the constant rule for 20 concepts: the lower the order, the more accurate the model.

Overall, the performance of the models with 3, 4, and 5 concepts are very close to each other, especially for $N_{SR} \geq 20$ and regardless of the order. Also, it can be confirmed from Table 17 which shows the summary of the best performance of the R-HFCM model for the SONDA dataset, extracted from Table 16, for the defined number of concepts.

Figures 16 and 17 show a better understanding of the model performance with respect to the map size, reservoir size, and the order in terms of RMSE and U. Accordingly, increasing the number of sub-reservoirs improves the accuracy of the model with a constant number of concepts except in the second order model where $k \in \{3, 4\}$ and $N_{SR} \in \{30, 40\}$. In reverse, raising the number of concepts for the models with a fixed number of sub-reservoirs degrades the accuracy. Furthermore, there exist slight differences in the performance of the models constructed with 3, 4, and 5 concepts when $N_{SR} \geq 20$

and $\Omega \in \{3, 4, 5\}$.

Table 16 – The effect of the order on the accuracy of the proposed R-HFCM model for different k and N_{SR}

k	**N_{SR}**	Ω	**RMSE**	**U**	**k**	**N_{SR}**	Ω	**RMSE**	**U**
		3	101.130	0.770			3	157.630	1.219
3	10	4	105.141	0.801	7	10	4	166.384	1.293
		5	112.640	0.861			5	175.647	1.369
		3	95.712	0.730			3	99.420	0.750
3	20	4	92.069	0.703	7	20	4	112.454	0.861
		5	92.495	0.705			5	124.518	0.955
		3	94.743	0.720			3	95.609	0.729
3	30	4	90.505	0.690	7	30	4	93.834	0.716
		5	88.590	0.677			5	98.114	0.748
		3	97.352	0.746			3	95.344	0.726
3	40	4	90.947	0.693	7	40	4	92.000	0.701
		5	88.690	0.676			5	91.859	0.700
		3	103.748	0.791			3	178.294	1.385
4	10	4	117.148	0.899	10	10	4	190.056	1.480
		5	131.019	1.011			5	201.068	1.567
		3	94.434	0.718			3	126.946	0.968
4	20	4	92.649	0.700	10	20	10	141.0465	1.086
		5	94.412	0.719			5	151.736	1.169
		3	96.095	0.732			3	100.079	0.763
4	30	4	89.881	0.684	10	30	4	109.373	0.835
		5	89.835	0.683			5	121.317	0.931
		3	102.318	0.780			3	98.114	0.748
4	40	4	90.099	0.686	10	40	4	96.961	0.739
		5	87.935	0.669			5	102.992	0.785
		3	121.505	0.933			3	211.071	1.645
5	10	4	141.587	1.092	20	10	4	218.188	1.702
		5	151.199	1.170			5	221.940	1.731
		3	95.659	0.720			3	179.111	1.394
5	20	4	95.170	0.727	20	20	20	191.092	1.488
		5	100.443	0.770			5	189.334	1.474
		3	92.975	0.708			3	155.260	1.198
5	30	4	90.670	0.689	20	30	4	165.824	1.287
		5	90.423	0.680			5	169.128	1.314
		3	92.736	0.705			3	130.405	0.998
5	40	4	89.612	0.682	20	40	4	145.985	1.126
		5	87.341	0.611			5	152.173	1.176

As Table 17 indicates, it is concluded that designing the R-HFCM model with 5 concepts is the best choice but as we discussed earlier the performance of the models with 3, 4, and 5 concepts are very close. Also, as can be seen from this table, the best accuracy is obtained for $N_{SR} = 40$ regardless of the size of the map but with the same order ($k \in \{3, 4, 5, 7\}$) or different order values ($k \in \{10, 20\}$). Finally, we increased the

Table 17 – The summary of the best performance of the model considering different k, N_{SR} and Ω

k	N_{SR}	Parameters	Ω	RMSE	U
3	30	31	5	88.590	0.677
4	40	41	5	87.935	0.669
5	**40**	**41**	**5**	**87.341**	**0.611**
7	40	41	5	91.859	0.700
10	40	41	4	96.961	0.739
20	40	41	2	117.598	0.897

Table 18 – The summary of the best model performance considering 60 sub-reservoirs and $k \in \{3, 4, 5\}$

k	N_{SR}	Parameters	Ω	RMSE	U
3	60	61	5	89.421	0.682
4	60	61	5	88.689	0.674
5	60	61	5	87.936	0.668

number of sub-reservoirs to 60 and executed the model for the best choices of concepts $k \in \{3, 4, 5\}$. Table 18 guarantees that there exist no improvements in the results by increasing the number of sub-reservoirs.

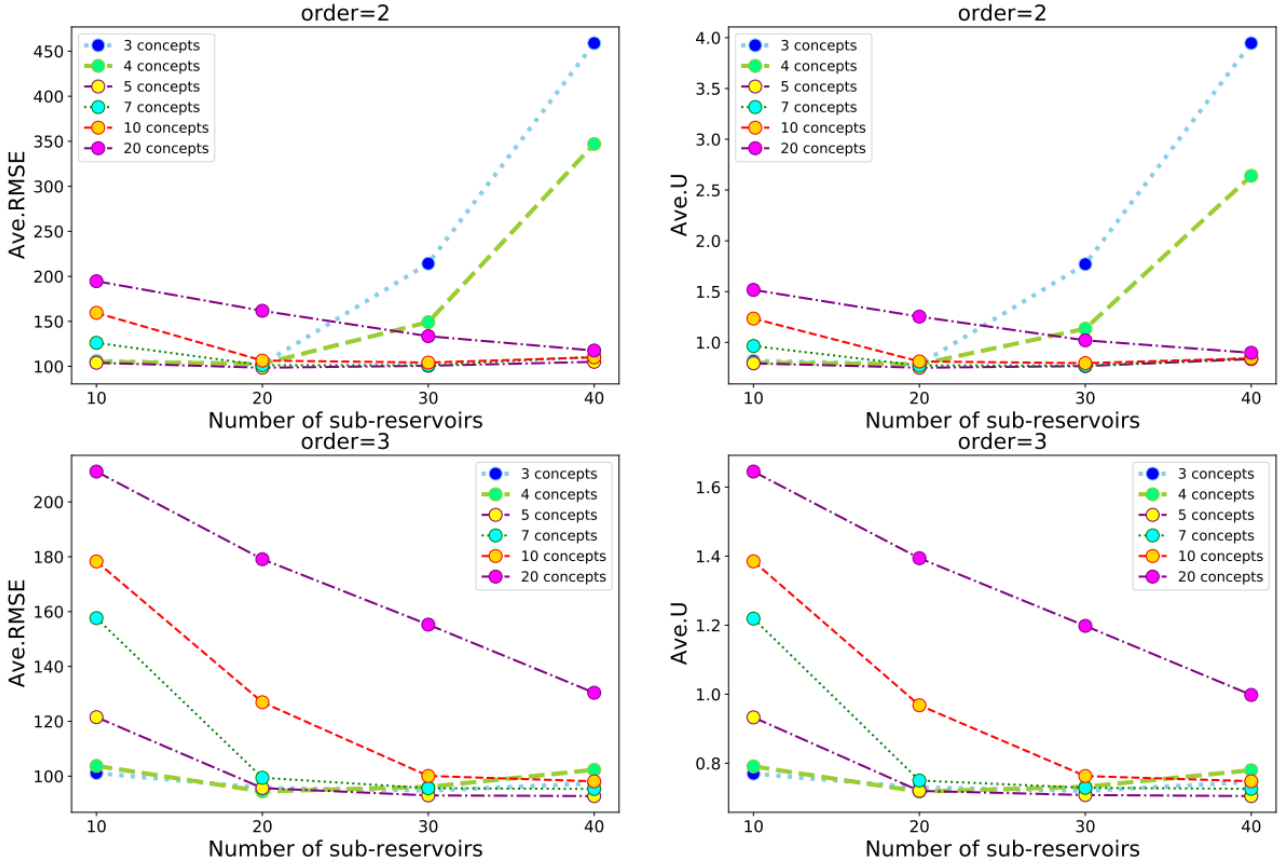


Figure 16 – Comparison of the model performance in terms of the average RMSE and U using different map sizes, size of reservoirs, and $\Omega = \{2, 3\}$ for SONDA data

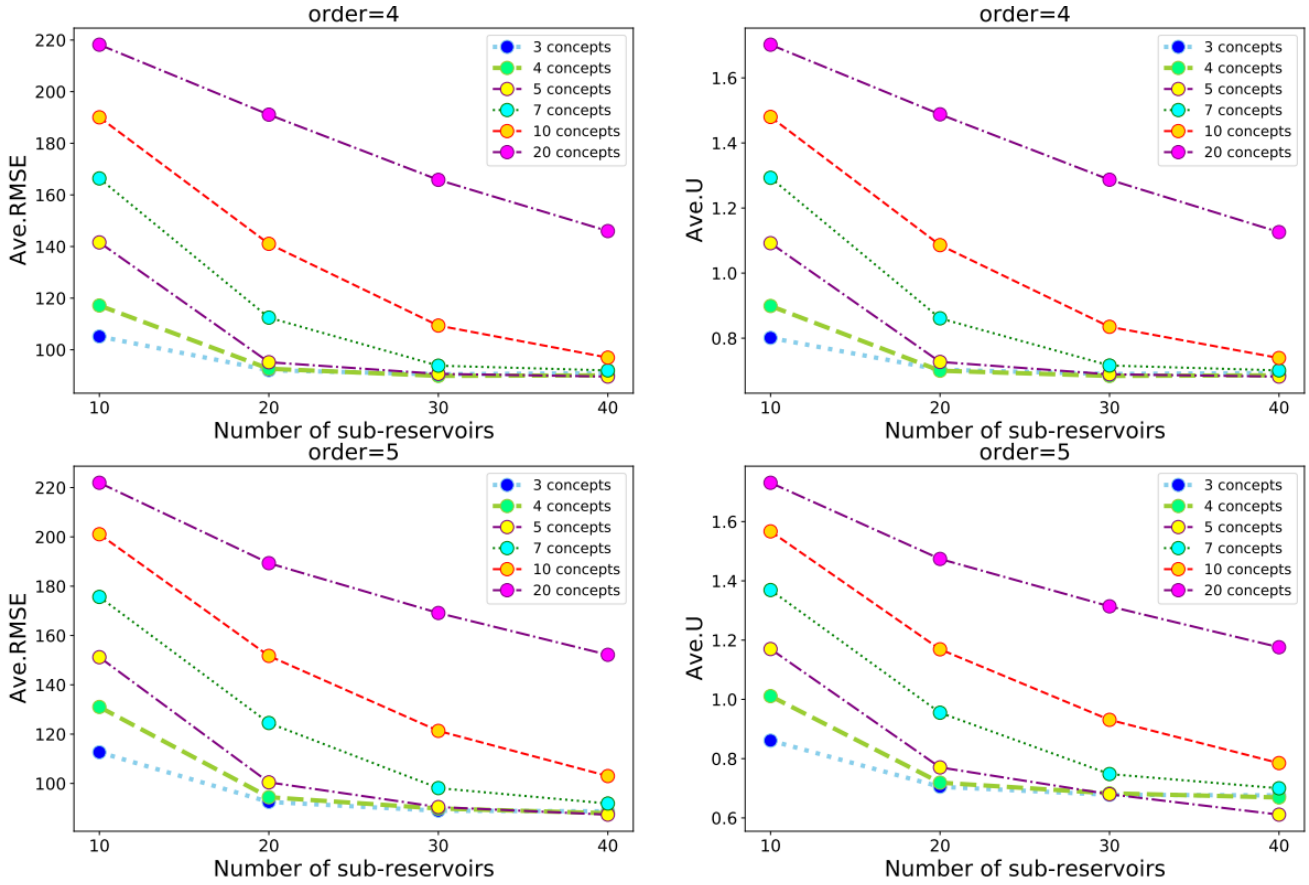


Figure 17 – Comparison of the model performance in terms of the average RMSE and U using different map sizes, size of reservoirs, and $\Omega = \{4, 5\}$ for SONDA data

4.3.3.4 Comparison with other Fuzzy Time Series methods

In this section, the proposed method is tested against other recent fuzzy time series forecasting methods available in the literature. In particular, the proposed model is tested against HOFTS, Weighted HOFTS (WHOFTS), and Probabilistic Weighted FTS (PWFTS) methods introduced by Silva (2019) and HFCM-FTS (ORANG et al., 2020). Bear in mind that the structure of each sub-reservoir in the R-HFCM model is a kind of HFCM-FTS model whereas the weights are generated randomly. Since the weights and biases are randomly chosen to meet the ESP condition according to the ESN reservoir computing, the R-HFCM results are obtained after 20 independent runs.

Accordingly, Table 19 compares the performance of the aforementioned methods with each other regarding the number of partitions in terms of RMSE and NRMSE defined by equations (3.6) and (4.14), respectively. Therefore, in order to have a better comparison, firstly we evaluate the performance of R-HFCM against the HFCM-FTS model.

$$NRMSE = \frac{RMSE}{y_{max} - y_{min}} \tag{4.14}$$

As the results indicate, unlike the HFCM-FTS model, the proposed R-HFCM

model is more accurate for each number of fuzzy sets. The other added value of the R-HFCM model is the reduction of training time in comparison to the HFCM-FTS model. It means that the R-HFCM is much faster than the HFCM-FTS model because it has fewer parameters. In other words, since the weights and biases matrices are determined randomly in the proposed model, the number of parameters only depends on the number of N_{SR}. For instance, with 10 concepts, the parameters of HFCM-FTS are more than 5 times that of the proposed R-HFCM model which can considerably decrease the training times. Thus, the least squares regression algorithm in the proposed R-HFCM model will solve a linear problem with fewer parameters compared to GA ($Pcruz$=0.5, $Pmut$=0.3, $Population$=50, $Generation$=30) in HFCM-FTS model which is a more time-consuming and iterative learning algorithm. Although the HFCM-FTS model has fewer parameters when compared with the R-HFCM model for $k = 3$, it is not comparable in terms of accuracy and training speed due to the GA-based learning process. As mentioned by Orang et al. (2020), in order to assess the efficacy of the HFCM-FTS model, more experiments are needed, as the GA parameters also have to be adjusted. For instance, increasing the number of generations or the population has an effect on the training time. Therefore, the results confirm the superiority of the proposed methodology against HFCM-FTS. Finally, despite the mentioned differences, their functions follow similar behavior: as the number of concepts increases, the performance of both models deteriorates.

Based on the results in Table 19, it can be inferred that, despite the success of the HFCM-FTS model in comparison to the WHOFTS and PWFTS models, when the number of concepts is low, the use of R-HFCM model will fill this gap. It means that the R-HFCM model outperforms the other methodologies regardless of the map size. In addition, as can be seen in Table 19, increasing the number of partitions leads to increasing the number of parameters of HOFTS, WHOFTS, and PWFTS models significantly when compared with the R-HFCM model. For instance, the number of parameters in the HOFTS, WHOFTS, and PWFTS models is 69 and 16 times that of the R-HFCM model, respectively, when $k = 20$ and $k = 10$. From another perspective, in contrast to R-HFCM and HFCM-FTS, increasing the number of concepts meliorates the performance of HOFTS, WHOFTS and PWFTS models.

Also, it is necessary to mention that increasing the number of concepts (fuzzy sets) to 40 or 60, does not improve the accuracy of the HOFT, WHOFTS, and PWFTS models. Therefore, R-HFCM ranked first in terms of accuracy in comparison to the other models. In conclusion, the optimal performance is achieved by R-HFCM employing $k = 5$ and $\Omega = 5$ followed by R-HFCM ($k = 3$, $\Omega = 5$) and R-HFCM ($k = 10$, $\Omega = 4$), respectively.

Table 19 – Evaluation of the accuracy of the proposed method against other models in terms of RMSE and NRMSE

k	Model	Parameters	RMSE	NRMSE
3	**R-HFCM**	**31**	**88.590**	**0.071**
	HFCM-FTS	24	116.850	0.094
	HOFTS	27	368.470	0.298
	WHOFTS	27	247.800	0.200
	PWFTS	27	158.700	0.125
5	**R-HFCM**	**41**	**87.341**	**0.070**
	HFCM-FTS	60	119.191	0.096
	HOFTS	123	342.640	0.277
	WHOFTS	123	173.090	0.140
	PWFTS	123	137.390	0.111
10	**R-HFCM**	**41**	**96.610**	**0.078**
	HFCM-FTS	220	157.401	0.127
	HOFTS	656	251.600	0.203
	WHOFTS	656	141.050	0.114
	PWFTS	656	135.430	0.109
20	**R-HFCM**	**41**	**117.598**	**0.095**
	HFCM-FTS	840	182.044	0.147
	HOFTS	2855	192.920	0.156
	WHOFTS	2855	134.370	0.108
	PWFTS	2855	135.420	0.109

4.3.3.5 Statistical testing

The Kruskall-Wallis method was used to statistically compare the R-HFCM, HFCM-FTS, HOFTS, WHOFTS, and PWFTS results. The obtained test statistic and p-value were 66.016 and 1.571e-13, respectively. For this p-value, H_0 is rejected at the predefined significance level ($\alpha = 0.05$) indicating that there is a significant difference between the means of at least one of the competitor models. Accordingly, the Wilcoxon method was used in the pairwise comparison as illustrated in Table 48. According to these results, the R-HFCM model performs statistically better than all other models. Interestingly, the PWFTS and WHOFTS methods share the second rank. Overall, Table 20 shows a summary of the statistical ranking of the models in which R-HFCM ranks first, while HOFTS is the worst method

Table 20 – The Summary of statistical ranking of models

Rank	Algorithm
1	R-HFCM
2	PWFTS
3	HCM-FTS
2	WHOFTS
4	HOFTS

4.3.4 Malaysia Case Study

In the second scenario, to test the utility of the proposed model, we consider the hourly electric load of the Malaysia dataset. 8760 samples of load data are selected as the model inputs as shown in Figure 11. Since the model has been designed to predict univariate time series, the hourly electric load time series is fed into the model. Thus, this section is divided into two sub-sections. In section 4.3.4.1 the performance of the proposed R-HFCM model is evaluated considering the influence of its hyperparameters in the model accuracy. In section 4.3.4.2, the model is compared with other state-of-the-art time series forecasting models in the literature.

4.3.4.1 Forecasting Hourly electric load dataset

Similar to the first case study, a sliding window of $2,000$ samples in the cross-validation method has been used but in this scenario, 90% of the window is for training and 10% for testing. The experiment is executed with $k \in \{3, 4, 5, 7, 10\}$ concepts, $\Omega \in \{2, 3, 4, 5\}$, $N_{SR} \in \{2, 5, 10, 20, 30, 40, 60, 70, 80\}$ sub-reservoirs, Softplus activation function and by considering bias terms to assess the model performance. According to Table 46 shown in section A, the performance of the proposed method is evaluated by considering RMSE, MAPE, and U as the accuracy metrics. Therefore, the influences of the effective parameters including reservoir size, map size, and the order over the model's accuracy are discussed as follows.

Firstly, we assess the effect of the reservoir size. As shown in Table 46, it can be said that, in general, increasing the number of sub-reservoirs decreases the prediction errors regardless of the number of concepts and the order value. However, there are some limited cases where the accuracy does not improve. For instance for the models with ($k = 3$, $\Omega = \{2, 3, 4\}$) and ($k = 10$, $\Omega = 2$), increasing the reservoir sizes from 70 to 80 actually reduces the accuracy.

In the second phase, the impact of the map size on the proposed model is evaluated. As can be seen in Table 46, increasing the number of concepts harms the model performance. Accordingly, the most accurate model is reached when $k = 3$ regardless of the reservoir size and the order value. Generally speaking, it can be said that raising the number of concepts has an inverse impact on the model accuracy. It means that with a constant number of sub-reservoirs, the fewer the number of concepts, the more accurate the model. In summary, as in the SONDA case, the accuracy of the model has positive and negative relation with N_{SR} and k respectively. Moreover, it is worth mentioning that, the model with the lowest map size ($k = 3$) performs better in comparison to the other cases and the most accurate R-HFCM is reached when $k = 3$, $N_{SR} = 80$ and $\Omega = 5$.

Finally, the impact of the order on the proposed model is examined. As deduced

from the results, it is clear that finding a fixed rule that explains the model behavior for the defined number of orders is not possible. Thus, in this step, the effect of the order is assessed for each map size separately in terms of RMSE, MAPE, and U metrics as follows:

1. **k=3**: As the results show, for $N_{SR} \leq 10$, $N_{SR} = 20$ and $N_{SR} \geq 30$, the best models are obtained by setting $\Omega = 2$, $\Omega = 3$ and $\Omega = 5$, respectively.

2. **k=4**:In this case, the optimum models for $N_{SR} \leq 10$, $N_{SR} \in \{20, 30\}$ and $N_{SR} \geq 40$ are generated by setting $\Omega = 2$, $\Omega = 3$ and $\Omega = 5$, respectively.

3. **k=5**: In this case, the most accurate models are obtained by selecting $\Omega = 2$, $\Omega = 3$ and $\Omega = 5$ for the models with $N_{SR} \leq 20$, $N_{SR} = 30$ and $N_{SR} \geq 40$, respectively.

4. **k=7**: In this case, the best performance could achieve by setting $\Omega = 2$, $\Omega = 3$ and $\Omega = 5$ when $N_{SR} \leq 20$, $N_{SR} \in \{30, 40\}$ and $N_{SR} \geq 60$, respectively.

5. **k=10**: This is the same as the model with 7 concepts. It means that for $N_{SR} \leq 20$, $N_{SR} \in \{30, 40\}$ and $N_{SR} \geq 60$, the orders could be set as $\Omega = 2$, $\Omega = 3$ and $\Omega = 5$, respectively, to reach the least prediction error.

In a nutshell, it can be concluded that the performance of the model has been upgraded by setting $\Omega = 5$ and $N_{SR} = \{70, 80\}$ for the given map sizes. On the other hand, the worst results are reached with the minimum number of sub-reservoirs regardless of the map size and order value. Also, constructing the second-order models for $N_{SR} \geq 30$ is considered the worst option. Moreover, for $k \in \{3, 4, 5, 7\}$ and $N_{SR} \geq 40$, the models perform more accurately with order values equal to five, three, four, and two, respectively. It means that fourth and second-order R-HFCMs are the worst cases in this category.

More precisely, Figures 18 and 19 represent the model performance for the given model order, sub-reservoirs, and concepts in terms of RMSE, MAPE, and U. In general, as the figures indicate, increasing the number of concepts and sub-reservoirs has an inverse and direct effect on the model accuracy, similar to the one observed in the SONDA dataset. Also, the performance of the models with 3, 4, and 5 concepts are very close especially for $N_{SR} \geq 30$.

The summary of the best performance of the model in terms of average RMSE, MAPE, and U is shown in Table 21. It can be seen that there are very small differences in the accuracy of the model using 3, 4, and 5 concepts, especially in terms of MAPE and U comparison. According to these results, extracted from Table 46, the maximum accuracy is achieved by setting $k = 5$, $N_{SR} = 80$ and $\Omega = 5$, while the maximum error is obtained by selecting 10 concepts in our model. Interestingly, constructing the model with $N_{SR} = 80$ and $\Omega = 5$ leads to the highest level of accuracy for the given number of concepts.

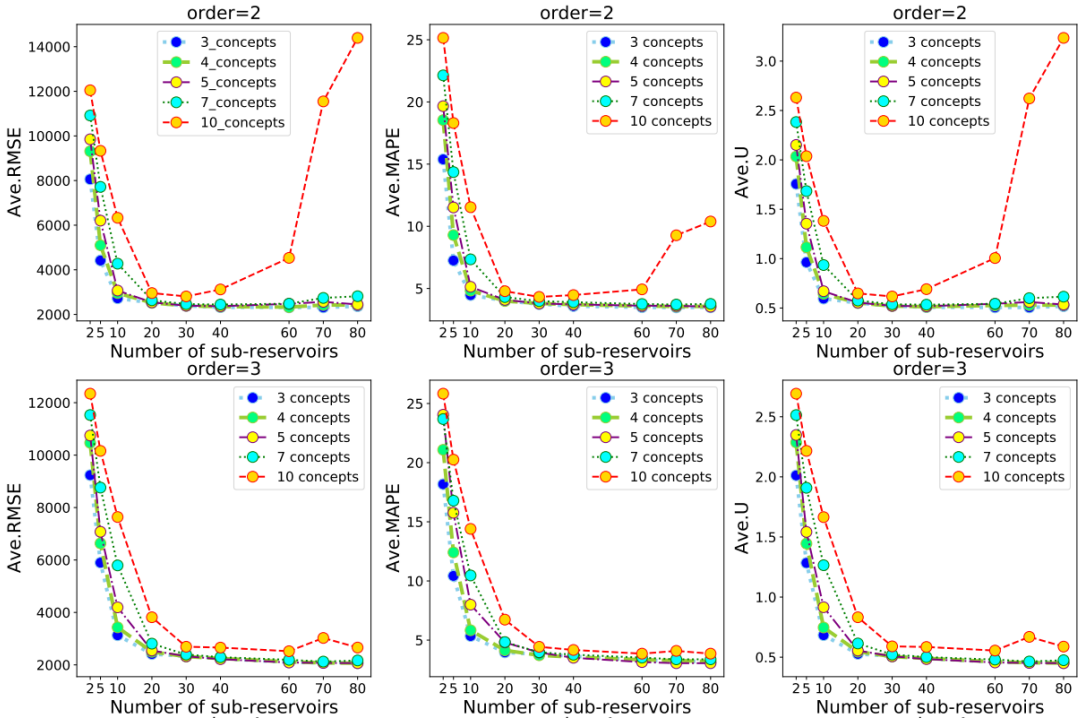


Figure 18 – Comparison of the model accuracy in terms of the average RMSE, MAPE, and U considering the different number of concepts and sub-reservoir when $\Omega \in \{2, 3\}$(Malaysia load data)

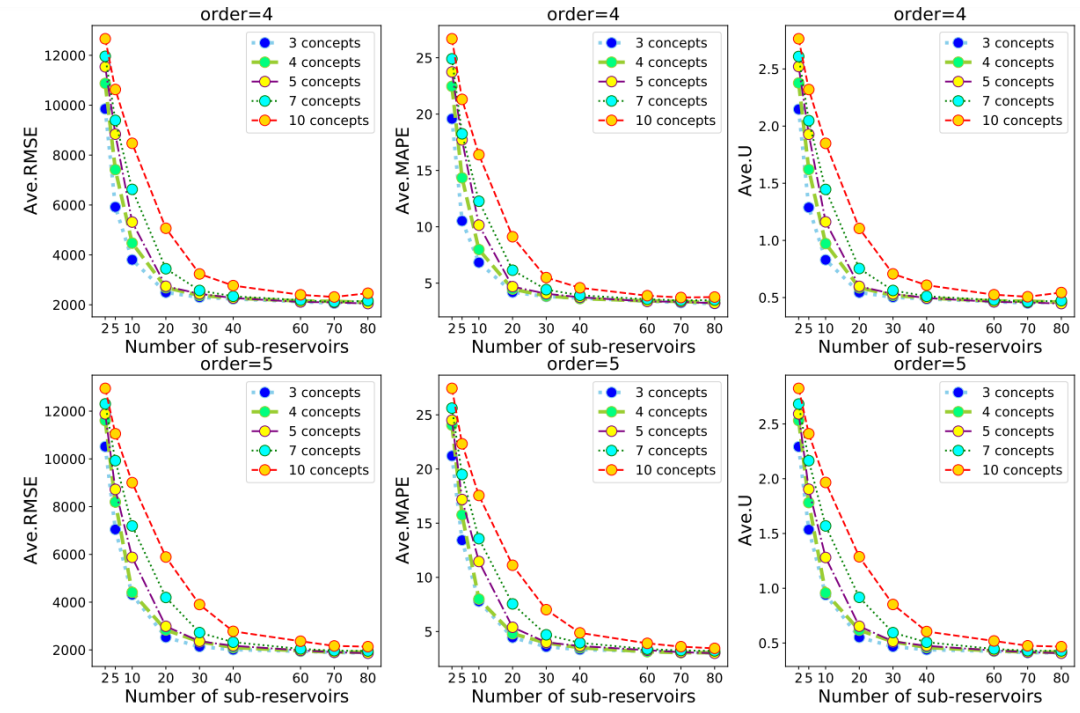


Figure 19 – Comparison of the model accuracy in terms of the average RMSE, MAPE, and U considering the different number of concepts and sub-reservoir when $\Omega \in \{4, 5\}$(Malaysia load data)

Table 21 – The summary of the best R-HFCM performance considering different map sizes $k \in \{3, 4, 5, 7, 10\}$

k	N_{SR}	Parameters	Order	RMSE	MAPE	U
3	**80**	**81**	5	**1870.096**	**3.033**	**0.409**
4	70	71	5	1904.884	3.048	0.417
5	80	81	5	1872.186	3.043	0.409
7	80	81	5	1955.056	3.213	0.427
10	80	81	5	2137.144	3.447	0.467

4.3.4.2 Comparison With Other Methods

In this section, the proposed R-HFCM model is tested against other investigated models in the literature whereas the target is to predict the hourly electric load. Accordingly, Table 22 shows the collection of the obtained results from the proposed R-HFCM method and other competitor models in terms of average MAPE, RMSE, and NRMSE. R-HFCM is executed considering $k = \{3, 4, 5, 10, 20\}$ concepts.

