

Analyzing time series data using graph theoretical and machine learning techniques

**Thesis submitted by
Kakuli Mishra**

DOCTOR OF PHILOSOPHY (Engineering)

**Department of Computer Science and Engineering,
Faculty Council of Engineering & Technology,
Jadavpur University
Kolkata, India
2023**

JADAVPUR UNIVERSITY
KOLKATA-700032, INDIA

INDEX NO. 206/17/E

1. Title of the Thesis: **Analyzing time series data using graph theoretical and machine learning techniques.**
2. Name, Designation & Institution of the Supervisors:
 - (a) Prof. Ujjwal Maulik
Professor
Department of Computer Science and Engineering
Jadavpur University, Kolkata-700032, India
 - (b) Dr. Srinka Basu
Assistant Professor
Department of Engineering and Technological studies
University of Kalyani, Kalyani- 741245, India

List of Publication

Papers in Journals

1. K. Mishra, S. Basu, and U. Maulik, "A dilated convolutional based model for time series forecasting," SN Computer Science, vol. 2, no. 2, pp. 1–11, Feb 2021. [Online]. Available: <https://doi.org/10.1007/s42979-021-00464-4>
2. Mishra, Kakuli, Srinka Basu, and Ujjwal Maulik. "SeqDTW: A Segmentation Based Distance Measure for Time Series Data." Transactions of the Indian National Academy of Engineering 6.3 (2021): 709-730.
3. K. Mishra, S. Basu, and U. Maulik, "Load profile mining using directed weighted graphs with application towards demand response management," Applied Energy, vol. 311, p. 118578, 2022.
4. K. Mishra, S. Basu, and U. Maulik, "Graft: A graph based time series data mining framework," Engineering Applications of Artificial Intelligence, vol. 110, p. 104695, 2022.
5. Mishra, Kakuli, Srinka Basu, and Ujjwal Maulik. "Indexing Demand Response potential at multiple temporal granularity using network theory based analysis", IEEE Transactions on Sustainable Computing (Communicated).
6. Mishra, Kakuli, Srinka Basu, and Ujjwal Maulik. "Time series classification using graph neural networks for demand side management in residential grids", Knowledge and Information Systems, Springer (Communicated).

Papers in Book Chapters

1. S. Basu, K. Mishra, and U. Maulik, "Analyzing load profiles in commercial buildings using smart meter data," *Towards Energy Smart Homes: Algorithms, Technologies, and Applications*, pp. 463–487, 2021.

Papers in Conference Proceedings

1. K. Mishra, S. Basu, and U. Maulik, "Danse: A dilated causal convolutional network based model for load forecasting," in *Pattern Recognition and Machine Intelligence*, B. Deka, P. Maji, S. Mitra, D. K. Bhattacharyya, P. K. Bora, and S. K. Pal, Eds. Cham: Springer International Publishing, 2019, pp. 234–241.
2. K. Mishra, S. Basu, and U. Maulik, "A segmentation based similarity measure for time series data," in *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*. Association for Computing Machinery, 2020, p. 351–352.
3. K. Mishra, S. Basu, and U. Maulik, "Mining representative load profiles in commercial buildings," in *Proceedings of the 7th International Conference on Advances in Energy Research*. Springer Singapore, 2021, pp. 1025–1036.
4. Mishra, Kakuli, Srinka Basu, and Ujjwal Maulik. "Aggregate load forecasting in residential smart grids using deep learning model", 10th International Conference on Pattern Recognition and Machine Intelligence (PReMI2023) (Accepted).

List of Presentations in National/International/Conferences:

1. K. Mishra, S. Basu, and U. Maulik, "Danse: A dilated causal convolutional network based model for load forecasting," in Pattern Recognition and Machine Intelligence, B. Deka, P. Maji, S. Mitra, D. K. Bhattacharyya, P. K. Bora, and S. K. Pal, Eds. Cham: Springer International Publishing, 2019, pp. 234–241.
2. K. Mishra, S. Basu, and U. Maulik, "A segmentation based similarity measure for time series data," in Proceedings of the 7th ACM IKDD CoDS and 25th COMAD. Association for Computing Machinery, 2020, p. 351–352.

PROFORMA-1

Statement of Originality

I... **Kakuli Mishra** registered on **01.09.2017**....do hereby declare that this thesis entitled**"Analyzing time series data using graph theoretical and machine learning techniques"**.....contains literature survey and original research work done by the undersigned candidate as part of Doctoral studies.

All information in this thesis have been obtained and presented in accordance with existing academic rules and ethical conduct. I declare that, as required by these rules and conduct, I have fully cited and referred all materials and results that are not original to this work.

I also declare that I have checked this thesis as per the "Policy on Anti Plagiarism, Jadavpur University, 2019", and the level of similarity as checked by iThenticate software is4...%.

Signature of Candidate: *Kakuli Mishra*

Date: **21.09.23**

Certified by Supervisor(s):
(Signature with date, seal)

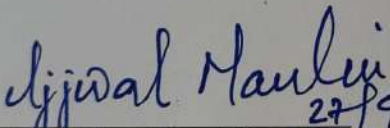
1. *dijwal Maubani*
27/9/23
Professor
Computer Sc. & Engg. Department
Jadavpur University
Kolkata-700032
2. *Sourin Basu*
21/09/23

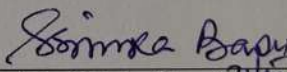
Asst. Professor
Department of
Engineering &
Technological Studies
University of Kalyani

PROFORMA-2

CERTIFICATE FROM THE SUPERVISORS

This is to certify that the thesis entitled **...Analyzing time series data using graph theoretical and machine learning techniques...** submitted by **....Smt Kakuli Mishra....**, who got her name registered on **...01.09.2017...** for the award of Ph.D. (Engg) degree of Jadavpur University, is absolutely based upon her own work under the supervision of **...Prof. Ujjwal Maulik,....Professor, Department of Computer Science and Engineering, Jadavpur University, Kolkata-700032, India, Dr. Srinka Basu, Assistant Professor, Department of Engineering and Technological studies, University of Kalyani, Kalyani-741245, India...** and that neither her thesis nor any part of the thesis has been submitted for any degree/diploma or any other academic award anywhere before.


27/9/23
1. (Prof. Ujjwal Maulik) Professor
Signature of the Supervisor
and date with Office Seal
Computer Sc. & Engg. Department
Jadavpur University
Kolkata-700032


21/05/23
2. (Dr. Srinka Basu)
Signature of the Supervisor
and date with Office Seal



Dedication

To my family

Acknowledgements

This thesis is a reward earned after a couple of hectic years. This had been always the first in my priority. I would express my sincere gratitude for my supervisors- Prof. Ujjwal Maulik and Dr. Srinka Basu. Your scope of research has reached many and has shown several societal benefits. Your process of research followed by the writing, reviewing has been much easier with your support. Your research group has a sheer dedication and a positive working environment and I am privileged to be a part of it. Thank you for giving me the opportunity.

I would like to thank my work mates- Ashmita, Rangan, Sohini, Soma di and all the other undergraduate and postgraduate students who has been a part in my research work. Our conversations, and arguments together in the lab is appreciable. We might have divergent paths after this, but I genuinely cherish the time we have spent together. I would also wish all the best to the undergraduate and the postgraduates for their upcoming professional careers.

To my father and late mother, all your values, ethics you taught me has helped to explore the absolute best in me. This Ph.D journey was eased with my father by my side. The experiences, views that you shared has widened my perspective towards maintaining a balanced work-life balance. Your emphasis on values and a belief in my potential has helped in shaping me as a researcher. To my late brother, though he cannot be here to witness this milestone, I am grateful for the time we had together. The moments like walking to school together, watching movies, sharing study space, to the shared laughter, all are woven to my memory forever. To my in laws, who firmly took up all the needful responsibilities when I was busy throughout these years.

I also owe a profound debt of gratitude to my husband Debajyoti Pathak, whose unwavering support have been a cornerstone of this success. I thank you for all the extra responsibilities that you took off, your understanding during the busy weekends and late nights. As I write this thesis, it reminds of your guidance, the countless discussions on future and our sacrifices together. This thesis is a reflection of your unwavering support and love. Lastly, to my princess Prakriti, who served as a constant reminder of the importance of work-life balance and perseverance. I am happy to create a legacy— a world where education and ambition is nurtured and celebrated.

Date

Kakuli Mishra

Contents

Acknowledgements	viii
List of Figures	xi
List of Tables	xvii
Abstract	xviii
1 Introduction	1
1.1 Introduction	1
1.1.1 Time series	1
1.1.2 TS components	3
1.2 TS analysis (TSA)	6
1.2.1 Challenges of TSA	11
1.3 Advanced computational methods for TS analysis	12
1.3.1 Graph theoretical techniques	12
1.3.2 ML based techniques	13
1.4 Thesis overview and contributions	14
1.4.1 Chapter 2	15
1.4.2 Chapter 3	15
1.4.3 Chapter 4	16
1.4.4 Chapter 5	16
1.4.5 Chapter 6	16
2 Clustering time series using directed graph	17
2.1 Introduction	17
2.2 Literature survey	18
2.3 Proposed Method	20
2.3.1 Definitions and Notations	20
2.3.2 Segmentation and symbolic representation	21
2.3.3 Graph based representation	22
2.3.4 Graph based analysis	23
2.3.5 Example	27
2.3.6 Time complexity analysis	28
2.4 Experiments	31
2.4.1 Data description and data preparation	31
2.4.2 Design of experiments	32
2.4.3 Baseline methods	33
2.4.4 Evaluation Metrics	35
2.5 Results and analysis	37

CONTENTS

2.5.1	Planted patterns	38
2.5.2	Clustering using path level analysis	38
2.5.3	Rare event detection using graph component analysis	47
2.6	Conclusions	50
3	Clustering time series subsequences using directed graph	52
3.1	Introduction	52
3.2	Literature	53
3.3	Methodology	54
3.3.1	Definitions and Notations	54
3.3.2	Segmentation and symbolic representation	56
3.3.3	Graph representation	57
3.3.4	Clustering	58
3.3.5	Working Example	62
3.3.6	Time complexity	62
3.4	Application to DR programs	63
3.4.1	Measuring Stability	63
3.4.2	Atypical pattern identification	64
3.5	Experiments	64
3.5.1	Data description	64
3.5.2	Experimental design	65
3.5.3	Evaluation metrics	66
3.5.4	Comparative methods	68
3.6	Results and Analysis	69
3.6.1	Planted quasi clique	69
3.6.2	Cluster quality	70
3.6.3	Cluster characterization	72
3.6.4	Application to DR programs	78
3.7	Conclusions	86
4	Clustering time series subsequences using multiple directed graph	88
4.1	Introduction	88
4.2	Related work	89
4.3	Method	90
4.3.1	Data preparation	90
4.3.2	Feature extraction and discretization	92
4.3.3	Multiple network and consensus network formation	92
4.3.4	Clustering in multiple network and consensus network . . .	93
4.3.5	Indexing the TS samples	97
4.3.6	Time complexity	99
4.4	Experimental design	99

CONTENTS

4.4.1	Data description	100
4.4.2	Comparative Study	100
4.4.3	Validation measures	100
4.4.4	Parameter tuning	101
4.5	Results and Observations	102
4.5.1	Cluster quality analysis	103
4.5.2	Validating Proposed method	105
4.5.3	Study on stable buildings	106
4.5.4	Study of appliances	108
4.6	Conclusions	110
5	Classification of time series using graph convolutional network	111
5.1	Introduction	111
5.2	Literature	112
5.3	Proposed method	113
5.3.1	Graph structure learning	114
5.3.2	Graph structure formation	116
5.3.3	Graph based classification	117
5.3.4	Toy example	118
5.3.5	Time complexity	119
5.4	Experiments	119
5.4.1	Validation measures	119
5.4.2	Baseline methods	121
5.5	Results and analysis	123
5.5.1	Model Parameters	123
5.5.2	Graph structure results	123
5.5.3	Comparative study	125
5.5.4	Energy consumption during benchmark vs test period	126
5.6	Conclusions	127
6	Conclusions and Future Scope of Research	128
	Bibliography	134

List of Figures

1.1	Subfigure (a) shows TS data constituting the number of births per month in New York city and its components, (b) shows the trend component, (c) shows the seasonality component and, subfigure (d) shows the random component.	5
1.2	Subfigure (a) shows the ACF plot for random component r_t of y_t and subfigure (b) shows the ACF plot for the original time series y_t	5
1.3	Taxonomy of TS analysis techniques in time domain, related to this thesis.	7
2.1	An example of Depth First Search (DFS) method used to find the path from a given source node to target node.	25
2.2	Illustration of the proposed TSA framework for TS clustering using a directed graph. The steps are: Segmentation, Feature Extraction, Symbolic representation, Graph Representation and Clustering.	28
2.3	Subfigures show seasonality in case of (a) London dataset (b) Ausgrid dataset (c) Stock market dataset and (d) Web traffic dataset.	31
2.4	Subfigures (a) and (b) show two different manually implanted sinusoidal patterns (highlighted in <i>red</i>) to 20 randomly chosen TS samples from London dataset. Subfigure (c) shows a table that report the support value and the temporally dependent patterns obtained by clustering and temporal pattern extraction using the proposed graph based framework when Pattern type 1 and Pattern type 2 is implanted. The symbols for the manually implanted patterns are highlighted in <i>red</i>	37
2.5	RI and ARI values obtained for the change in subsequence length in case of Deg-WG, Deg-UWG, Ei-WG and Ei-UWG graph representation on the proposed framework.	40
2.6	Subfigure (a) shows temporal subsequence patterns obtained in case of Ei-WG graph representation for London dataset and subfigure (b) shows the respective graph nodes for London dataset.	41
2.7	Temporal pattern and the respective symbols discovered from each cluster, shown in sequential order, in case of Ei-WG graph. For each dataset, the temporal patterns are illustrated using different colors in each window and is also marked by dotted lines. <i>SL</i> in the table is the subsequence length.	42
2.8	Subfigure (a) shows temporal subsequence patterns obtained in case of Ei-WG graph representation for Ausgrid dataset and subfigure (b) shows the respective graph nodes for Ausgrid dataset.	44

LIST OF FIGURES

2.9	Subfigure (a) shows temporal subsequence patterns obtained in case of Ei-WG graph representation for Stock market dataset and subfigure (b) shows the respective graph nodes for Stock market dataset.	45
2.10	Subfigure (a) shows temporal subsequence patterns obtained in case of Ei-WG graph representation for Web traffic dataset and subfigure (b) shows the respective graph nodes for Web traffic dataset. . . .	46
2.11	Subfigure (a) shows temporal dependent rare events obtained in case of Ei-WG graph representation for London dataset and subfigure (b) shows the respective graph nodes for London dataset.	47
2.12	Subfigure (a) shows temporal dependent rare events obtained in case of Ei-WG graph representation for Ausgrid dataset and (b) shows the respective graph nodes for Ausgrid dataset.	48
2.13	Subfigure (a) shows temporal dependent rare events obtained in case of Ei-WG graph representation for Stock market dataset and subfigure (b) shows the respective graph nodes for Stock market dataset.	49
2.14	Subfigure (a) shows temporal dependent rare events obtained in case of Ei-WG graph representation for Web traffic dataset and subfigure (b) shows the respective graph nodes for Web traffic dataset.	50
3.1	A flowchart of the proposed graph based subsequence clustering method with application to Demand Response (DR) program for identification of stable consumers and identification of atypical patterns.	55
3.2	(a) A working example of the proposed TS subsequence clustering, shown for time series dataset T , with three time series t_1 and t_2 and t_3 , each with $n = 16$ and $l = 4$. The PAA is shown for the respective time series. The points in the lines denote the start and end of subsequences. x-axis is the time units and y-axis is the power consumption values.(b) The features obtained from each subsequence in T . α_{opt} is the alphabet size obtained, shown in <i>bold</i> . (c) The B^4 represents the breakpoints for $\alpha_{opt} = 4$ (d) Symbolic representation of the features (\tilde{T}). (e) The graph representation obtained from \tilde{T} (f) The cluster labels obtained using the proposed quasi clique clustering approach, for $\psi_{min} = 0.3$. (g) Stability of t_1 t_2 and t_3 . The cluster labels of the subsequences are illustrated with different colors. . . .	61
3.3	Cluster quality measures, DB index and the ED for London and Ausgrid datasets achieved using the proposed clustering method. .	70

LIST OF FIGURES

- 3.4 Graph structure obtained from London and Ausgrid dataset for (a) WL 12 hours and $\beta = 0.5$ and (b) WL 4 hours and $\beta = 0.65$. Clustered regions highlight the vertices and edges using different colors. . . . 71
- 3.5 Subfigure (a) shows the clusters, grouped into shapes A through shape G for the London dataset with a cluster centroid (in black). x-axis shows the time units of the subsequence (in hours) and y-axis is the energy consumed (in kWh). 72
- 3.5 Subfigure (b) shows the month wise occurrence of cluster labels in London dataset using heatmaps where x-axis shows the time, y-axis shows cluster labels. Total number of subsequences corresponding to a given time and belonging to a particular cluster is denoted using different colors in log scale. 73
- 3.6 Subfigures show the consecutive occurrence of patterns that fall into the same cluster for (a) two subsequent time segments and (b) three subsequent time segments in London dataset. The red dashed lines denote the time segment boundaries. The cluster numbers are shown in the panel of each plot. 75
- 3.7 Subfigure (a) shows the respective cluster samples in *skyblue*, grouped into shapes A through shape E for Ausgrid dataset and the cluster centroid is shown in *black* lines. x-axis shows time units of subsequence (in hours) and y-axis is the energy consumed (in kWh). 76
- 3.7 Subfigure (b) shows the month wise occurrences of cluster labels of Ausgrid dataset using heatmap where x-axis shows the time, y-axis is cluster label. The total number of subsequences corresponding to a given time and belonging to a particular cluster is denoted using different colors in log scale. 77
- 3.8 (a) Consecutive occurrence of patterns that fall into the same cluster for a) two subsequent time segments and b) three subsequent time segments in Ausgrid dataset. The red dashed lines denote the time segment boundaries. The cluster numbers are shown in the panel of each plot. 78

3.9	Subfigures (a) and (c) show the entropy values (along Y-axis) of the buildings against the number of clusters per building (along X-axis) for (a) London and (c) Ausgrid datasets. The density plots for entropy values and number of clusters per building are shown along the right and top axis respectively. The thick dotted lines show the mean entropy and mean number of clusters that break the plots into four regions: A) low entropy and low number of clusters per building, B) high entropy and low number of clusters per building, C) high entropy and high number of clusters per building, D) low entropy and high number of clusters per building. In subfigure (a) the buildings #137, #181 and #198 are highlighted for their typical positions while in subfigure (c) the buildings #14, #95, #145, are highlighted for their typical positions. The load data in subfigure (b) show the pattern for buildings #137, #181 and #198 of London grid dataset and in subfigure (d) show the pattern for buildings #14, #95, #145 of Ausgrid dataset.	79
3.10	Atypical patterns from (a) London and (b) Ausgrid datasets. Each subsequence is bounded by dotted lines. The <i>black</i> represent normal subsequence and the <i>red</i> represent atypical pattern. x-axis shows date-month-time of occurrence and y-axis shows the load consumption values (in kWh). The building number and the symbol corresponding to the atypical pattern is shown at the top.	82
3.11	Subfigure (a) shows the average load consumption of the subsequences (WL=1 hour), of top-25 buildings during the benchmark and the test period in case of CER-IRISH dataset. Subfigure (b) shows percent change during test period to that of the benchmark period in case of the top-25 buildings in case of CER-IRISH dataset.	85
4.1	A flow chart of the proposed TS subsequence clustering using multiple directed graphs. Step A, is data cleaning, segmentation and normalization. Step B is the feature extraction from the LSTM-AE model. Step C is the multiple network and consensus network formation followed by the clustering of networks based on quasi-clique partitioning given in step D. Residential building indexing using <i>SampEn</i> and the stable set is obtained in step E.	91
4.2	Normalized representative TS profiles (RTP) obtained from clusters in multiple network (respective seasons) and from the consensus network for (a) IRISH (b) CU-BEMS datasets	104

LIST OF FIGURES

4.3	Subfigure (a) shows mean <i>SampEn</i> and variation of TPS at short term granularity- the dotted horizontal lines denote the average of mean <i>SampEn</i> for the stable and unstable set, the highlighted region in the plot shows the days of the summer season. Subfigure (b) shows mean <i>SampEn</i> at medium term and subfigure (c) shows a boxplot for the TPS distribution in stable and unstable set at long term granularity, all for CER-IRISH data set.	107
4.4	The % change in cluster labels in CER-IRISH data set out of all the observed changes during (a) short term granularity and (b) medium term granularity.	108
4.5	Subfigure (a) shows mean <i>SampEn</i> and variation of TPS for short term granularity analysis. The dotted horizontal lines denote the average of mean <i>SampEn</i> for the stable and unstable set. The inset plot shows the mean <i>SampEn</i> and TPS for the medium term granularity. Subfigure (b) shows barplot of % of days when an appliance is identified as stable in each floor during the short term granularity and for number of stable set appliances in each floor during the long term granularity, all in case of CU-BEMS data set.	109
5.1	An overview of the proposed GCN framework. The framework is made of two levels- Level 1 (GRU-ATN) and Level 2 (GCN). An elaborate representation of Level 1 and Level 2 is shown.	114
5.2	A toy example that shows the working principle of proposed GCN where, (a) shows the input subsequences obtained for a TS sample, the input and output for the GRU-ATN used at Level 1 (b) is the graph structure formed and (c) is the vertex and the edge features to be used as input to GCN.	118
5.3	Predicted versus the actual results of a subsequence for the residential home that shows lowest <i>SMAPE</i> and <i>MAE</i> value in case of the proposed method.	124
5.4	(a) A graph structure obtained from the proposed framework for the residential home with lowest <i>SMAPE</i> and <i>MAE</i> value. The undirected edges show attention coefficient values (b) the subsequence patterns of the respective vertices of the graph shown.	124
5.5	The power consumed by the residential homes predicted using the proposed GCN framework, during the test period- Jul' 2010 to Nov'2010 over that of the benchmark period- Jul' 2009 to Nov'2009.	126
5.6	Load factor of the predicted set of residential homes for DSM operation during the benchmark and the test period. The x-axis denote the residential house number (<i>House number</i>) while the y-axis denote the load factor.	127

List of Tables

2.1	(a) An example to show the working principle of the proposed framework for a TS dataset T , with two TS samples t_1 and t_2 . The segments of t_1 and t_2 are shown (assuming $L = 3$) (b) Feature extraction from TS dataset T . (c) Symbolic representation $\tilde{T}_{K \times q}$. The number of alphabets for symbolic representation is shown in <i>bold</i> . (d) A graph based representation obtained from $\tilde{T}_{K \times q}$	29
2.2	Data description of the datasets used in experiments.	33
2.3	Comparative results of RI and ARI in case of clustering in the proposed framework and the existing baseline clustering methods measured in terms. The best values of RI and ARI are highlighted in bold. The abbreviations used for the algorithms are mentioned in section 2.4.2 and section 2.4.3	39
2.4	The p-values of Wilcoxon signed rank test performed on the RI and ARI for the different clustering techniques. The columns corresponding to clustering on proposed framework is highlighted in bold.	40
2.5	The highest support values obtained from temporal patterns. The best support values are highlighted in bold	41
3.1	Actual and predicted cluster labels of the vertices, in the planted quasi clique to a time series dataset.	69
3.2	Cluster quality results and the p-values obtained from post-hoc Nemenyi test, between pairwise comparisons of state-of-art clustering techniques with the proposed method. The best cluster quality and the paired comparisons which vary significantly are highlighted in bold.	69
3.3	The Table shows accuracy results for different values of k . The top- k signify the accuracy achieved for the buildings with highest k stability scores and the last- k signify the k least stability scores. . .	84
4.1	Hyperparameters of the LSTM-AE model– number of layers, hidden units, number of epochs, batch size and activation functions. The best MSE scores are reported in bold.	102
4.2	Comparative results of cluster quality indices for the proposed and baseline methods. WT is the p-value for Wilcoxon signed rank test. The best values are highlighted in bold.	103

LIST OF TABLES

4.3 Comparative study on accuracy measures (F1- score and AUC) for the proposed and state-of-the-art clustering methods. The *Change* % denote increase ↑ or decrease ↓ in electricity consumption of the stable set selected using the proposed and state-of-the-art methods. The best scores are highlighted in bold face. The % *improvement* in the last row highlight the % improvement in F1-score and AUC when compared with the best performing state-of-the-art clustering. 105

5.1 Accuracy results of the proposed framework with the change in parameters like– Layers, Batch size (BS), number of hidden units (HU) and the optimizer in Level 1 and Level 2 (*Level 1-Level 2*). The best accuracy result is highlighted in bold. 122

5.2 Comparative study of the proposed framework in terms of accuracy and F1-score. The best values obtained are highlighted in bold. . . 125

Abstract

Time series (TS) is an integral part of the modern life that helps to understand the several aspects of the daily activities. For example, the trend in weather time series gives an idea about the climate conditions of a place, the trend in infectious disease time series data gives an idea about rate at which the disease rises and the preventive measures to be taken. Several data collection devices like sensors, data loggers, data streaming platforms are used to accumulate the time series data from multiple domain. TS analysis on the historic data provide a basis to derive the underlying patterns and anomalies. It further helps in evaluation and deciding the performance of the systems, models and identify the areas that require improvements.

Several research questions arise for time series data mining operation and this thesis intends to address the following 1. how to preserve the pattern based information when using the dimensionality reduction or the TS data representation techniques 2. how to identify the frequently occurring temporal patterns from the historic time series datasets at the same or different time granularity 3. how to identify the atypical patterns from the historic time series datasets 4. out of the many time series samples present, how to identify the time series samples that have a regularity in patterns.

This thesis aim to contribute towards the above discussed research questions in case of TS datasets. It is also important to consider the space and time constraints that will be required to mine each of the patterns present in the entire TS dataset. In this regard, the first study of this thesis presents a TS dataset representation technique that can preserve the pattern based information. A weighted directed graph structure is formed from a dimensionally reduced TS dataset. A graph based clustering approach is proposed that group the TS samples based on similar temporally dependent subsequent patterns. The graph path intend to capture the temporal patterns observed in TS dataset. The frequently occurring longest temporal patterns from each cluster is identified. The temporally dependent atypical patterns are extracted from the TS dataset using the simple graph component analysis. The next study of this thesis propose a TS subsequence clustering approach in case of TS dataset. The proposed clustering technique can identify the dense coherent substructures which actually represent the subsequently co-occurring patterns of the time series samples. The cluster labels are further used to compute the regularity of TS samples. Regularity denote how frequently does the cluster labels change in each TS sample. The temporally dependent atypical patterns are extracted using the vertex degree analysis on the graph structure. The third contributory work of this thesis propose a multiple graph structure formation to capture the subsequently co-occurring patterns at multiple time granularity in TS dataset. An automated feature extraction from the subsequences is shown using

Abstract

state-of-the-art model. The cluster labels are obtained using quasi-clique based subsequence clustering approach applied at each granular level. The cluster labels are further used to compute the regularity of TS samples at each granular level. It is shown that which of the subsequent patterns are common at different granular level and which of them are different. The final contributory work of this thesis is to classify the TS samples based on the regularity in patterns. Unlike the previous work where a graph structure is designed, in this work, the graph structure is formed automatically by learning which of the previous subsequences are important to predict the future subsequence. The attention values obtained from the state-of-the-art model define the importance of the subsequences when predicting the future subsequence values, that are used for edge formation. The learnt graph structure is used to train the Graph convolutional network (GCN) based classifier model.

The clustering, classification techniques employed to solve the above research questions is useful in the fields of engineering, finance and utility domain. For example, in case of stock market TS data, the opening and the closing prices of a stock differ, and also there exist several companies with similar rise/fall in patterns in their opening or closing stock prices. Clustering and identification of such companies help in designing accurate recommender systems. Study of the frequently occurring temporal patterns in case of stock market help in identifying the companies which have similar patterns during a certain range of time. Identifying the companies that follow similar rise/falls in pattern at the different time granularity help in identifying the differences or the similarity between the given time granularities and study the change in market.

Considering the smart grids utility, the historic TS dataset consist of electricity power consumption for each residential/commercial building. Considering each building as a consumer, shows different behavioral usage of appliances that cause pattern changes. Clustering such consumers help the electricity utility providers in decision making. Study of the frequently occurring temporal patterns in case of smart grids, it help to identify the appliances, devices that are dependent in nature, or are used subsequently. The DSM is the current state-of-the art decision making system in smart grids that help in maintaining a demand-supply equilibrium. The regularity measure as shown in this thesis is important choosing the appropriate consumers for DSM operations. Identifying the TS samples that have temporal patterns that are different from the majority other consumers is important and that gives idea about theft, both in case of the stock market and smart grids. An automated process, like a classifier, that can automatically find the consumers that are suitable for DSM operations, will reduce the workload of electricity utility providers.

1

Introduction

1.1 Introduction

Time series (TS) is a sequence of values generated from varying sources each one being recorded at a specific time. In case of a discrete TS, the set of time stamp at which observations are collected, form a discrete set. For example when observations are made at intervals of seconds, hours, days, months or years.

1.1.1 Time series

The recent advances in technology help in collection of temporal data from diverse domains like– astronomy [1], medical science [2, 3, 4], finance [5, 6], engineering, utility [7], environmental science [8], and many more.