FTS-CNN was introduced by Sadaei et al. (2019) as a combination of FTS and convolutional neural networks (CNN). This methodology is composed of three phases including processing the time series to make them ready for converting multivariate time series into multi-channel images, creating image time series, and in the last step, the proper model (image size=32, batch size=100, number of epochs=20, learning rate=0.001) for the prediction is selected. The other aforementioned baseline models in Table 22 are explained briefly here (SADAEI et al., 2019). Autoregressive Integrated Moving Average (SARIMA)(1,0,3)(2,1,2) performs perfectly among other linear models with the mentioned value of parameters. PWFTS, Weighted FTS (WFTS), and Integrated Weighted Fuzzy Time Series (IWFTS) are some new FTS techniques available in the pyFTS library (SILVA et al., 2019). In addition, Long Short-Term Memory (LSTM) was introduced by Hochreiter e Schmidhuber (1997) which is one of the most important deep learning models and a member of the RNNs family. Based on the Sadaei et al. (2019), LSTM is tested with regards to the various former lags covering 168, 72, 48, and 24 for LSTM, LSTM model 3, LSTM model 2, and LSTM model 1, respectively, to predict the target time series at time $t + 1$. Table 22 compares the accuracy of the models in terms of average MAPE, RMSE, and NRMSE defined by equations (4.12), (3.6) and (4.14), respectively.

As Table 22 suggests, the performance of R-HCM and FTS-CNN are very close in terms of MAPE metrics. Therefore, in terms of MAPE, the best performance belonged to the R-HFCM($k = 3$) and FTS-CNN followed by R-HFCM($k = 4$) and R-HFCM($k = 5$). However, FTS-CNN outperforms all models considering RMSE (or NRMSE) metrics. Overall, it can be concluded that FTS-CNN has better performance in comparison to the R-HFCM model. Even though FTS-CNN is located in the first rank, the results for R-HFCM with k=$\{3, 4, 5\}$ are still promising because there is no significant difference

Table 22 – Comparison of the proposed R-HFCM method with other models in terms of average MAPE, average RMSE and NRMSE

Methods	MAPE	RMSE	NRMSE
R-HFCM ($k = 3$)	3.033	1870.096	0.033
R-HFCM ($k = 4$)	3.048	1904.884	0.034
R-HFCM($k = 5$)	3.043	1872.186	0.033
R-HFCM ($k = 7$)	3.213	1955.056	0.034
R-HFCM ($k = 10$)	3.447	2137.144	0.038
FTS-CNN	3.020	1777.990	0.032
LSTM	3.710	2194.190	0.039
LSTM model 1	4.550	2689.420	0.048
LSTM model 2	4.110	2413.600	0.043
LSTM model 3	3.930	2317.880	0.041
WFTS with differences	5.330	2930.360	0.052
IWFTS with differences	5.327	2961.170	0.052
PWFTS with differences	4.590	2987.400	0.053
PWFTS	3.860	2230.910	0.039
SARIMA	4.680	2763.660	0.049

among the forecasting errors and their performance is very close. Also, the results indicate the superiority of the proposed R-HFCM models in comparison to the other competitor models. For instance, the proposed R-HFCM model could achieve 24.08% 22.07%, 21.91%, 15.4%, and 7.62% better accuracy than LSTM, when the number of R-HFCM concepts are $k = \{5, 3, 4, 7, 10\}$, respectively. Besides, the R-HFCM model is like a shallow model with fewer parameters and a cheaper computational cost than the FTS-CNN and LSTM methods. These results indicate that the proposed R-HFCM model, which is equipped with a fast and non-iterative learning algorithm, is competitive with other FTS methods and some state-of-the-art methods.

4.3.5 Other case studies

According to Figures 12, 13 and 14, we employed eight additional univariate time series from NREL, GEFCom2012, and PJM datasets to assess the efficiency of our proposed model. In all cases, 8000 samples of the time series have been utilized using sliding window cross-validation as explained in section 4.3.2.

Table 47, which is shown in section A, represents the accuracy of the R-HFCM model in terms of average RMSE considering six univariate time series including DHHL_1, DHHL_2 and DHHL_3 taken from NREL datasets and the zones 1, 2 and 3 electric load time series taken from GEFCom2012 dataset. As shown in Table 47, the experiments are executed with $k \in \{3, 4, 5\}$, $\Omega \in \{3, 4, 5\}$, $N_{SR} \in \{20, 40, 60\}$, Softplus activation function and considering bias term. In the following, according to the obtained results, we evaluate the model performance for NREL, GEFCom2012, and PJM datasets.

4.3.5.1 NREL dataset(DHHL_1, DHHL_2, DHHL_3)

As indicated in Table 47, in terms of order, the accuracy of the model is improved by raising the order value. Therefore, it can be said that regardless of both the number of concepts and sub-reservoirs, the more accurate model is obtained by setting $\Omega = 5$. On the other hand, for a specific number of concepts, designing the model with $N_{SR} = 20$ leads to the highest forecasting error when we increase the order value. Obviously, for each time series in this category, the more accurate model is achieved by choosing $N_{SR} = 40$ when $k = \{3, 5\}$ and $\Omega \in \{3, 4\}$. Meanwhile, the best results are obtained by setting $N_{SR} = 60$ and $\Omega = 5$. Additionally, in the case of $k = 4$, the most accurate model is achieved by setting $N_{SR} = 40$ for any number of orders. With a constant size of the reservoir, the model with 3 concepts shows the best performance with one exception when $k = 5$ and $N_{SR} = 60$, which is more accurate than the models with $k = \{3, 4\}$.

4.3.5.2 GEFCom2012 dataset (zone1, zone2, zone3)

In order to evaluate the model accuracy, Table 47 also lists the performance of the proposed model employing the GEFCom2012 dataset using a different number of concepts, sub-reservoirs, and order. In this case study, in general, raising the order value improves the performance of the model when $N_{SR} \in \{40, 60\}$ so that the worst case is associated with $\Omega = 3$. For $N_{SR} = 20$, we have some scenarios based on the zones. In zone 1, the optimal accuracy is achieved when $\Omega = 5$ and $k = 3$. In contrast, increasing the order value degrades the model accuracy when $k \in \{4, 5\}$. In zones 2 and 3, the performance of the model is improved by increasing the order. It means that the higher the order, the more accurate the model. However, in the case of $k = 3$, the best accuracy is achieved by adjusting $\Omega = 4$.

It is worth noting that with the constant number of sub-reservoirs, the accuracy of the model is upgraded by adjusting $k = 4$. For instance, suppose that zone 1 is the input data. For each number of defined sub-reservoirs, the best values of RMSE are 753.464, 716.364, and 732.174, which are achieved when $k = 4$ for 20, 40, and 60 sub-reservoirs, respectively. On the other side, for the constant number of concepts, we can consider the following scenarios:

1. k=3: increasing the number of N_{SR} from 20 to 60 declines the accuracy.
2. k=4: depends on the order such that for $\Omega = 3$ the highest accuracy is provided with $N_{SR} = 20$. For $\Omega \in \{4, 5\}$, the best performance is achieved with 40 sub-reservoirs.
3. k=5: it behaves similar to the model with $k = 4$ concepts.

Table 23 – The summary of the best R-HFCM performance in terms of average RMSE

Data	k	N_{SR}	order	RMSE
NREL(DHHL_1)	3	60	5	86.421
NREL(DHHL_2)	5	60	5	86.154
NREL(DHHL_3)	5	60	5	89.050
GEFCom2012(zone1)	4	40	5	716.364
GEFCom2012(zone2)	4	40	5	5531.027
GEFCom2012(zone3)	4	40	5	5837.610

Table 23 summarizes the optimal performance of the R-HFCM model exploiting NREL and GEFCom2012 datasets. As can be seen from this table, $\Omega = 5$ is the best choice of order for both datasets. Interestingly, for GEFCom2012 the highest accuracy is obtained with $k = 4$ and $N_{SR} = 40$ while for NREL we have highest accuracy when $N_{SR} = 60$ with k={3,5,5} for NREL(DHHL_1), NREL(DHHL_2) and NREL(DHHL_3), respectively.

4.3.5.3 PJM dataset (AEP and DEOK)

In this case, Table 24 provides just the best results for both AEP and DEOK time series. In all cases, the best model performance is achieved by setting $N_{SR} = 20$ regardless of the map sizes. In other words, increasing the number of sub-reservoirs greater than 20 has an inverse effect on the model accuracy. A more detailed look at the table reveals that for AEP data, R-HFCM can produce the highest accuracy when $k = 4$, whereas, for DEOK, the most accurate performance is achieved when $k = 3$. Also, it is noticeable that these optimal results have been achieved when $\Omega = 5$ except in one case for DEOK data with $k = 3$ and $N_{SR} = 20$, in which the best performance is obtained by setting $\Omega = 4$.

4.3.5.4 Comparison with other methods

In this section, the performance accuracy of the proposed R-HFCM is quantitatively compared to some of the baseline models. As listed in Table 25, R-HFCM is tested against LSTM, PWFTS, and ESN models, and the prediction error was measured in terms of RMSE defined by equation (3.6) and NRMSE calculated via equation (4.14).

As we discussed earlier, the PWFTS method was developed recently as a powerful FTS technique in comparison to many statistic forecasting methods, other FTS, and machine learning methods. Another selected method is LSTM which has been applied in time series forecasting applications by many researchers. The best results for LSTM are presented in Table 25 when 24 former lags were selected. The last competitor model is ESN as explained in section 4.1. Note that the ESN model was implemented using

Table 24 – The summary of the R-HFCM performance in terms of average RMSE (AEP and DEOK data)

PJM(AEP)			PJM(DEOK)		
k	N_{SR}	**RMSE**	**k**	N_{SR}	**RMSE**
3	20	641.305	3	20	136.783
	40	878.101		40	178.636
	60	1802.222		60	325.730
4	20	624.151	4	20	202.148
	40	644.408		40	470.219
	60	1011.252		60	942.807
5	20	782.124	5	20	175.145
	40	1143.738		40	183.175
	60	1462.928		60	312.400

the PyESN[6] library. In this study, various settings were tested and the best results were recorded in Table 25.

The superiority of R-HFCM over competitor models can be recognized visually when looking at the error values given in Table 25. As reported in this table, the first rank belonged to R-HFCM for any case studies. PWFTS performs better than R-HFCM with a slight difference when DEOK is fed to the model. As can be seen from Table 25, LSTM performs differently. More clearly, for the NREL time series, the second rank belonged to LSTM while for the remaining time series, LSTM is the worst choice and it becomes more acute for GEFCom2012 (zone 2) and GEFCom2012 (zone 3) with high prediction errors. On the other hand, PWFTS ranks second after R-HFCM except where NREL datasets are injected into the model as inputs. As the results represent, the third place goes to the ESN model, except when NREL datasets are fed to the model, which leads to the worst-case results.

4.3.5.5 Statistical testing

The Kruskall-Wallis method was employed to compare the R-HFCM, PWFTS, LSTM and ESN method results on the datasets NREL (DHHL_1, DHHL_2, DHHL_3), GEFCom2012 (zone1, zone2, zone3) and PJM (AEP and DEOK), as presented in Table 49.

Based on this table, H_0 has been rejected for all datasets. Thereby, a *post-hoc* test was exploited to compare all the models with each other as specified in Table 50. The results in this table confirm that R-HFCM is a winning model compared to other models regardless of the type of datasets. More precisely, Table 44 summarizes the rank of each model for each dataset separately. According to the calculated average ranks, as reported

[6] ⟨https://pypi.org/project/pyEsn/⟩

Table 25 – Evaluation of the accuracy of the proposed method against other models

Data	Model	RMSE	NRMSE
NREL(DHHL_1)	**R-HFCM**	**86.421**	**0.077**
	LSTM	88.191	0.078
	PWFTS	105.883	0.094
	ESN	331.415	0.294
NREL(DHHL_2)	**R-HFCM**	**86.154**	**0.076**
	LSTM	90.692	0.081
	PWFTS	110.269	0.097
	ESN	438.782	0.389
NREL(DHHL_3)	**R-HFCM**	**89.050**	**0.076**
	LSTM	92.093	0.079
	PWFTS	111.736	0.096
	ESN	440.594	0.378
GEFCom2012(zone 1)	**R-HFCM**	**716.364**	**0.023**
	LSTM	9952.014	0.3221
	PWFTS	939.671	0.030
	ESN	5258.725	0.171
GEFCom2012(zone 2)	**R-HFCM**	**5531.027**	**0.029**
	LSTM	139798.901	0.746
	PWFTS	6826.440	0.036
	ESN	34911.848	0.186
GEFCom2012(zone 3)	**R-HFCM**	**5837.610**	**0.028**
	LSTM	162303.694	0.802
	PWFTS	7365.544	0.036
	ESN	37669.955	0.186
PJM(AEP)	**R-HFCM**	**624.151**	**0.044**
	LSTM	7071.920	0.498
	PWFTS	674.244	0.047
	ESN	3038.478	0.214
PJM(DEOK)	R-HFCM	136.783	0.038
	LSTM	1176.827	0.329
	PWFTS	**136.183**	**0.038**
	ESN	718.079	0.201

in Table 44, the first rank belongs to the R-HFCM model followed by PWFTS, LSTM, and ESN, respectively.

Table 26 – The summary of the rank of models considering eight various datasets

Dataset	R-HFCM	PWFTS	LSTM	ESN
PJM(DEOK)	**1**	2	3	4
PJM(AEP)	**1**	2	4	3
GEFCom2012(zone1)	**1**	2	4	3
GEFCom2012(zone2)	**1**	2	4	3
GEFCom2012(zone3)	**1**	2	4	3
NREL(DHHL_1)	**1**	3	2	4
NREL(DHHL_2)	**1**	3	2	4
NREL(DHHL_3)	**1**	3	2	4
AVG.RANK	**1**	2.375	3.125	3.5

In brief, R-HFCM as a new hybrid FTS model combining HFCM and ESN reservoir computing was tested over 10 different univariate time series to predict hourly solar and electrical load consumption. The proposed model has been designed based on different sets of parameters such as map sizes, reservoir size, and order value. As the results indicate, the relationship between the accuracy and the number of concepts are opposite of each other. That is, in general, the most accurate model is achieved with the lowest number of concepts ($k = \{3, 4, 5\}$). On the other side, the optimal performance of the presented model is reached by increasing the size of the reservoir adjusting $N_{SR} \geq 20$ for different datasets. Interestingly, the model performs with a higher level of accuracy by selecting $\Omega = 5$ in most of the cases. Moreover, according to the obtained results, it can be concluded that the R-HFCM model is more accurate than other baseline and recent FTS models such as ESN, LSTM, and PWFTS. Although the accuracy of the proposed R-HFCM model is slightly inferior to that of FTS-CNN, the results are very close and promising as we discussed earlier.

4.4 Chapter's Highlights

This chapter provides a new randomized-based method to predict univariate time series using FCM. It introduces a novel FCM forecasting model, called R-HFCM, which combines reservoir computing (ESN) and HFCM.

The R-HFCM model is a group of random HFCM-FTS models integrating the concepts of FCM and ESN. More specifically, R-HFCM is composed of three layers employing random HFCM-FTS models as the components of the reservoir unit called sub-reservoirs. Since the structure of each sub-reservoir is the same as HFCM-FTS, the main focus of this model is on training the output layer using least squares. The weights of each sub-reservoir in the internal layer are initialized randomly so that the ESP condition

of ESN is met. Then, the least squares regression method is applied to the output units to generate the final predicted value. The model performance has been evaluated using 10 univariate time series from 5 different public datasets. Based on the results, the size of the reservoirs, the map size, and the model order have considerable influence on the model accuracy. Meanwhile, the activation function has the least effect on the accuracy of the proposed method.

Compared with the HFCM-FTS, the R-HFCM offers advantages such as high speed and accuracy while working with a similar number of concepts. It is important to highlight that as the number of concepts increases, the accuracy of the prediction, interpretability, and readability of the model deteriorates. Moreover, the number of parameters in R-HFCM directly depends on N_{SR}. Thereby, the least squares method will solve a linear problem with fewer variables than GA, which can be regarded as an effective element in the training speed of the model. Also, the obtained results verify the efficiency and effectiveness of the R-HFCM compared to other state-of-the-art models and methods listed in the literature, including PWFTS, WHOFTS, FTS-CNN, LSTM, and ESN.

Since the model has been designed to predict univariate time series, predicting multivariate time series, either low dimension or high dimension, can be considered as the possibility for the model extension. Thus, the key attention of the next chapters is to introduce Multiple Input Multiple Output (MIMO) methods with the ability to handle the above-mentioned issues.

5 PROPOSED MO-RHFCM TECHNIQUE

Multivariate time series forecasting methodologies are categorized into Multiple Input Single Output (MISO) and Multiple Input Multiple Output (MIMO) methods. Unlike MISO models with only one target variable (or endogenous), MIMO models employ more than one endogenous which means that all the variables can be set as target variables. To the best of our knowledge, there is a significant gap in the literature regarding MIMO FCM-based forecasting models. Hence, the key objectives of this chapter are to introduce a novel randomized multivariate FCM technique to predict multiple outputs labeled as MO-RHFCM. MO-RHFCM is a hybrid model merging the concepts of HFCM, multivariate FTS (MVFTS) (SILVA, 2019), and ESN trained via the least squares algorithm.

The contributions of this chapter are summarized as follows:

1. Developing a new FCM-based forecasting model integrating the concepts of MVFTS, HFCM, and ESN termed MO-RHFCM, which is trained with time-effective learning inspired by reservoir computing;
2. The first proposal of multi-output time series forecasting with randomized multivariate FCMs in the literature according to the best of our knowledge;
3. Validation of MO-RHFCM, which is tested on two low-dimensional data sets. The experimental results demonstrate that the proposed model is efficient and competitive in comparison to the other existing approaches.

The remainder of this chapter is structured as follows: Section 5.1 introduces the proposed method in detail; Section 5.2 describes the case studies used to test our methodology; Section 5.3 represents the experimental results and discussion; and finally the conclusions and some possibilities of future work are drawn in Section 5.4.

5.1 Proposed MO-RHFCM model

This section introduces a new FCM-based time series forecasting model to predict multivariate time series. Figure 20 presents the general structure of the proposed MO-RHFCM technique.

MO-RHFCM is a randomized-based hybrid technique where the least squares algorithm is used to train the model. In terms of structure, as shown in Figure 20, MO-RHFCM consists of three components:

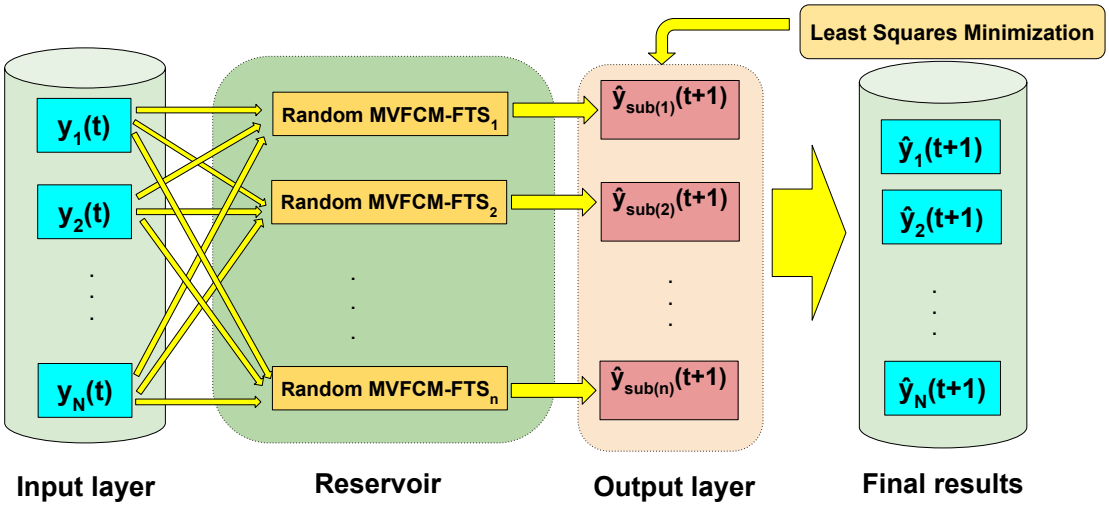


Figure 20 – General structure of the proposed MO-RHFCM method.

1. **Input Layer**: Given a multivariate time series $Y^{T\times N}$, where $N \in \mathrm{N}^+$ is the number of variables and T is the number of samples, and each sample given by $y(t) = [y_1(t), \ldots, y_N(t)]$, for $t = 1..T$, the input of the model are the $\Omega \in \mathrm{N}^+$ last lags of the time series Y;

2. **Reservoir**: the reservoir consists of a group of $N_{SR} \in \mathrm{N}^+$ sub-reservoirs represented by random multivariate HFCM-FTS termed Random MVFCM-FTS. Each random MVFCM-FTS layer is fed by all the variables in the external multivariate time series from the input layer;

3. **Output Layer**: The obtained independent outputs from each sub-reservoir (represented by $\hat{y}_{sub(j)}(t+1)$,$j = 1 \ldots N_{SR}$) are fed to the output layer;

4. **Final Result**: Shows the final predicted values for time $t+1$.

As Figure 20 displays, the reservoir structure of the proposed model consists of a number of sub-reservoirs with no links between them, unlike deep learning models with hierarchical stacking structure.

In the proposed MO-RHFCM model the weights inside each sub-reservoir are randomly initialized similarly to the R-HFCM model. More specifically, it can be said that MO-RHFCM is the extended form of the R-HFCM model. Thus, MO-RHFCM is also a type of ESN in which only the output layer is trainable whereas the reservoir parameters are initialized randomly and remain fixed during the training process. It means that, in our model, the reservoir weights are initialized based on the ESP condition in the ESN reservoir computing, and then the least squares minimization algorithm is applied to the output units to generate the final prediction.

The proposed MO-RHFCM method is divided into two procedures: the Training procedure and the Forecasting procedure, detailed in sections 5.1.1 and 5.1.2.

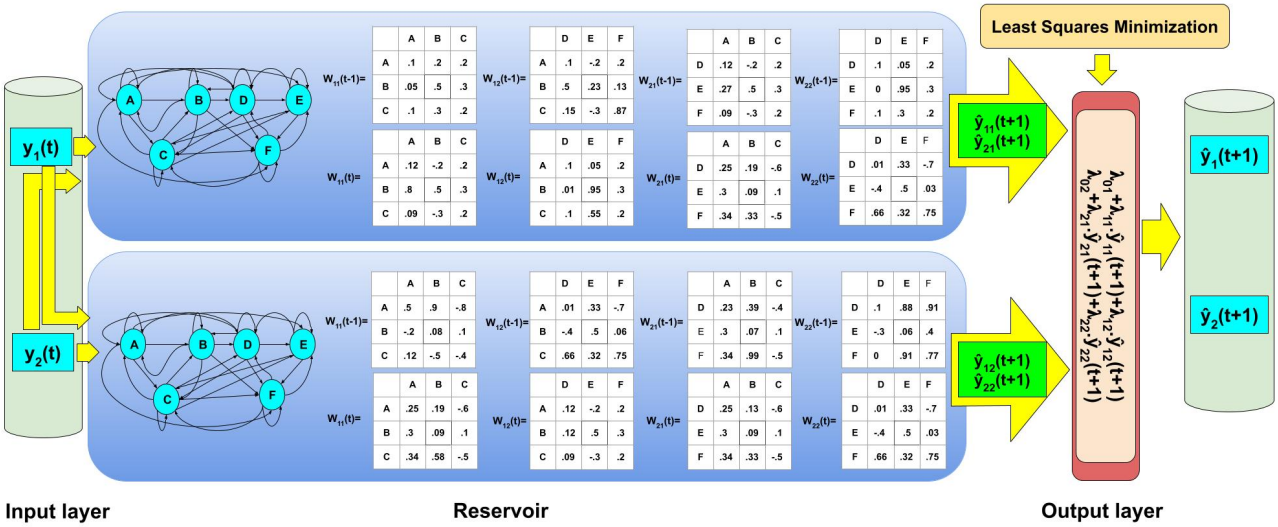


Figure 21 – A simple example of the proposed MO-RHFCM mechanism considering $N = 2$, $N_{SR} = 2$,$\Omega = 2$ and $k = 3$. Firstly, the original time series is fed to the reservoir. Interestingly, each variable is presented by one FCM with $k = 3$ concepts. A, B, and C are used to model variable y_1 while variable y_2 is modeled by another FCM with D, E, and F. Since $\Omega = 2$, eight randomly weight matrices are set in each sub-reservoir. After that, the generated outputs are injected into the output layer. Finally, the least squares algorithm is applied to train the output layer and produce the final predicted values.

5.1.1 Training Procedure

The main aim of the training procedure is to find appropriate weights for the output layer given a crisp training set. It means that the training dataset is fed to each sub-reservoir to generate the inputs for the output layer. Then, the least-squares algorithm is used to find the optimal least-squares coefficients. It can be said that the training procedure is divided into two steps: constructing the structure of each sub-reservoir and determining the least squares coefficients as detailed in the following.