Based on the frequency of data collection, the TS can be categorized into the following:

- Regular intervals: In this case, the data is collected at equally spaced time intervals. For example, data recorded for count of four wheeler/two wheeler in a traffic signal, hourly stock market index, daily web traffic count, are the regular TS.
- Irregular intervals: In this case, the data is either collected at varying time interval or in bursts [9]. For example, the ATM transaction data collected from a bank customer, the account opening at the banks, are irregular TS.

On the contrary, considering the number of variables observed, the TS can be categorized into the following dichotomy:

- Univariate: TS that consist of a single variable. For example– the daily average temperature of a place collected over a period of time is a univariate TS.

CHAPTER 1. INTRODUCTION

- Multivariate: TS that consist of multiple daily variables that are possibly correlated. For example– the daily average temperature, daily average wind speed and average humidity of a place, collected over a period of time combinedly forms a multivariate TS.

In this thesis, we restrict our focus on a set of possibly interdependent, discrete, regular, univariate TS, referred as the TS dataset. Some of the examples of TS datasets encountered in the fields of engineering, utility, finance, that are used in this thesis are briefly discussed below:

1. CER-IRISH: This dataset consists of half-hourly electricity load consumption of the residential and commercial buildings of Ireland [10]. The Commission for Energy Regulation (CER), an independent body of Ireland (IRISH), collected the load consumption data during the year 2009–2010 [10] with the help of the smart meters. A customer behavior trial is conducted by CER to study the impact of several energy efficiency measures like the time of usage tariffs (ToU) and Demand Side Management (DSM) stimuli that were designed by the statistical advisors. To conduct the trial, the data collection time is divided into two periods– a benchmark period of six months– 1st Jul to 31st Dec 2009 and test period– 1st Jan to 31st Dec 2010 [10]. The energy efficiency measures– ToU and DSM operations are effective during the test period only. At the end of benchmark period, the residential homes are categorized into test group and the control group. During the benchmark period, all the buildings followed their usual load consumption regime without any DSM plans. During the test period, buildings in the test group were asked to try different time of use tariffs and DSM stimuli while the buildings of control group followed their usual consumption regime.
2. London household dataset: The smart meter London household data [11] is collected for a project initiated by United Kingdom (UK) power networks. The households are categorized into– affluent, comfortable community, financially stretched, urban adversity and not private households. The households of affluent category have a lavish lifestyle [11]. The households of comfortable community are either of rural community, old age people with tidy neighborhood, smaller houses [11]. The households categorized as financially stretched belong to students, striving families, semi-skilled workers, poorer pensioners [11]. The households of urban adversity belong to young families living in low cost flats or low cost terraces, poorer families with many children [11]. The households belonging to active/inactive communal population, is categorized into not private households [11]. Each household consist of half-hourly load consumption values during the period of 2011-2014, and the household characteristics information.

3. Ausgrid solar home electricity dataset: The Ausgrid Smart Home was created by Ausgrid and Sydney Water, that aim to trial the next generation of energy efficient technologies and distributed generation. They plan for a grid system where the homes can meet their own electricity requirements without compromising their lifestyle. In this regard, they have installed rooftop solar system into some of the residential houses in Sydney. Each residential house has a postcode associated with their location. The data consists of half-hourly power readings of the solar electricity generated by the rooftop solar system installed in the respective residential houses, starting from the year 2010 – 2013 [12].
4. Stock market data: The National stock exchange India (NSE) provide real time data that includes information on stock prices, trading volume, historical price charts, and more. The dataset used in this thesis consist of daily closing stock price of various companies during March 2013 to November 2017. Each company belong to one of the four different stock indices- NIFTY 500¹, NASDAQ², NYSE² and S&P³.
5. Web traffic: Web traffic data represent the number of visits to a website or a web page. This dataset, originally accumulated to organize a Kaggle competition, consists the daily views of different wikipedia articles during 2015 to 2017 [13]. Depending on the mode of access of the wikipedia article, the dataset has three different sources of traffic- all access, mobile access and desktop access.
6. CU-BEMS: This dataset captures the electricity consumption of a seven storeyed office building in Bangkok Thailand [14] in half-hourly interval for various appliances– light and plug loads, during the year 2018 to 2019. Each floor of the buildings has several zones– Zone 1, Zone 2, Zone 3 and Zone 4, and each zone has the electricity consumption readings of lights and plugs. The uniqueness of the CU-BEMS dataset described in the paper [14] is the breakdown of building-level electricity consumption into each zone and each floor of the building.

1.1.2 TS components

A time series $\{y_t\}_{t=1}^n$ is said to be stationary if the joint probability distribution of observations $y_t, y_{t+1}, \dots, y_{t+q}$ is same as that of the joint probability distribution of observations $y_{t+k}, y_{t+k+1}, \dots, y_{t+k+q}$, where the observations are separated by k^{th} lag [15].

¹<https://www.nseindia.com/>

²<https://www.nseindia.com/>

³<https://www.nseindia.com/>

CHAPTER 1. INTRODUCTION

Irrespective of the domain from which the data is accumulated, the TS has several components that exhibit the overall pattern throughout the TS. The components are discussed below:

- Trend: A continuous increase or decrease in TS pattern which may or may not be linear in nature.
- Seasonality: The repeating cycles observed across different time granularities.
- Random: This is an irregular residual obtained after removing the trend and seasonality from the TS.

Common approaches to decompose the TS into its components is the additive and the multiplicative decomposition [15]. For TS y_t , the trend and seasonality can be obtained using the additive decomposition, shown in Equation 1.1 or the multiplicative decomposition, shown in Equation 1.2.

$$y_t = s_t + t_t + r_t \quad (1.1)$$

$$y_t = s_t \times t_t \times r_t \quad (1.2)$$

where, t_t is the trend component, s_t is the seasonal component and r_t is a random component. For example, consider y_t consist of the number of births per month in New York city, during the year January 1946 to December 1959⁴, as shown in Fig. 1.1, the trend, seasonality and random components are shown in Fig. 1.1 (b), (c) and (d) respectively.

A TS that exhibits the trend and seasonality is a nonstationary TS, because, in that case the mean value change linearly with time. Converting a nonstationary TS to a stationary require removal of the TS components and an approach to this is differencing the TS using $(y_t - y_{t-k})$, where k is the lag.

The similarity between a TS and the lagged version of it can be measured using the autocorrelation coefficient, given in Equation 1.3.

$$\gamma_k = \frac{Cov(y_t, y_{t+k})}{Var(y_t)} \quad (1.3)$$

where $Cov(.,.)$ signify the covariance and $Var(.)$ signify the variance. The collection of γ_k for $k = 0, 1, 2, \dots, n-1$, is called the autocorrelation function (ACF). Therefore the ACF can compute the autocorrelation between y_t and y_{t-k} .

Fig. 1.2 shows the ACFs generated for y_t and its random component r_t . The ACF of Fig. 1.2 (a) decays to 0 and then show a sinusoidal pattern around 0, while

⁴<http://robjhyndman.com/tsdldata/data/nybirths.dat>

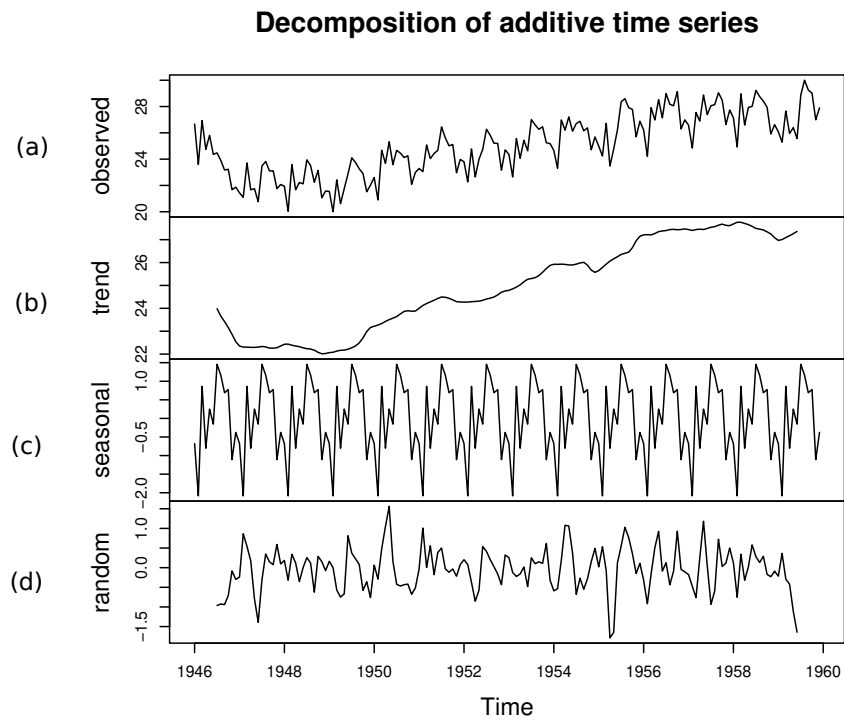


Figure 1.1: Subfigure (a) shows TS data constituting the number of births per month in New York city and its components, (b) shows the trend component, (c) shows the seasonality component and, subfigure (d) shows the random component.

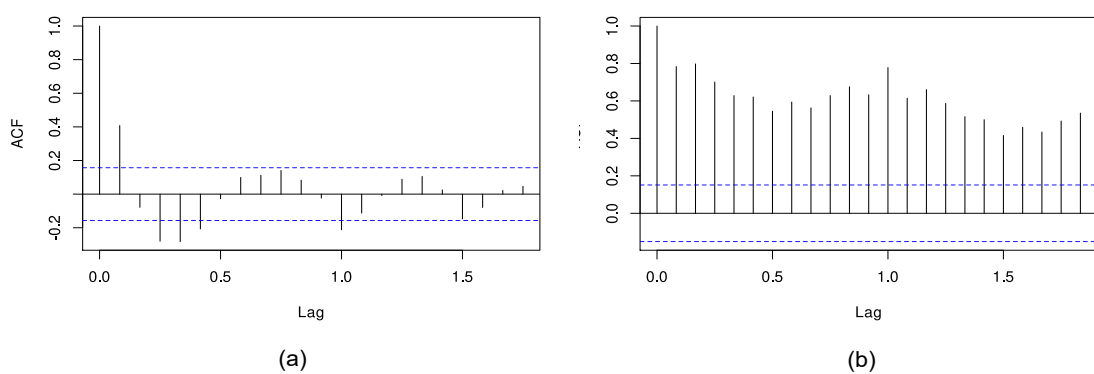


Figure 1.2: Subfigure (a) shows the ACF plot for random component r_t of y_t and subfigure (b) shows the ACF plot for the original time series y_t .

CHAPTER 1. INTRODUCTION

the ACF in Fig. 1.2 (b) does not decay to 0. This nature of pattern in ACF plots, distinguish the stationary TS from the non-stationary [15, 16]. An approach to find the autocorrelation between y_t and y_{t-k} , removing the effect of $y_{t-1}, y_{t-2}, \dots, y_{t-k-1}$, is called the partial autocorrelation function (PACF). Hence, ACF and PACF are the tools to determine the length of seasonality in TS [16].

1.2 TS analysis (TSA)

TSA comprises of methods to study and summarize the time dependence and extract meaningful insights about the underlying data patterns that helps to solve several real life applications. For example, the ecologists study the ecological TS to derive causal dependency amongst the environmental factors due to several factors including population surge and urbanization [17], the epidemiologist study the infectious disease in an area to plan for the corrective measures, social scientist study the national data on employment, unemployment, earnings, and other labor market topic [17].

TSA is an age old problem. John Graunt examined the mortality in infancy and childhood due to rickets, that broke out during the 17th century, and during that time, he first formulated the temporal data and named it as *life tables* [18]. The European merchants and financiers continued the use of quantitative reasoning to study profits and losses in the changing markets during the 17th and 18th century [19]. It was 19th century when the modeling and investigation of the temporal phenomenon began. Yule, Gilbert, Eugen and Aleksandr during the 19th century constructed the basic building blocks for modeling stationary stochastic processes [19]. Correlation and regression analysis was first used by the financial centers to quantify the risks involved in their stock holdings [19]. During the early 20th century, the statisticians used the graph and plots to summarize the results of their statistical analysis, for example the ACF plot [19]. The harmonic analysis on TS is one of the earliest method used during the 20th century to determine the periodicity of annual measurements of the sunspot activity [20]. Later during the 20th century, Box and Jenkins popularized the autoregressive integrated moving average (ARIMA) model by using an iterative modeling procedure [21].

A wide range of models followed ARIMA, such as, autoregressive integrated moving average with exogenous variables (ARIMAX), or seasonal decomposition of time series (STL). Selecting an appropriate model require domain knowledge and study on the data. Majority of the above models assume that the given TS data is stationary in nature and cannot model the trend or seasonality in it [22]. The above discussed model require a complete data on a given time range and data handling techniques have to be used in case of missing values [23]. The statistical techniques do not work well in case if the TS has outliers [23].

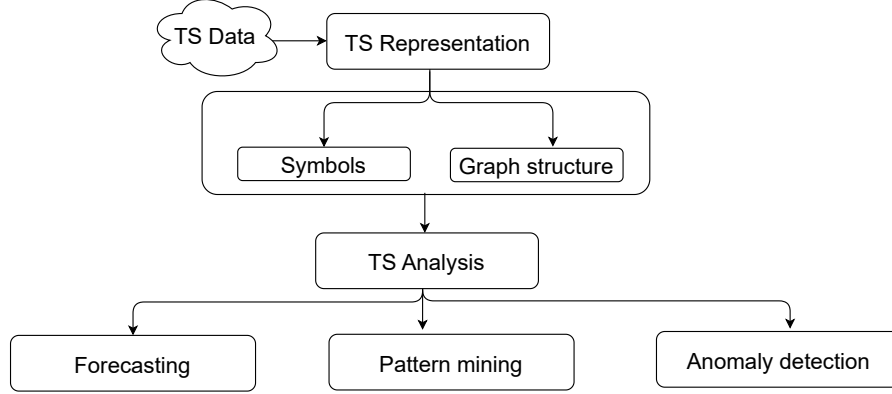


Figure 1.3: Taxonomy of TS analysis techniques in time domain, related to this thesis.

We show a taxonomy of TSA in Fig. 1.3, with respect to the time domain. Advantage of using the time domain representation is that it facilitates analysis of TS components by observing the data points over time. Each of the analysis technique and the related literature is discussed below.

1. TS representation: TSA depend on TS representation that can be broadly categorized as the data-adaptive [24, 25] and non-data adaptive [26] representation. As the name signifies, the data adaptive techniques are data dependent, that is, the parameters for dimensionality reduction are based on the TS data type, while in case of non data adaptive techniques, the parameters remain the same for every TS data type. The hierarchical organization of data adaptive and non data adaptive TS representation technique is available in [26]. The TS representation is categorized into (a) symbolic representation and, (b) graph representation, as shown in the taxonomy of TSA in Fig. 1.3.