5.1.1.1 Randomized Model Initialization

Each matrix $\mathbf{W}^t$, for $t = 1, \ldots, \Omega$, is a $k \times k$ matrix where, $w_{ij}^t \in \mathbb{R}$ is the weight between the concepts C_i and C_j at the time lag t, and Ω is the order of the model. As mentioned earlier, in the proposed model the weight matrices of each MVFCM-FTS are randomly chosen from a uniform distribution over the interval [-1,1]. Then, they are scaled according to the ESN reservoir computing to preserve the ESP condition as described by the equation 4.5. It is worth observing that each sub-reservoir (or MVFCM-FTS) consists of $N^2 \times \Omega$ weight matrices with $k \times k$ dimensions. For instance, Figure 21 shows a simple structure of our proposed MO-RHFCM model considering $N_{SR} = 2$,$\Omega = 2$ and $k = 3$. Suppose a time series with two variables $Y(t) = [y_1(t), y_2(t)]$ which are fed to both sub-reservoirs. Each variable in each sub-reservoir is represented by one FCM using three concepts. Concepts A, B, and C with their relations are used to represent $y_1(t)$ while the concepts D, E, and F and their relations are used to present $y_2(t)$. Thus, we have eight (or

$2^2 \times 2$) randomly weight matrices with the dimensions of 3×3. Note that the map size for each variable is chosen equally in our model. $\mathbf{W_{11}}$ represents weight matrix for $y_1(t)$, $\mathbf{W_{22}}$ represents weight matrix for $y_2(t)$, $\mathbf{W_{12}}$ and $\mathbf{W_{21}}$ demonstrate weight matrix among both variables. Also, it is worth noting that the model contains the bias weights which are computed using equation 4.6 while they are discarded from Figure 21 just because of the ease of notation.

5.1.1.2 Partitioning

Since MO-RHFCM is composed of a group of MVFCM-FTS models, they must share the same linguistic variables, i.e., for each $y_i \in Y$ the group of fuzzy sets defined over the Universe of Discourse $U_{y_i} = [\min(y_i), \max(y_i)]$.

In this step, U_{y_i} is split into k even length, evenly spaced, overlapped intervals, where, mp_i, $i = 1..k$, are the midpoints of each interval. Then, a fuzzy set C_i (i.e. a concept) is defined with midpoint mp_i and a membership function μ_{C_i}. In each sub-reservoir, a group of the k concepts form the linguistic variable C, such that $C_i \in C$, $\forall i = 1, \ldots, k$. Here grid partitioning is used and the number of concepts is the same as the number of partitions.

Figure 4 shows a simple example of the FCM structure used for each variable in this model, with triangular membership function μ_{C_i}.

5.1.1.3 Fuzzification

In this stage, the values of the original numerical time series ($Y(t)$) are transformed to a fuzzy series A which represents the activation state $a_i(t)$ of each concept $C_i \in C$ for each time series sample. Then, for each sample in each time series $y(t) \in y_i(t)$ the corresponding fuzzified sample $a(t) \in A$ is the set of activations $a_i(t) \in a(t)$ such that $a_i(t) = \mu_{C_i}(y(t))$, for $i = 1, \ldots, k$.

5.1.1.4 Activation

In this proposed model the state values of the concepts of each variable depend on the state values of the related concepts and weights among the concepts of the associated variable as well as the state values and causal relations among the concept of the relevant variable and other variables. Thus, for each MVFCM-FTS with index $j = 1 \ldots N_{SR}$, the state value of each concept for each variable (y_1, y_2,..., y_N) in time $t + 1$ can be defined by the equations (5.1) to (5.3).

$$\mathbf{a}_{y_1(j)}(t+1) = f\left(\mathbf{w}^{01} + \sum_{l=1}^{\Omega} \mathbf{W_{11}^l} \cdot \mathbf{a_{y_1}}(t-l+1) + \mathbf{W_{12}^l} \cdot \mathbf{a_{y_2}}(t-l+1) + ... + \mathbf{W_{1N}^l} \cdot \mathbf{a_{y_N}}(t-l+1)\right) \quad (5.1)$$

$$\mathbf{a}_{y_2(j)}(t+1) = f\left(\mathbf{W}^{02} + \sum_{l=1}^{\Omega} \mathbf{W_{21}^l} \cdot \mathbf{a_{y_1}}(t-l+1) + \mathbf{W_{22}^l} \cdot \mathbf{a_{y_2}}(t-l+1) + ... + \mathbf{W_{2N}^l} \cdot \mathbf{a_{y_N}}(t-l+1)\right) \quad (5.2)$$

. . .

$$\mathbf{a}_{y_N(j)}(t+1) = f\left(\mathbf{W}^{0N} + \sum_{l=1}^{\Omega} \mathbf{W_{N1}^l} \cdot \mathbf{a_{y_1}}(t-l+1) + \mathbf{W_{N2}^l} \cdot \mathbf{a_{y_2}}(t-l+1) + ... + \mathbf{W_{NN}^l} \cdot \mathbf{a_{y_N}}(t-l+1)\right) \quad (5.3)$$

where $\mathbf{w}^{01}, \mathbf{w}^{02}, ..., \mathbf{w}^{0N}$ are bias weight vectors, $\mathbf{W_{11}}, \mathbf{W_{22}}, ..., \mathbf{W_{NN}}$ represent the weight matrices among the concepts of variables y_1, y_2,..., y_N, respectively. $\mathbf{W_{12}},\ldots,\mathbf{W_{1N}}$ respectively represent the causal relations among the concepts of variable y_1 and the concepts of variables y_2,y_3,...,y_N. The relations among the concepts of variable y_2 and the concepts of variables y_1,y_3,...,y_N are respectively represented by $\mathbf{W_{21}},\mathbf{W_{23}},\ldots,\mathbf{W_{2N}}$, and $\mathbf{W_{N1}},\mathbf{W_{N2}},\ldots,\mathbf{W_{N(N-1)}}$ present the relations among the concepts of variable y_N and the concepts of variables y_1,y_2,...,y_{N-1}. Based on these equations, the activation level of the nodes of a specific variable at the moment $t+1$ depends on the activation degree of all concepts of this variable at $\{t, t-1, .., t-\Omega+1\}$ moments as well as the state values of the nodes of other variables at $\{t, t-1, .., t-\Omega+1\}$ moments. Also, it is worth reminding that the number of concepts for each variable is determined equally so that all the weight matrices have the same dimensions $(k \times k)$.

5.1.1.5 Defuzzification

After calculating the activation level of each concept, in this step, the defuzzification is carried out to produce the output related to each sub-reservoir. Therefore, the forecast values for each sub-reservoir ($\hat{y}_{sub(j)}(t+1)$, $j = 1 \ldots N_{SR}$) at time $t+1$ in numeric terms can be calculated via the equations (5.4) to (5.6).

$$\hat{y}_{1j}(t+1) = \hat{y}_{y_1(j)}(t+1) = \frac{\sum_{i=1}^{k} a_{y_1(j)i}(t+1) \cdot mp_i}{\sum_{i=1}^{k} a_{y_1(j)i}(t+1)} \quad (5.4)$$

$$\hat{y}_{2j}(t+1) = \hat{y}_{y_2(j)}(t+1) = \frac{\sum_{i=1}^{k} a_{y_2(j)i}(t+1) \cdot mp_i}{\sum_{i=1}^{k} a_{y_2(j)i}(t+1)} \quad (5.5)$$

...

$$\hat{y}_{Nj}(t+1) = \hat{y}_{y_N(j)}(t+1) = \frac{\sum_{i=1}^{k} a_{y_N(j)i}(t+1) \cdot mp_i}{\sum_{i=1}^{k} a_{y_N(j)i}(t+1)} \quad (5.6)$$

where $a_{y_1(j)}(t+1)$,$a_{y_2(j)}(t+1)$,$\ldots$,$a_{y_N(j)}(t+1)$ are the activation calculated from the previous step for each concept at time $t+1$ and mp_i is the center of each concept $C_i \in C$. Simply, the equation (5.7) is used to generate the predicted values for $\hat{y}_{y_v(j)}(t+1)$, $v = 1 \ldots N$,$j = 1 \ldots N_{SR}$.

$$\hat{y}_{y_v(j)}(t+1) = \frac{\sum_{i=1}^{k} a_{y_v(j)i}(t+1) \cdot mp_i}{\sum_{i=1}^{k} a_{y_v(j)i}(t+1)} \tag{5.7}$$

Accordingly, the predicted outputs for each sub-reservoir directly depend on the dimension (N) of the original injected input to our model. Hereupon, the reservoir layer generates $N \times j$ predicted values which are fed to the output layer.

5.1.1.6 Least Squares coefficients determination

Given the outputs $\hat{y}_{y_v(j)}(t+1)$, $v = 1 \dots N$,$j = 1 \dots N_{SR}$ for each input sample $y(t) \in y_v(t)$, $t = 1 \dots T$, a design matrix $X \in \mathbb{R}^{N_{SR} \times T}$ is created for each variable to represent the linear system $Y = \lambda X$, where $\lambda = [\lambda_0, \dots, \lambda_j]$ is the coefficient vector. Thus, $\lambda_{tot} = [\lambda_{01}, \lambda_{11}, \dots, \lambda_{1N}, \lambda_{02}, \lambda_{21} \dots, \lambda_{2N}, \dots, \lambda_{0N}, \lambda_{j1}, \dots, \lambda_{jN}]$ form the whole set of coefficients in our proposed model with respect to the number of variables as well as the size of reservoir. Then, the Least Squares method is employed to solve this linear system and find the λ_{tot} vector of coefficients that minimizes the Mean Squared Error.

For instance Figure 21 demonstrates a simple structure of the proposed MO-RHFCM technique considering $N = 2$ and $N_{SR} = 2$. Therefore, each sub-reservoir generates two outputs. In this case, the least squares learning method is employed to solve two equations to find six coefficients ($N \times N_{SR} + 2$ biases=6).

5.1.2 Forecasting Procedure

The main goal of this stage is to compute the predicted crisp values $\hat{y}_{sub}(t+1)$ of each sub-reservoir as well as the final predicted values $\hat{y}_1(t+1)$,$\hat{y}_2(t+1)$,. . .,$\hat{y}_N(t+1)$, given the linguistic variables C, the weight matrices $\mathbf{W}$, the activation function f and the numerical input time series $Y(t)$. The steps of the forecasting procedure are fully described as follows.

5.1.2.1 Fuzzification

The same as Step 3 in the Training procedure.

5.1.2.2 Activation

The same as Step 4 in the Training procedure.

5.1.2.3 Defuzzification

1. Sub-reservoir forecasts defuzzification: The same as Step 5 in the Training procedure.
2. Final forecast defuzzification:

   The predicted value at time $t+1$ is computed through the linear combination of the obtained outputs of all sub-reservoirs and the least squares coefficients. The final predicted values for each variable $(y_v, v = 1, \ldots, N)$ is described by the equation (5.8).

$$\hat{y}_v(t+1) = \lambda_{0v} + \lambda_{v1} \cdot \hat{y}_{v1}(t+1) + \lambda_{v2} \cdot \hat{y}_{v2}(t+1) + \ldots + \lambda_{vN_{SR}} \cdot \hat{y}_{vN_{SR}}(t+1) \quad (5.8)$$

   which can be formulated as follows:

$$\hat{y}_v(t+1) = \lambda_{0v} + \sum_{j=1}^{N_{SR}} \lambda_{vj} \cdot \hat{y}_{vj}(t+1) \quad (5.9)$$

   For instance, as highlighted in Figure 21, the final predicted values for each variable are calculated via the equations (5.10) and (5.11).

$$\hat{y}_1(t+1) = \lambda_{01} + \lambda_{11} \cdot \hat{y}_{11}(t+1) + \lambda_{12} \cdot \hat{y}_{12}(t+1)) \quad (5.10)$$

$$\hat{y}_2(t+1) = \lambda_{02} + \lambda_{21} \cdot \hat{y}_{21}(t+1) + \lambda_{22} \cdot \hat{y}_{22}(t+1)) \quad (5.11)$$

   where $\{\lambda_{01}, \lambda_{11}, \lambda_{12}, \lambda_{02}, \lambda_{21}, \lambda_{22}\}$ denote the obtained least square coefficients from training procedure, $\hat{y}_{11}(t+1)$ and $\hat{y}_{21}(t+1)$ are the generated outputs for the first sub-reservoir, $\hat{y}_{12}(t+1)$ and $\hat{y}_{22}(t+1)$ are the generated outputs for the second sub-reservoir and $\hat{y}_1(t+1)$ and $\hat{y}_2(t+1)$ are the final predicted values for $y_1(t)$ and $y_2(t)$ respectively. In addition, it is worth observing that $\hat{y}_{11}(t+1)$ and $\hat{y}_{12}(t+1)$ are measured using the equation 5.4 and $\hat{y}_{21}(t+1)$ and $\hat{y}_{22}(t+1)$ are measured using the equation 5.5.

5.2 Computational Experiments

This section is mainly focused on evaluating the accuracy of the proposed MO-RHFCM approach. The first subsection provides a brief description of the case studies as well as evaluation metrics. Then, the efficiency of the proposed MO-RHFCM model is verified by applying two low-dimensional time series. All the experiments are implemented with Python 3.6.12 using open-source packages such as Scikit-Learn, Keras, Tensorflow, PyTorch, Pandas, Numpy, and pyFTS.

5.2.1 Case studies

To assess the efficacy of the proposed MO-RHFCM technique, in this experiment, two data sets with two variables including the Brazilian SONDA dataset and the Malaysian dataset have been employed as explained earlier in section 4.3.1. In the case of SONDA, the model was applied to predict solar radiation and wind speed in Brazil. In the Malaysia dataset, we used the proposed model to predict electrical load consumption and temperature in the city of Johor in Malaysia.

It is worth noting that some pre-processing techniques are applied to clean the original data set by removing outliers and missing values. Also, it is worth observing that 8760 samples of each variable of each dataset are used in our experiments.

5.2.2 Experimental Methodology

To evaluate the performance of the proposed methodology, we utilized two comparison criteria of the forecasting accuracy including RMSE described in equation (3.6), Mean Absolute Error (MAE) described in equation (5.12), NRMSE described in equation (4.14), and the coefficient of determination (R^2), described in equation (5.13), where RSS stands for the residual sum of squares, TSS stands for a total sum of squares, $y(t)$ and $\hat{y}(t)$ stand for the actual and forecast values, respectively.

$$MAE = \frac{1}{T}\sum_{i=1}^{T} \mid y_i - \hat{y}_i \mid \tag{5.12}$$

$$R^2 = 1 - \frac{RSS}{TSS} = 1 - \frac{\sum_{i=1}^{T}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{T}(y_i - \bar{y})^2} \tag{5.13}$$

These forecasting accuracy metrics were calculated using the sliding window cross-validation technique, where 90% of each window is used as a training set and 10% for testing. For each variable $y_i(t) \in Y$ of each data set $d_j \in \mathcal{D}$, the samples were divided into 35 windows $w \in W$. For each window, we train and test the models $m \in \mathcal{M}$ using respectively the training and test subsets. Noteworthy that the accuracy of the models is evaluated over the test data computing the average values of accuracy metrics obtained for all 35 windows.

To compare the performance of the models statistically, the Kruskal-Wallis test was conducted with the confidence level of $\alpha = 0.05$ in terms of the average NRMSE of all variables in each data set for each window. In this method, the null hypothesis H_0 stands for the equality of average NRMSE errors of all methods and H_1 presents an alternative hypothesis such that at least one of the averages is different from the others. When H_0 is

Table 27 – The summary of the best performance of the model for SONDA dataset considering different k, N_{SR} and Ω

k	Ω	N_{SR}	Variable	RMSE	MAE	R2	NRMSE
3	5	100	glo_avg	**95.011**	**59.199**	**0.895**	**0.096**
			ws_10m	**0.339**	**0.257**	**0.682**	0.120
4	5	100	glo_avg	96.224	59.896	0.893	0.097
			ws_10m	0.341	0.259	0.675	**0.119**
5	5	100	glo_avg	96.783	61.449	0.892	0.098
			ws_10m	0.340	0.259	0.678	0.120
6	4	100	glo_avg	99.500	60.997	0.886	0.100
			ws_10m	0.339	**0.257**	**0.682**	**0.119**
7	4	80	glo_avg	100.365	62.869	0.884	0.101
			ws_10m	0.338	0.257	**0**.682	0.119
8	4	100	glo_avg	101.021	62.815	0.883	0.102
			ws_10m	0.343	0.261	0.674	0.120
9	4	100	glo_avg	101.711	63.842	0.881	0.103
			ws_10m	0.343	0.261	0.672	0.120
10	4	80	glo_avg	102.822	65.317	0.879	0.104
			ws_10m	0.341	0.259	0.678	**0.119**
15	3	80	glo_avg	108.577	66.843	0.865	0.110
			ws_10m	0.348	0.263	0.664	0.122
20	2	80	glo_avg	114.156	67.826	0.853	0.116
			ws_10m	0.355	0.265	0.655	0.124
30	2	100	glo_avg	122.112	70.493	0.829	0.123
			ws_10m	0.369	0.272	0.611	0.129
40	2	100	glo_avg	156.429	96.557	0.730	0.160
			ws_10m	0.392	0.299	0.571	0.137

not accepted, it is needed to apply *post hoc* tests to compare the equality of each pair of means. In this investigation, we selected the Wilcoxon approach for the *post hoc* test.

5.3 Results

The goal of this section is to present the obtained experimental results of our proposed model and to compare our approach against other forecasting techniques that are tested over the mentioned case studies.

This section is divided into four sub-sections. Subsection 5.3.1 is focused on presenting a discussion and analysis of our proposed model's hyperparameters to reach the best forecasting accuracy. The second subsection 5.3.2 provides a brief introduction to the baseline models as well as hyperparameter tuning. Subsection 5.3.3 is responsible for comparing the accuracy of the proposed model with other baseline models. Finally, the models are compared statistically in subsection 5.3.4.

5.3.1 Parameter setting for MO-RHFCM

This part is organized to analyze the influence of effective hyper-parameters on the accuracy of the proposed method. The number of concepts (k), the number of layers or sub-reservoirs (N_{SR}), and the order value (Ω) are the most influential parameters on the model forecasting accuracy.

The experiments are executed with $k \in \{3, 4, 5, 6, 7, 8, 9, 10, 15, 20, 30, 40\}$, $N_{SR} \in \{2, 5, 10, 15, 20, 30, 40, 60, 80, 100\}$ and $\Omega \in \{2, 3, 4, 5, 6\}$. Since the effectiveness of the proposed model is sensitive to the proper choice of these hyper-parameters, the goal is to set them optimally to achieve the best performance in terms of prediction accuracy. The best configuration of these hyperparameters is obtained empirically (through trial and error) based on the training data set. More specifically, models are executed by varying these hyper-parameters. Also, it is worth mentioning that in this investigation the activation function in FCM is set to the sigmoid function.

Tables 27 and 28 summarize the obtained results with the optimal combinations of the parameters adjusted separately for each variable for SONDA and Malaysia time series datasets, respectively. In accordance with the results recorded in these tables, it is clear that the best performance of the model is achieved by increasing the size of the reservoir ($N_{SR} \in \{80, 100\}$). In contrast, there is a reverse relation between map size and forecasting accuracy. To be more precise, the optimal forecasting accuracy is obtained with the least number of concepts.

To have a better comparison, Table 29 represents a summary of the forecasting accuracy of the model in terms of average NRMSE. As can be seen, for both cases, the most accurate results are obtained only with $k = 3$, although there are very slight differences in the performance of models considering $k \in \{3, 4, 5\}$. In addition, the results demonstarte that the accuracy of the model is degraded as the size of the map increases.

5.3.2 Baseline models

To confirm the validity of our proposed model, a comparison against some common machine learning and deep learning models is reported. Hereupon, eight baseline models including popular machine learning and deep learning methods are coded which are briefly described. Then, the results of the proposed model are compared with the results of these competitor models in subsection 5.3.3.

In this investigation, two different types of RNNs Long Short-term Memory (LSTM) and Gated Recurrent Unit (GRU) as well as Multilayer perceptron (MLP) as a fully connected class of feedforward artificial neural network are used as competitor models which are implemented using Keras framework.

Table 28 – The summary of the best performance of the model for Malaysia dataset considering different k, N_{SR} and Ω

k	Ω	N_{SR}	Variable	RMSE	MAE	R2	NRMSE
3	5	100	temperature load	**0.886** **1958.516**	**0.598** **1401.437**	**0.894** **0.981**	**0.088** **0.044**
4	5	100	temperature load	0.894 1962.226	0.604 1412.620	0.892 0.980	0.089 **0.044**
5	5	100	temperature load	0.889 2011.188	0.606 1440.706	0.893 0.979	0.089 0.045
6	3	100	temperature load	0.895 2164.534	0.609 1564.875	0.892 0.976	0.089 0.049
7	5	100	temperature load	0.896 2098.788	0.613 1519.511	0.891 0.978	0.089 0.047
8	3	100	temperature load	0.893 2148.369	0.608 1550.375	0.893 0.977	0.089 0.048
9	3	100	temperature load	0.905 2285.157	0.615 1597.223	0.890 0.972	0.090 0.051
10	3	100	temperature load	0.907 2397.055	0.618 1673.326	0.889 0.968	0.090 0.054
15	3	100	temperature load	0.916 2423.465	0.628 1738.890	0.887 0.969	0.091 0.054
20	2	100	temperature load	0.980 2623.41	0.646 1817.464	0.861 0.964	0.098 0.059
30	2	100	temperature load	1.022 3231.658	0.659 1987.116	0.844 0.923	0.102 0.072
40	2	100	temperature load	1.038 3249.561	0.674 2028.324	0.818 0.917	0.104 0.073

Table 29 – Evaluation of the model's performance in terms of average NRMSE

Dataset	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$	$k=9$	$k=10$	$k=15$	$k=20$	$k=30$	$k=40$
SONDA	**0.108**	0.1085	0.109	0.1095	0.110	0.111	0.1115	0.1115	0.1160	0.120	0.126	0.1485
Malaysia	**0.066**	0.0665	0.067	0.158	0.0755	0.068	0.073	0.0695	0.0925	0.0765	0.075	0.071

Gradient Boosting (GB), eXtreme Gradient Boosting (XGB), and Light Gradient Boosting machine (LGBM) are the other popular machine learning regressor algorithms that are coded using the Sklearn library as baselines in this work.

Additionally, Random Forest (RF) (BREIMAN, 2001) and Extremely Randomized Trees(Extra Trees) (GEURTS; ERNST; WEHENKEL, 2006) were selected as other baseline models implemented using the Sklearn library. Table 30 reports the optimal values of hyper-parameters for these models using the Randomized Search technique.

5.3.3 Comparison against baselines

The key objective of this section is to make a comparison between the accuracy of the proposed MO-RHFCM approach and baseline techniques which are detailed in Table 31.

Table 30 – Optimal parameters of each dataset for baseline models

Model	Hyperparameters	SONDA	Malaysia
LGBM	Max depth	9	23
	estimator	20	112
	Num.leaves	21	49
	learning rate	0.2	0.05
GB	Max depth	2	4
	estimator	100	72
	min samples leaf	5	9
	min samples split	3	19
	learning rate	0.5	0.2
XGB	Max depth	5	5
	estimator	100	20
	gamma	2	3
	reg alpha	60	140
	learning rate	0.05	0.9
GRU	activation function	ReLU	ReLU
	batch size	16	16
	neuron	67	85
LSTM	activation function	ReLU	ReLU
	batch size	32	16
	neuron	75	58
MLP	activation function	ReLU	ReLU
	batch size	64	64
	neuron1	63	28
	neuron2	13	23
	neuron3	39	91
RF	max depth	27	11
	min leaf nodes	30	20
	min samples leaf	3	9
	min samples split	7	10
	estimator	100	300
Extra Trees	max depth	12	26
	min samples leaf	8	4
	min samples split	10	2
	estimator	300	50

The reported results in Table 31 confirm the outperformance of our proposed MO-RHFCM model compared with other baseline techniques. However, in a few cases, MO-RHFCM is a loser with very slight differences. Thus, as can be observed, ranking the methods with respect to the obtained results is difficult due to the unstable forecasting accuracy of the methods for each variable. As such, the average NRMSE over all variables for every data set has been calculated to compare the methodologies more precisely as shown in Table 32. This Table suggests the superior performance of our proposed method followed by Extra Trees for both case studies.

5.3.4 Statistical testing

As explained in section 5.2.2, Kruskall-Wallis is utilized to statistically compare the MO-RHFCM technique with other baseline techniques. According to the listed p-values

Table 31 – Evaluation of the accuracy of the proposed method against other baseline models

Dataset	**SONDA**					**Malaysia**				
Model	Variable	RMSE	MAE	R^2	NRMSE	Variable	RMSE	MAE	R^2	NRMSE
MO-RHFCM(k=3)	glo_avg	**95.011**	59.199	**0.895**	**0.096**	temperature	0.886	**0.598**	**0.894**	**0**.088
	ws_10m	**0.339**	**0.258**	**0.682**	0.120	load	**1958.516**	**1401.437**	**0.981**	**0.044**
MO-RHFCM(k=4)	glo_avg	96.224	59.896	0.893	0.097	temperature	0.894	0.604	0.892	0.089
	ws_10m	0.341	0.259	0.675	0.120	load	1962.226	1412.620	0.980	**0.044**
MO-RHFCM(k=5)	glo_avg	96.783	61.449	0.892	0.098	temperature	0.900	0.613	0.890	0.089
	ws_10m	0.340	0.259	0.678	0.120	load	2149.452	1549.505	0.976	0.048
GB	glo_avg	112.402	63.931	0.856	0.114	temperature	0.938	0.643	0.881	0.093
	ws_10m	0.360	0.274	0.640	0.126	load	2383.030	1699.112	0.971	0.053
XGB	glo_avg	125.851	80.644	0.825	0.128	temperature	1.214	0.879	0.803	0.120
	ws_10m	0.417	0.326	0.519	0.146	load	2899.048	2074.125	0.958	0.065
LGBM	glo_avg	104.549	57.285	0.873	0.106	temperature	0.900	0.605	0.886	0.091
	ws_10m	0.343	0.260	0.674	0.120	load	2167.382	1509.273	0.978	0.048
LSTM	glo_avg	108.087	67.365	0.867	0.109	temperature	2899.048	2074.125	0.887	0.091
	ws_10m	0.338	0.258	0.684	**0.119**	load	0.914	0.645	0.963	0.062
GRU	glo_avg	113.196	75.026	0.853	0.115	temperature	0.901	0.628	0.890	0.09
	ws_10m	0.344	0.261	0.670	0.121	load	2650.018	1938.483	0.965	0.06
MLP	glo_avg	108.880	67.858	0.864	0.110	temperature	**0.884**	0.603	0.894	0.088
	ws_10m	0.340	0.258	0.682	0.119	load	2392.469	1679.621	0.971	0.054
RF	glo_avg	103.224	**55.545**	0.877	0.104	temperature	0.926	0.621	0.884	0.092
	ws_10m	0.347	0.262	0.667	0.121	load	2630.708	1886.509	0.965	0.059
Extra Tree	glo_avg	101.643	57.645	0.881	0.103	temperature	0.896	**0.598**	0.891	0.089
	ws_10m	0.343	0.259	0.676	0.120	load	2077.621	1424.353	0.978	0.047

Table 32 – Evaluation of the models' accuracy in terms of average NRMS over all variables for each dataset

Dataset	$k=3$	$k=4$	$k=5$	RF	Extra Trees	GB	LGBM	XGB	LSTM	GRU	MLP
SONDA	**0.108**	0.1085	0.109	0.1125	0.1115	0.120	0.1130	0.1370	0.1140	0.1180	0.1145
Malaysia	**0.066**	0.0665	0.067	0.0755	0.068	0.073	0.0695	0.0925	0.0765	0.075	0.071

and test statistics in Table 33, H_0 has been rejected for all data sets. Accordingly, *post-hoc* test is employed to compare all the models. The obtained results from the Wilcoxon test indicate the superior performance of the proposed MO-RHFCM model in comparison to other baseline models. More transparently, a summary of the statistical ranking of the models for each data set according to the obtained results obtained from the Wilcoxon test is reported in Table Table 34 based on the times in which H_0 can not be rejected. As Table 34 details, the first rank belongs to the MO-RHFCM model for both case studies followed by Extra Trees.

Table 33 – Kruskal-Wallis mean comparison test results

data set	**Statistic**	**p-value**	**Result**
Malaysia	108.470	7.850e-20	H_0 is rejected
SONDA	56.541	2.212e-09	H_0 is rejected

Table 34 – The summary of the ranking of the forecasting models

Dataset	MO-RHFCM(k=3)	RF	Extra Trees	GB	LGBM	XGB	LSTM	GRU	MLP
Malaysia	**1**	6	2	5	3	9	7	6	4
SONDA	**1**	3	2	8	3	9	3	7	5

In summary, it can be claimed that our proposed model focused on two main aspects of FCM-based time series forecasting models by designing a new FCM structure

as reservoir computing to predict multiple outputs where the training phase is accelerated. The obtained results showed the proposed model is robust with high-performance accuracy in comparison to other ML and DL models. Also, as can be seen in Table 35, our method is less complex, cheaper, and more parsimonious than deep learning models due to the lower number of parameters. In our model, the number of parameters is only determined by the number of least squares coefficients calculated by $(N \times N_{SR}) + N_{SR} = 2 \times 100 + 100 = 300$.

Table 35 – Comparison among the number of parameters of our proposed model with deep learning methods

dataset	MO-RHFCM	LSTM	GRU
Malaysia	**300**	14038	22612
SONDA	**300**	23100	2430

5.4 Chapter's Highlights

In this chapter, a new hybrid forecasting method was implemented to predict low-dimensional multivariate time series referred to as MO-RHFCM. This method is inspired by a combination of the concepts of MVFTS, reservoir computing (ESN) and HFCM trained via least squares algorithm as time effective learning technique. MO-RHFCM was suggested as a MIMO FCM-based forecasting model to predict multiple outputs.

Structurally, MO-RHFCM consists of three layers: the input layer, the reservoir, and the output layer. Similar to the R-HFCM technique, the reservoir includes a group of sub-reservoirs presented by random MVFCM-FTS. Each random MVFCM-FTS is fed by all the variables from the input layer. Then, each layer inside the reservoir unit generates independent outputs which are fed to the output layer. The number of outputs from each sub-reservoir directly depends on the number of variables of the original time series. Finally, the least squares algorithm is applied to train the output layer and generate the final predicted values for each variable. It is worth mentioning that MO-RHFCM is a kind of ESN because weights inside each sub-reservoir are initialized randomly and kept constant during the training process. That is the reservoir weights are initialized based on the ESP condition in the ESN reservoir computing, then the least squares minimization algorithm is applied to the output, units to generate the final prediction.

To examine the effectiveness of the proposed MO-RHFCM method, we used two low-dimensional datasets including SONDA and Malaysia datasets only with two variables. The obtained results indicate the validity of our proposed model compared with other baseline machine learning and deep learning methods. Since MO-RHFCM was designed to predict low-dimensional time series, another new MIMO FCM-based technique is introduced in the next chapter to predict high-dimensional time series.

6 PROPOSED MIMO-FCM TECHNIQUE

Despite the remarkable success of FCMs in the field of time series forecasting and analysis, handling large-scale multivariate time series is still one of the main challenging topics in this area. Few studies can be found in this direction, for instance, Shanchao e Liu (2018) developed a wavelet-High Order FCM (wavelet-HFCM) model trained via ridge regression algorithm to predict large-scale multivariate time series.

Likewise, presenting a novel MIMO FCM-based technique with the ability to handle high-dimensional data sets with promising results in Internet of Things (IoT) applications can be considered one of the objectives of this chapter. More clearly, IoT means a worldwide network of interconnected objects uniquely addressable, based on standard communication protocols (GUBBI et al., 2013). In this context, data is recorded from different sensor nodes in the form of streaming time series. Thereby, a high-dimensional time series is generated where each dimension represents the measurements recorded by a sensor node.

Smart buildings and smart cities are considered as the most popular IoT applications, which are employed as case studies in this investigation. It is worth mentioning that several machine learning and deep learning models have been introduced in the literature to enhance the IoT forecasting accuracy (CANDANEDO; FELDHEIM; DERAMAIX, 2017; CHAMMAS; MAKHOUL; DEMERJIAN, 2019; CHAI et al., 2019; SAJJAD et al., 2020; KHAN et al., 2020; PARHIZKAR; RAFIEIPOUR; PARHIZKAR, 2021).

The proposed model has been equipped with least squares regression to predict multiple outputs with promising forecasting accuracy. In terms of structure, the MIMO-FCM model represents a hybrid model composed of cascade randomized HFCMs (CR-HFCM) with respect to the number of principal components. More precisely, MIMO-FCM combines the concepts of HFCM, ESN, reservoir computing, and Kernel Principal Component Analysis (KPCA) (KIM; FRANZ; SCHÖLKOPF, 2005).

In summary, the following contributions are brought to life in this chapter:

1. Developing a new FCM-based forecasting model integrating the concepts of embedding (KPCA), FCM, and ESN termed MIMO-FCM, which is trained with time-effective learning inspired by reservoir computing;

2. Introducing multiple output FCM-based forecasting model to predict high-dimensional time series with more than 100 variables for the first time in the literature according to the best of our knowledge;

3. Validation of MIMO-FCM, which is tested on three high-dimensional IoT data sets. The experimental results demonstrate that the proposed model is efficient and competitive in comparison to the other existing approaches.

The remainder of this chapter is structured as follows: Section 6.1 presents a brief description of the dimensionality reduction; Section 6.2 introduces the proposed method in details; Section 6.3 describes three IoT case studies used to test our methodology; Section 6.4 represents the experimental results and discussion; and finally the conclusions and some possibilities of future work are drawn in Section 6.5.

6.1 Preliminaries

Dimensionality reduction

Different types of dimensionality reduction (embedding) techniques have been introduced in the literature covering feature selection and feature extraction. Despite feature selection techniques in which a subset of the original variables are selected, in feature extraction, a set of new variables is generated through some linear or non-linear mapping techniques from the existing input variables. The goal of feature extraction is to map N-dimensional variables measured (i.e. time series) over T time steps into the reduced M-dimensional feature space with $M \ll N$.

However there exist several techniques used for the purpose of reducing dimensions, PCA and KPCA are the most common strategies which have been widely used and are explained briefly in the following.

PCA is a method to reduce the number of dimensions in a data set while the most significant information is kept. It means that PCA calculates the principal components by projecting high-dimensional data onto its main components of variation known as principal components containing the maximum variance. In summary, PCA contains the following steps: computing the covariance matrix of all variable dimensions, calculating the eigenvectors and eigenvalues of the covariance matrix, and data projection along the eigenvectors.

Since PCA extracts only a reduced linearly uncorrelated set of variables, KPCA is employed as nonlinear PCA to extract nonlinear correlation. Therefore, KPCA can handle nonlinear feature extraction utilizing a proper mapping function (Φ) named the kernel function. The polynomial function, Gaussian radial basis function (RBF), and sigmoid function are examples of kernel functions that can be used in KPCA.

6.2 Proposed MIMO-FCM model

In this investigation, a new FCM-based time series forecasting model is introduced to predict multiple output high-dimensional multivariate time series. Figure 22 presents the general structure of the proposed MIMO-FCM technique.

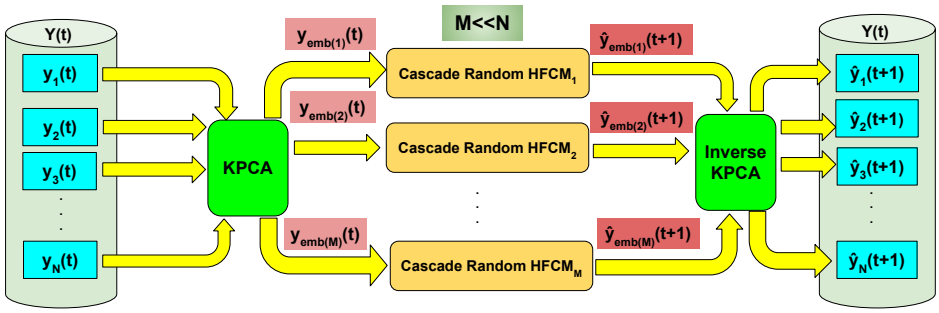


Figure 22 – General structure of the proposed MIMO-FCM method. Firstly, the original data set is transmitted through KPCA to create M principal components. Then each component is predicted by one univariate cascade randomized HFCM unit. Finally, inverse KPCA is applied to reconstruct the original time series.

Our proposed method is mainly focused on two challenging aspects of FCM-based time series forecasting models: designing a new FCM structure as well as speeding up the training phase. As such, in the first step, the general structure of MIMO-FCM is explained, then our strategy for learning FCM is detailed.

Structurally speaking, as shown in Figure 22, the MIMO-FCM model is a combination of KPCA and FCM. In more details, the original data set is transmitted through KPCA data transformation to create M principal components, $\mathbf{y}_{emb} = (y_{emb(1)}, y_{emb(2)}, ..., y_{emb(M)})$, defined by a user in a new feature space such that $M \ll N$ where N is the number of variables of the original time series. After that, the obtained components are fed as inputs to R-HFCM models independently. Then, the predicted outputs $(\hat{y}_{emb(1)},\hat{y}_{emb(2)},\ldots,\hat{y}_{emb(M)})$ from M models are fed to the inverse KPCA operation, which is responsible to calculate the final predicted value for each original feature.

Figure 23 illustrates the internal structure of each univariate model. As can be seen from this figure, each model consists of three layers: input layer, intermediate or reservoir layer, and output layer. Each principal component $(y_{emb(i)}(t),\ i = \{1, 2, ..., M\})$ is injected to each model individually. It means that for each component there will be a corresponding CR-HFCM model. The CR-HFCM is composed of successive randomized HFCM block reservoirs (L). In this structure, $y_{ebm(M)}(t)$ only is fed to the first block, and its calculated output is considered as input for the second block as well as the output layer. This process is repeated for all subsequent blocks which means that the output of the previous block $(L-1)$ is considered as both the input of the next one (L) and the input for the output layer. It is worth mentioning that L acts in the same role as N_{SR} in

our previous models.

Regarding the training phase, since the structure of each sub-reservoir in the reservoir layer is designed based on the R-HFCM model, as introduced in Orang et al. (2022), the model is mainly concentrated on training the output layer using least squares regression. From another perspective, it can be said that the model is a kind of ESN in which only the output layer is trainable, while the reservoir parameters are initialized randomly and remain unalterable during the training process.

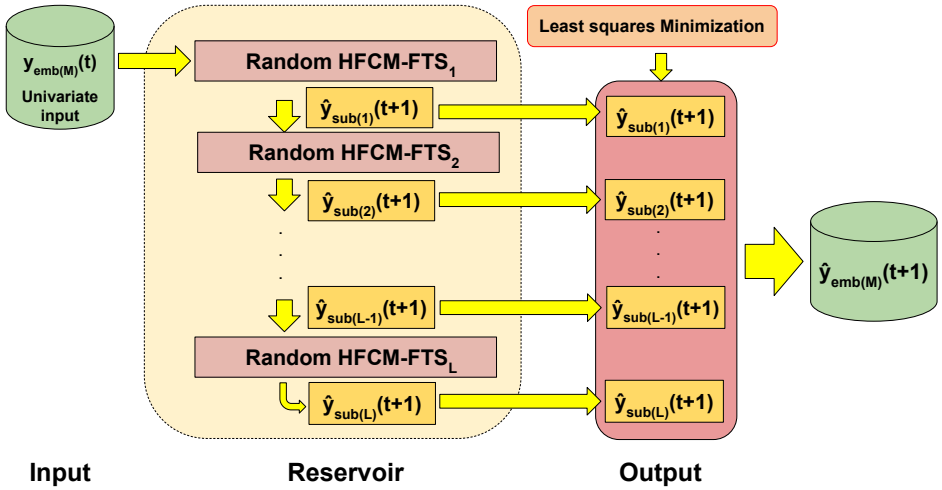


Figure 23 – Internal structure of each univariate CR-HFCM unit. Each CR-HFCM is composed of three layers: the input layer, the reservoir, and the output layer. The reservoir consists of L sub-reservoirs in which weights are initialized randomly so that the Echo State Property (ESP) condition is satisfied in the ESN reservoir computation. Each extracted principal component is only fed into the first block to generate the input for both the second block and the output layer and so on. Accordingly, the generated output from each randomized HFCM block is injected as input to the next successive block as well as to the output layer. Finally, the least squares algorithm is employed to train the output layer and generate the predicted value for each principal component.

On the other hand, the weights of each sub-reservoir in the reservoir layer are chosen randomly to meet the ESP condition in ESN and then each sub-reservoir generates its output independently ($\hat{y}_{sub(1)}(t+1)$,$\hat{y}_{sub(2)}(t+1)$,...,$\hat{y}_{sub(L)}(t+1)$, where $L = N_{SR}$). The least squares minimization algorithm is applied to the output units to generate the final predicted value for each principal component ($\hat{y}_{emb(1)}(t+1)$,$\hat{y}_{emb(2)}(t+1)$,...,$\hat{y}_{emb(M)}(t+1)$). Finally, based on Figure 22, Inverse KPCA is exploited to reconstruct and predict all the original variables ($\hat{y}_1(t+1)$,$\hat{y}_2(t+1)$,...,$\hat{y}_N(t+1)$) from the predicted principal components ($\hat{y}_{emb(1)}(t+1)$,$\hat{y}_{emb(2)}(t+1)$,...,$\hat{y}_{emb(M)}(t+1)$).

The proposed MIMO-FCM method is divided into two procedures: the Training procedure and the Forecasting procedure. These procedures are detailed in the next sections.

6.2.1 Training Procedure

The main objective of the training procedure is to train the output layer in each CR-HFCM to find the least square coefficients, given a crisp training embedded data set. It means that the training data set from each component is fed to the CR-HFCM model to generate the inputs for the output layer. Then, the least square algorithm is employed to train the output layer to find the optimal least squares coefficients. Accordingly, the training procedure is divided into two steps: constructing the structure of each block(sub-reservoir) in each CR-HFCM and determining the least squares coefficients as described in the following.

6.2.1.1 Embedding

In this stage, RBF kernel PCA is applied to extract M principal components from the original high-dimensional time series ($Y = [y_1(t), y_2(t), \ldots, y_N(t)]$ where $Y \in \mathbb{R}^{T\times N}$) accordingly to the following steps. Firstly, the kernel similarity matrix is constructed with the equation (6.1).

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \parallel \mathbf{x}_i - \mathbf{x}_j \parallel_2^2) \tag{6.1}$$

where γ is the kernel coefficient. Since it is not guaranteed that the kernel matrix is centered, in the second phase the centered kernel matrix $\widetilde{M} = M - 1_T M - K 1_M + 1_T M 1_T$ is computed where 1_T is an $T \times T$ matrix with all values equal to $1/T$. The third step includes eigenvalue problem-solving based on $\tilde{M}\alpha_i = \lambda_i T \alpha_i$. Finally, eigenvectors of the centered kernel matrix which correspond to the largest eigenvalues, are obtained.

After applying KPCA according to the above-mentioned process, each component in $Y_{emb} = (y_{emb(1)}(t), y_{emb(2)}(t), \ldots, y_{emb(M)}(t))$, where $Y_{emb} \in \mathbb{R}^{T\times M}$, is fed to the first layer in the reservoir of CR-HFCM individually, as we discussed earlier.

6.2.1.2 Randomized model initialization

The goal of this step is to initialize the weight matrix for each sub-reservoir in CR-RHFCM in Figure 23. It is worth mentioning that the weights and bias inside each sub-reservoir are generated randomly similar to the R-HFCM initialization technique as discussed in section 4.2.1.

6.2.1.3 Partitioning

CR-HFCM is composed of randomized HFCM models with the same linguistic variable, i.e., the group of fuzzy sets defined over the Universe of Discourse $U = [\min(Y_{emb}), \max(Y_{emb})]$.

The goal of the partitioning process is to split U into k even length and evenly spaced and overlapped intervals, where mp_i, $i = 1..k$, is the midpoint of each interval. Then, a fuzzy set C_i (i.e. a concept) is defined with midpoint mp_i and a membership function μ_{C_i}. In each sub-reservoir, the group of the k concepts form the linguistic variable C, such that $C_i \in C$, $\forall i = 1, \ldots, k$. Here grid partitioning is used in which the number of concepts is equal to the number of partitions.

Figure 4 shows a simple example of the FCM structure used in this model, with triangular membership function μ_{C_i}.

6.2.1.4 Fuzzification

The aim of the fuzzification step is converting crisp time series (Y_{emb}) into a fuzzy series A which represents the activation state $a_i(t)$ of each concept $C_i \in C$ for each time series sample. Then, for each sample $y_{emb}(t) \in Y_{emb}$ the corresponding fuzzified sample $a(t) \in A$ is the set of activations $a_i(t) \in a(t)$ such that $a_i(t) = \mu_{C_i}(y_{emb}(t))$, for $i = 1, \ldots, k$.

6.2.1.5 Activation

For each sub-reservoir with index $j = 1, \ldots, L$, the state value of each concept in time $t + 1$ can be defined by the equation (6.2).

$$\mathbf{a}_j(t+1) = f\left(\mathbf{w}^0 + \sum_{l=1}^{\Omega} \mathbf{W}^l \cdot \mathbf{a}(t-l+1)\right) \tag{6.2}$$

6.2.1.6 Defuzzification

After calculating the activation level of each concept, in this step, the defuzzification is carried out to produce the output related to each sub-reservoir. Therefore, the forecast values for each sub-reservoir $j = 1, \ldots, L$ at time $t + 1$ in numeric terms can be calculated via the equation (6.3).

$$\hat{y}_j(t+1) = \frac{\sum_{i=1}^{k} a_{ji}(t+1) \cdot mp_i}{\sum_{i=1}^{k} a_{ji}(t+1)} \tag{6.3}$$

where $a_j(t+1)$ is the activation calculated from the previous step for each concept at time $t+1$ and mp_i is the center of each concept $C_i \in C$.

6.2.1.7 Least Squares coefficients determination

Given the outputs ($\hat{y}_{sub(1)}(t+1)$, $\hat{y}_{sub(2)}(t+1)$,..., $\hat{y}_{sub(L)}(t+1)$ where $t = 1, ..., T$) from each sub-reservoir, a design matrix $X \in \mathbb{R}^{L \times T}$ is created to represent the linear system $Y = \lambda X$, where $\lambda = [\lambda_0, ..., \lambda_j]$ is the coefficient vector. Then, the Least Squares method is employed to solve this linear system and find the λ vector of coefficients that minimizes the Mean Squared Error.

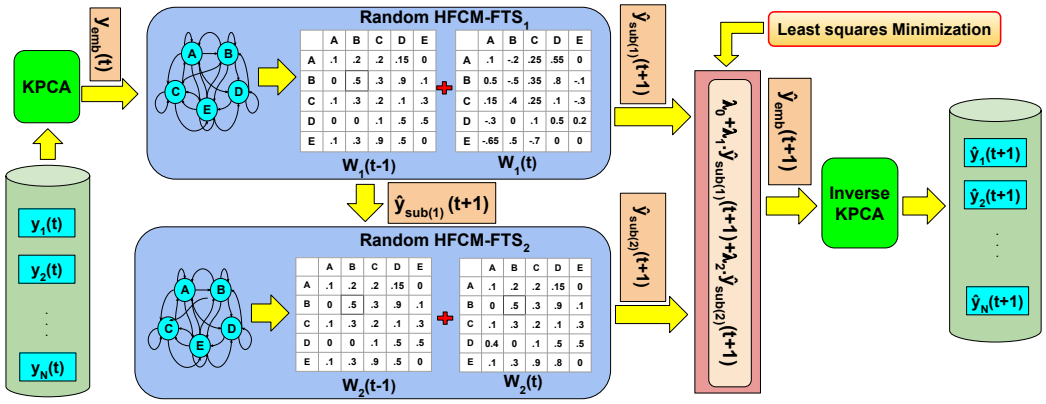


Figure 24 – A simple example of the proposed MIMO-FCM mechanism considering $\Omega = 2$, $M = 1$, $N_{SR} = 2$.

A simple structure of the MIMO-FCM model considering $\Omega = 2$,$M = 1$,$N_{SR} = 2$ has been shown in Figure 24. Accordingly, KPCA is used to translate the original high-dimensional time series $Y \in \mathbb{R}^{T \times N}$ into $y_{emb} \in \mathbb{R}^{T \times 1}$. The obtained y_{emb} is fed to the first sub-reservoir to generate its output with regards to the weight matrices at times t and $t-1$ for each sub-reservoir. It is worth noting that the model contains the bias weights while they are discarded from Figure 24 just because of the ease of notation.

6.2.2 Forecasting Procedure

The main objective of this stage is calculating the predicted crisp value for each sub-reservoir ($\hat{y}_{sub(L)(t+1)}$ and $N_{SR} = 1, 2, ..., L$), the predicted crisp value of principal components ($\hat{y}_{emb(1)}(t+1)$,$\hat{y}_{emb(2)}(t+1)$,...,$\hat{y}_{emb(M)}(t+1)$) as well as employing the inverse KPCA unit to predict the final crisp values of all the variables ($\hat{y}_1(t+1)$,$\hat{y}_2(t+1)$,...,$\hat{y}_N(t+1)$), given the linguistic variable C, the weight matrices $\mathbf{W}^t$, the activation function f and a crisp input $Y(t)$. The steps of the forecasting process are fully explained as follows:

6.2.2.1 Embedding:

In this step, the KPCA transformation is used for test data.

6.2.2.2 Fuzzification:

The same as Step 4 in the Training procedure.

6.2.2.3 Activation:

The same as Step 5 in the Training procedure.

6.2.2.4 Defuzzification

(A) **Sub-reservoir forecasts deffuzification**: The same as Step 6 in the Training procedure.

(B) **Final forecast defuzzification for each principal component:**

The predicted value of each principal component at time $t+1$ is computed through the linear combination of the obtained outputs of all sub-reservoirs and the least squares coefficients. The final predicted value is described using the equation (6.4).

$$\hat{y}_{emb(M)}(t+1) = \lambda_0 + \sum_{j=1}^{N_{SR}} \lambda_j \cdot \hat{y}_{sub(j)}(t+1) \tag{6.4}$$

For instance, as Figure 24 demonstrates, since $M = 1$ and $N_{SR} = 2$, the final output is estimated by the equation (6.5).

$$\hat{y}_{emb}(t+1) = \lambda_0 + \lambda_1 \cdot \hat{y}_{sub(1)}(t+1) + \lambda_2 \cdot \hat{y}_{sub(2)}(t+1) \tag{6.5}$$

where $\{\lambda_0, \lambda_1, \lambda_2\}$ denote the obtained least square coefficients from the training procedure and $\hat{y}_{sub(1)}(t+1)$ and $\hat{y}_{sub(2)}(t+1)$ are the generated outputs for the first and second sub-reservoirs respectively according with equation (6.3).

Obviously, as Figure 24 shows, the number of least squares coefficients directly depends on the number of sub-reservoirs (N_{SR}). To be more precise, the number of least squares coefficients in this model is equal to $N_{SR}+1$. Thereby, for $N_{SR} = n$, the final predicted value is described by the equation (6.6) using $n+1$ least-squares coefficients.

$$\hat{y}_{emb}(t+1) = \lambda_0 + \lambda_1 \cdot \hat{y}_{sub(1)}(t+1) + \lambda_2 \cdot \hat{y}_{sub(2)}(t+1) + ... + \lambda_n \cdot \hat{y}_{sub(n)}(t+1) \tag{6.6}$$

(C) **Final predicted crisp values for original variables of each data set:**

In the last step, as can be seen from figures 22 and 24, inverse KPCA is applied to construct the final predicted crisp values ($\hat{y}_1(t+1)$,$\hat{y}_2(t+1)$,...,$\hat{y}_N(t+1)$) from the obtained predicted values for each component.

6.3 Computational Experiments

This section is mainly dedicated to evaluating the accuracy of our proposed approach. The first subsection provides a brief description of the case studies as well as evaluation metrics. Then, the efficiency of the proposed MIMO-FCM model is verified by applying some high-dimensional IoT data sets. All the experiments are implemented with Python 3.6.12 using open-source packages such as Scikit-Learn, Keras, Tensorflow, PyTorch, Pandas, Numpy, and pyFTS.

6.3.1 Case studies

The effectiveness of the proposed MIMO-FCM method is assessed on two types of IoT applications, which include smart buildings and air quality monitoring (smart city). Accordingly, three public data sets are used to evaluate the performance of the proposed approach, as presented in the following subsections. It is worth mentioning that some pre-processing techniques are applied to clean the original data set by removing outliers and missing values.

As the first case study, we used the UCI Appliances energy prediction data set (AE-DS) which is accessible at the UCI ML repository (DUA; GRAFF, 2017). The data set includes 10-minute sample information collected for about 4.5 months. A Zigbee wireless sensor network was used to monitor the house temperature and humidity conditions which were merged with the weather data collected from the nearest airport weather station. The data set contains 19,735 samples and 27 variables.

In the second case study, Smart home with Weather Information data set (SH-DS) is employed which is published on the Kaggle platform (KAGGLE, 2021). The data set consists of time-span house appliances in KW from a smart meter and weather conditions of a particular region, ranging from January 2016 to December 2016 at a frequency of 1 minute with 29 variables including 18 electricity variables, 10 weather variables, and 1 temporal feature. It should be noted that the original time series is resampled from 1 to 10 minutes, before injecting to the model.

The UCI Beijing Multi-Site Air-Quality data set (AQB) (ZHANG et al., 2017) was considered as the final case study which consists of sets of hourly meteorological and air pollutants information from 12 nationally-controlled air-quality monitoring sites/stations: Aotizhongxin, Changping, Dingling, Dongsi, Guanyuan, Gucheng, Huairou, Nongzhanguan, Shunyi, Tiantan, Wanliu, and Wanshouxigong. In each air quality station, meteorological data are matched with the nearest weather station from the China Meteorological Administration. The data set contains 35,065 instances with multi-variables in each station over a four-year period from March 1st, 2013 to February 28th, 2017.

6.3.2 Experimental methodology

To evaluate the performance accuracy of the proposed approach, three comparison criteria for the forecasting accuracy including RMSE described in equation (3.6), MAE described in equation (5.12) and NRMSE, described in equation (4.14) are employed, where $y(t)$ and $\hat{y}(t)$ stand for the actual and forecast values, respectively.

These forecasting accuracy metrics were calculated using the sliding window cross-validation technique, where 75% of each window is used as a training set and 25% for testing. For each variable $y_i(t) \in Y$ of each data set $d_j \in \mathcal{D}$, the samples were divided into 30 windows $w \in W$. For each window, we train and test the models $m \in \mathcal{M}$ using respectively the training and test subsets. Noteworthy that the accuracy of the models is evaluated over the test data computing the average values of accuracy metrics obtained for all 30 windows.

To compare the performance of the models statistically, the Kruskal-Wallis test was conducted with the confidence level of $\alpha = 0.05$ in terms of the average NRMSE of all variables in each data set for each window. In this method, the null hypothesis H_0 stands for the equality of average NRMSE errors of all methods and H_1 presents an alternative hypothesis such that at least one of the averages is different from the others. When H_0 is not accepted, it is needed to apply *post hoc* tests to compare the equality of each pair of means. In this investigation, we selected the Wilcoxon approach for *post hoc* test.

6.4 Results

In this section, we present the obtained experimental results of our proposed model and provide a comparison of our approach against other forecasting models that are tested over the mentioned case studies.