Keogh et. al [26] first introduced the concept of Piecewise Aggregate Approximation (PAA) [27] in TS, which was later extended to the Symbolic Aggregate Approximation (SAX) [26]. PAA is a statistical tool in which the TS is initially divided into segments of equal length, each of which is approximated by the mean value. However, statistical measure like mean alone is not sufficient to represent the discriminatory information of TS segments. To overcome this, additional statistical measures like standard deviation, minimum or maximum [28] are used alongwith mean. Segment based approaches suffer from another challenge is the determination of appropriate segment length. Recent researchers proposed the use of automatic segment length detection and the variable length segment based approaches [29, 30].

SAX [26] opens up the scope for the dictionary based algorithms [31, 32] for

CHAPTER 1. INTRODUCTION

TS representation. The dictionary based algorithms forms a histogram of the words or the patterns obtained from SAX. SAX can explicitly represent the TS by a sequence of symbols in which the order of occurrence of the pattern is maintained but the temporal information, the subsequently co-occurring pattern based information is lost.

The graph representation open up the scope to explore the TS patterns based on the temporal information. The visibility graph (VG) [33] represent each time instant as graph vertex and the edge connectivity is decided on the visibility rule [33]. For example there exist an edge connection between the two vertices if one vertex is visible from the other. Therefore in addition to storing the sequential information of TS, graph structure also can store the subsequently co-occurring pattern based information, using the vertices and the edges. The graph learning techniques and the properties of the graph structure will also be applicable for the TSA. A TS of length n require $O(n^2)$ adjacency matrix to represent the VG [33]. A better approach for VG is shown in [34] where each subsequence is represented as a vertex. While the edge connections can be based on the similarity between the vertices [34], several different edge connection techniques exist [35, 36]. Zou et. al [37] review some other graph structure representation from TS that have replaced the VG.

2. Forecasting: Exploiting the auto correlation property of the TS, the future values can be predicted using the historic TS data. In general forecasting problems can be categorized as- short term, medium term and long term. Short term consider daily, weekly and monthly forecasts, medium term consider one to two year of forecasts and forecasts beyond that are the long term forecasts. The autoregressive models (AR) are popular technique used for forecasting the linear TS [20]. Bayesian estimation is used as a parameter estimation technique for the AR models. Another popular method for forecasting is the exponential smoothing that uses the weighted averages of past to predict the future values [38]. The ARIMA model, Holt-Winters' exponential smoothing are a few among many that can capture the trend and seasonal components of TS [39, 23]. The state space models discussed in [40] provide a framework for modeling the linear TS using a state equation that describe how a TS component changes over time and the observation equation of the state space model links the latent state to the observed data. During the 1980s, the researchers realized the need for higher accuracy in forecasting methods based on the non-linear models [41]. The artificial neural networks (ANNs) represent the non-linear relationship of the TS data using the hidden layers each of which consist of a number of hidden units. The final output obtained from the output layer gives the forecasted values. ANNs

are universal approximators that can approximate large class of functions with higher accuracy. However, ANNs are not designed to capture the auto-regressive property of TS. This leads to the recent developments for non-linear TSA models. The deep learning models such as the convolutional models [42], recurrent neural networks [43] and long short-term memory networks [43] are some of the models that have gained prominence.

In general, the challenges of nonlinear models include (a) model selection [39] and (b) hyperparameter selection. The ongoing research on nonlinear models focus on developing automated methods for reducing the number of parameters to be learned for automatic model selection.

3. Pattern mining: The pattern mining techniques aim to discover the frequent patterns, and the subsequently co-occurring patterns from TS [44]. The pattern length obtained from TS can either be of fixed length [45] or variable length [46]. A benchmark approach for frequent pattern mining is proposed in [45] where a matrix profile is obtained from pairwise Euclidean distance between the subsequences and the subsequences with lower distance value than a threshold are considered as frequent pattern. Gao et. al [47] propose a hierarchical technique to discover the variable length frequent patterns that are longer than a predefined threshold length. The frequent pattern discovery is a computationally intensive process which require distance computation between the subsequences, but a data parallel processing framework can reduce the burden [48]. The authors in [48] show the frequent pattern and rare pattern discovery from each TS sample in which the distance matrix for each TS is stored in a shared memory. Considering the frequent pattern discovery with respect to entire TS dataset require pairwise distance computation between the subsequences of all the existing TS samples. This will require a larger distance matrix space and also the distance measures like DTW and Euclidean distances used in [48] cause loss of pattern based information of the subsequences.

The frequently occurring temporal patterns are observed to occur at multiple time steps in the entire TS dataset. Two different patterns are said to be subsequently co-occurring if they occur in subsequent time steps. Mining frequently occurring temporal patterns and the subsequent patterns, can reveal important insights about the sequential dependency, like, the behavioral analysis of the elderly individuals living alone at home so that alarm can be raised in case of emergency situations [49]. In literature, the co-occurring patterns are discovered from heterogenous data sources, for example, measure of average traffic speed in case of a foggy weather, within a certain locality, during a certain time of the day [50]. Li et. al [51] propose a visual

interface to discover the co-occurring patterns from the spatial TS within a specified set of parameter constraints. Another visual interface tool for spatio-temporal data is proposed in [50]. There exists countless possibilities to define the co-occurrence relationships of interest [51] when considering heterogeneous data, so generalizing the method to summarize it on a visual interface is the ongoing area of research. Analysis and discovery of the co-occurring patterns from a single data source with multiple TS data samples, is not abundant in literature.

Considering the graph structure representation for pattern mining, the authors in [52] discover the common subsequences between two different sequences of human motion capture data. The graph structure is obtained from a sample distance matrix of two different sequences. The graph structure henceforth ignores several important pattern information. Therefore, this approach [52] cannot be trivially extended for frequently co-occurring pattern discovery. The authors in [53] propose a graph structure for knowledge discovery in building operational data. The graph nodes in [53] represent the different operational modes of the electrical appliance in a building and the edges indicate the change in modes. A graph representation as shown in [53] however cannot be trivially extended for mining frequent patterns or the co-occurring patterns of the electrical appliance.

4. Anomaly discovery: Anomaly detection on TS, can be categorized as- point anomaly, subsequence anomaly and TS anomaly [54]. Given a TS, the unusual data points of TS are called the point anomaly [55]. A TS subsequence is called anomaly when it is observed to be maximally different from the other TS subsequences [56]. Given a TS dataset, a TS is said to be anomalous whose behavior is significantly different from the other TS samples of the TS dataset [57]. The statistical techniques like entropy [58], standard deviation [59] is used for point anomaly and subsequence anomaly detection in case of a TS. These statistical techniques when extended for subsequence anomaly detection in case of TS dataset may not be able to identify the anomalous behavior based on the pattern dependency.

SAX is used to detect the anomalous subsequences of the TS [56, 60]. The SAX representation on a TS data gives a sequence of symbols and considering a fixed sliding window length, a sequence that do not match with any of the previous sequence is said to be likely an anomalous subsequence. However this method strictly considers no repetition of the symbols.

To detect the anomalous TS, the authors in [61] compute a vector of features and apply principal component analysis (PCA) on it. Bivariate outlier detection is applied to get the anomalous TS [61]. Anomalous TS can also be

obtained using the pairwise distance measures between the TS samples [62], which is rather a computationally expensive method. Series2Graph proposed in [63] is a novel graph structure representation that discover both single and recurrent varying length TS subsequence anomaly in an unsupervised way. It first embeds the subsequences in a vector space and the graph nodes are obtained by analyzing the overlapping trajectories and the edges represent the transition between the nodes [63]. Mean and standard deviation of the subsequences is used for distance computation using the nearest neighbor rule [63]. Only the densest parts of the two dimensional vector space is represented as nodes, which might ignore the other patterns from TS [63].

1.2.1 Challenges of TSA

This thesis consider TSA for TS datasets. The above discussed works on pattern mining and anomaly discovery mostly have shown promising results in case of TS samples and in different TS domain. But the motive is to discover the frequently co-occurring patterns and the anomalous patterns with respect to entire TS dataset. Therefore, the TS representation plays a major role so that the temporal dependency of the patterns across all the TS samples is embedded. As discussed earlier, the symbolic representation embed the sequential pattern information, the graph representation can embed the subsequent patterns and their complex relationships. However, it is challenging to design a TS representation so that the subsequently co-occurring patterns can be identified from the TS dataset. It also is challenging to design a framework that can identify the similarity or the changing behavior of the subsequently co-occurring patterns at multiple time granularity.

Most of the above discussed works on TSA, have the constraints that do not allow extending it for TS datasets. Considering the visual interface methods for pattern mining [50] are specifically designed to visualize the co-occurring patterns from spatio-temporal data, each of which has different features values. The co-occurring pattern is based on user-defined instances [50], which do not serve the purpose when considering a TS dataset obtained from a single data source. The pattern mining in [53] is designed for a public building operational data in which the graph nodes ignore the actual pattern shape. This TS representation approach is not suitable when aiming to discover the subsequently co-occurring patterns. Also, obtaining a single graph for multiple such buildings increases the space complexity.

This thesis aim to discover the TS subsequence anomaly or temporally dependent anomalous pattern, with respect to entire TS dataset. The statistical methods shown in [58, 59] are designed for point anomaly detection and extending it for TS subsequence anomaly cannot discover the subsequently co-occurring pattern

CHAPTER 1. INTRODUCTION

based anomaly. The method in [63] detect the TS subsequence anomaly in every TS sample from the graph structure formed by embedding the subsequences to a vector space. Extending this approach to a TS dataset becomes computationally intensive process, because each TS sample is converted to a graph structure hence it cannot be used to discover subsequently co-occurring pattern which is anomalous.

1.3 Advanced computational methods for TS analysis

Due to the recent success of graph structure in modelling the dependencies in the TS and due to the recent advancement in ML based techniques in wide fields of science and technology, in this thesis, we focus on the graph theory based and ML based techniques for TSA. Below we discuss how these two computational techniques helped in TSA across domains.

1.3.1 Graph theoretical techniques

A graph is an ordered pair $G(V, E)$ where V is the set of vertices and E is the set of edges. While the edge weight, edge direction and betweenness are edge properties, degree, vertex betweenness and Eigen vector centrality are the vertex properties. The graph properties has proved to be informative in analyzing the pairwise interactions in complex systems like predicting the protein-protein interaction [64], traffic prediction [65], smart grids [53]. The application of graph node degree analysis is shown in case shield tunneling construction management which is generally required during the metro construction [66]. The VG is obtained from of shield tunneling parameters like total thrust, torque, penetration. The degree analysis help the industry practitioners in shield tunneling construction management, such as performance, efficiency, and safety [66]. The VGs limit only to visibility rule for edge formation, henceforth a more concrete graph structure formation is required that can minimize the pattern loss. The application of vertex connectivity is shown in TS animation view count (TS-AVC) dataset. The TS-AVC shows a local relationship, that reflect the variance in view counts among episodes of a single animated series, and a global relationship that reflect the variability in total view counts among different animated series. This variance is captured using distance measures that is represented using the edge weight. But, a single graph node for an entire TS sample as shown in [67], is not desirable to capture the TS patterns.

The graph structure opens up the scope of applying the Graph neural network (GNN) models for TSA [68]. The GNN models can explicitly model the intra and the inter temporal dependencies, that is, the dependencies between the different patterns of a TS sample and dependencies between patterns of different

1.3. ADVANCED COMPUTATIONAL METHODS FOR TS ANALYSIS

TS samples. For example, to summarize the overnight movement of stock prices, a relational Graph Convolutional Network (GCN) approach is proposed in [69]. Each stock corresponds to a graph vertex and the edge connection is determined by the correlation between the stocks filtered by a threshold and the graph structure obtained is fed to the GCN model that predicts the stock movement [69]. Authors [67] use a Temporal convolutional network (TCN) [70] that capture the local relationship and the Graph convolutional network (GCN) that capture the global relationship in case of TS-AVC dataset.

The graph properties help in identifying the dense, sparse or loosely connected subgraph structures. Authors in [71] construct a graph structure from a hand-crafted feature set and use a community detection algorithm to obtain TS clusters with different trend patterns. The graph nodes of a subgraph are connected densely and the nodes of different subgraphs are sparsely or loosely connected [71]. The different densely connected subgraphs show a similar functional behavior. The concept of connectivity of the nodes introduce the idea of clique for community detection from graph. As it is difficult to find cliques in real life applications, relaxation of cliques called the quasi-clique is introduced [72]. The edge connectivity or the edge weight density is relaxed in case of quasi-clique structure [73, 74]. Clique and the relaxed clique structures are abundantly explored in case of social networks [75], biological networks [72], crime activities [72].

1.3.2 ML based techniques

The ML based techniques help to develop the algorithms that enable learning from the previously recorded TS values to infer the future values, or the frequent and anomalous patterns. ML techniques have been broadly categorized as supervised and unsupervised learning. The supervised learning algorithms are used for classification of the labeled TS data while the unsupervised learning algorithms are used for clustering of the unlabeled TS data.

The traditional clustering techniques like K-Means, Hierarchical clustering solely rely on the distance measures used. The pairwise distance measures applied on raw TS data is a time consuming process and it also cannot discover the clusters formed due to different pattern shifts [76]. To capture the pattern information from electricity consumption data, the authors in [77] obtain cumulative load consumption and then apply K-Means and Hierarchical clustering techniques on it. The cumulative TS in [77] is defined as the sum of load consumed over a predefined time window. The association rules help to find the patterns in the form of $A \rightarrow B$, that means B occur after A [78]. Association rule mining in case of smart grids and buildings gives an idea about the consumption behavior of the consumers due to the different appliance usage. The authors in [78] use k-shape clustering to discover the association rules in case of building energy consumption data. But

CHAPTER 1. INTRODUCTION

the above clustering techniques do not consider how to mine the subsequently co-occurring patterns in case of the TS datasets. Considering the anomalous pattern discovery using clustering techniques, Ane et. al [54] review some clustering approaches where the similar subsequences are placed into the same cluster while the subsequences which are farthest from the cluster centroid are the detected anomalous subsequences. In [79] the authors distinguish between the regular patterns and the anomalous patterns in case of underwater sensor dataset, using DBSCAN algorithm.