This section is divided into three sub-sections. Subsection 6.4.1 is focused on presenting a discussion and analysis of our proposed model's hyperparameters to reach the best forecasting accuracy. The second subsection 6.4.2 provides a brief introduction of the baseline models as well as hyper-parameter tuning and subsection 6.4.3 is responsible for comparing the accuracy of the proposed model with other baseline models.

In order to prove the potential of our presented approach in dealing with high-dimensional MIMO time series, we considered two scenarios for AQB data set. Firstly, six sites comprising Aotizhongxin, Changping, Dingling, Dongs, Guanyuan and Gucheng have been selected to predict all 66 variables (i.e AQB-6). In the next scenario (i.e AQB-12), 12 stations are considered to predict all 132 variables.

6.4.1 Parameter setting for MIMO-FCM

The aim of this part is to analyze the influence of effective hyper-parameters on the accuracy of the proposed method. The number of concepts (k), the number of components (M), the number of layers or sub-reservoir (N_{SR}), the order value (Ω) and the kernel coefficient of KPCA (γ) are the most influential parameters on the model forecasting accuracy.

The experiments are executed with $k \in \{3, 4, 5, 7, 10, 20\}$, $M \in \{1, 2, 3, 4, 5, 7, 8, 10\}$, $N_{SR} \in \{2, 3, 4, 5, 7, 10, 20\}$, $\Omega \in \{2, 3, 4, 5, 6\}$ and $\gamma \in \{0.1, 0.3, 0.4, 0.5, 0.7, 0.8, 0.9\}$. Since the effectiveness of the proposed model is sensitive to the proper choice of these hyper-parameters, the goal is to set them optimally to achieve the best performance in terms of prediction accuracy. The best configuration of these hyperparameters is obtained empirically (through trial and error) based on the training data set. More specifically, models are executed by varying these hyper-parameters. Also, it is worth mentioning that in this investigation the activation function in FCM is set to ReLU (3.2).

Table 36 summarizes the optimal combinations of the parameters adjusted separately for each case study. Table 36 exhibits that the most accurate models are obtained utilizing the minimum values of hyper-parameters as the strengths of the proposed model. This means that in MIMO-FCM, the models with the fewest concepts ($k = 3, 4$) and sub-reservoirs ($N_{SR} = 2$) have the lowest forecasting errors. It is worth observing that increasing the number of blocks (sub-reservoirs) has a negative effect on our model performance. In addition, in all case studies, the highest forecasting accuracy is achieved by setting $k = 1$, $N_{SR} = 2$, and $\gamma = 0.9$.

It is worth mentioning that selecting the optimal values of hyper-parameters using the grid search technique would be a better choice which is planned for future work.

Table 36 – Optimal parameters of our proposed model for each data set

data set	k	Ω	N_{SR}	M	γ
AE-DS	3	4	2	1	0.9
SH-DS	3	3	2	1	0.9
AQB-6	3	4	2	1	0.9
AQB-12	4	4	2	1	0.9

6.4.2 Baseline models

In this section, a comparison against several common machine learning and deep learning models is reported to emphasize the validity of our proposed model. Thereby, six baseline models including popular machine learning and deep learning methods are imple-

mented in this investigation, which are briefly described. Then, the results of the proposed model are compared with the results of these competitor models in subsection 6.4.3.

In this work, three different types of RNNs including Vanilla RNN, Long Short-term Memory (LSTM) (HOCHREITER; SCHMIDHUBER, 1997) and Gated Recurrent Unit (GRU) (CHUNG et al., 2014) are used as competitor models which are implemented using PyTorch framework. Various experiments were conducted to adjust the best hyper-parameters for these models whereas the optimal values were obtained with layers = 1, epochs = 300, batch size = 64, learning rate = 0.001, weight decay = 0.1 and optimizer = Adam. Also, the number of hidden neurons is determined considering the number of dimensions and instances of each data set, varying from 3 to 467.

In addition, stacked encoder-decoder LSTM (SLSTM) was implemented as another type of vanilla LSTM that is more complex and designed for sequence-to-sequence problems. The encoder stage reads the input and produces an expressive fixed-length embedded representation, while the decoder interprets the latent space used to predict the output. The best performance could be yielded with epochs = 25, batch size = 32, neurons = 200, and optimizer = Adam, after various settings.

Random Forest (RF) (BREIMAN, 2001) and Support vector regression (SVR) (DRUCKER et al., 1996) were selected as other baseline models. Table 37 reports the optimal values of hyper-parameters for these models using the Randomized Search technique.

Table 37 – Optimal parameters of each data set for RF and SVR models

Model	Hyperparameters	AE-DS	SH-DS	AQB-6	AQB-12
RF	Max depth	20	15	30	30
	Max leaf nods	10	20	30	30
	Min samples leaf	6	3	2	2
SVR	C	5.76	17.07	25.71	25.71
	ϵ	0.07	0.15	0.05	0.05

6.4.3 Comparison against baselines

The main purpose of this section is to provide a comparison among the obtained results for the proposed model and baseline methods considering the above-mentioned case studies. In other words, this section can be divided into three subsections such that in each subsection the models are run by feeding three different high-dimensional IoT time series as explained in the following. Since AQB6 and AQB12 data sets contain many variables, it is impractical to report the prediction results obtained for all variables of these data sets. Hereby, in these cases, the forecasting accuracy of MIMO models for two variables ($PM2.5$ and $PM10$) are presented and evaluated in terms of accuracy metrics.

Table 38 – Evaluation of the accuracy of the proposed method against other baseline models in terms of Average RMSE and average MAE (AE-DS data set)

Metric	AVG.RMSE							AVG.MAE						
Variable	MFCM	SLSTM	RF	SVR	LSTM	GRU	RNN	MFCM	SLSTM	RF	SVR	LSTM	GRU	RNN
Appliance	**66.31**	79.55	72.91	86.81	78.06	77.56	75.02	**33.54**	47.93	42.74	58.11	46.76	49.34	47.67
lights	**5.19**	6.60	5.36	7.06	5.90	5.67	5.92	**2.92**	4.38	3.65	5.12	4.19	3.98	4.30
T1	**0.17**	0.41	0.25	0.70	0.29	0.27	0.26	**0.13**	0.35	0.19	0.56	0.23	0.21	0.21
RH_1	**0.55**	1.22	0.83	2.05	1.06	1.01	0.96	**0.32**	0.93	0.53	1.66	0.76	0.73	0.71
T2	0.36	0.65	**0.31**	1.04	0.51	0.49	0.5	0.24	0.54	**0.21**	0.84	0.40	0.39	0.40
RH_2	**0.63**	1.47	0.74	2.20	0.99	0.94	0.95	**0.41**	1.20	0.52	1.82	0.77	0.75	0.73
T3	**0.12**	0.55	0.37	0.72	0.35	0.34	0.317	**0.08**	0.45	0.27	0.60	0.27	0.26	0.24
RH_3	**0.31**	0.99	0.56	1.49	0.74	0.67	0.66	**0**.21	0.82	0.39	1.23	0.57	0.52	0.52
T4	**0.17**	0.69	0.32	0.85	0.41	0.43	0.41	**0.12**	0.58	0.25	0.72	0.32	0.34	0.32
RH_4	**0.32**	1.09	0.61	1.90	0.82	0.69	0.69	**0.22**	0.89	0.41	1.57	0.66	0.56	0.56
T5	**0.14**	0.46	0.28	0.63	0.33	0.31	0.32	**0.09**	0.36	0.20	0.50	0.25	0.24	0.25
RH_5	**2.11**	5.28	3.09	6.94	4.97	4.34	4.85	**0.93**	3.94	1.95	5.27	3.54	3.24	3.54
T6	1.04	1.61	**0.89**	2.70	1.42	1.24	1.43	**0.66**	1.27	0.66	2.17	1.15	1.01	1.17
RH_6	3.72	7.53	**2.14**	10.12	5.67	5.64	5.16	2.44	6.19	**1.41**	8.20	4.54	4.60	4.16
T7	**0.10**	0.37	0.22	0.58	0.25	0.22	0.24	**0.07**	0.31	0.16	0.48	0.20	0.17	0.19
RH_7	**0.56**	1.84	1.03	2.89	1.14	1.10	1.02	**0.39**	1.55	0.80	2.45	0.93	0.91	0.86
T8	**0.13**	0.62	0.23	0.70	0.39	0.34	0.37	**0.09**	0.53	0.17	0.58	0.32	0.28	0.31
RH_8	**0.87**	1.66	**0.87**	2.80	1.48	1.43	1.46	**0.61**	1.37	0.63	2.31	1.21	1.17	1.21
T9	**0.06**	0.30	0.22	0.40	0.16	0.14	0.15	**0.04**	0.26	0.17	0.34	0.13	0.11	0.12
RH_9	**0.53**	1.42	0.62	2.19	1.10	0.90	0.94	**0.36**	1.21	0.47	1.89	0.89	0.72	0.76
T_out	0.79	1.29	**0.71**	2.24	1.04	0.91	1.01	**0.51**	1.10	0.52	1.84	0.84	0.73	0.81
Press	**0.42**	2.57	1.23	3.40	1.51	1.61	1.45	**0.28**	2.17	0.97	3.03	1.30	1.38	1.24
RH_out	3.28	6.31	**1.85**	8.57	4.31	4.19	4.11	2.18	5.24	**1.32**	6.76	3.36	3.30	3.26
wind speed	0.49	1.40	**0.40**	1.55	1.12	1.03	1.07	0.36	1.17	**0.27**	1.25	0.91	0.84	0.89
visibility	4.31	6.31	**2.39**	7.99	7.50	6.30	7.14	3.08	5.14	**1.60**	6.59	6.02	4.97	5.78
Tdewpoint	**0.39**	1.29	0.70	1.94	1.03	0.88	0.94	**0.27**	1.05	0.52	1.58	0.84	0.71	0.77

However, the full results are publicly accessible via the link address given as follows: ⟨https://github.com/OMIDUFMG2019/MIMO-FCM⟩.

Tables 38, 39, 40 and 41 compare the forecasting accuracy of the proposed model with competitors for the given data sets in terms of RMSE and MAE. In all tables, MFCM represents short abbreviations of the MIMO-FCM model. It is worth mentioning that Tables 40 and 41 record only the obtained results for two variables. More precisely, all the 66 and 132 variables have been predicted while only PM2.5 and PM10 (i.e. the most important air quality indexes) have been chosen to present in our method. Based on the obtained results, the superiority of our proposed model can be recognized visually when looking at the error values in all these tables.

As Table 38 details, in the case of SE-DS, the highest accuracy is achieved by the MIMO-FCM method for 73% and 81% of variables in terms of average RMSE and MAE, respectively. In the second scenario, the SH-DS dataset is fed to the models to assess their forecasting accuracy as shown in Table 39. In this case, MIMO-FCM is the most accurate model for 59.25% (16 out of 27) and 78% (20 out of 27) of variables in terms of average RMSE and MAE, respectively. It is worth highlighting that for some variables, RF performs better than our proposed model with very slight differences. Finally, the performance accuracy of our proposed model is quantitatively compared with other baseline models feeding the AQB data set considering two scenarios. Firstly, six stations consisting of Aotizhongxin, Changping, Dingling, Dongs, Guanyuan, and Gucheng have been selected to predict 66 variables. The listed results in this table confirm the superiority

Table 39 – Evaluation of the accuracy of the proposed method against other baseline models in terms of Average RMSE and average MAE (SH-DS data set)

Metric	AVG.RMSE							AVG.MAE						
Variable	MFCM	SLSTM	RF	SVR	LSTM	GRU	RNN	MFCM	SLSTM	RF	SVR	LSTM	GRU	RNN
Use	0.54	0.56	**0.51**	0.65	0.55	0.53	0.53	**0.33**	0.38	0.34	0.47	0.38	0.37	0.37
Gen	**0.03**	0.04	**0.03**	0.07	0.04	0.039	0.03	0.02	0.03	**0.01**	0.05	0.03	0.02	0.02
Dishwasher	0.14	0.14	**0.11**	0.15	0.12	0.12	0.13	0.03	0.04	**0.02**	0.07	0.06	0.05	0.05
Furnace1	0.09	0.09	**0.08**	0.11	0.09	0.09	0.09	**0.05**	0.06	0.06	0.08	0.06	0.06	0.06
Furnace2	0.10	0.10	**0.09**	0.12	0.098	0.09	0.09	**0.06**	0.07	0.07	0.09	0.07	0.07	0.07
H.office	**0.03**	0.04	0.04	0.06	0.04	0.04	0.04	**0.01**	0.02	0.02	0.03	0.03	0.03	0.03
Fridge	0.06	**0**.05	**0.05**	0.06	0.06	**0**.05	**0**.05	**0.03**	0.04	**0**.03	0.05	0.04	0.04	0.04
WineCellar	**0.02**	0.03	0.03	0.04	0.03	0.03	0.03	**0.01**	0.02	0.02	0.02	0.02	0.02	0.02
G.door	0.01	0.01	0.09	0.09	**8e-3**	**8e-3**	**8e-3**	**2e-3**	**2e-3**	3e-3	3e-3	3e-3	3e-3	3e-3
Kitchen12	**4e-3**	6e-3	7e-3	8e-3	5e-3	5e-3	5e-3	**1e-3**	2e-3	2e-3	4e-3	2e-3	2e-3	2e-3
Kitchen14	0.05	**0.04**	0.05	**0.04**	**0**.04	**0**.04	**0**.04	**0**.01	**0.01**	0.02	0.02	0.02	0.02	0.02
Kitchen38	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Barn	0.07	0.08	0.07	0.09	0.07	**0.06**	0.07	**0.02**	0.03	**0**.02	0.04	0.03	0.03	0.03
Well	0.08	0.06	0.06	0.07	**0.06**	**0.06**	**0.06**	**0**.02	**0.02**	0.03	0.03	0.03	0.03	0.03
Microwave	0.07	0.06	0.07	0.06	**0.05**	**0.05**	**0.05**	**0.01**	**0.01**	0.02	0.02	0.02	0.02	0.02
Livingroom	**0.04**	0.05	0.05	0.06	0.05	0.05	0.05	**0.01**	0.02	0.02	0.04	0.03	0.02	0.03
Solar	**0.03**	0.04	**0.03**	0.07	0.04	0.04	0.04	0.02	0.03	**0.01**	0.05	0.03	0.03	0.02
Temperature	**1.21**	2.47	1.63	4.97	2.25	1.98	1.81	**0.63**	1.70	0.98	3.49	1.69	1.46	1.36
Humidity	0.03	0.05	**0.02**	0.09	0.05	0.05	0.05	**0.01**	0.03	**0.01**	0.06	0.04	0.03	0.03
Visibility	**0.25**	0.55	0.38	0.91	0.45	0.38	0.42	**0.11**	0.33	0.17	0.57	0.31	0.27	0.30
App.Temperature	**1.38**	2.81	2.00	5.68	2.64	2.32	2.09	**0.72**	1.93	1.19	3.99	1.98	1.69	1.55
Pressure	**0.61**	2.42	1.11	3.71	1.59	1.27	1.39	**0.33**	1.83	0.71	2.76	1.21	0.93	1.06
WindSpeed	**0.78**	1.20	0.65	1.80	1.17	0.99	1.09	**0.41**	0.82	0.37	1.31	0.85	0.71	0.79
Cloud Cover	**0.06**	0.09	0.07	0.17	0.09	0.07	0.08	**0.03**	0.06	0.04	0.12	0.06	0.05	0.06
WindBearing	**28.98**	38.15	28.84	54.55	35.92	33.68	36.07	**11.43**	24.42	14.83	39.48	23.42	21.12	23.70
PrecipIn.	**2e-3**	5e-3	4e-3	7e-3	0.04	0.04	0.04	**1e-3**	2e-3	**1e-3**	3e-3	3e-3	3e-3	3e-3
Dew Point	**0.95**	2.60	1.81	4.88	1.99	1.76	1.87	**0.52**	1.83	1.13	3.52	1.53	1.36	1.48
Preciprob.	**0.03**	0.06	0.05	0.10	0.09	0.08	0.08	**0.01**	0.03	0.02	0.06	0.06	0.05	0.05

Table 40 – PM2.5 and PM10 prediction accuracy of our proposed models and baseline methods over six stations data (AQB-6) in terms of RMSE and MAE

Metric	AVG.RMSE							AVG.MAE						
Variable(Station)	MFCM	SLSTM	RF	SVR	LSTM	GRU	RNN	MFCM	SLSTM	RF	SVR	LSTM	GRU	RNN
PM2.5(Aotizhongxin)	**22.15**	32.06	26.15	47.73	35.97	35.10	32.10	**12.78**	21.34	15.36	32.14	25.96	25.83	23.65
PM10(Aotizhongxin)	**34.39**	46.95	42.52	66.84	54.05	51.00	47.25	**20.34**	30.98	26.15	46.55	38.35	36.22	33.64
PM2.5(Changping)	**20.79**	29.93	23.73	42.28	34.78	32.11	31.01	**11.96**	20.20	14.51	28.71	25.68	23.38	23.03
PM10(Changping)	**31.41**	41.94	36.83	57.02	49.98	46.13	45.52	**18.43**	27.75	22.52	39.49	35.58	32.23	32.41
PM2.5(Dingling)	**19.58**	29.36	23.44	41.43	37.15	33.93	32.83	**10.85**	19.88	13.97	28.27	27.64	25.11	23.65
PM10(Dingling)	**29.53**	39.28	34.44	53.15	47.80	45.03	45.10	**16.45**	25.59	20.56	36.79	33.93	31.48	32.52
PM2.5(Dongsi)	**21.99**	35.04	26.95	50.72	37.41	35.02	32.94	**13.17**	23.56	16.06	34.95	27.04	25.80	24.14
PM10(Dongsi)	**33.58**	47.26	41.60	67.73	55.16	52.47	50.95	**19.71**	31.01	25.10	47.48	38.96	37.61	37.03
PM2.5(Guanyuan)	**21.77**	33.07	25.92	47.11	36.68	33.39	30.51	**12.98**	22.18	15.29	31.96	27.12	24.76	22.86
PM10(Guanyuan)	**32.65**	43.23	37.66	62.45	53.18	48.99	46.66	**19.33**	28.73	22.93	43.28	38.48	35.56	33.29
PM2.5(Gucheng)	**24.12**	37.24	28.55	49.07	39.18	37.85	33.48	**13.68**	25.17	16.66	33.41	28.82	27.43	23.76
PM10(Gucheng)	**39.77**	51.96	44.39	71.51	59.47	57.17	55.39	**24.85**	36.63	29.24	51.95	43.72	42.32	41.16

of our proposed technique for all six stations. In the next scenario, the experiments are executed for all 12 stations to predict 132 variables and the results have been stored in Table 41. Again, the obtained results demonstrate the excellent performance of the proposed MIMO-FCM model for all stations. However, in three cases, RF provides better performance in comparison to our proposed model with very slight differences.

As can be seen from the results, it can be concluded that ranking the methods with respect to the obtained values of RMSE and MAE is difficult due to the unstable forecasting accuracy of the methods for each variable. As such, in order to have a more precise comparison among the methodologies, the average NRMSE over all variables for every data set has been calculated as shown in Table 42. As this table suggests, our proposed model outperforms the other competitor models for the given input time series.

Table 41 – PM2.5 and PM10 prediction accuracy of our proposed models and baseline methods over all stations data (AQB-12) in terms of RMSE and MAE

Metric	**AVG.RMSE**							**AVG.MAE**						
Variable(Station)	MFCM	SLSTM	RF	SVR	LSTM	GRU	RNN	MFCM	SLSTM	RF	SVR	LSTM	GRU	RNN
PM2.5(Aotizhongxin)	**22.07**	31.37	25.35	36.56	49.77	40.82	55.67	**12.41**	21.34	15.63	26.35	36.37	31.61	45.27
PM10(Aotizhongxin)	32.41	38.86	**31.92**	47.64	51.85	51.09	69.04	**19.89**	26.65	20.73	34.74	38.42	39.31	52.60
PM2.5(Changping)	**19.54**	30.85	21.75	34.26	35.97	39.20	47.90	**11.66**	22.34	14.02	25.17	27.80	29.61	38.36
PM10(Changping)	28.13	36.67	**28.05**	42.81	45.12	44.64	55.21	**17.65**	26.45	18.73	31.90	33.83	34.36	42.26
PM2.5(Dingling)	**17.55**	28.41	20.57	31.32	36.99	36.18	48.04	**10.08**	20.49	13.21	23.14	27.23	28.41	37.25
PM10(Dingling)	**24.51**	34.22	25.81	39.27	42.46	41.98	50.24	**14.89**	24.84	17.50	29.34	32.90	33.38	39.77
PM2.5(Dongsi)	**23.01**	31.67	25.05	38.57	41.75	46.89	61.41	**13.45**	21.80	15.54	27.81	32.47	38.31	48.84
PM10(Dongsi)	33.75	41.18	**33.03**	49.73	57.01	55.54	75.85	**20.27**	28.33	21.48	35.48	42.00	41.37	58.31
PM2.5(Guanyuan)	**20.59**	29.96	23.31	35.66	37.89	37.16	52.70	**12.26**	20.90	14.54	25.76	28.25	28.17	41.28
PM10(Guanyuan)	**28.37**	36.09	29.22	45.17	50.63	52.61	72.88	**17.91**	25.48	19.38	32.90	40.58	40.57	57.13
PM2.5(Gucheng)	**20.86**	32.94	24.06	36.60	41.44	43.92	55.27	**12.75**	23.32	15.76	26.62	30.60	31.87	42.78
PM10(Gucheng)	**35.86**	44.96	37.17	54.44	55.69	60.31	75.40	**23.77**	32.48	26.66	40.50	42.53	46.99	60.03
PM2.5(Huairou)	**18.26**	30.00	21.07	31.35	37.62	36.95	46.17	**10.56**	21.35	13.44	22.82	28.20	27.21	35.98
PM10(Huairou)	**31.11**	38.60	31.58	43.62	48.47	47.32	57.54	**18.51**	26.99	20.82	32.03	36.01	35.93	45.26
PM2.5(Nongzhanguan)	**22.34**	32.01	25.08	37.47	40.96	44.21	66.66	**13.11**	22.07	15.63	26.96	31.02	35.36	56.01
PM10(Nongzhanguan)	**30.58**	38.75	31.41	46.79	48.91	53.68	67.26	**19.04**	27.10	20.82	33.91	36.96	40.83	52.00
PM2.5(Shunyi)	**21.93**	35.11	24.66	38.44	49.61	43.51	53.88	**12.86**	24.77	16.00	28.00	37.76	32.71	40.19
PM10(Shunyi)	**29.85**	42.31	31.90	47.06	52.32	50.98	61.46	**18.48**	30.47	21.29	34.15	39.34	39.23	47.94
PM2.5(Tiantan)	**21.66**	29.57	23.43	35.39	42.77	40.35	52.71	**13.02**	20.28	14.81	25.33	31.14	31.26	40.61
PM10(Tiantan)	**31.87**	39.49	32.64	46.63	49.95	48.29	66.24	**19.77**	27.14	21.81	33.58	38.27	35.56	52.29
PM2.5(Wanliu)	**20.17**	32.53	23.98	36.08	38.91	46.01	58.76	**12.07**	22.61	15.19	25.75	29.16	34.20	47.04
PM10(Wanliu)	**30.52**	38.05	31.56	46.37	48.98	48.98	74.51	**19.81**	26.85	21.59	34.05	37.16	38.00	60.57
PM2.5(Wanshouxigong)	**23.16**	33.43	26.76	39.55	43.02	41.79	56.57	13.91	23.31	17.18	28.43	32.55	31.89	44.77
PM10(Wanshouxigong)	**33.15**	42.18	34.53	50.02	52.35	52.84	67.78	**21.04**	29.82	23.68	36.76	39.20	40.11	54.08

In brief, the obtained results confirm the efficacy and efficiency of our proposed model employing four high-dimensional time series containing 27, 29, 66, and 132 variables from three data sets in IoT applications. In other words, for all case studies, the ideal forecasting accuracy has been achieved using MIMO-FCM over other competing algorithms with regard to average NRMSE. Additionally, the experimental results justify that the performance of RNN models is close to each other in most cases. The other noteworthy point is that RNN models are less accurate than our proposed model. One of the possible hypotheses is related to the number of samples of data sets. In other words, the more samples, the more accurate the RNN models. More specifically, stacked LSTM is classified as the worst model because of the insufficient harmony associated with the number of samples and the complexity of the model. However, promising results have been generated by SLSTM in comparison to the SVR approach.

Table 42 – Evaluation of the models' accuracy in terms of average NRMSE

Methods	AE-DS	SH-DS	AQB-6	AQB-12
MFCM	**0.086**	**0.0882**	**0.0973**	**0.109**
RF	0.138	0.266	3.227	0.176
SVR	0.394	1.596	1.077	0.274
SLSTM	0.273	0.615	0.507	0.171
LSTM	0.176	0.111	0.153	0.193
GRU	0.162	0.105	0.144	0.189
RNN	0.164	0.106	0.142	0.228

6.4.4 Statistical testing

As described in section 6.3.2, we used the Kruskall-Wallis method for statistical comparison of our proposed model with other baseline methods. Based on the reported p-values and test statistics in Table 43, H_0 has been rejected for all data sets. Hereupon, all the models are compared using *post-hoc* test. The obtained results from the Wilcoxon test reveal that MIMO-FCM is a winning model compared to other baseline models. More clearly, Table 44 summarizes the statistical ranking of the models (based on the times in which H_0 can not be rejected) for each data set according to the obtained results from the Wilcoxon test. As Table 44 details, the first rank belongs to the MIMO-FCM model for all case studies followed by RF. The complete Wilcoxon test results are also available in the supplementary material.

In summary, it can be said that our proposed model is robust and effective with a high ability to predict high dimensional time series. Additionally, MIMO-FCM focuses on both aspects by designing a new frame of randomized FCMs as reservoir computing that speeds the training process. Furthermore, as shown in Table 45, the number of parameters of our model is considerably lower than deep learning models which results in less complexity of our model. The reason is associated the number of parameters in our model depends only on the number of least squares coefficients which is equal to the number of sub-reservoirs ($N_{SR} = 2$) plus one. Thereby, MIMO-FCM is more parsimonious as well as more interpretable than deep learning models.

Table 43 – Kruskal-Wallis mean comparison test results

data set	**Statistic**	**p-value**	**Result**
AE-DS	146.661	3.929e-29	H_0 is rejected
SH-DS	106.861	9.255e-21	H_0 is rejected
AQB-6	95.494	2.181e-18	H_0 is rejected
AQB-12	127.301	4.755e-25	H_0 is rejected

Table 44 – The summary of the ranking of the forecasting models

data set	MFCM	SLSTM	LSTM	GRU	RNN	RF	SVR
AE-DS	**1**	6	5	3	4	2	7
SH-DS	**1**	6	4	2	2	2	7
AQB-6	**1**	3	5	3	3	2	7
AQB-12	**1**	3	5	4	7	2	4

6.5 Chapter's Highlights

In this chapter, a new hybrid forecasting method was introduced to predict high dimensional multivariate time series called MIMO-FCM. The method is inspired by integrating the concepts of reservoir computing (ESN), HFCM, and embedding transformation.

Table 45 – Comparison among the number of parameters of our proposed model with deep learning methods

data set	MFCM	LSTM	GRU	RNN
AE-DS	**3**	7132	3199	1413
SH-DS	**3**	2687612	707532	244268
AQB-6	**3**	4152	2739	1353
AQB-12	**3**	2172	1761	939

MIMO-FCM was adopted as a MIMO FCM-based forecasting approach to predict multiple outputs.