Classification of TS samples based on the regularity of the patterns involves determining whether a given time series follows a predictable and consistent pattern or if it exhibits anomalous behavior. The application areas of regularity in patterns is shown in [80] to mine the sleep-wake patterns, in [81] for the demand side management in smart grids. The handcrafted features limits the performance accuracy of the classifier, therefore in [82], authors combine multiple feature extraction techniques and classifiers as well. Authors in [83] learn the frequent temporal patterns from each class based on several metrics such as the frequency of patterns in each class, or the average number of their instances. They then obtain a feature matrix to induce the classifier. The authors in [32] propose an ensemble approach for TS classification, where the trend information alongwith the mean of the segments is used for symbolic representation. The regularity or the consistency of the patterns depend on accurately mining a co-occurring sequence of patterns. When considering the TS dataset, such co-occurring patterns should be identified with respect to the multiple TS samples. Therefore employing multiple feature extraction or the ensemble methods is not suitable for identifying the regularity.

1.4 Thesis overview and contributions

The research objective of this thesis is univariate time series analysis in time domain for pattern mining and anomaly detection. The existing pattern mining and anomaly detection techniques are not well defined to capture the subsequently co-occurring patterns, the regularity of the TS samples in case of TS datasets. The symbolic representation of TS dataset to construct a graph structure can indeed perform multiple TSA tasks at multiple time granularity. The research questions addressed in this thesis are:

1. How to represent the large TS datasets so that the pattern based information are retained?
2. How to identify the frequently occurring longest temporal patterns from TS dataset ?
3. How to extract the atypical patterns from TS dataset ?

1.4. THESIS OVERVIEW AND CONTRIBUTIONS

4. How to identify the co-occurring patterns at multiple time granularity ?
5. How to measure the regularity of the patterns in TS dataset ?
6. How to classify the TS dataset based on regularity of TS patterns ?

The contribution of this thesis to address the above research questions are discussed in the following sections.

1.4.1 Clustering time series using directed graph

The existing TS dataset representation techniques based on the segmentation strategy typically suffer from information loss. Considering the graph structure for TS representation, most of the work use the VGs and its variants [84, 85, 86, 34, 87, 88, 89]. These VGs cannot be trivially extended to represent TS dataset because they cannot embed the subsequent co-occurring pattern based information for multiple TS samples.

In Chapter 2 of this thesis, a TS representation technique is shown for TS datasets that can retain the pattern based information. Based on the vertex and edge properties, several different weighted directed graph structure is obtained from the entire TS dataset. The graph path based clustering is applied on the best graph structure. The proposed method can extract the frequently occurring longest temporal patterns from each cluster, using statistical measure. The proposed method can also extract the atypical patterns using a graph component analysis.

1.4.2 Clustering time series subsequences using directed graph

Clustering of subsequences in TS dataset, aim to group the subsequences so that the clusters formed can actually represent the subsequently co-occurring patterns of TS samples. The existing subsequence TS clustering focus on capturing either the shape or the structure of the patterns [90, 91, 92, 93, 94] but do not consider their pattern dependency.

In Chapter 3 of this thesis, a novel quasi clique based TS subsequence clustering is shown. To overcome the limitations of the existing SAX technique, a modified SAX based dimensionality reduction approach is proposed for encoding the TS subsequences. The symbols obtained from SAX are used to obtain a single weighed directed graph structure that can retain the pattern based information. The cluster labels assigned to subsequences are used to measure the regularity of patterns in TS datasets. The vertex degree measure is used to extract the temporally dependent atypical patterns from TS dataset.

CHAPTER 1. INTRODUCTION

1.4.3 Clustering time series subsequences using multiple directed graph

The co-occurring patterns in case of TS datasets can either be same or different at the different time granularities. It is important to mine the patterns at the different time granularity because of several applications, like, the demand side management in case of smart grids. Identifying the abrupt changes in TS components [5, 95], identifying the homogeneous regions based on the TS components [96, 97] are common in literature, but identifying the commonality or the changes in subsequently co-occurring patterns at multiple time granularity for multiple TS samples is a new perspective.

In Chapter 4 of this thesis, a method is proposed to identify the subsequently co-occurring patterns at multiple time granularity. Unlike the handcrafted feature extraction, a state-of-the-art learning technique is used to automatically extract features from TS subsequences. The features are converted into symbols from which multiple weighted directed network and consensus network is obtained. A maximal quasi-clique based network clustering method on directed weighted multiple network and consensus network is shown. A sample entropy based measure is shown to measure the regularity of TS samples at multiple time granularity.

1.4.4 Classification of time series using graph convolutional network

The classification of TS datasets rely on TS representation using handcrafted features or ensemble of features [98, 99, 100, 101]. The literature also has model based [101, 102, 103] and distance based measures [104, 105] for classification. The importance of classifying TS datasets on the basis of regularity help to group the TS samples based on their consistency in the pattern. An application of this approach is shown smart grids and buildings in [106].

The Chapter 5 of this thesis propose a method to classify the TS dataset based on regularity of the TS patterns. An automated graph structure formation is proposed using the state-of-the-art encoder-decoder technique. The vertices represent unique subsequences and the edges are obtained by applying a threshold on the attention values of the encoder-decoder model. The attention values define the co-occurring patterns formed in the TS sample. The edge weight define the similarity between the co-occurring patterns. The graph structure is the input to Graph Convolutional Network, that classify the TS sample whether it regular or not.

1.4.5 Conclusions and future scope

Chapter 6 of this thesis discusses the conclusions the proposed contributory work and the applications with respect to the TS dataset in each chapter. It also discusses the future scope of this thesis.

Clustering time series using directed graph

2.1 Introduction

TS clustering deals with identifying the similar data points based on the pointwise similarity or the similar patterns formed, based on the structural similarity. The pointwise similarity is measured mainly using TS distance measures [107], while the structural similarity can be obtained by study and discovery of the features in TS [108, 60].

The clustering results of pointwise similarity depend on the choice of distance measures and the cluster quality vary with the change in distance measures [108]. Deciding the suitable distance measure that would capture the data characteristics has been an important area of research [108]. It is observed that the distances concerned with the phase and amplitude invariances perform exceptionally well in TS clustering [109]. In case of clustering based on structural similarity, the high dimensional TS is the cause of fluctuating data patterns and high correlation between the features impose a challenge on designing efficient clustering techniques. Therefore, TS data representation is important so that the feature based information can be retained. SAX [26], a time domain TS representation cause loss of information [110, 111]. The SAX approach however diversifies TS analysis for temporal pattern discovery.

In this chapter, we propose a novel Graph based Time series data mining framework (*Graft*). The framework uses equal length segmentation to extract the subsequence features. Features are then converted into symbols using SAX approach. Unique symbols are used for graph structure formation. The graph structure help to represent TS data into a unique structure of nodes and edges that can capture the temporal nature of co-occurring patterns. The whole TS clustering approach aim to cluster the TS data based on frequently appearing longest temporal patterns. A simple yet effective graph based approach for identification of temporal dependent rare events is also shown.

2.2 Literature survey

The whole TS data clustering methods has shown improvement in prediction accuracy [112, 113], but it has been rarely used for analysis of temporal co-occurring patterns or the discovery of rare patterns from a TS dataset. The matrix profile based approach developed in [114] applies all pair similarity search between all the possible TS subsequences and the unique features and behaviors obtained gives distinguishable clusters. Chin et. al [114] uses the Euclidean distance between the subsequences and obtains the distance profile to discover the frequently occurring patterns and the rarely occurring pattern in a TS sample. To extend the method for an entire TS dataset, the matrix profile becomes computationally expensive. Several improvements were done to reduce the run time for frequently occurring pattern discovery [115, 116] but they do not consider TS dataset. However, the matrix profile is considered as one of the foundational techniques for frequent pattern discovery [117, 118]. The temporal co-occurring pattern signify the discovery of dependent patterns across the time and the repetition of such patterns in multiple TS samples is called frequently co-occurring patterns. Discovery of temporally dependent co-occurrence patterns help in accurate forecasting and identify the similar characteristics in the TS samples [5]. The matrix profile [114] can extract only the motifs or discords based on the distance profile, not the temporally dependent co-occurring patterns.

Singh et. al [119] unveil the importance of temporal pattern discovery in case of energy consumption patterns of electrical appliance of residential buildings and also discuss how it is useful to built accurate forecasting models; Matloob et.al [120] propose a sequence mining approach that can generate a sequence of events for each patient and identify the anomalies to detect the insurance claim related frauds in healthcare systems; Park et.al [121] discover a set of frequent rules to differentiate between the normal events and the deviant events for a die-casting manufacturing process dataset.

The pairwise similarity between TS samples, or between the TS subsequences cannot unveil temporal information of the patterns. However, the graph structure can store temporal information of the patterns in the form of nodes and edges. The transformation of TS to complex network domain is shown for the first time in [122] where the authors construct a graph from pseudoperiodic TS. Each unique cycle in the series is represented by a node in the graph and the node link define the correlation measure. Later in [123], Zhang et. al, test an extensive range of network topology statistics on two pseudo periodic TS data in order to distinguish between their dynamic regimes. Several improvements on the NVGs [33] are proposed, for example, the weighted VGs can capture the pairwise similarity between the TS and help in accurate forecasting [124]; Stephen et al. [34] construct a temporal network from the NVGs where each TS subsequence corresponds to a node in the graph

and nodes are connected based on the visibility rule. Small et.al [86] represent each subsequence as a permutation of integers on the basis of increasing order of values in the subsequence. The nodes are linked if the corresponding integer permutations occur in succession. The authors in [86] claim that their proposed representation can capture the non-linear dynamics of TS.

TS to graph representation is shown also in forecasting problems, where complex network is used to select the variables that retains the causal relationship, based on which the prediction is obtained [125]. Several graph properties like number of nodes, number of links, betweenness centrality, are used as features to train the regression model in [126] and forecast the political crisis in a country. For trend clustering in stock market data [5], the authors extract noise, gradient, maximum and minimum values as features to be represented as graph nodes and the communities are detected using a traditional community detection algorithm [5]. A community detection is shown in [127] where the TS samples are represented as graph nodes and the weights between the links are based on similarity measures between the TS. But representing a TS by a single node, causes information loss. Yan et. al [88] study the hourly traffic flow congestion using a complex network for each hour.

The existing works on analysis of TS patterns cannot address the questions like 1. which patterns occur in sequence termed as co-occurring patterns or temporally dependent patterns 2. what are the longest of such subsequently occurring patterns, 3. how to identify frequently appearing longest co-occurring patterns in different groups of TS samples 4. what are the rare patterns from the perspective of frequently appearing co-occurrence pattern analysis.

To address the above mentioned research questions, we first consider a data structure that can model the information about the longest common co-occurring patterns over the TS data base. One of the popular data structure to capture pairwise relation is graph. Mapping each TS sample to a graph can cause redundancy and increase computational burden. Hence a single weighted directed graph structure is formed. The next two research question can be addressed by using a data mining approach that can extract the longest common paths. Though there are plenty of graph clustering algorithms available, there are limited or no work on clustering the graph paths so as to identify the longest common paths. To address this issue, we propose a unique graph path based clustering approach that would group the samples based on the frequently appearing longest temporal pattern occurrences. The final and obvious question that appear is how to identify the rare patterns from the perspective of analysis of longest temporal pattern occurrences. For this we use a simple graph component analysis based method to detect the temporally dependent rare event patterns that occur in the dataset. The objective of the proposed framework is that other graph analysis based method can be applied

CHAPTER 2. CLUSTERING TIME SERIES USING DIRECTED GRAPH

to detect the rare events depending on the application and apriori knowledge.

2.3 Proposed Method

In this section, we discuss the definitions, notations and the proposed graph based TS data mining framework.

2.3.1 Definitions and Notations

Definition 1 *TS dataset*: The TS dataset is represented as a matrix $T = \mathbb{R}^{K \times N}$ with K different TS samples each of length N . Each row vector of the matrix corresponds to TS, $t_j = \{t_j^1, t_j^2, \dots, t_j^N\}$.

Definition 2 *TS subsequences*: TS subsequences are the ordered sequence of values obtained by sliding a fixed length window L across $t_j \in T$. For example, a subsequence at time instant p will be $\psi_{t_j}^p = \{t_j^p \dots t_j^{(p+L-1)}\}$. For two subsequences $\psi_{t_j}^p$ and $\psi_{t_j}^r$, the sliding window follows the rule: $r = p + L - 1$, to obtain the non-overlapping subsequences. This is done to avoid the trivial matches but capture maximum possible patterns from the data.

Definition 3 *Temporally dependent subsequences*: : Two subsequences, $\psi_{t_j}^p = \{t_j^p \dots t_j^q\}$ and $\psi_{t_j}^r = \{t_j^r \dots t_j^s\}$, will be called as temporally dependent subsequence if, $p < r$ and $q < s$, where p and r are the start time and q and s are the end time of the subsequences.

The length of temporally dependent pattern ranges from a minimum value of 1 to maximum ξ .

Definition 4 *Graph*: The directed graph is represented as $G = (v, \epsilon)$ where v corresponds to the set of nodes and ϵ corresponds to the set of edges.

The weighted graph is represented as $G = (v, \epsilon, w)$, where $w : \epsilon \rightarrow R$, is a function which maps the edges to their real values R .

Definition 5 *Path*: The path between the vertices v_i to v_j is represented as $p_{(i,j)} = \{v_\lambda\}_{\lambda=i}^j$. The length of path is the number of edges present on it.

In case of a weighted directed graph and a path $p_{(i,j)}$, where the vertices $\{v_\lambda\}_{\lambda=i}^j$ lies on the path, the weight of the path is defined as the summation of edge weights connecting v_i to v_j , given as $\sum_{z=i}^j \omega_z$

The steps of the proposed graph based TS mining framework is detailed below. The terms pattern and subsequences have been used interchangeably in this chapter.

Algorithm 1 Symbolic representation: T

Input: $F_{\tau \times K \times q}$ Output: $\widetilde{T}_{K \times q}$

- 1: Initialize L
- 2: $q = N/L$, is the number of subsequences
- 3: $\tau = [\mu(\theta), sd(\theta), \max \theta, \min \theta]$
- 4: **for** t_j in $1 : K$ **do**
- 5: **for** s in $1 : q$ **do**
- 6: **for** each z in τ **do**
- 7: $\theta = \{t_j^s \dots t_j^{(s+L-1)}\}$
- 8: $F(\tau, j, s) = \tau[z](\theta)$
- 9: **end for**
- 10: **end for**
- 11: **end for**
- 12: Initialize $\widetilde{T}_{K \times q}$
- 13: Initialize the breakpoint table $B = \{\beta_{al}\}_{al=3}^{20}$
- 14: **for** each z in τ **do**
- 15: $columnVec \leftarrow F(\tau, j, q)[z]_{j=1}^K$
- 16: $med \leftarrow median(columnVec)$
- 17: $a[z] = \arg \min(|med - B|)$
- 18: **end for**
- 19: $a \leftarrow \max(a)$
- 20: Convert values in $F_{\tau \times K \times q}$, to symbols with alphabet size a .
- 21: $\widetilde{T}_{K \times q}$ is the obtained symbolic representation of T .

2.3.2 Segmentation and symbolic representation

Given a TS dataset T , each TS $t_j \in T$ is broken down into non-overlapping subsequences of a fixed length L , defined in Definition 2 and each subsequence is represented by a feature vector. For $T = \mathbb{R}^{K \times N}$, the tensor for features is given as $F_{\tau \times K \times q}$ where K is the number of TS samples, q is the number of subsequences obtained and τ is a feature vector comprising of- mean, standard deviation, time of maximum and minimum as discussed below:

- Mean: To obtain a reduced representation, the average value of the subsequence is considered [26].
- Standard deviation: It is insufficient to characterise the subsequences only on the basis of mean [110]. The standard deviation (SD) is used to determine the deviation of the subsequence values from its mean.
- Time of maximum and minimum: The time of occurrence of the maximum and minimum value in a subsequence captures the time of rise and fall in pattern [111], thus indirectly incorporating the shape of the subsequence.

CHAPTER 2. CLUSTERING TIME SERIES USING DIRECTED GRAPH

The above mentioned features have been used in our proposed method, however, any other attribute set which captures the underlying pattern from the subsequences, can be chosen for the feature representation.

In case of SAX [26], assuming that the normalized TS data follow a Gaussian distribution, it is partitioned into equiprobable regions and unique alphabet is assigned to each equiprobable region. However, the number of alphabets in the Gaussian curve plays a vital role in space complexity. The method for alphabet size estimation of each feature is shown in algorithm 1. The input to algorithm 1 is the TS dataset T . Steps 4 to 10 shows computing the features— mean (μ), standard deviation (sd), time of maximum value (max), time of minimum value respectively (min), for each TS $\{t_j\}_{j=1}^K$ and for each subsequence s . Median, being the middle element, maintain a symmetricity in the values across the time instants. The alphabet for which the partitioning points in Gaussian curve is closest to the median of the feature values, is chosen for the symbolic representation. Steps 13 to 18 is for the symbolic representation of extracted features. The alphabet set size is chosen in the range, $al = \{3, 20\}$. For example, an alphabet size, $al = 3$, signify that alphabets $\{a, b, c\}$ will be used to convert the features into symbols. B is the lookup table where each column stores a sorted list of breakpoints for the alphabet al , that divide the Gaussian distribution into equal sized areas. In step 14 to 16, we compute the median value of each feature. The variable *columnVec* store the feature values from which we compute the median, as shown in steps 15 and 16 respectively, and obtain the alphabet size from al , for which atleast one of its breakpoints is closest to the computed median. The vector $a[z]$ in step 17 stores the alphabet size for each feature that corresponds to minimum value of $|med - B|$. To ensure maximum discretization, the maximum value from vector a is chosen for symbolic representation, shown in step 19. The matrix $\tilde{T}_{K \times q}$, store the symbols for each subsequence of the TS dataset T . In section 2.3.5 we show an example of the symbolic representation in a sample timeseries dataset.

2.3.3 Graph based representation

The symbolic data $\tilde{T}_{K \times q}$, obtained from previous step is converted to a graph structure. The proposed graph based representation is different from the existing because we obtain a single graph for an entire TS dataset unlike the others where the nodes of the graph either represent the segments of a TS or an entire TS. Both the weighted and unweighted graph is used in our work.

Two different subsequences with the same symbol denotes they have similar pattern, therefore the unique symbols from $\tilde{T}_{K \times q}$ form the graph nodes. This approach helps to keep the minimal number of nodes in the graph thereby removing the redundant pattern based information from the entire TS dataset. The directional edges are based on the time of first occurrence of the symbols in the TS.

2.3. PROPOSED METHOD

Subsequent occurrences of same symbol is represented by self loops. Each unique $sym \in \widetilde{T}_{k \times q}$ is a mapping to $v \in G$, given as $f : sym \rightarrow v$, and, $\exists \epsilon_{ij} \in G$, if the subsequences i and j occur subsequently.

2.3.4 Graph based analysis

For $T = \mathbb{R}^{K \times N}$ and its graph based representation G , we analyze G for clustering of TS and rare event detection.

Algorithm 2 Extract path from the graph: *pathDiscovery*(G)

Input: *sou* and *tar* as the source and target nodes of G

Output: G_p

```

1:  $G_p \leftarrow NULL$ , stores all the paths from sou to tar
2: for tar in  $1 : (|V| - 1)$  do
3:   while there exists path from sou to tar do
4:      $G_p[] \leftarrow pathAlgorithm(G, sou, tar)$ 
5:      $G \leftarrow G - G_p[]$ 
6:   end while
7: end for
8: return ( $G_p$ )

```

Clustering using path level analysis

At this level of analysis, we discover unique paths in G and then use them as features to cluster the TS samples in T . Each path of the graph is a sequence of unique symbols, that corresponds to the temporal patterns in TS. Therefore path in the graph act as discriminative features that can identify the homogenous groups of TS. The steps are explained below.

- **Path discovery:** As it is intractable to discover all possible paths from the graph, we choose a source node as the central node of the graph and the nodes other than source is considered as the target. The initial node of the path is the source and the final node is the target. The minimum path length is 2. The maximum path length depends on the length of subsequences.

The steps to extract path from G is outlined in algorithm 2. Input to algorithm is the graph with source and target nodes, represented as *sou* and *tar* respectively. Steps 3 to 6 is repeated until all paths are visited. After each iteration, the path obtained from *pathAlgorithm*(.) is removed from G to discover the other possible paths from source to target. No two paths in G_p should have more than 60% match in symbols.

CHAPTER 2. CLUSTERING TIME SERIES USING DIRECTED GRAPH

Algorithm 3 Path algorithm: $pathAlgorithm(G, sou, tar)$

Input: G, sou, tar

Output: $path$

```
1: while not all nodes marked Visited do
2:    $Visited \leftarrow sou$ 
3:    $Stack \leftarrow sou$ 
4:   for  $i$  adjacent of  $sou$  do
5:     if  $i \neq tar$  then
6:        $Visited[] \leftarrow i$ 
7:       Update  $Stack$  and  $Out$ 
8:        $dfs(G, i, tar)$ 
9:     else
10:       $path \leftarrow Visited - Out$ 
11:    end if
12:  end for
13: end while
14: return ( $path$ )
```

The function $pathAlgorithm(.)$ in step 4 is the algorithm 3 used for path discovery in case of weighted and unweighted graph. In case of weighted directed graph, the paths are obtained using the traditional Dijkstra's shortest path algorithm [128]. Edit distance (ED) is used to compute the edge weights. ED is the minimum number of changes required to convert one symbol to the other. The maximum value of ED is equal to the number of features used and the minimum is 0. Lower ED denotes better match between two subsequent nodes. The weight of the path in case of Dijkstra's algorithm is defined in Definition 5. The Dijkstra's algorithm always computes atleast one path with minimum weight from the source to target nodes.

In case of unweighted directed graph, we modify the traditional depth first search (DFS) algorithm [129] to obtain the paths from the graph. Algorithm 3 for the modified DFS approach is explained with an example shown in Fig. 2.1. Input to the algorithm is G with the source $sou = p$ and the target $tar = t$ nodes. Visited nodes are represented as $Visited$ and Out stores the order in which the nodes are popped out of $Stack$ in the traditional DFS. DFS starts from p and is continued until t is visited shown in step 1. Each time a new node is traversed, is marked into the $Visited$ and $Stack$ list. Once the target t is visited in Step 4 of Fig. 2.1, the algorithm stops however output from the DFS after the last step is $pqrts$ as shown in Fig. 2.1. We subtract the nodes in Out from $Visited$ as the given order, until the target t is obtained. The final output from the algorithm 3 is pt . Reason of choosing DFS over

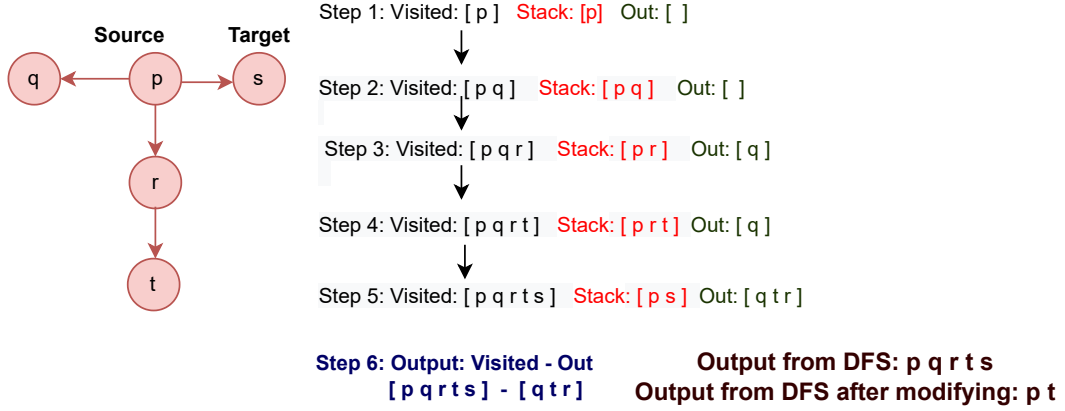


Figure 2.1: An example of Depth First Search (DFS) method used to find the path from a given source node to target node.

BFS is that DFS traverse across the depth which makes the path discovery easier but BFS traverse across the levels, so to reach the target nodes at the deeper levels, each time all the previous level nodes has to be traversed hence the path obtained is not valid. For instance, in case of the graph shown in Fig. 2.1, considering p as the source node and q as target, the BFS path is pq but when the target is t , one of the BFS traversal is $pqrst$ which is not a valid path.

- **Clustering:** We compute the similarity between the extracted paths and the symbolically represented TS data using the longest common subsequence (LCSS) distance. The reason of choosing LCSS is because it can find the matching subsequences between the symbolically represented TS and the paths which in turn helps to identify the temporal patterns in the data.

In the graph G , with source as sou and target as tar , the set of paths obtained between sou and tar is $G_p = \{p_{(sou,tar)}\}_{tar=1}^{v-1}$. The steps for the clustering is outlined in algorithm 4. The input to the algorithm is $\tilde{T}_{K \times q}$ and the set of paths G_p . The matrix $D_{K \times G_p}$, stores the LCSS distance between each path in G_p and each TS $\{t_j\}_{j=1}^K$. Steps 5 and 6 shows the distance matrix computation. To obtain a normalized value the LCSS is divided by q . Motivated by [130], to avoid the use of redundant features in clustering, we employed a pruning strategy. The steps for pruning is shown in step 12 to 15. For k clusters and for each path $p \in G_p$, we compute the rand index (RI) score, assuming the cluster labels obtained from $j - 1$ paths as the true labels. Lesser the change in ri_score signifies a stable clustering, and hence we choose the cluster labels

CHAPTER 2. CLUSTERING TIME SERIES USING DIRECTED GRAPH

Algorithm 4 Clustering: $cluster(\widetilde{T}_{K \times q}, G_p)$

Input: $\widetilde{T}_{K \times q}, G_p$

Output: Cluster labels of T , given as cls

```

1:  $q = N/L$ 
2: Initialize  $D_{K \times G_p}$ 
3: for each  $j$  in  $1 : K$  do
4:   for  $t$  in  $G_p$  do
5:      $LCSS = length(LCSS(t_j, G_p[t]))$ 
6:      $D[j, G_p[t]] \leftarrow \frac{LCSS}{q}$ 
7:   end for
8: end for
9:  $k \leftarrow$  number of clusters
10:  $ri\_score \leftarrow NULL$ 
11:  $cl[1] \leftarrow kMeans(D[1 : K, 1], k)$ 
12: for  $j$  in  $2 : G_p$  do
13:    $cl[j] \leftarrow kMeans(D[1 : K, 1 : j], k)$ 
14:    $ri\_score[j] \leftarrow 1 - RI(cl[j - 1], cl[j])$ 
15: end for
16:  $cls \leftarrow cl[\arg \min(ri\_score)]$ 
17: return ( $cls$ )

```

with minimum ri_score .

To avoid the effect of random choices of cluster centers in K -means algorithm, the K -Means is run over a maximum of 500 iterations and the best results are reported. The number of clusters k is chosen using the elbow rule.

Algorithm 5 Temporal pattern discovery: $TP(c_k)$

Input: c_k

Output: temporal pattern TP

```

1:  $TP[] \leftarrow NULL$ , is a list, initially empty
2: for  $y$  in  $1 : |c_k|$  do
3:    $tp \leftarrow c_k[y]$ 
4:   for  $z$  in  $1 : |c_k|$  do
5:      $tp \leftarrow LCSS(tp, c_k[z])$ 
6:     Append  $tp$  to  $TP$ 
7:   end for
8: end for
9:  $TP \leftarrow TP[\arg \min(|TP|)]$ 
10: return( $TP$ )

```

- **Temporally dependent subsequences:** The temporally dependent subse-

quences are discovered from the labeled T . For each cluster k , we obtain the temporal pattern which occurs in majority of the cluster samples. The minimum length of the discovered temporal pattern is 1. Assuming the cluster set $C = c_1, c_2, \dots, c_k$, algorithm 5 shows the steps for temporal pattern discovery for an arbitrary chosen cluster c_k . For each TS sample in cluster c_k , we obtain the longest common subsequence between each pair of TS and append to the list TP . The minimum length subsequence from TP will be the temporal pattern of c_k , because it occurs in every TS sample of c_k , given in step 9. The algorithm is repeated for each cluster c_k to obtain the temporal patterns of respective clusters.

Rare event detection using graph component analysis

We define the temporally dependent rare events as the sequence of events which rarely occurs in T , or their occurrence is restricted only to some of the TS samples. This study is different from the TS subsequence anomaly discovery techniques in literature where single event without temporal dependency is obtained from the TS. The strongly connected components (SCC) of G is given as $SC(G) = \{g_i\}_{i=1}^c$. We obtain a subset of components, $SC_{sub}(G) \subseteq SC(G)$, such that, $cs < ct$, $\forall g_i \in SC_{sub}(G)$, where cs is the component score and ct is the threshold chosen for graph component analysis. The component score cs is discussed further in section 2.4.4. This denotes that $\forall g_i \in SC_{sub}(G)$ is either connected to $g_j \in SC(G) - SC_{sub}(G)$ or to another $g_j \in SC_{sub}(G)$ or to both. The path formed by the nodes connecting the components in $SC_{sub}(G)$, forms the rare events in the T . Advantage of the proposed rare event detection using the graph component analysis is that we obtain the temporally dependent rare events.

We summarize the steps for the proposed graph based data mining framework in Fig. 2.2. The TS samples of T is shown using different colors. In the segmentation step, the subsequences are obtained from each TS sample and are shown using dotted lines. Tensor $F_{\tau \times K \times q}$, is the input to the symbolic representation technique. We obtain $\tilde{T}_{K \times q}$ as the matrix of symbols shown in Fig. 2.2 and construct the graph G . Due to space constraints, a subset of graph has been shown as an example, which is the input to clustering algorithm. The frequently occurring longest temporal patterns are obtained from the clusters. Temporally dependent rare patterns are obtained from entire TS dataset.