The structure of MIMO-FCM consists of a certain number of univariate models based on the defined number of principal components returned by KPCA. In other words, KPCA is applied to reduce the dimension of the original time series by extracting some principal components. It is worth noting that each principal component is fed only as input to the first block(sub-reservoir) of the CR-HFCM. Since the CR-HFCM has a sequential architecture, the output from each sub-reservoir is considered as input for the next sub-reservoir as well as input for the output layer. In the next step, the least squares minimization technique is applied to train the output layer and generate the final predicted value for each principal component. Finally, the inverse KPCA is applied over all the obtained predicted values from each principal component to reach the final predicted values.

Three high-dimensional non-stationary IoT data sets, covering 27, 29, 66, and 132 variables, were employed to evaluate the validity of our proposed method against some popular deep learning and machine learning algorithms. The obtained results verify the superior performance of our proposed model with low values of the effective hyper-parameters including the number of components, map size, and reservoir size. In future work, we intend to replace the univariate HFCM block with a multivariate one. Also, future work intends to implement deep models by extending the shallow structure.

7 CONCLUSION AND FUTURE WORKS

The task of TSF is to predict the future values of a phenomenon using historical observation. Among plenty of forecasting methods presented in the literature, the usage of FTS techniques is growing. FTS was introduced as the main substitution for traditional TSF models to mitigate some of their drawbacks including the expensive cost of computation, lack of scalability and explainability, and insufficient ability to deal with uncertainty and complex problems in the real world.

Based on the presented investigations in the literature, weighted FTS methods improve the forecasting accuracy significantly in comparison to the weightless FTS model. Nevertheless, the right way to assign weights to each fuzzy rule remains an open challenge.

FCMs as a weighted FTS approach has achieved remarkable success in the area of time series forecasting by extracting knowledge in FTS models. This weighted FTS model is a kind of interpretable RNN combining the concepts of fuzzy logic and neural networks in which the weights are learned from the data. In this model, the neurons have special meanings, called concepts in the FCM terminology, with regard to the problem under investigation; whereas the edges represent the causal relationships between a pair of concepts. The usage of these weighted knowledge-based models continues to grow in popularity more specifically due to their high potential to design more complex ML solutions as well as their high ability to deal with uncertainties.

Likewise, a variety of univariate and multivariate FCM-based times series forecasting methodologies have been presented in the literature. Designing a proper structure of FCMs and learning weight matrices are considered the main steps of time series forecasting using FCMs. Based on the literature, common techniques to construct the structure of FCMs include granularity, membership values representation, Fuzzy c-means clustering, and a combination of wavelet transformation with empirical mode decomposition (EMD).

Despite the remarkable success of FCMs in the field of time series forecasting and analysis, there are still some challenging topics in this area. Regarding learning weights, according to the literature, the major proportion of studies have focused on applying population-based methods to adjust weights matrices due to some of the benefits including lower simulation error, higher functionality, robustness, and generalization abilities in comparison to the Hebbian-based methods. For instance, GA and PSO have been exploited widely. However, some limitations including a time-consuming, large number of learning hyper-parameters, a large number of learning processes, and convergence issues confined the usage of population-based strategies. These issues encouraged the researchers to employ other time-efficient learning techniques such as ridge regression and Bayesian ridge

regression.

Handling high-dimensional non-stationary multivariate time series is considered another main issue in this area. Few studies can be found in this direction. Additionally, there is a significant gap in the literature regarding MIMO FCM-based methods.

Therefore, the core concentration of this thesis was to design new FCM-based forecasting methodologies to predict univariate as well as low/high dimensional multivariate time series to cope with some of these challenges. The proposed approaches introduce new HFCM forecasting techniques integrating the concepts of FCM and FTS focusing on both challenging aspects of FCM-based time series forecasting models: designing new FCM frameworks to predict univariate and multivariate time series as well as speeding up the training phase.

In my first proposed approach, a hybrid univariate forecasting model named HFCM-FTS was introduced. HFCM-FTS combines HFCM and HOFTS, where the weight matrices associated with the state transitions are learned via the GA from the data. The objective of FCM is to find the weight matrices that model the causal relations among the concepts defined in the Universe of Discourse. The model has been tested over public data to predict solar radiation in Brazil.

Despite the success of the HFCM-FTS model when the number of concepts is low, the proposed training methodology was very time-consuming as this number increased. Thus, to tackle the aforementioned problem, we suggested a novel univariate time series forecasting technique composed of a group of randomized HFCM-FTS models labeled R-HFCM. The novelty of the proposed R-HFCM model is relevant to merging the concepts of FCM and ESN as an efficient and particular family of RC models, where the LS algorithm is applied to train the model. The structure of R-HFCM consists of the input layer, reservoir layer(composed of some number of sub-reservoirs), and output layer so that only the output layer is trainable while the weights of each sub-reservoir components are selected randomly and kept constant during the training process. Thus, it is worth observing that R-HFCM is much faster than HFCM-FTS. Although R-HFCM provides advantages in both terms of building a new structure and training way, it also examines the impact of effective parameters on forecasting accuracy.

In the next step, MO-RHFCM and MIMO-FCM (or MFCM) models were introduced to fill the absence of MIMO models in the literature. Firstly, we developed MO-RHFCM to predict low-dimensional multivariate time series. MO-RHFCM is a hybrid method integrating the concepts of MVFTS, HFCM, and ESN using LS as the learning algorithm. As an extended frame of R-HFCM, MO-RHFCM consists of three layers including an input layer, an intermediate layer(reservoir), and an output layer where the least squares algorithm is applied to train the output layer in the model. In MO-RHFCM, each sub-reservoir is constructed by a multivariate MVFCM-FTS model. MVFCM-FTS

is a combination of MVFTS and randomized HFCM with the same weight initialization strategy. In this model, each sub-reservoir is fed by all the variables. Therefore, each sub-reservoir generates multiple outputs which depend on the number of variables in the input time series. One further feature that sets MO-RHFCM apart from the majority of discussed techniques in the literature is that each variable is not characterized by a single concept. It means that each variable is presented by one FCM considering the user-defined number of concepts which can be considered an efficient solution for modeling the dynamics of complex systems.

Since MO-RHFCM was constructed to predict multivariate datasets with low dimensions, in the second scenario we developed another MIMO technique named MFCM with the ability to handle high-dimensional multivariate datasets. More specifically, MFCM is a hybrid method combining kernel principal component (KPCA), HFCM, and ESN. Firstly, the original data set is transmitted through KPCA to create principal components defined by the user. Then each component is fed to each sub-reservoir to be predicted by one univariate cascade randomized HFCM unit. Finally, inverse KPCA is applied to reconstruct the original time series. Interestingly, the underlying model within MFCM is composed of cascaded R-HFCM models with different architectures such that only the output layer is trainable using the LS algorithm. The weights inside each sub-reservoir are selected randomly and remain fixed during the training process. To compare the proposed MFCM technique with some baseline methods, real-world high-dimensional time series in IoT applications with more than 100 variables were employed to evaluate the performance of the proposed method.

The obtained results show the success of our proposed models using several univariate datasets as well as multivariate datasets (either low-dimensional or high-dimensional time series) with competitive and promising results when compared to other FTS methods and some state-of-the-art machine learning and deep learning methods. Also, noteworthy that our proposed models are cheap computationally and more parsimonious than other deep-learning models.

7.1 Summary of contributions

Thus, the continuations of this investigation can be listed as follows:

1. A comprehensive review of up-to-date FCM-based time series forecasting techniques until mid-2022;

2. Proposal of a new hybrid forecasting method based on FCM, named HFCM-FTS, as a combination of HFCM and HOFTS, where weights are learned via the GA;

3. First introduction of randomized FCMs as reservoir computing called R-HFCM. This leads to a hybrid univariate FTS forecasting model trained via LS as time-effective learning;

4. A study of the effect of the main influential elements on the accuracy performance of the proposed models;

5. The first proposal of MIMO FCM-based forecasting methods to predict:

   (a) low-dimensional multivariate time series termed MO-RHFCM combining the concepts of multivariate FTS (MVFTS), randomized HFCM, and ESN trained via LS;

   (b) high-dimensional multivariate time series labeled MFCM integrating the concepts of embedding (KPCA), FCM, and ESN trained via LS;

6. Evaluating the effectiveness of the proposed: (a) univariate techniques as well as MO-RHFCM model to predict solar radiance and electric load consumption; (b) MFCM model on four high-dimensional non-stationary time series with more than 100 variables in IoT applications (smart cities and smart buildings).

7.2 Methods limitations and future investigations

Although this study covered some gaps in the field of TSF using FCM, there are still some limitations and gaps. Some of these limitations are summarized as follows.

Our proposed methods still comprise weaknesses in dealing with nonstationarity, outliers, and concept drifts. The other limitation of our proposed models is related to the forecasting horizon because they have been constructed to forecast one step ahead (H=1). Also, our proposed models do not provide the possibility of probabilistic and interval forecasting. The final limitation is relevant to the lack of hyperparameters (Hp) optimization because they are tuned empirically in our methods.

Therefore, some future research directions should be stated as follows, although some of them are derived from the methods' limitations:

1. Extending our one-step ahead forecasting models to multi-steps ahead;

2. Enabling our models to handle outliers and concept drift challenges in non-stationary time series;

3. To the best of our knowledge, there is no probabilistic FCM-based forecasting method in the literature. As such, we intend to equip our randomized methods with these capabilities;

4. The use of an optimization technique to find a well-performing HP configuration of the proposed models on the given data set;

5. Extending our shallow MIMO techniques to deep randomized HFCM model;

6. Apply and develop our proposed methods in other applications such as hydrology, medicine, etc;

7. Investigating the interpretability of the proposed methods.

7.3 Publications

1. Orang, Omid; Lima e Silva, Petrônio Cândido; Guimarães, Frederico Gadelha. Time Series Forecasting Using Fuzzy Cognitive Maps: A Survey. **Artificial Intelligence Review (AIRE)**; 2022.

2. Orang, Omid; Lima e Silva, Petrônio Cândido; Silva, Rodrigo; Guimarães, Frederico Gadelha. Randomized High Order Fuzzy Cognitive Maps as Reservoir Computing Models: A first introduction and applications. **Neurocomputing**, V. 512, P. 153-177, 2022.

3. Bitencourt, Hugo Vinicius; Orang, Omid; Souza, Luiz Augusto Facury; Lima e Silva, Petrônio Cândido; Guimarães, Frederico Gadelha. An Embedding-based Non-Stationary Fuzzy Time Series Method for Multiple Output High-dimensional Multivariate Time Series Forecasting in IoT Applications. **Neural Computing and Applications (NCAA)**, 2022.

4. Lucas, Patricia; Orang, Omid; Lima e Silva, Petrônio Cândido; Mendes, Eduardo; Guimarães,Frederico Gadelha . A Tutorial on Fuzzy Time Series Forecasting Models: Recent Advances and Challenges; **SBIC**-Learning and Nonlinear Models, Journal of the Brazilian Society on Computational Intelligence, 2022.

5. Orang, Omid; Silva,Rodrigo; Lima e Silva, Petrônio Cândido; Guimarães, Frederico Gadelha. Solar Energy Forecasting With Fuzzy Time Series Using High-Order Fuzzy Cognitive Maps. **FUZZ-IEEE:** IEEE International Conference on Fuzzy Systems, 2020.

REFERENCES

AGUILAR, J. A dynamic fuzzy-cognitive-map approach based on random neural networks. *International Journal of Computational Cognition*, v. 1, n. 4, p. 91–107, 2003.

AGUILAR, J. A survey about fuzzy cognitive maps papers. *International Journal of Computational Cognition*, v. 3, 01 2005.

AHMADI, S. et al. Learning fuzzy cognitive maps using imperialist competitive algorithm. *Neural Comput. Appl.*, Springer-Verlag, Berlin, Heidelberg, v. 26, n. 6, p. 1333–1354, ago. 2015. ISSN 0941-0643.

AHMADI, S. et al. A first study of fuzzy cognitive maps learning using cultural algorithm. In: . [S.l.: s.n.], 2014. p. 2023–2028.

AL-GUNAID, M. A. et al. A survey of fuzzy cognitive maps forecasting methods. In: IEEE. *2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA)*. [S.l.], 2017. p. 1–6.

ALIZADEH, S.; GHAZANFARI, M. Learning fcm by chaotic simulated annealing. *Chaos, Solitons & Fractals*, Elsevier, v. 41, n. 3, p. 1182–1190, 2009.

ALIZADEH, S. et al. Learning fcm by tabu search. *International Journal of Computer Science*, Citeseer, v. 2, n. 2, p. 142–149, 2007.

ALVES, M. et al. An extension of nonstationary fuzzy sets to heteroskedastic fuzzy time series. In: . [S.l.: s.n.], 2018.

AMIRKHANI, A. et al. A review of fuzzy cognitive maps in medicine: Taxonomy, methods, and applications. *Computer methods and programs in biomedicine*, Elsevier, v. 142, p. 129–145, 2017.

ANDREOU, A.; MATEOU, N.; ZOMBANAKIS, G. Evolutionary fuzzy cognitive maps: A hybrid system for crisis management and political decision making. In: . [S.l.: s.n.], 2003.

ANDREOU, A.; MATEOU, N.; ZOMBANAKIS, G. Soft computing for crisis management and political decision making: The use of genetically evolved fuzzy cognitive maps. *Soft Comput.*, v. 9, p. 194–210, 02 2005.

AXELROD, R. (Ed.). *Structure of Decision: The Cognitive Maps of Political Elites*. [S.l.]: Princeton Legacy Library, 1976.

BAGDATLI, M. E. C.; DOKUZ, A. S. Modeling discretionary lane-changing decisions using an improved fuzzy cognitive map with association rule mining. *Transportation Letters*, v. 13, n. 8, p. 623–633, 2021. ISSN 1942-7867.

BAYKASOğLU, A.; DURMUSOGLU, Z. D. U.; KAPLANOGLU, V. Training fuzzy cognitive maps via extended great deluge algorithm with applications. *Comput. Ind.*, v. 62, p. 187–195, 2011.

BEENA, P.; GANGULI, R. Structural damage detection using fuzzy cognitive maps and hebbian learning. *Appl. Soft Comput.*, v. 11, p. 1014–1020, 2011.

BOSE, M.; MALI, K. Designing fuzzy time series forecasting models: A survey. *International Journal of Approximate Reasoning*, Elsevier, v. 111, p. 78–99, 2019.

BOUTALIS, Y.; KOTTAS, T. L.; CHRISTODOULOU, M. Adaptive estimation of fuzzy cognitive maps with proven stability and parameter convergence. *IEEE Transactions on Fuzzy Systems*, IEEE, v. 17, n. 4, p. 874–889, 2009.

BREIMAN, L. Random forests. *Machine learning*, Springer, v. 45, n. 1, p. 5–32, 2001.

BUENO, S.; SALMERON, J. L. Benchmarking main activation functions in fuzzy cognitive maps. *Expert Systems with Applications*, Elsevier, v. 36, n. 3, p. 5221–5229, 2009.

CAI, Y. et al. Creating an immersive game world with evolutionary fuzzy cognitive maps. *IEEE computer graphics and applications*, IEEE, v. 30, n. 2, p. 58–70, 2009.

CANDANEDO, L. M.; FELDHEIM, V.; DERAMAIX, D. Data driven prediction models of energy use of appliances in a low-energy house. *Energy and Buildings*, v. 140, p. 81–97, 2017. ISSN 0378-7788.

CARVALHO, J.; TOMé, J. Rule based fuzzy cognitive maps - expressing time in qualitative system dynamics. In: . [S.l.: s.n.], 2001. v. 1, p. 280 – 283. ISBN 0-7803-7293-X.

CARVALHO, J.; TOMé, J. Qualitative optimization of fuzzy causal rule bases using fuzzy boolean nets. *Fuzzy Sets and Systems*, v. 158, p. 1931–1946, 09 2007.

CHAI, M. et al. Pv power prediction based on lstm with adaptive hyperparameter adjustment. *Ieee Access*, IEEE, v. 7, p. 115473–115486, 2019.

CHAMMAS, M.; MAKHOUL, A.; DEMERJIAN, J. An efficient data model for energy prediction using wireless sensors. *Comput. Electr. Eng.*, v. 76, p. 249–257, 2019.

CHEN, Y.; MAZLACK, L.; LU, L. Inferring fuzzy cognitive map models for gene regulatory networks from gene expression data. In: . [S.l.: s.n.], 2012. p. 1–4. ISBN 978-1-4673-2559-2.

CHEN, Y. et al. Inferring causal networks using fuzzy cognitive maps and evolutionary algorithms with application to gene regulatory network reconstruction. *Applied Soft Computing*, Elsevier, v. 37, p. 667–679, 2015.

CHI, Y.; LIU, J. Learning large-scale fuzzy cognitive maps using a hybrid of memetic algorithm and neural network. In: . [S.l.: s.n.], 2014. p. 1036–1040.

CHI, Y.; LIU, J. Learning of fuzzy cognitive maps with varying densities using a multiobjective evolutionary algorithm. *IEEE Transactions on Fuzzy Systems*, v. 24, p. 1–1, 01 2015.

CHUNG, J. et al. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

CHUNMEI, L.; YUE, H. Cellular automata learning of fuzzy cognitive map. In: . [S.l.: s.n.], 2012. p. 334–338. ISBN 978-1-4673-0944-8.

CHUNYING, Z. et al. Research of rough cognitive map model. *Communications in Computer and Information Science*, v. 144, 01 2011.

DICKERSON, J. A.; KOSKO, B. Virtual worlds as fuzzy cognitive maps. In: *Proceedings of the 1993 IEEE Virtual Reality Annual International Symposium*. USA: IEEE Computer Society, 1993. (VRAIS '93), p. 471–477. ISBN 0780313631.

DING, F.; LUO, C. Interpretable cognitive learning with spatial attention for high-volatility time series prediction. *Applied Soft Computing*, Elsevier, p. 108447, 2022.

DING, Z.; LI, J. J. D. First study of fuzzy cognitive map learning using ants colony optimization. Citeseer, 2011.

DRUCKER, H. et al. Support vector regression machines. *Advances in neural information processing systems*, v. 9, 1996.

DUA, D.; GRAFF, C. *UCI Machine Learning Repository*. 2017.

EDEN, C. Analyzing cognitive maps to help structure issues or problems. *European Journal of Operational Research*, v. 159, n. 3, p. 673–686, 2004. ISSN 0377-2217.

EDEN, C.; ACKERMANN, F. Cognitive mapping expert views for policy analysis in the public sector. *European Journal of Operational Research*, v. 152, p. 615–630, 02 2004.

EDEN, C.; ACKERMANN, F.; CROPPER, S. The analysis of cause maps. *Journal of Management Studies*, v. 29, n. 3, p. 309–324, 5 1992. ISSN 0022-2380.

FELIX, G. et al. A review on methods and software for fuzzy cognitive maps. *Artificial intelligence review*, Springer, v. 52, n. 3, p. 1707–1737, 2019.

FENG, G. et al. The learning of fuzzy cognitive maps with noisy data: A rapid and robust learning method with maximum entropy. *IEEE Transactions on Cybernetics*, v. 51, p. 2080–2092, 2021.

FENG, G.; LU, W.; YANG, J. The modeling of time series based on least square fuzzy cognitive map. *Algorithms*, v. 14, n. 3, 2021. ISSN 1999-4893.

FENG, G.; LU, W.; YANG, J. Time series modeling with fuzzy cognitive maps based on partitioning strategies. In: *2021 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. [S.l.: s.n.], 2021. p. 1–6.

FENG, G. et al. Long-term prediction of time series using fuzzy cognitive maps. *Engineering Applications of Artificial Intelligence*, Elsevier, v. 102, p. 104274, 2021.

FROELICH, W.; JUSZCZUK, P. Predictive capabilities of adaptive and evolutionary fuzzy cognitive maps-a comparative study. In: *Intelligent systems for knowledge management*. [S.l.]: Springer, 2009. p. 153–174.

FROELICH, W.; PAPAGEORGIOU, E. I. Extended evolutionary learning of fuzzy cognitive maps for the prediction of multivariate time-series. In: *Fuzzy cognitive maps for applied sciences and engineering*. [S.l.]: Springer, 2014. p. 121–131.

FROELICH, W. et al. Application of evolutionary fuzzy cognitive maps to the long-term prediction of prostate cancer. *Appl. Soft Comput.*, v. 12, p. 3810–3817, 2012.

FROELICH, W.; PEDRYCZ, W. Fuzzy cognitive maps in the modeling of granular time series. *Knowledge-Based Systems*, v. 115, 10 2016.

FROELICH, W.; SALMERON, J. L. Evolutionary learning of fuzzy grey cognitive maps for the forecasting of multivariate, interval-valued time series. *International Journal of Approximate Reasoning*, v. 55, n. 6, p. 1319–1335, 2014. ISSN 0888-613X.

GAO, R.; DU, L.; YUEN, K. F. Robust empirical wavelet fuzzy cognitive map for time series forecasting. *Engineering Applications of Artificial Intelligence*, Elsevier, v. 96, p. 103978, 2020.

GEURTS, P.; ERNST, D.; WEHENKEL, L. Extremely randomized trees. *Machine learning*, Springer, v. 63, n. 1, p. 3–42, 2006.

GHAZANFARI, M. et al. Comparing simulated annealing and genetic algorithm in learning fcm. *Applied Mathematics and Computation*, Elsevier, v. 192, n. 1, p. 56–68, 2007.

GLYKAS, M. *Fuzzy cognitive maps. Advances in theory, methodologies, tools and applications*. [S.l.: s.n.], 2010. v. 247.

GREGOR, M.; GROUMPOS, P. Training fuzzy cognitive maps using gradient-based supervised learning. In: . [S.l.: s.n.], 2013. v. 412, p. 547–556. ISBN 9783642411427.

GREGOR, M.; GROUMPOS, P. Tuning the position of a fuzzy cognitive map attractor using backpropagation through time. In: . [S.l.: s.n.], 2013. v. 1.

GREGOR, M.; GROUMPOS, P. P. Training fuzzy cognitive maps using gradient-based supervised learning. In: SPRINGER. *IFIP International Conference on Artificial Intelligence Applications and Innovations*. [S.l.], 2013. p. 547–556.

GROUMPOS, P. P. Modelling business and management systems using fuzzy cognitive maps: A critical overview. *IFAC-PapersOnLine*, v. 48, n. 24, p. 207–212, 2015. ISSN 2405-8963. 16th IFAC Conference on Technology, Culture and International Stability TECIS 2015.

GUBBI, J. et al. Internet of things (iot): A vision, architectural elements, and future directions. *Future Gener. Comput. Syst.*, v. 29, n. 7, p. 1645–1660, 2013.

HAJEK, P.; FROELICH, W.; PROCHAZKA, O. Intuitionistic fuzzy grey cognitive maps for forecasting interval-valued time series. *Neurocomputing*, Elsevier, v. 400, p. 173–185, 2020.

HAJEK, P.; PROCHAZKA, O.; FROELICH, W. Interval-valued intuitionistic fuzzy cognitive maps for stock index forecasting. In: IEEE. *2018 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS)*. [S.l.], 2018. p. 1–7.

HAN, S. I.; LEE, J. myung. Fuzzy echo state neural networks and funnel dynamic surface control for prescribed performance of a nonlinear dynamic system. *IEEE Transactions on Industrial Electronics*, v. 61, p. 1099–1112, 2014.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural Computation*, v. 9, n. 8, p. 1735–1780, 1997.

HOMENDA, W.; JASTRZEBSKA, A. Clustering techniques for fuzzy cognitive map design for time series modeling. *Neurocomputing*, v. 232, 12 2016.

HOMENDA, W.; JASTRZEBSKA, A.; PEDRYCZ, W. Joining concept's based fuzzy cognitive map model with moving window technique for time series modeling. In: . [S.l.: s.n.], 2014. p. 397–408.

HOMENDA, W.; JASTRZEBSKA, A.; PEDRYCZ, W. Modeling time series with fuzzy cognitive maps. In: . [S.l.: s.n.], 2014. p. 2055–2062.

HOMENDA, W.; JASTRZEBSKA, A.; PEDRYCZ, W. Time series modeling with fuzzy cognitive maps: Simplification strategies - the case of a posteriori removal of nodes and weights. In: *CISIM*. [S.l.: s.n.], 2014.

HUANG, D.; SHEN, Z. A curious learning model with elm for fuzzy cognitive maps. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, v. 21, 10 2013.

HUERGA, A. V. A balanced differential learning algorithm in fuzzy cognitive maps. In: *In: Proceedings of the 16th International Workshop on Qualitative Reasoning 2002*. [S.l.: s.n.], 2002.

IAKOVIDIS, D.; PAPAGEORGIOU, E. Intuitionistic fuzzy cognitive maps for medical decision making. *IEEE transactions on information technology in biomedicine : a publication of the IEEE Engineering in Medicine and Biology Society*, v. 15, p. 100–7, 01 2011.

JAEGER, H. The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, Bonn, v. 148, n. 34, p. 13, 2001.

JETTER, A. Fuzzy cognitive maps for engineering and technology management: What works in practice? In: . [S.l.: s.n.], 2006. p. 498 – 512.

JUSZCZUK, P.; FROELICH, W. Learning fuzzy cognitive maps using a differential evolution algorithm. *Pol. J. Environ. Stud*, v. 12, n. 3B, p. 108, 2009.

KAGGLE. *Smart Home Dataset with weather Information*. 2021. Accessed on 28 Ago 2021.

KETIPI, M. K. et al. Multi-criteria decision making using fuzzy cognitive maps – preliminary results. *Procedia Manufacturing*, v. 51, p. 1305–1310, 2020. ISSN 2351-9789. 30th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2021).

KHAN, Z. A. et al. Electrical energy prediction in residential buildings for short-term horizons using hybrid deep learning strategy. *Applied Sciences*, Multidisciplinary Digital Publishing Institute, v. 10, n. 23, p. 8634, 2020.

KIM, D.-H. Cognitive maps of policy makers on financial crises of south korea and malaysia: A comparative study. *International Review of Public Administration*, Routledge, v. 9, n. 2, p. 31–38, 2004.

KIM, K. I.; FRANZ, M. O.; SCHÖLKOPF, B. Iterative kernel principal component analysis for image modeling. *IEEE Trans. Pattern Anal. Mach. Intell.*, v. 27, n. 9, p. 1351–1366, 2005.

KLEIN, J. H.; COOPER, D. F. Cognitive maps of decision-makers in a complex game. *Journal of the Operational Research Society*, Taylor Francis, v. 33, n. 1, p. 63–71, 1982.

KOK, K. The potential of fuzzy cognitive maps for semi-quantitative scenario development, with an example from brazil. *Global Environmental Change*, v. 19, p. 122–133, 02 2009.

KONAR, A.; CHAKRABORTY, U. K. Reasoning and unsupervised learning in a fuzzy cognitive map. *Information Sciences*, Elsevier, v. 170, n. 2-4, p. 419–441, 2005.

KOSKO, B. Fuzzy cognitive maps. *International Journal of Man-Machine Studies*, v. 24, n. 1, p. 65 – 75, 1986. ISSN 0020-7373.

KOTTAS, T. L.; BOUTALIS, Y. S.; CHRISTODOULOU, M. A. Fuzzy cognitive network: A general framework. *Intelligent Decision Technologies*, IOS Press, v. 1, n. 4, p. 183–196, 2007.

KOTTAS, T. L.; BOUTALIS, Y. S.; CHRISTODOULOU, M. A. Fuzzy cognitive networks: Adaptive network estimation and control paradigms. In: *Fuzzy Cognitive Maps*. [S.l.]: Springer, 2010. p. 89–134.

KOULOURIOTIS, D.; DIAKOULAKIS, I.; EMIRIS, D. Learning fuzzy cognitive maps using evolution strategies: a novel schema for modeling and simulating high-level behavior. In: IEEE. *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No. 01TH8546)*. [S.l.], 2001. v. 1, p. 364–371.

KYRIAKARAKOS, G. et al. A fuzzy cognitive maps–petri nets energy management system for autonomous polygeneration microgrids. *Applied Soft Computing*, Elsevier, v. 12, n. 12, p. 3785–3797, 2012.

LAUREANO-CRUCES, A.; RAMIREZ, J.; TERAN, A. Evaluation of the teaching-learning process with fuzzy cognitive maps. In: . [S.l.: s.n.], 2004. v. 3315, p. 922–931.

LEE, M. H.; JAVEDANI, H. et al. A weighted fuzzy integrated time series for forecasting tourist arrivals. In: SPRINGER. *International Conference on Informatics Engineering and Information Science*. [S.l.], 2011. p. 206–217.

LEMKE, C.; GABRYS, B. Meta-learning for time series forecasting and forecast combination. *Neurocomputing*, Elsevier, v. 73, n. 10-12, p. 2006–2016, 2010.

LI, S.-J.; SHEN, R.-M. Fuzzy cognitive map learning based on improved nonlinear hebbian rule. In: IEEE. *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 04EX826)*. [S.l.], 2004. v. 4, p. 2301–2306.

LI, X. et al. A novel team-centric peer selection scheme for distributed wireless p2p networks. In: . [S.l.: s.n.], 2009. p. 1 – 5.

LI, Y.; LI, F. Pso-based growing echo state network. *Applied Soft Computing*, v. 85, p. 105774, 2019. ISSN 1568-4946.

LIN, C. An immune algorithm for complex fuzzy cognitive map partitioning. In: *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation*. [S.l.: s.n.], 2009. p. 315–320.

LIU, J.; CHI, Y.; ZHU, C. A dynamic multiagent genetic algorithm for gene regulatory network reconstruction based on fuzzy cognitive maps. *IEEE Transactions on Fuzzy Systems*, IEEE, v. 24, n. 2, p. 419–431, 2015.

LIU, L.; LIU, J. Inferring gene regulatory networks with hybrid of multi-agent genetic algorithm and random forests based on fuzzy cognitive maps. *Appl. Soft Comput.*, v. 69, p. 585–598, 2018.

LIU, P.; LIU, J.; WU, K. Cnn-fcm: System modeling promotes stability of deep learning in time series prediction. *Knowledge-Based Systems*, v. 203, p. 106081, 2020. ISSN 0950-7051.

LIU, Z.; LIU, J. A robust time series prediction method based on empirical mode decomposition and high-order fuzzy cognitive maps. *Knowledge-Based Systems*, v. 203, p. 106105, 2020. ISSN 0950-7051.

LU, W.; YANG, J.; LI, Y. Control method based on fuzzy cognitive map and its application on district heating network. 08 2010.

LU, W.; YANG, J.; LIU, X. The linguistic forecasting of time series based on fuzzy cognitive maps. In: IEEE. *2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS)*. [S.l.], 2013. p. 649–654.

LU, W.; YANG, J.; LIU, X. Numerical prediction of time series based on fcms with information granules. *International Journal of Computers Communications Control*, v. 9, p. 313, 04 2014.

LU, W. et al. The modeling and prediction of time series based on synergy of high-order fuzzy cognitive map and fuzzy c-means clustering. *Knowledge-Based Systems*, v. 70, p. 242–255, 07 2014.

LUO, X.; WEI, X.; ZHANG, J. Game-based learning model using fuzzy cognitive map. In: *Proceedings of the First ACM International Workshop on Multimedia Technologies for Distance Learning*. New York, NY, USA: Association for Computing Machinery, 2009. (MTDL '09), p. 67–76. ISBN 9781605587578.

MA, Q.; SHEN, L.; COTTRELL, G. W. Deep-esn: A multiple projection-encoding hierarchical reservoir computing framework. *arXiv preprint arXiv:1711.05255*, 2017.

MADEIRO, S.; ZUBEN, F. V. Gradient-based algorithms for the automatic construction of fuzzy cognitive maps. In: . [S.l.: s.n.], 2012. v. 1, p. 344 –349.

MAHMOUD, T. A. et al. Direct adaptive control for nonlinear systems using a tsk fuzzy echo state network based on fractional-order learning algorithm. *Journal of the Franklin Institute*, v. 358, n. 17, p. 9034–9060, 2021. ISSN 0016-0032.

MAKRIDAKIS, S.; WINKLER, R. L. Averages of forecasts: Some empirical results. *Management science*, Informs, v. 29, n. 9, p. 987–996, 1983.

MARTENS, M. Measuring and forecasting sp 500 index-futures volatility using high-frequency data. *Journal of Futures Markets*, v. 22, p. 497 – 518, 06 2002.

MATEOU, N. H.; MOISEOS, M.; ANDREOU, A. S. Multi-objective evolutionary fuzzy cognitive maps for decision support. In: IEEE. *2005 IEEE Congress on Evolutionary Computation*. [S.l.], 2005. v. 1, p. 824–830.

MIAO, Y. et al. Dynamical cognitive network–an extension of fuzzy cognitive map. *Fuzzy Systems, IEEE Transactions on*, v. 9, p. 760 – 770, 11 2001.

MIAO, Y. et al. Dynamical cognitive network-an extension of fuzzy cognitive map. *IEEE transactions on Fuzzy Systems*, IEEE, v. 9, n. 5, p. 760–770, 2001.

MLS, K. et al. Interactive evolutionary optimization of fuzzy cognitive maps. *Neurocomputing*, v. 232, p. 58–68, 2017.

MORRIS, R. G. D.o. hebb: The organization of behavior, wiley: New york; 1949. *Brain Research Bulletin*, v. 50, 1999.

NAIR, A.; RECKIEN, D.; MAARSEVEEN, M. V. A generalised fuzzy cognitive mapping approach for modelling complex systems. *Applied Soft Computing*, Elsevier, v. 84, p. 105754, 2019.

NANNAN, Z.; CHAO, L. Adaptive online time series prediction based on a novel dynamic fuzzy cognitive map. *J. Intell. Fuzzy Syst.*, IOS Press, NLD, v. 36, n. 6, p. 5291–5303, jan 2019. ISSN 1064-1246.

NAPOLES, G.; BELLO, R.; VANHOOF, K. How to improve the convergence on sigmoid fuzzy cognitive maps? *Intelligent Data Analysis*, v. 18, p. 77–88, 01 2014.

NAPOLES, G. et al. Two-steps learning of fuzzy cognitive maps for prediction and knowledge discovery on the hiv-1 drug resistance. *Expert Systems with Applications*, v. 41, n. 3, p. 821–830, 2014. ISSN 0957-4174. Methods and Applications of Artificial and Computational Intelligence.

NAPOLES, G. et al. Deterministic learning of hybrid fuzzy cognitive maps and network reduction approaches. *Neural Networks*, Elsevier, v. 124, p. 258–268, 2020.

NAPOLES, G. et al. On the convergence of sigmoid fuzzy cognitive maps. *Information Sciences*, Elsevier, v. 349, p. 154–171, 2016.

NAPOLES, G. et al. Learning and convergence of fuzzy cognitive maps used in pattern recognition. *Neural Processing Letters*, Springer, v. 45, n. 2, p. 431–444, 2017.

OIKONOMOU, P.; PAPAGEORGIOU, E. I. Particle swarm optimization approach for fuzzy cognitive maps applied to autism classification. In: SPRINGER. *IFIP International Conference on Artificial Intelligence Applications and Innovations*. [S.l.], 2013. p. 516–526.

ORANG, O. et al. Randomized high order fuzzy cognitive maps as reservoir computing models: A first introduction and applications. *Neurocomputing*, 2022. ISSN 0925-2312.

ORANG, O.; SILVA, P. C. Lima e; GUIMARAES, F. G. Time series forecasting using fuzzy cognitive maps: a survey. *Artificial Intelligence Review*, Springer, p. 1–62, 2022.

ORANG, O. et al. Solar energy forecasting with fuzzy time series using high-order fuzzy cognitive maps. In: *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. [S.l.: s.n.], 2020. p. 1–8.

PANIGRAHI, S.; BEHERA, H. S. A study on leading machine learning techniques for high order fuzzy time series forecasting. *Engineering Applications of Artificial Intelligence*, Elsevier, v. 87, p. 103245, 2020.

PAPAGEORGIOU; POCZETA; LASPIDOU. Hybrid model for water demand prediction based on fuzzy cognitive maps and artificial neural networks. In: *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. [S.l.: s.n.], 2016. p. 1523–1530.

PAPAGEORGIOU, E. A weight adaptation method for fine-tuning fuzzy cognitive map causal links. *Soft Computing*, v. 9, p. 846–857, 01 2005.

PAPAGEORGIOU, E.; GROUMPOS, P. A weight adaptation method for fuzzy cognitive map learning. *Soft Comput.*, v. 9, p. 846–857, 11 2005.

PAPAGEORGIOU, E. et al. A concept reduction approach for fuzzy cognitive map models in decision making and management. *Neurocomputing*, v. 232, 12 2016.

PAPAGEORGIOU, E.; POCZETA, K. Application of fuzzy cognitive maps to electricity consumption prediction. In: . [S.l.: s.n.], 2015.

PAPAGEORGIOU, E.; POCZETA, K. A two-stage model for time series prediction based on fuzzy cognitive maps and neural networks. *Neurocomputing*, v. 232, 12 2016.

PAPAGEORGIOU, E. et al. Exploring an ensemble of methods that combines fuzzy cognitive maps and neural networks in solving the time series prediction problem of gas consumption in greece. *Algorithms*, v. 12, p. 235, 11 2019.

PAPAGEORGIOU, E.; POCZETA, K.; LASPIDOU, C. Application of fuzzy cognitive maps to water demand prediction. In: . [S.l.: s.n.], 2015.

PAPAGEORGIOU, E. et al. Fuzzy cognitive maps and multi-step gradient methods for prediction: Applications to electricity consumption and stock exchange returns. In: _____. [S.l.: s.n.], 2015. v. 39, p. 501–511. ISBN 978-3-319-19856-9.

PAPAGEORGIOU, E.; STYLIOS, C.; GROUMPOS, P. Fuzzy cognitive map learning based on nonlinear hebbian rule. In: SPRINGER. *Australasian Joint Conference on Artificial Intelligence*. [S.l.], 2003. p. 256–268.

PAPAGEORGIOU, E.; STYLIOS, C.; GROUMPOS, P. Active hebbian learning algorithm to train fuzzy cognitive maps. *International Journal of Approximate Reasoning*, v. 37, p. 219–249, 11 2004.

PAPAGEORGIOU, E. I. Learning algorithms for fuzzy cognitive maps—a review study. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, IEEE, v. 42, n. 2, p. 150–163, 2011.

PAPAGEORGIOU, E. I. A new methodology for decisions in medical informatics using fuzzy cognitive maps based on fuzzy rule-extraction techniques. *Appl. Soft Comput.*, Elsevier Science Publishers B. V., NLD, v. 11, n. 1, p. 500–513, jan. 2011. ISSN 1568-4946.

PAPAGEORGIOU, E. I. Fuzzy cognitive maps for applied sciences and engineering - from fundamentals to extensions and learning algorithms. In: *Fuzzy Cognitive Maps for Applied Sciences and Engineering*. [S.l.: s.n.], 2014.

PAPAGEORGIOU, E. I.; FROELICH, W. Multi-step prediction of pulmonary infection with the use of evolutionary fuzzy cognitive maps. *Neurocomputing*, v. 92, p. 28–35, 2012. ISSN 0925-2312. Data Mining Applications and Case Study.

PAPAGEORGIOU, E. I.; GROUMPOS, P. P. A new hybrid method using evolutionary algorithms to train fuzzy cognitive maps. *Appl. Soft Comput.*, Elsevier Science Publishers B. V., NLD, v. 5, n. 4, p. 409–431, jul. 2005. ISSN 1568-4946.

PAPAGEORGIOU, E. I.; SALMERON, J. L. A review of fuzzy cognitive maps research during the last decade. *IEEE Transactions on Fuzzy Systems*, IEEE, v. 21, n. 1, p. 66–79, 2012.

PAPAGEORGIOU, E. I.; STYLIOS, C.; GROUMPOS, P. P. Unsupervised learning techniques for fine-tuning fuzzy cognitive map causal links. *International Journal of Human-Computer Studies*, Elsevier, v. 64, n. 8, p. 727–743, 2006.

PAPAKOSTAS, G.; KOULOURIOTIS, D. Classifying patterns using fuzzy cognitive maps. In: *Fuzzy cognitive maps*. [S.l.]: Springer, 2010. p. 291–306.

PAPAKOSTAS, G. et al. Towards hebbian learning of fuzzy cognitive maps in pattern classification problems. *Expert Systems with Applications*, v. 39, p. 10620–10629, 09 2012.

PAPAKOSTAS, G. A. et al. Training fuzzy cognitive maps by using hebbian learning algorithms: a comparative study. In: IEEE. *2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*. [S.l.], 2011. p. 851–858.

PARHIZKAR, T.; RAFIEIPOUR, E.; PARHIZKAR, A. Evaluation and improvement of energy consumption prediction models using principal component analysis based feature reduction. *Journal of Cleaner Production*, v. 279, p. 123866, 2021. ISSN 0959-6526.

PARK, K. S.; KIM, S. H. Fuzzy cognitive maps considering time relationships. *International Journal of Human-Computer Studies*, Elsevier, v. 42, n. 2, p. 157–168, 1995.

PARSOPOULOS, K.; PAPAGEORGIOU, E.; VRAHATIS, M. A first study of fuzzy cognitive maps learning using particle swarm optimization. In: . [S.l.: s.n.], 2002.

PEDRYCZ, W.; JASTRZEBSKA, A.; HOMENDA, W. Design of fuzzy cognitive maps for modeling time series. *IEEE Transactions on Fuzzy Systems*, IEEE Press Piscataway, NJ, USA, v. 24, n. 1, p. 120–130, 2016.

PETALAS, Y. et al. Fuzzy cognitive maps learning using memetic algorithms. In: *Proceedings of the international conference of "Computational Methods in Sciences and Engineering"(ICCMSE 2005)*. [S.l.: s.n.], 2005. p. 1420–1423.

POCZETA, K.; PAPAGEORGIOU, E. I. Implementing fuzzy cognitive maps with neural networks for natural gas prediction. In: IEEE. *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*. [S.l.], 2018. p. 1026–1032.

POCZETA, K.; PAPAGEORGIOU, E. I.; GEROGIANNIS, V. C. Fuzzy cognitive maps optimization for decision making and prediction. *Mathematics*, MDPI, v. 8, n. 11, p. 2059, 2020.

POCZETA, K.; PAPAGEORGIOU, E. I.; YASTREBOV, A. Application of fuzzy cognitive maps to multi-step ahead prediction of electricity consumption. In: *2018 Conference on Electrotechnology: Processes, Models, Control and Computer Science (EPMCCS)*. [S.l.: s.n.], 2018. p. 1–5.

POCZETA, K.; YASTREBOV, A.; PAPAGEORGIOU, E. I. Learning fuzzy cognitive maps using structure optimization genetic algorithm. In: IEEE. *2015 federated conference on computer science and information systems (FedCSIS)*. [S.l.], 2015. p. 547–554.

POCZKETA, K.; YASTREBOV, A. Monitoring and prediction of time series based on fuzzy cognitive maps with multi-step gradient methods. In: SPRINGER. *International Conference on Automation*. [S.l.], 2015. p. 197–206.

QIANG, S.; LELAND, R. P.; CHISSOM, B. S. Fuzzy stochastic fuzzy time series and its models. *Fuzzy Sets and Systems*, v. 88, n. 3, p. 333 – 341, 1997. ISSN 0165-0114.

RAJARAM, T.; DAS, A. Modeling of interactions among sustainability components of an agro-ecosystem using local knowledge through cognitive mapping and fuzzy inference system. *Expert Syst. Appl.*, Pergamon Press, Inc., USA, v. 37, n. 2, p. 1734–1744, mar. 2010. ISSN 0957-4174.

RAMIREZ-BAUTISTA, J. A. et al. Classification of plantar foot alterations by fuzzy cognitive maps against multi-layer perceptron neural network. *Biocybernetics and Biomedical Engineering*, v. 40, n. 1, p. 404–414, 2020. ISSN 0208-5216.

REN, Z. Learning fuzzy cognitive maps by a hybrid method using nonlinear hebbian learning and extended great deluge algorithm. *Proceedings of the 23rd Midwest Artificial Intelligence and Cognitive Science Conference, MAICS 2012*, v. 841, 01 2012.

REZAEE, M. J.; YOUSEFI, S.; BABAEI, M. Multi-stage cognitive map for failures assessment of production processes: An extension in structure and algorithm. *Neurocomputing*, v. 232, 12 2016.

RUAN, D.; HARDEMAN, F.; MKRTCHYAN, L. Using belief degree-distributed fuzzy cognitive maps in nuclear safety culture assessment. *Advances in Intelligent and Soft Computing*, v. 124, p. 1–6, 03 2011.

SADAEI, H. J. et al. Short-term load forecasting by using a combined method of convolutional neural networks and fuzzy time series. *Energy*, v. 175, p. 365–377, 2019. ISSN 0360-5442.

SADAEI, H. J. et al. Short-term load forecasting method based on fuzzy time series, seasonality and long memory process. *Int. J. Approx. Reasoning*, Elsevier Science Inc., USA, v. 83, n. C, p. 196–217, abr. 2017. ISSN 0888-613X.

SAJJAD, M. et al. A novel cnn-gru-based hybrid approach for short-term residential load forecasting. *IEEE Access*, v. 8, p. 143759–143768, 2020.

SALMERON, J. Augmented fuzzy cognitive maps for modelling lms critical success factors. *Knowledge-Based Systems*, v. 22, p. 275–278, 05 2009.

SALMERON, J.; PAPAGEORGIOU, E. A fuzzy grey cognitive maps-based decision support system for radiotherapy treatment planning. *Knowledge-Based Systems*, v. 30, p. 151–160, 06 2012.

SALMERON, J.; PAPAGEORGIOU, E. Fuzzy grey cognitive maps and nonlinear hebbian learning in process control. *Applied Intelligence*, v. 41, 12 2013.

SALMERON, J. L. Supporting decision makers with fuzzy cognitive maps. *Research-Technology Management*, Routledge, v. 52, n. 3, p. 53–59, 2009.

SALMERON, J. L. Modelling grey uncertainty with fuzzy grey cognitive maps. *Expert Systems with Applications*, Elsevier, v. 37, n. 12, p. 7581–7588, 2010.

SALMERON, J. L.; FROELICH, W. Dynamic optimization of fuzzy cognitive maps for time series forecasting. *Know.-Based Syst.*, Elsevier Science Publishers B. V., NLD, v. 105, n. C, p. 29–37, ago. 2016. ISSN 0950-7051.

SALMERON, J. L. et al. Learning fuzzy cognitive maps with modified asexual reproduction optimisation algorithm. *Knowledge-Based Systems*, Elsevier, v. 163, p. 723–735, 2019.

SHAN, D.; LU, W.; YANG, J. The data-driven fuzzy cognitive map model and its application to prediction of time series. *International Journal of Innovative Computing, Information and Control*, v. 14, p. 1583–1602, 10 2018.

SHANCHAO, Y.; LIU, J. Time-series forecasting based on high-order fuzzy cognitive maps and wavelet transform. *IEEE Transactions on Fuzzy Systems*, PP, p. 1–1, 04 2018.

SHEN, F.; LIU, J.; WU, K. Evolutionary multitasking fuzzy cognitive map learning. *Knowledge-Based Systems*, v. 192, 11 2019.

SHEN, F.; LIU, J.; WU, K. Multivariate time series forecasting based on elastic net and high-order fuzzy cognitive maps: A case study on human action prediction through eeg signals. *IEEE Transactions on Fuzzy Systems*, v. 29, n. 8, p. 2336–2348, 2021.

SILVA, P. C. L. *Scalable Models for Probabilistic Forecasting with Fuzzy Time Series*. Tese (Doutorado) — UFMG, 09 2019.

SILVA, P. C. Lima e et al. Probabilistic forecasting with seasonal ensemble fuzzy time-series. In: *XIII Brazilian Congress on Computational Intelligence, Rio de Janeiro*. [S.l.: s.n.], 2017.

SILVA, P. C. Lima e et al. *PYFTS/pyFTS: Stable version 1.7*. [S.l.]: Zenodo, 2019. - Type hints - New methods - Performance improvements - Bugfixes.

SILVA, P. C. Lima e et al. Forecasting in non-stationary environments with fuzzy time series. *Applied Soft Computing*, v. 97, p. 106825, 2020. ISSN 1568-4946.

SILVA, P. C. Lima e et al. Probabilistic forecasting with fuzzy time series. *IEEE Transactions on Fuzzy Systems*, v. 28, n. 8, p. 1771–1784, 2020.

SILVA, P. C. Lima e; SADAEI, H. J.; GUIMARãES, F. G. Interval Forecasting with Fuzzy Time Series. In: *IEEE Symposium Series on Computational Intelligence (IEEE SSCI 2016)*. Athens, Greece: [s.n.], 2016. ISBN 9781509042401.

SINGH, P. A brief review of modeling approaches based on fuzzy time series. *International Journal of Machine Learning and Cybernetics*, Springer, v. 8, n. 2, p. 397–420, 2017.

SONG, H. et al. Implementation of fuzzy cognitive maps using fuzzy neural network and application in prediction of time series. *IEEE T. Fuzzy Systems*, v. 18, p. 233–250, 04 2010.

SONG, H. et al. An extension to fuzzy cognitive maps for classification and prediction. *IEEE T. Fuzzy Systems*, v. 19, p. 116–135, 02 2011.

SONG, H. et al. Design of fuzzy cognitive maps using neural networks for predicting chaotic time series. *Neural networks : the official journal of the International Neural Network Society*, v. 23, p. 1264–75, 12 2010.

SONG, Q.; CHISSOM, B. Forecasting enrollments with fuzzy time series—part ii. *Fuzzy Sets and Systems*, v. 62, p. 1–8, 02 1994.

SONG, Q.; CHISSOM, B. S. Forecasting enrollments with fuzzy time series—part i. *Fuzzy sets and systems*, Elsevier, v. 54, n. 1, p. 1–9, 1993.

STACH, W.; KURGAN, L.; PEDRYCZ, W. A survey of fuzzy cognitive map learning methods. *Issues in soft computing: theory and applications*, Exit, p. 71–84, 2005.

STACH, W.; KURGAN, L.; PEDRYCZ, W. Parallel learning of large fuzzy cognitive maps. In: . [S.l.: s.n.], 2007. p. 1584–1589.

STACH, W.; KURGAN, L.; PEDRYCZ, W. Data-driven nonlinear hebbian learning method for fuzzy cognitive maps. In: IEEE. *2008 IEEE international conference on fuzzy systems (IEEE world congress on computational intelligence)*. [S.l.], 2008. p. 1975–1981.

STACH, W.; KURGAN, L.; PEDRYCZ, W. Numerical and linguistic prediction of time series with the use of fuzzy cognitive maps. *Fuzzy Systems, IEEE Transactions on*, v. 16, p. 61 – 72, 03 2008.

STACH, W.; KURGAN, L.; PEDRYCZ, W. A divide and conquer method for learning large fuzzy cognitive maps. *Fuzzy Sets and Systems*, v. 161, p. 2515–2532, 10 2010.

STACH, W. et al. Genetic learning of fuzzy cognitive maps. *Fuzzy sets and systems*, Elsevier Science, v. 153, n. 3, p. 371–401, 2005.

STACH, W.; PEDRYCZ, W.; KURGAN, L. A. Learning of fuzzy cognitive maps using density estimate. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, IEEE, v. 42, n. 3, p. 900–912, 2012.

STYLIOS, C. D.; GROUMPOS, P. P. Modeling complex systems using fuzzy cognitive maps. *Trans. Sys. Man Cyber. Part A*, IEEE Press, v. 34, n. 1, p. 155–162, jan. 2004. ISSN 1083-4427.

SZWED, P. Classification and feature transformation with fuzzy cognitive maps. *Applied Soft Computing*, v. 105, p. 107271, 2021. ISSN 1568-4946.

TABER, R. Knowledge processing with fuzzy cognitive maps. *Expert systems with applications*, Elsevier, v. 2, n. 1, p. 83–87, 1991.

TABER, R.; YAGER, R. R.; HELGASON, C. M. Quantization effects on the equilibrium behavior of combined fuzzy cognitive maps. *Int. J. Intell. Syst.*, v. 22, p. 181–202, 2007.

TAN, C. O.; ÖZESMI, U. A generic shallow lake ecosystem model based on collective expert knowledge. *Hydrobiologia*, v. 563, p. 125–142, 2005.

TEALAB, A. Time series forecasting using artificial neural networks methodologies: A systematic review. *Future Computing and Informatics Journal*, Elsevier, v. 3, n. 2, p. 334–340, 2018.

TSADIRAS, A. K. Comparing the inference capabilities of binary, trivalent and sigmoid fuzzy cognitive maps. *Inf. Sci.*, Elsevier Science Inc., USA, v. 178, n. 20, p. 3880–3894, out. 2008. ISSN 0020-0255.

TSADIRAS, A. K.; MARGARITIS, K. G. An experimental study of the dynamics of the certainty neuron fuzzy cognitive maps. *Neurocomputing*, Elsevier, v. 24, n. 1-3, p. 95–116, 1999.

TSAIH, R.; HSU, Y.; LAI, C. C. Forecasting sp 500 stock index futures with a hybrid ai system. *Decis. Support Syst.*, Elsevier Science Publishers B. V., NLD, v. 23, n. 2, p. 161–174, jun. 1998. ISSN 0167-9236.

VANHOENSHOVEN, F. et al. Pseudoinverse learning of fuzzy cognitive maps for multivariate time series forecasting. *Applied Soft Computing*, Elsevier, v. 95, p. 106461, 2020.

VANHOENSHOVEN, F. et al. Fuzzy cognitive maps employing arima components for time series forecasting. In: . [S.l.: s.n.], 2018. p. 255–264. ISBN 978-3-319-59420-0.

VARGAS, C. L.; SALMERON, J. Dynamic risks modelling in erp maintenance projects with fcm. *Information Sciences*, v. 256, 06 2012.

VLIET, M. van; KOK, K.; VELDKAMP, T. Linking stakeholders and modellers in scenario studies: The use of fuzzy cognitive maps as a communication and learning tool. In: . [S.l.: s.n.], 2010.

WANG, C. et al. Learning large-scale fuzzy cognitive maps using an evolutionary many-task algorithm. *Applied Soft Computing*, v. 108, p. 107441, 04 2021.

WANG, J. et al. Deep fuzzy cognitive maps for interpretable multivariate time series prediction. *IEEE Transactions on Fuzzy Systems*, IEEE, 2020.

WANG, Y. et al. A new adaptive fuzzy cognitive map-based forecasting model for time series. In: *2019 IEEE 14th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*. [S.l.: s.n.], 2019. p. 1112–1118.

WANG, Y. et al. Training novel adaptive fuzzy cognitive map by knowledge-guidance learning mechanism for large-scale time-series forecasting. *IEEE Transactions on Cybernetics*, p. 1–12, 2021.

WANG, Y. et al. The trend-fuzzy-granulation-based adaptive fuzzy cognitive map for long-term time series forecasting. *IEEE Transactions on Fuzzy Systems*, p. 1–1, 2022.