2.3.5 Example

We explain the steps of segmentation, symbolic representation and the graph representation of the proposed framework using an example as shown in Table 2.1. All the notations used in Table 2.1 are detailed in the section 2.3.

CHAPTER 2. CLUSTERING TIME SERIES USING DIRECTED GRAPH

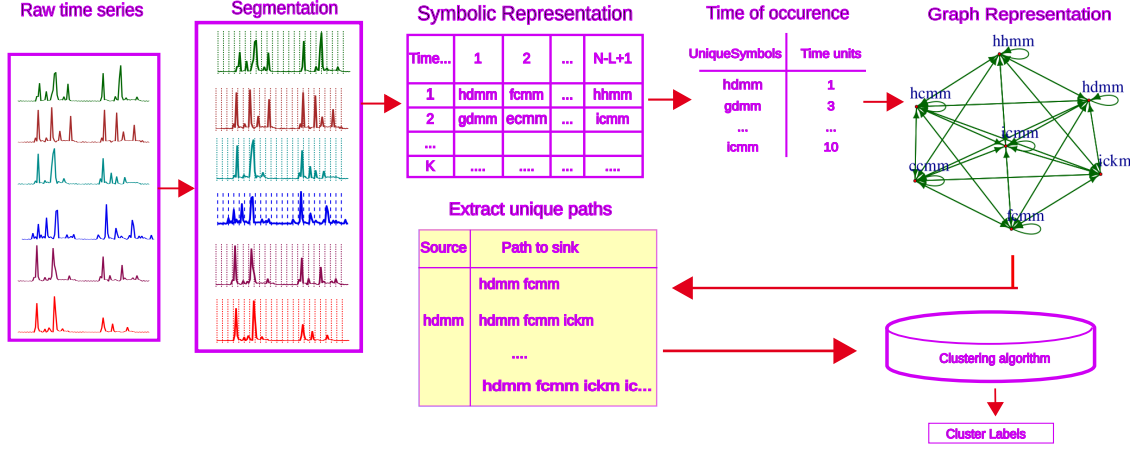


Figure 2.2: Illustration of the proposed TSA framework for TS clustering using a directed graph. The steps are: Segmentation, Feature Extraction, Symbolic representation, Graph Representation and Clustering.

In Table 2.1 (a), we show a sample TS dataset T . For simplicity, we assume two TS t_1 and t_2 in T . For the TS t_j the time instants are given as $\{t_j^1, \dots, t_j^9\}$. We assume the segment length $L = 3$. The subsequences are referred as *segments* in the table 2.1 (a).

In Table 2.1 (b), we show the values of the features extracted from each subsequence of TS t_1 and t_2 , given as μ, sd, max and min . The median value med obtained for each feature are also shown in the feature. To decide the alphabet size, we obtain $|med - B|$ for each feature. a is a vector which stores the alphabet size for which $|med - B|$ is minimum. Table 2.1 (b) returns the maximum alphabet size from the vector of alphabet sizes in a .

The features are converted to symbols using the SAX technique, with the alphabet size a . In Table 2.1 (c), we show the symbolic representation corresponding to the feature values of Table 2.1 (b).

For the graph based representation, we obtain unique symbols from $\tilde{T}_{K \times q}$. Three unique symbols— $eehg, eegh$ and $eehh$ forms the graph as shown in Table 2.1 (d). A self loop in $eegh$ denotes temporal occurrence in t_1 .

2.3.6 Time complexity analysis

Time complexity for the graph based data mining framework is governed by-segmentation, feature extraction, symbolic representation, graph representation, and the clustering and rare event detection. Given the TS dataset $T = \mathbb{R}^{K \times N}$, and, $L \ll N$, $G(v, \epsilon)$ is the graph based representation of TS T , below we discuss the

2.3. PROPOSED METHOD

Table 2.1: (a) An example to show the working principle of the proposed framework for a TS dataset T , with two TS samples t_1 and t_2 . The segments of t_1 and t_2 are shown (assuming $L = 3$) (b) Feature extraction from TS dataset T . (c) Symbolic representation $\tilde{T}_{K \times q}$. The number of alphabets for symbolic representation is shown in *bold*. (d) A graph based representation obtained from $\tilde{T}_{K \times q}$

t_j	t_j^1	t_j^2	t_j^3	t_j^4	t_j^5	t_j^6	t_j^7	t_j^8	t_j^9
t_1	0.30	0.31	0.30	0.28	0.15	0.133	0.131	0.13	0.12
t_2	0.02	0.0	0.02	0.0	0.01	0.008	0.003	0.024	0.0
segment	1			2			3		

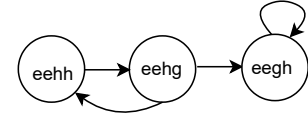
(a)

	segment	μ	sd	max	min
t_1	1	0.30	0.005	2	1
	2	0.19	0.08	3	3
	3	0.12	0.001	3	3
t_2	1	0.01	0.01	3	2
	2	0.0	0.009	2	1
	3	0.0	0.01	2	3
med	-	0.07	0.011	2	2.5
$ med - B $	-	$ 0.07 - B $	$ 0.011 - B $	$ 2 - B $	$ 2.5 - B $
a	-	8	3	8	8
$a \leftarrow \max(a)$	-				8

(b)

	segment	μ	sd	max	min	symbols
t_1	1	e	e	h	g	eehg
	2	e	e	g	h	eegh
	3	e	e	g	h	eegh
t_2	1	e	e	h	h	eehh
	2	e	e	h	g	eehg
	3	e	e	h	h	eehh

(c)



(d)

time complexity of each operation and then show the time complexity for graph analysis at the two different levels.

Segmenting the TS data into equal length subsequences takes a constant time. The number of subsequences in TS dataset T is $K \times (N/L)$. The time complexity to obtain the features from a subsequence is $O(L)$. Because $K \gg N \gg L$, the overall complexity of feature extraction is, $O(K)$.

The symbolic representation using algorithm 1 is based on median value computation. For $q = N/L$, the time complexity to compute the median value is $K \times O(q \log q)$. As $K \gg N \gg L$, the time complexity becomes $O(K)$.

The graph representation require– unique symbols and the edge weight computation. Considering $K \times (N/L)$ as total number of subsequences, the time complexity to obtain unique symbols is $O(K^2)$. Assuming f number of features used to represent the subsequence, the time complexity for edge weight computation

CHAPTER 2. CLUSTERING TIME SERIES USING DIRECTED GRAPH

using edit distance is $O(f^2)$. In the worst case, if all the vertices are connected, the time complexity for edge weight computation is $O(v^2 f^2)$. Considering total number of vertices $v = K \times (N/L)$, it becomes $O(K^2)$. Hence, the total time for graph representation is $O(K^2) + O(K^2) = O(K^2)$.

Below we discuss the time complexity for analyses performed on the graph representation.

- Clustering using path level analysis: Time complexity for path level analysis is decided by the path discovery algorithm, clustering and the temporally dependent subsequence extraction. The path discovery require a source node and we extract the source using two methods. In the first case, when highest out-degree is chosen a source, the time complexity is $O(v^2)$. In the second case, when the highest Eigen-vector centrality is chosen as source, the time complexity is $O(v^3)$. In the worst case, maximum number of vertices in graph is $|v| = (N/L) \times K = K$, when $K \gg N \gg L$. The overall time complexity for source node extraction using the first and the second method is, $O(K^2)$ and $O(K^3)$ respectively.

The time complexity to extract the path from source to target using the DFS based approach is $O(\epsilon + v)$. The maximum number of vertices in the obtained graph in worst case is $|v| = (N/L) \times K = K$, when $K \gg N \gg L$. Considering G as a fully connected graph, maximum number of edges in a graph is $v \times (v - 1)/2$ edges. Therefore time complexity due to modified DFS is $O(K^2 + K) = O(K^2)$. The time complexity of the Dijkstra's algorithm is $O(v + \epsilon \log v) = O((N/L) \times K + K^2 \log((N/L) \times K)) = O(K^2 \log K)$. The worst case combination is, choosing the Eigen-vector centrality as source and Dijkstra's algorithm for path extraction from source to target, $O(K^3 + K^2 \log K) = O(K^3)$. The best case is choosing highest out-degree as source and modified DFS for path extraction, that is $O(K^2 + K^2) = O(K^2)$. The K -means clustering is applied, and the time complexity for K timeseries samples is $O(K^2)$. Therefore the clustering in case of the proposed graph based framework in worst case is $O(K^2)$.

The temporal pattern extraction is, LCSS string matching between the symbolic TS and the paths extracted. In the worst case, the maximum path length is based on maximum number of vertices in the graph, that is $(N/L) \times K$ and the length of symbolically represented TS is N/L , the time complexity due to the LCSS string matching algorithm is $O((N/L) \times K \times N/L) = O(K^2)$ because $K \gg N$.

- Rare event detection using graph component analysis: Time complexity for graph component analysis is decided by the algorithm used for SCC computation, which is $O(v + \epsilon)$. Since the maximum number of edges in

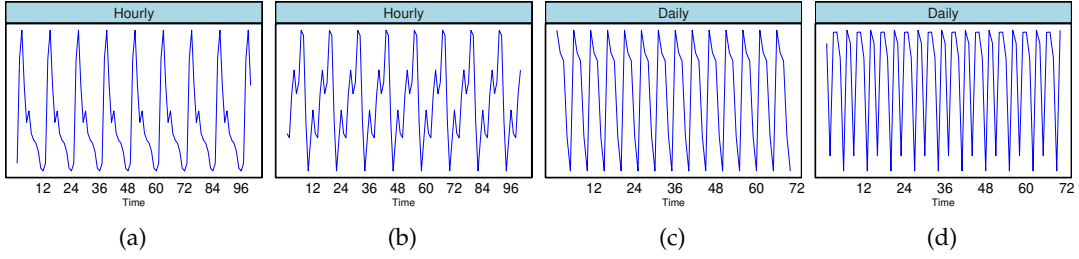


Figure 2.3: Subfigures show seasonality in case of (a) London dataset (b) Ausgrid dataset (c) Stock market dataset and (d) Web traffic dataset.

a complete graph is $v \times (v - 1)/2$ edges, the time complexity for graph component analysis in the worst case is $O(K^2)$.

2.4 Experiments

In this section, we discuss the datasets used for the experiments, design of experiment, the baseline methods for comparison and the evaluation metrics.

2.4.1 Data description and data preparation

The proposed graph based framework is validated on– London household data discussed in section 1.1.1, Ausgrid dataset discussed in section 1.1.1, Stock market discussed in section 1.1.1 and Web traffic dataset discussed in section 1.1.1.

In case of London household, we extract the houses which belong to one of the top three categories- adversity, affluent and comfortable, which are considered as true labels in our experiment. Each category has 70 different houses and each house has six month half hourly power consumption values. Linear interpolation is used to impute the missing values in the data.

In Ausgrid dataset, the postcodes of the houses are taken as the true labels in our experiment. Duplicate values at a time instant are replaced by their averages.

In case of stock market data, the stock index to which the company belongs is considered as the true labels in our experiment. Linear interpolation is used to impute the missing values in the data.

In case of web traffic data, the type of access to the wikipedia article is considered as the true labels in our experiment.

Table 2.2 details the number of TS samples, data size and the subsequence length (SL) for all the datasets used. The above chosen datasets demonstrate typical characteristics of TS including trend, seasonality, repetitive patterns and also contain ground truth information about the cluster labels. The datasets are log normalized before carrying out the experiments.

CHAPTER 2. CLUSTERING TIME SERIES USING DIRECTED GRAPH

2.4.2 Design of experiments

In this section, we discuss the experimental design to analyze the proposed framework and the graph parameters.

1. Planted patterns: The performance of the proposed method is evaluated on the TS samples with manually implanted patterns. The clustering and temporal pattern extraction is performed and the results obtained are reported.
2. Clustering using path level analysis: The proposed graph based framework is experimented on- (a) weighted graph (WG) and (b) unweighted graph (UWG). The source nodes of the graph are chosen using two different methods (a) node with the highest out degree (b) node with highest Eigen vector centrality. Hence we obtain four different combinations (a) Deg-WG: a weighted graph (WG) in which the source node is the one with highest out degree (Deg) (b) Deg-UWG: unweighted graph (UWG) in which the source node is the one with highest out degree (Deg) (c) Ei-WG: a weighted graph (WG) in which the source node is the one with Eigen vector centrality (Ei) and (d) Ei-UWG: unweighted graph (UWG) in which the source node is the one with Eigen vector centrality (Ei).

The objective of path level analysis is (a) clustering and (b) discover the temporally dependent subsequences from clusters. The optimal number of clusters is decided using elbow rule in within cluster sum of squares (WSS) plot. The cluster quality metric used in this experiment is RI and ARI. Also a statistical test is performed to check whether there is significant difference between the clustering on proposed framework and other clustering measures used.

The objective of temporally dependent subsequences, is to characterize the discovered clusters by the temporal patterns that occur in majority of cluster samples. None of the existing state-of-the-art clustering techniques discussed above, can characterize the clusters by temporal patterns, hence no comparison could be shown with respect to the temporal pattern discovery.

The best graph representation out of Deg-WG, Deg-UWG, Ei-WG and Ei-UWG, is based on the study of- RI and ARI indices, statistical test and support values of temporal pattern.

3. Rare event detection using graph component analysis: The objective of the graph component analysis is to extract the rare event or the temporally dependent rare events from the TS dataset. A metric called component score is computed, to identify the rare events.

Table 2.2: Data description of the datasets used in experiments.

Data	#Samples	Size	SL
London	210	8785	{12, 18, 24, ... 66}
Ausgrid	216	8832	{12, 18, 24, ... 66}
Stock Market	207	1023	{5, 10, 15, ... 50}
Web traffic	3000	802	{7, 14, 21, ... 49}

4. Configuration setup: The nodes and the extracted path from graph are the main parameters that govern the clustering results and analyses.

Number of nodes changes with the changing subsequence length (SL). The SL used in the experiment for each dataset is decided based on the periodicity in the dataset. For visualization purpose, in Fig. 2.3, we show the periodicity in the datasets. Reason of choosing periodicity is because the periodic length alongwith the feature vector of subsequences can capture the distinct patterns.

For aperiodic TS, each subsequence will correspond to unique symbols based on the SL chosen. If the minimum SL= 2 for the aperiodic data, there will atleast be one path corresponding to the TS and a connected graph is obtained. In Table 2.2 we report the range of SL used for each dataset. For example, in the London dataset, as the minimum periodicity is observed in 12 hours, the subsequence lengths chosen are {12, 18, 24, ... 66}. To determine the best SL, we analyse the cluster quality with the changing SL. The cluster quality metrics are explained later in the section.