WEI, Z.; LU, L.; YANCHUN, Z. Using fuzzy cognitive time maps for modeling and evaluating trust dynamics in the virtual enterprises. *Expert Systems with Applications*, v. 35, p. 1583–1592, 11 2008.

WOJCIECH, F.; JUSZCZUK, P. Predictive capabilities of adaptive and evolutionary fuzzy cognitive maps - a comparative study. In: _____. [S.l.: s.n.], 2009. v. 252, p. 153–174. ISBN 978-3-642-04169-3.

WU, K.; LIU, J. Robust learning of large-scale fuzzy cognitive maps via the lasso from noisy time series. *Know.-Based Syst.*, Elsevier Science Publishers B. V., NLD, v. 113, n. C, p. 23–38, dez. 2016. ISSN 0950-7051.

WU, K. et al. Time series prediction using sparse autoencoder and high-order fuzzy cognitive maps. *IEEE Transactions on Fuzzy Systems*, IEEE, v. 28, n. 12, p. 3110–3121, 2019.

XIROGIANNIS, G.; GLYKAS, M. Fuzzy cognitive maps in business analysis and performance-driven change. *Engineering Management, IEEE Transactions on*, v. 51, p. 334 – 351, 09 2004.

XIXI, Y.; DING, F.; LUO, C. Time series prediction based on high-order intuitionistic fuzzy cognitive maps with variational mode decomposition. *Soft Computing*, v. 26, p. 1–13, 01 2022.

YAN-CHUN, Z.; WEI, Z. R. An integrated framework for learning fuzzy cognitive map using rcga and nhl algorithm. *2008 4th International Conference on Wireless Communications, Networking and Mobile Computing*, p. 1–5, 2008.

YANG, Z.; LIU, J. Learning of fuzzy cognitive maps using a niching-based multi-modal multi-agent genetic algorithm. *Applied Soft Computing*, v. 74, p. 356–367, 2019. ISSN 1568-4946.

YANG, Z.; LIU, J. Learning fuzzy cognitive maps with convergence using a multi-agent genetic algorithm. *Soft Computing*, v. 24, p. 4055–4066, 2020.

YANG, Z.; LIU, J.; WU, K. Learning of boosting fuzzy cognitive maps using a real-coded genetic algorithm. In: IEEE. *2019 IEEE Congress on Evolutionary Computation (CEC)*. [S.l.], 2019. p. 966–973.

YAO, X.; WANG, Z. An intelligent interconnected network with multiple reservoir computing. *Applied Soft Computing*, v. 78, p. 286–295, 2019. ISSN 1568-4946.

YE, N. et al. Learning fuzzy cognitive maps using decomposed parallel ant colony algorithm and gradient descent. In: IEEE. *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*. [S.l.], 2015. p. 78–83.

YESIL, E. et al. Fuzzy cognitive maps learning using artificial bee colony optimization. *2013 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, p. 1–8, 2013.

YESIL, E.; URBAS, L. Big bang - big crunch learning method for fuzzy cognitive maps. 11 2010.

YU, H.-K. Weighted fuzzy time series models for taiex forecasting. *Physica A: Statistical Mechanics and its Applications*, Elsevier, v. 349, n. 3-4, p. 609–624, 2005.

YU, T. et al. A new fuzzy cognitive maps classifier based on capsule network. *Knowledge-Based Systems*, v. 250, p. 108950, 2022. ISSN 0950-7051.

YUAN, K. et al. Time series forecasting based on kernel mapping and high-order fuzzy cognitive maps. *Knowledge-Based Systems*, Elsevier, v. 206, p. 106359, 2020.

ZAMORA-MARTINEZ, F. et al. On-line learning of indoor temperature forecasting models towards energy efficiency. *Energy and Buildings*, Elsevier, v. 83, p. 162–172, 2014.

ZHANG, H.; SHEN, Z.; MIAO, C. Train fuzzy cognitive maps by gradient residual algorithm. In: IEEE. *2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*. [S.l.], 2011. p. 1815–1821.

ZHANG, S. et al. Cautionary tales on air-quality improvement in beijing. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, The Royal Society Publishing, v. 473, n. 2205, p. 20170457, 2017.

ZHANG, S. et al. Deep fuzzy echo state networks for machinery fault diagnosis. *Trans. Fuz Sys.*, IEEE Press, v. 28, n. 7, p. 1205–1218, jul 2020. ISSN 1063-6706.

ZHANG, W.; ZHANG, X.; SUN, Y. A new fuzzy cognitive map learning algorithm for speech emotion recognition. *Mathematical Problems in Engineering*, v. 2017, p. 1–12, 07 2017.

ZHENG, K. et al. Long-short term echo state network for time series prediction. *IEEE Access*, IEEE, v. 8, p. 91961–91974, 2020.

ZOU, X.; LIU, J. A mutual information-based two-phase memetic algorithm for large-scale fuzzy cognitive map learning. *IEEE Transactions on Fuzzy Systems*, v. 26, n. 4, p. 2120–2134, 2018.

A Long Tables- Complete R-HFCM results for Malaysia, NREL, and GEFCom2012

Table 46 represents the obtained results for the proposed R-HFCM model considering various values of order Ω, map size k and reservoir size N_{SR} in terms of average RMSE, MAPE and U metrics for Malaysia dataset.

Table 47 shows a summary of the obtained results using R-HFCM considering various values of Ω, k, and N_{SR} in terms of average RMSE for NREL and GEFCom2012 datasets.

Table 46 – Complete results of the proposed R-HFCM for different values of Ω, k, and N_{SR} in terms of average RMSE, MAPE, and U metrics (Malaysia data).

Begin of Table											
k	**Ω**	**N_{SR}**	**RMSE**	**MAPE**	**U**	**k**	**Ω**	**N_{SR}**	**RMSE**	**MAPE**	**U**
3	2	2	8057.492	15.383	1.756	4	2	2	9314.501	18.539	2.034
		5	4416.291	7.250	0.963			5	5100.143	9.293	1.115
		10	2721.366	4.467	0.595			10	2955.623	4.829	0.644
		20	2520.703	4.017	0.551			20	2519.803	4.134	0.551
		30	2385.128	3.739	0.521			30	2384.070	3.796	0.520
		40	2323.078	3.580	0.508			40	2350.703	3.702	0.513
		60	2313.620	3.489	0.506			60	2329.441	3.592	0.526
		70	2314.204	3.476	0.506			70	2422.967	3.551	0.530
		80	2365.847	3.482	0.518			80	2422.717	3.561	0.530
3	3	2	9235.783	18.176	2.012	4	3	2	10470.710	21.085	2.286
		5	5897.559	10.426	1.284			5	6634.820	12.414	1.445
		10	3129.842	5.343	0.683			10	3431.517	5.830	0.747
		20	2412.889	3.961	0.527			20	2498.325	3.199	0.546
		30	2333.533	3.804	0.509			30	2315.558	3.718	0.505
		40	2240.754	3.637	0.489			40	2245.132	3.562	0.490
		60	2100.133	3.297	0.46			60	2116.214	3.337	0.462
		70	2078.911	3.168	0.454			70	2076.996	3.266	0.454
		80	2096.302	3.231	0.460			80	2064.286	3.191	0.451
3	4	2	9850.804	19.581	2.147	4	4	2	10872.103	22.463	2.377
		5	5920.061	10.522	1.288			5	7416.954	14.344	1.620
		10	3804.708	6.834	0.830			10	4473.214	7.977	0.972
		20	2494.743	4.187	0.544			20	2687.664	4.471	0.586
		30	2299.614	3.818	0.503			30	2378.232	3.866	0.520
		40	2235.445	3.674	0.488			40	2269.199	3.629	0.496
		60	2124.831	3.4011	0.464			60	2150.956	3.422	0.470
		70	2065.715	3.184	0.452			70	2141.944	3.371	0.469
		80	2123.815	3.273	0.465			80	2092.803	3.269	0.459
		2	10515.150	21.210	2.292			2	11607.423	24.051	2.532
		5	7045.649	13.428	1.535			5	8193.999	15.761	1.782
		10	4314.591	7.782	0.940			10	4410.328	7.996	0.958

Continuation of Table 46

k	Ω	N_{SR}	RMSE	MAPE	U	k	Ω	N_{SR}	RMSE	MAPE	U
		20	2527.866	4.444	0.552			20	2833.951	4.801	0.617
3	5	30	2137.298	3.593	0.467	4	5	30	2343.558	3.950	0.512
		40	2010.003	3.335	0.439			40	2100.606	3.505	0.459
		60	1943.688	3.178	0.425			60	1961.00	3.1405	0.429
		70	1895.067	3.103	0.413			70	1904.884	3.048	0.417
		80	1870.096	3.033	0.409			80	1911.764	3.049	0.417
		2	9849.642	19.661	2.151			2	10916.727	22.136	2.383
		5	6204.021	11.516	1.354			5	7716.306	14.360	1.684
		10	3069.797	5.134	0.670			10	4275.027	7.342	0.935
		20	2540.412	4.084	0.555			20	2634.038	4.308	0.576
5	2	30	2390.658	3.792	0.522	7	2	30	2462.800	3.982	0.538
		40	2368.729	3.723	0.518			40	2444.084	3.898	0.534
		60	2474.040	3.614	0.543			60	2481.736	3.750	0.543
		70	2563.532	3.604	0.561			70	2739.217	3.710	0.600
		80	2447.504	3.540	0.535			80	2815.926	3.760	0.615
		2	10748.072	22.080	2.348			2	11529.570	23.683	2.514
		5	7076.185	13.441	1.542			5	8761.517	16.783	1.909
		10	4193.412	7.542	0.915			10	5793.352	10.465	1.265
		20	2546.370	4.170	0.555			20	2809.531	4.833	0.614
5	3	30	2327.434	3.797	0.507	7	3	30	2393.709	3.972	0.522
		40	2217.739	3.579	0.484			40	2294.180	3.766	0.501
		60	2090.421	3.281	0.456			60	2192.973	3.518	0.480
		70	2074.233	3.212	0.452			70	2125.343	3.375	0.464
		80	2065.736	3.153	0.450			80	2169.000	3.376	0.474
		2	11536.614	23.738	2.521			2	11959.080	24.904	2.609
		5	8838.086	17.710	1.928			5	9392.704	18.249	2.048
		10	5315.146	10.145	1.1622			10	6626.356	12.268	1.446
		20	2742.238	4.690	0.598			20	3445.598	6.147	0.754
5	4	30	2440.293	4.061	0.532	7	4	30	2574.640	4.423	0.562
		40	2271.706	3.745	0.496			40	2340.258	3.919	0.511
		60	2114.182	3.378	0.462			60	2186.663	3.569	0.478
		70	2097.599	3.344	0.457			70	2117.565	3.464	0.463
		80	2047.606	3.215	0.447			80	2160.816	3.497	0.473
		2	11884.637	24.509	2.592			2	12302.511	25.617	2.682
		5	8722.499	17.185	1.904			5	9927.018	19.490	2.165
		10	5876.603	11.464	1.281			10	7192.6139	13.568	1.569
		20	2989.220	5.394	0.653			20	4196.601	7.5486	0.917
5	5	30	2362.917	4.003	0.515	7	5	30	2726.0522	4.699	0.595
		40	2173.468	3.663	0.474			40	2323.451	3.960	0.507
		60	1981.024	3.278	0.432			60	2042.421	3.377	0.446
		70	1900.525	3.092	0.414			70	1969.191	3.253	0.430
		80	1872.116	3.043	0.409			80	1955.056	3.213	0.427
		2	12046.617	25.150	2.631			2	12662.432	26.684	2.764
		5	9333.250	18.306	2.037			5	10630.154	21.306	2.320
		10	6330.156	11.535	1.382			10	8472.634	16.398	1.849

Continuation of Table 46

k	Ω	N_{SR}	RMSE	MAPE	U	k	Ω	N_{SR}	RMSE	MAPE	U
		20	2957.031	4.781	0.648			20	5067.756	9.116	1.105
		30	2806.552	4.321	0.617			30	3236.711	5.493	0.707
10	2	40	3119.889	4.470	0.691	10	4	40	2769.708	4.587	0.608
		60	4534.300	4.928	1.006			60	2400.924	3.887	0.525
		70	11547.010	9.276	2.622			70	2313.466	3.726	0.507
		80	14393.648	10.386	3.235			80	2464.105	3.765	0.544
		2	12343.679	25.831	2.692			2	12953.864	27.442	2.825
		5	10160.819	20.253	2.216			5	11054.099	22.3242	2.411
		10	7638.937	14.400	1.665			10	9005.943	17.557	1.966
		20	3816.230	6.730	0.832			20	5890.285	11.116	1.287
10	3	30	2692.0748	4.233	0.590	10	5	30	3900.382	7.018	0.852
		40	2659.655	4.166	0.585			40	2777.466	4.877	0.605
		60	2521.522	3.854	0.555			60	2364.608	3.811	0.519
		70	3026.792	4.089	0.669			70	2166.487	3.604	0.475
		80	2661.02	3.855	0.588			80	2137.144	3.447	0.467

End of Table

Table 47 – Complete results of the proposed R-HFCM for different values of order, map size and reservoir size in terms of average RMSE (NREL and GEFCom2012 datasets).

Begin of Table

NREL(DHHL_1)				NREL(DHHL_2)				NREL(DHHL_3)			
k	N_{SR}	Ω	RMSE	k	N_{SR}	Ω	RMSE	k	N_{SR}	Ω	RMSE
		3	91.891			3	93.058			3	96.114
3	20	4	90.152	3	20	4	91.315	3	20	4	93.770
		5	90.189			5	90.983			5	93.572
		3	89.671			3	90.764			3	94.206
3	40	4	87.932	3	40	4	89.039	3	40	4	91.369
		5	86.687			5	87.593			5	89.856
		3	92.359			3	93.162			3	99.313
3	60	4	88.029	3	60	4	89.500	3	60	4	90.427
		5	86.421			5	87.206			5	89.634
		3	91.929			3	93.287			3	96.287
4	20	4	90.796	4	20	4	91.613	4	20	4	94.256
		5	91.455			5	92.085			5	94.231
		3	91.183			3	91.893			3	96.247
4	40	4	87.327	4	40	4	89.910	4	40	4	92.3987
		5	86.798			5	88.675			5	90.168
		3	102.594			3	96.277			3	99.341
4	60	4	88.960	4	60	4	90.665	4	60	4	93.137
		5	86.936			5	88.843			5	90.984
		3	92.512			3	94.017			3	96.590
5	20	4	92.833	5	20	4	95.055	5	20	4	96.340

Continuation of Table 47

k	N_{SR}	Ω	RMSE	k	N_{SR}	Ω	RMSE	k	N_{SR}	Ω	RMSE
		5	98.437			5	97.622			5	101.391
		3	90.468			3	91.909			3	94.634
5	40	4	88.849	5	40	4	89.561	5	40	4	91.090
		5	87.823			5	88.770			5	90.582
		3	92.068			3	93.049			3	95.483
5	60	4	87.950	5	60	4	89.355	5	60	4	91.753
		5	86.647			5	86.154			5	89.050
GEFCom2012(zone 1)				**GEFCom2012(zone 2)**				**GEFCom2012(zone 3)**			
		3	950.226			3	6043.718			3	6782.452
3	20	4	779.530	3	20	4	5964.476	3	20	4	6452.415
		5	764.601			5	6071.217			5	6719.522
		3	4347.515			3	8994.901			3	8789.426
3	40	4	1277.452	3	40	4	6210.837	3	40	4	6566.107
		5	834.066			5	6136.744			5	6402.676
		3	19132.751			3	15940.346			3	23764.711
3	60	4	1208.587	3	60	4	7900.542	3	60	4	7520.464
		5	1147.397			5	7181.690			5	6871.729
		3	753.464			3	5711.891			3	6153.115
4	20	4	769.604	4	20	4	5753.411	4	20	4	6423.982
		5	847.805			5	6167.448			5	6736.355
		3	1027.872			3	6037.326			3	6889.979
4	40	4	738.129	4	40	4	5608.940	4	40	4	5984.054
		5	716.364			5	5531.027			5	5837.610
		3	2237.307			3	9965.255			3	10904.258
4	60	4	864.482	4	60	4	6312.521	4	60	4	6903.886
		5	732.174			5	5676.048			5	6061.141
		3	828.133			3	5844.780			3	6281.373
5	20	4	963.518	5	20	4	6592.041	5	20	4	7028.272
		5	1008.525			5	7591.047			5	8239.029
		3	973.594			3	5910.525			3	6720.246
5	40	4	805.513	5	40	4	5734.343	5	40	4	6284.737
		5	816.769			5	5556.940			5	6279.130
		3	2048.510			3	8388.159			3	7968.098
5	60	4	934.968	5	60	4	6526.877	5	60	4	6551.153
		5	839.134			5	5922.283			5	6271.211

End of Table

B Complete statistical results for SONDA, NREL, GEFCom2012, and PJM data

Complete statistical results for evaluating our proposed R-HFCM model against other methods

Table 48 – Wilcoxon *post hoc* test results for SONDA dataset.

Comparison	Statistic	p-value	Result
R-HFCM vs PWFTS	0.0	0.999999	H_0 is accepted
R-HFCM vs HFCM-FTS	55.0	0.999922	H_0 is accepted
R-HFCM vs HOFTS	0.0	0.999999	H_0 is accepted
R-HFCM vs WHOFTS	0.0	0.999999	H_0 is accepted
PWFTS vs R-HFCM	496.0	5.867e-07	H_0 is rejected
PWFTS vs HFCM-FTS	124.0	0.992451	H_0 is accepted
PWFTS vs HOFTS	0.0	0.999999	H_0 is accepted
PWFTS vs WHOFTS	203.0	0.811069	H_0 is accepted
HFCM-FTS vs R-HFCM	441.0	7.774e-05	H_0 is rejected
HFCM-FTS vs PWFTS	372.0	0.007548	H_0 is rejected
HFCM-FTS vs HOFTS	0.0	0.999999	H_0 is accepted
HFCM-FTS vs WHOFTS	361.0	0.013400	H_0 is rejected
HOFTS vs R-HFCM	496.0	5.877e-07	H_0 is rejected
HOFTS vs PWFTS	496.0	5.877e-07	H_0 is rejected
HOFTS vs HFCM-FTS	496.0	5.877e-07	H_0 is rejected
HOFTS vs WHOFTS	496.0	5.877e-07	H_0 is rejected
WHOFTS vs R-HFCM	496.0	5.870e-07	H_0 is rejected
WHOFTS vs PWFTS	293.0	0.188930	H_0 is accepted
WHOFTS vs HFCM-FTS	135.0	0.986599	H_0 is accepted
WHOFTS vs HOFTS	0.0	0.999999	H_0 is accepted

Table 49 – Kruskal-Wallis mean comparison test results for datasets NREL (DHHL_1, DHHL_2, DHHL_3), GEFCom2012 (zone1, zone2, zone3) and PJM (AEP and DEOK)

Dataset	Statistic	p-value	Result
PJM(DEOK)	86.912	1.008e-18	H_0 is rejected
PJM(AEP)	98.453	3.341e-21	H_0 is rejected
GEFCom2012 (zone1)	100.789	1.051e-21	H_0 is rejected
GEFCom2012 (zone2)	106.725	5.558e-23	H_0 is rejected
GEFCom2012 (zone3)	106.972	4.918e-23	H_0 is rejected
NREL(DHHL_1)	77.547	1.030e-16	H_0 is rejected
NREL(DHHL_2)	80.421	2.492e-17	H_0 is rejected
NREL(DHHL_3)	79.768	3.441e-17	H_0 is rejected

Table 50 – Wilcoxon *post hoc* test results for datasets NREL (DHHL_1, DHHL_2, DHHL_3), GEFCom2012 (zone1, zone2, zone3) and PJM (AEP and DEOK)

Dataset	Comparison	Statistic	Adjusted p-vale	Result
DEOK	R-HFCM vs PWFTS	86.0	0.999249	H_0 is accepted
	R-HFCM vs LSTM	30.0	0.999990	H_0 is accepted
	R-HFCM vs ESN	31	0.999989	H_0 is accepted
	PWFTS vs R-HFCM	410.0	0.000750	H_0 is rejected
	R-HFCM vs LSTM	0.0	0.999990	H_0 is accepted
	R-HFCM vs ESN	31	0.999989	H_0 is accepted
	LSTM vs R-HFCM	466.0	9.688e-06	H_0 is rejected
	LSTM vs PWFTS	496.0	5.870e-07	H_0 is rejected
	LSTM vs ESN	158.0	0.961107	H_0 is accepted
	ESN vs R-HFCM	465.0	1.057e-5	H_0 is rejected
	ESN vs PWFTS	496.0	5.870e-07	H_0 is rejected
	ESN vs LSTM	338.0	0.038892	H_0 is rejected
AEP	R-HFCM vs PWFTS	157.0	0.962730	H_0 is accepted
	R-HFCM vs LSTM	0.0	0.999999	H_0 is accepted
	R-HFCM vs ESN	0.0	0.999999	H_0 is accepted
	PWFTS vs R-HFCM	339.0	0.037269	H_0 is rejected
	PWFTS vs LSTM	0.0	0.999999	H_0 is accepted
	PWFTS vs ESN	0.0	0.999999	H_0 is accepted
	LSTM vs R-HFCM	496.0	5.870e-07	H_0 is rejected
	LSTM vs PWFTS	496.0	5.870e-07	H_0 is rejected
	LSTM vs ESN	485.0	1.705e-06	H_0 is rejected
	ESN vs R-HFCM	496.0	5.870e-07	H_0 is rejected
	ESN vs PWFTS	496.0	5.870e-07	H_0 is rejected
	ESN vs LSTM	11.0	0.999998	H_0 is accepted
GEFCom2012 (zone1)	RHFCM vs PWFTS	0.0	0.999999	H_0 is accepted
	R-HFCM vs LSTM	0.0	0.999999	H_0 is accepted
	R-HFCM vs ESN	0.0	0.999999	H_0 is accepted
	PWFTS vs R-HFCM	496.0	5.870e-07	H_0 is rejected
	PWFTS vs LSTM	0.0	0.999999	H_0 is accepted
	PWFTS vs ESN	0.0	0.999999	H_0 is accepted
	LSTM vs R-HFCM	496.0	5.870e-07	H_0 is rejected
	LSTM vs PWFTS	496.0	5.870e-07	H_0 is rejected
	LSTM vs ESN	475.0	4.325e-06	H_0 is rejected
	ESN vs R-HFCM	496.0	5.870e-07	H_0 is rejected
	ESN vs PWFTS	496.0	5.870e-07	H_0 is rejected
	ESN vs LSTM	21.0	0.999995	H_0 is accepted
GEFCom2012 (zone2)	R-HFCM vs PWFTS	1.0	0.999999	H_0 is accepted
	R-HFCM vs LSTM	0.0	0.999999	H_0 is accepted
	R-HFCM vs ESN	0.0	0.999999	H_0 is accepted
	PWFTS vs R-HFCM	495.0	6.480e-07	H_0 is rejected
	PWFTS vs LSTM	0.0	0.999999	H_0 is accepted
	PWFTS vs ESN	0.0	0.999999	H_0 is accepted

Continued on next page

Table 50 – continued from previous page

Dataset	Comparison	Statistic	Adjusted p-value	Result
GEFCom2012 (zone2)	LSTM vs R-HFCM	496.0	5.870e-07	H_0 is rejected
	LSTM vs PWFTS	496.0	5.870e-07	H_0 is rejected
	LSTM vs ESN	496.0	5.870e-07	H_0 is rejected
	ESN vs R-HFCM	496.0	5.870e-07	H_0 is rejected
	ESN vs PWFTS	496.0	5.870e-07	H_0 is rejected
	ESN vs LSTM	0.0	0.999995	H_0 is accepted
GEFCom2012 (zone3)	R-HFCM vs PWFTS	3.0	0.999999	H_0 is accepted
	R-HFCM vs LSTM	0.0	0.999999	H_0 is accepted
	R-HFCM vs ESN	0.0	0.999999	H_0 is accepted
	PWFTS vs R-HFCM	493.0	7.887e-07	H_0 is rejected
	PWFTS vs LSTM	0.0	0.999999	H_0 is accepted
	PWFTS vs ESN	0.0	0.999999	H_0 is accepted
	LSTM vs R-HFCM	496.0	5.870e-07	H_0 is rejected
	LSTM vs PWFTS	496.0	5.870e-07	H_0 is rejected
	LSTM vs ESN	496.0	5.870e-07	H_0 is rejected
	ESN vs R-HFCM	496.0	5.870e-07	H_0 is rejected
	ESN vs PWFTS	496.0	5.870e-07	H_0 is rejected
	ESN vs LSTM	0.0	0.999995	H_0 is accepted
NREL(DHHL_1)	R-HFCM vs PWFTS	1.0	0.999999	H_0 is accepted
	R-HFCM vs LSTM	29.0	0.9999991	H_0 is accepted
	R-HFCM vs ESN	0.0	0.999999	H_0 is accepted
	PWFTS vs R-HFCM	495.0	6.480e-07	H_0 is rejected
	PWFTS vs LSTM	406.0	0.0009799	H_0 is accepted
	PWFTS vs ESN	0.0	0.999999	H_0 is accepted
	LSTM vs R-HFCM	467.0	8.867e-06	H_0 is rejected
	LSTM vs PWFTS	90.0	0.999020	H_0 is accepted
	LSTM vs ESN	0.0	0.999999	H_0 is accepted
	ESN vs R-HFCM	496.0	5.870e-07	H_0 is rejected
	ESN vs PWFTS	496.0	5.870e-07	H_0 is rejected
	ESN vs LSTM	0.0	0.999995	H_0 is accepted
NREL(DHHL_2)	R-HFCM vs PWFTS	0.0	0.999999	H_0 is accepted
	R-HFCM vs LSTM	109.0	0.996774	H_0 is accepted
	R-HFCM vs ESN	0.0	0.999999	H_0 is accepted
	PWFTS vs R-HFCM	496.0	5.870e-07	H_0 is rejected
	PWFTS vs LSTM	492.0	8.697e-07	H_0 is rejected
	PWFTS vs ESN	0.0	0.999999	H_0 is accepted
	LSTM vs R-HFCM	387.0	0.003225	H_0 is rejected
	LSTM vs PWFTS	4.0	0.999999	H_0 is accepted
	LSTM vs ESN	0.0	0.999999	H_0 is accepted
	ESN vs R-HFCM	496.0	5.870e-07	H_0 is rejected
	ESN vs PWFTS	496.0	5.870e-07	H_0 is rejected
	ESN vs LSTM	496.0	5.870e-07	H_0 is rejected
NREL(DHHL_3)	R-HFCM vs PWFTS	0.0	0.999999	H_0 is accepted
	R-HFCM vs LSTM	116.0	0.995155	H_0 is accepted

Continued on next page

Table 50 – continued from previous page

Dataset	Comparison	Statistic	Adjusted p-value	Result
NREL(DHHL_3)	R-HFCM vs ESN	0.0	0.999999	H_0 is accepted
	PWFTS vs R-HFCM	496.0	5.870e-07	H_0 is rejected
	PWFTS vs LSTM	492.0	8.697e-07	H_0 is rejected
	PWFTS vs ESN	0.0	0.999999	H_0 is accepted
	LSTM vs R-HFCM	380.0	0.004844	H_0 is rejected
	LSTM vs PWFTS	0.0	0.999999	H_0 is accepted
	LSTM vs ESN	0.0	0.999999	H_0 is accepted
	ESN vs R-HFCM	496.0	5.870e-07	H_0 is rejected
	ESN vs PWFTS	496.0	5.870e-07	H_0 is rejected
	ESN vs LSTM	496.0	5.870e-07	H_0 is rejected