In order to obtain the longer paths from DFS technique, we choose a criteria for each discovered path p , where, $\delta < |p| < q$, $\forall p \in G_p$, where $\delta = 80\% \times q$. However, δ is reduced by 5%, when no path of desired length exists. When no path is obtained after the decrease, we choose the immediate next source node.

For the graph component analysis, the value of threshold, ct is based on component size distribution. For our experiments, ct is set to 0.9. The machine specifications where the proposed framework is implemented are Intel i7 CPU with 3.4GHz clock speed, Ubuntu 18.04 LTS, 64-bit operating system. The experimental results are carried out using R-3.5.0.

2.4.3 Baseline methods

The primary and the first step of mining using the proposed framework is the clustering approach. Hence, it is important that a good quality clusters are

CHAPTER 2. CLUSTERING TIME SERIES USING DIRECTED GRAPH

obtained before further analysis on temporal pattern discovery and the temporal rare event discovery. In this regard, we perform a comparative analysis of the cluster quality with some of the conventional clustering algorithms like– hierarchical clustering using complete linkage (HC-C), hierarchical clustering using single linkage (HC-S), *K*-means, *K*-Shape and the recent state-of-the art techniques like- SAX-Distance (SAX), minimum spanning tree (MST) based clustering, HC on matrix profile based distance (MP). We discuss the baseline methods below,

SAX distance based clustering (SAX) A straightforward way of TS representation is to represent each subsequence using its mean value [26]. SAX distance is used to compute the pairwise similarity. *K*-Means clustering is applied on the computed SAX distance.

Clustering via community detection (Walktrap) Since the proposed method is a graph based representation, we compare it with a graph based community detection technique, called as walktrap. Authors in [127] have discovered the clusters using community detection algorithms applied on ϵ - nearest neighbor. The authors discuss that very small values of ϵ generates many network components and very large value generates fully connected network, we choose the intermediary based on the best rand index. In order to keep the comparisons on same scale, Euclidean distance is chosen for pairwise distance computation. The dataset is min-max normalized before the experiments.

Minimum spanning tree (MST) based clustering : MST is defined as the sub-graph that covers all the vertices of the graph such that the sum of edge weights is minimized. Authors in [131] have used MST based clustering approach, where each node of the graph represents a TS from the TS dataset and the edge weight is the similarity between two different TS. To keep the comparisons at same scale, Euclidean distance has been used as the similarity measure. Since this is a supervised approach, the number of clusters is same as the true labels of the dataset.

Other methods The results have been shown for the standard clustering algorithms like *K*-means using Euclidean distance [132], *K*-shape clustering [133], HC with single and complete linkages [134] and the clustering using matrix profile based distance [118] (MP). Because *K*-means gives different results on different runs, the best out of 500 runs have been reported.

For comparisons on same scale, the dataset is normalized in every clustering algorithm and the number of clusters are scored using the elbow rule. The number of clusters in HC is chosen using the Silhouette score.

Time complexity comparisons: The TS dataset $T = \mathbb{R}^{K \times N}$ with K TS samples each of length N , L is the length of subsequences and $L \ll N$, $q = N/L$ is the number of subsequences, $G(v, \epsilon)$ is the graph based representation of TS T , below we discuss the time complexity of the baseline methods discussed above for comparison with the clustering on proposed graph based framework.

The time complexity for HC using complete and single linkage and K -means is $O(K^2)$. The time complexity for K -shape clustering is $O(K \times N^2)$ [133]. The time complexity of SAX method is the maximum of piecewise aggregate approximation (PAA) of the subsequences, the SAX distance and the K -means clustering. Because $K \gg N \gg L$, time complexity for SAX based clustering is $O(K^2)$.

Walktrap and MST clustering technique depends on the number of vertices and edges of the graph. The time complexity for walktrap algorithm is $O(\epsilon v^2)$. If $|\epsilon| = v \times (v - 1)/2$ and $|v| = N/L \times K$, the time complexity for walktrap algorithm is reduced into $O(K^3)$. In case of MST, the time complexity is $O(|v| + |\epsilon|)$. If $|v| = N/L \times K$, it becomes $O(K^2)$. The time complexity to compute the matrix profile is $O(K^2)$. Because HC is applied on MPDist, the total time complexity is $O(K^2) + O(K^2) = O(K^2)$.

Therefore, the time complexity reported for the clustering from proposed graph based representation is similar to the K -means, MST, SAX and the MP clustering approaches. The runtime of the SAX distance based clustering is 20 seconds, the K -shape clustering is 10 seconds, HC clustering is 5 seconds and the proposed K -means based clustering is 0.08 seconds. The proposed method shows the minimum runtime as compared to the other methods.

2.4.4 Evaluation Metrics

In the clustering using path level analysis, the proposed graph based data mining framework is evaluated using the rand index (RI) and adjusted rand index (ARI) and the temporally dependent subsequences are evaluated using the support values. A statistical test is performed to check if there exists significant difference between the RI and ARI obtained from proposed framework and the other cluster techniques. In the graph component analysis, evaluation is done using the component score. All the evaluation metrics are discussed below.

Rand index (RI) Rand index is a measure of similarity between the predicted and the ground truth labels. It is computed by counting the number of sample pairs that falls to the same or different clusters in the predicted cluster set as that of the ground truth [135]. It is given as:

$$RI = (a + b)/(a + b + c + d) \quad (2.1)$$

CHAPTER 2. CLUSTERING TIME SERIES USING DIRECTED GRAPH

where, a is the number of sample pairs that have same predicted cluster labels as that of ground truth labels, b is the number of sample pairs that have different predicted cluster labels as that of ground truth labels, c is the number of sample pairs that have same predicted cluster labels but different ground truth labels and d is the number of sample pairs that have different predicted cluster labels but same ground truth labels.

Adjusted rand index (ARI) ARI is corrected-for chance version of the RI [136]. Consider two cluster sets $\{A_i\}_{i=1}^k$ and $\{B_j\}_{j=1}^k$ be represented in a contingency table and the values of the table computed as, $n_{ij} = |A_i \cap B_j|$, that is, number of elements common in A_i and B_j clusters. So, the ARI is computed as

$$ARI = \frac{\sum_{ij} \binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2} \right] / \binom{n}{2}}{\frac{1}{2} \left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2} \right] - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2} \right] / \binom{n}{2}} \quad (2.2)$$

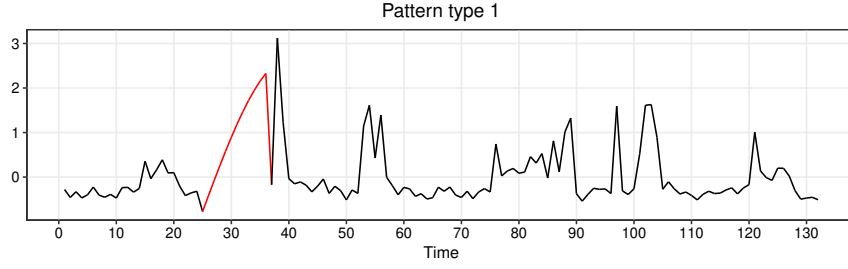
where, a_i and b_j corresponds to the sum in the i^{th} row and j^{th} column of the contingency table respectively. Greater the value of ARI, better is the clustering.

Statistical test Wilcoxon signed rank test [137] is a non-parametric test performed to determine if the pairs of values are statistically different. Because the RI and the ARI values are continuous and non-normally distributed, we choose the Wilcoxon test to obtain the paired differences between each clustering techniques. The test returns p-value that decides whether the paired values are significantly different from each other or not. Considering, $\alpha = 0.05$, according to the null hypothesis, no significant difference exists between the paired values if p-value $> \alpha$, the alternate hypothesis is satisfied otherwise.

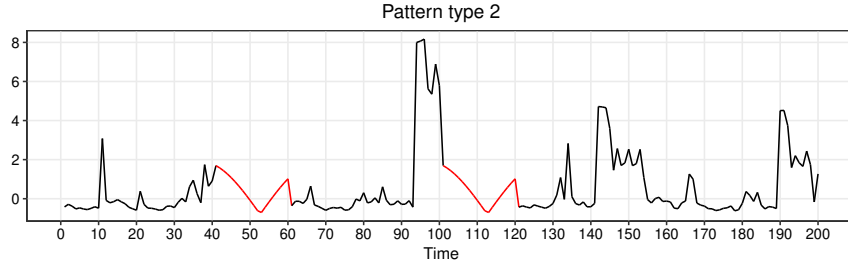
Support value The number of cluster samples that contains the temporal pattern out of the cluster size is the support value. Assuming the cluster set $C = c_1, c_2 \dots c_k$, the support value for cluster c_k is given as $S_k = n_c/n$ where n_c is the number of cluster samples with the detected temporal patterns and $|c_k| = n$. The maximum support value is 1 and the minimum is 0.

Component score The component score (cs) is the ratio of the number of edges in a graph component (ϵ_{g_i}) by the total number of edges in the graph. It is give as $cs = \frac{\epsilon_{g_i}}{\epsilon}$. A lesser component score denotes the occurrence of rare events with respect to that component.

2.5. RESULTS AND ANALYSIS



(a)



(b)

	Clusters	Support	Temporally dependent patterns
Pattern type 1	1	0.8	qrtq
	2	0.6	qrtq → jott
	3	0.91	qrtq
Pattern type 2	1	0.5	gktq → lptt → npqt → gltt → hntt → npqt → iott → gltt → hott → iptt
	2	0.9	npqt → npqt

(c)

Figure 2.4: Subfigures (a) and (b) show two different manually implanted sinusoidal patterns (highlighted in *red*) to 20 randomly chosen TS samples from London dataset. Subfigure (c) shows a table that report the support value and the temporally dependent patterns obtained by clustering and temporal pattern extraction using the proposed graph based framework when Pattern type 1 and Pattern type 2 is implanted. The symbols for the manually implanted patterns are highlighted in *red*.

2.5 Results and analysis

In this section we discuss the results obtained from the different experiments discussed in section 2.4.2.

CHAPTER 2. CLUSTERING TIME SERIES USING DIRECTED GRAPH

2.5.1 Planted patterns

To evaluate the performance of the proposed method, we manually implant sinusoidal patterns to 20 randomly chosen TS from the London dataset. Pattern type 1 shown in Fig. 2.4 (a) is the dataset which has a single pattern implanted at a time and pattern type 2 shown in Fig. 2.4 (b) is the dataset which has periodic pattern, occurring after some time interval. Due to space constraints, we illustrate only one TS samples from each dataset. We perform the experiments on both the datasets and report the support values obtained from temporal pattern in the table in Fig. 2.4 (c). The symbols obtained for pattern type 1 and 2 are *qrtq* and *npqt* respectively.

It is observed that the proposed graph based framework can successfully identify the manually implanted single pattern and the periodic pattern present in the TS samples. The results in the table of Fig. 2.4 (c) are the support values obtained from the temporal pattern symbols in the column *temporally dependent patterns*. We obtain 3 clusters from dataset of pattern type 1 and 2 clusters from dataset of pattern type 2. In case of pattern type 2, the two different temporal patterns are detected. The first temporal pattern obtained from pattern type 2 denotes the occurrence of a loop in the corresponding graph and the second temporal pattern denotes the occurrence of self loop.

2.5.2 Clustering using path level analysis

In this case we evaluate the clustering method and the temporal patterns discovered from the clusters.

Clustering comparisons

To evaluate the performance of the proposed framework for clustering TS database we carryout a comparative study on the datasets with ground truth information. In Table 2.3, we report the best RI and the ARI values obtained by clustering on the proposed graph based framework and the other baseline clustering techniques- HC (complete), HC (single), K-shape, K-means, SAX, Walktrap, MST, MP, discussed in section 2.4. The best results obtained are highlighted in bold. The highest RI and ARI values indicate that the cluster obtained using proposed method matches the ground truth information the best.

The RI and the ARI values obtained from the four graph representations of the proposed framework are significantly greater than that of the clusters obtained by using the other techniques: HC (complete), HC (single), K-shape, K-means, Walktrap, MST and MP. The performance of the SAX method and the proposed method using Ei-WG graph representation are comparable on the London dataset when measured in terms of ARI. On the Ausgrid dataset the ARI score of K-means

2.5. RESULTS AND ANALYSIS

Table 2.3: Comparative results of RI and ARI in case of clustering in the proposed framework and the existing baseline clustering methods measured in terms. The best values of RI and ARI are highlighted in bold. The abbreviations used for the algorithms are mentioned in section 2.4.2 and section 2.4.3

Dataset	London		Ausgrid		Stock market		Web traffic	
	RI	ARI	RI	ARI	RI	ARI	RI	ARI
HC-C	0.42	0.001	0.37	0.01	0.4	0.05	0.34	0
HC-S	0.34	0.001	0.18	0	0.31	0.01	0.34	0
K-shape	0.54	0.002	0.61	0.007	0.61	0.005	0.63	0.046
K-means	0.58	0.001	0.67	0.08	0.43	0.076	0.67	0.25
SAX	0.61	0.04	0.73	0.04	0.59	0.02	0.66	0.1
Walktrap	0.5	0.01	0.51	0.06	0.7	0.2	0.69	0.2
MST	0.34	0	0.19	0.001	0.26	0	0.33	0
MP	0.36	0.003	0.25	0.001	0.29	0.002	0.34	0.001
Deg-WG	0.61	0.007	0.73	0.08	0.64	0.09	0.74	0.26
Deg-UWG	0.63	0.01	0.75	0.07	0.71	0.17	0.7	0.26
Ei-WG	0.62	0.04	0.73	0.07	0.72	0.2	0.71	0.3
Ei-UWG	0.62	0.01	0.73	0.04	0.74	0.18	0.7	0.25

is same as that of the proposed method using Deg-WG graph representation. On the Stock market dataset, Walktrap and the proposed method with Ei-WG graph representation score same ARI. Among the other baseline methods, while SAX and Walktrap show high RI scores for all the data sets, K-means follows them closely. It may noted that the highest RI score ranges between 0.63-0.75 across the datasets, while the highest ARI range is 0.04-0.3 across the datasets. This may be because of the number of clusters.

We plot the change in RI and ARI with changing subsequence length (SL), shown in Fig. 2.5. The plot shows highest values of ARI from Ei-WG graph representation in case of London, Stock market and the Web traffic datasets. This is because the paths formed from Ei-WG is better as compared to other graph based representation. The edges in Ei-WG are weighted, so, according to Dijkstra's algorithm, it forms the path with more similar sequence of nodes. Also the source has high Eigen vector centrality, which signifies, the neighbour nodes of source also has high Eigen vector scores.

Table 2.4 shows the results for Wilcoxon signed rank test. The Deg-WG, Deg-UWG and Ei-UWG show the p-values that are lesser than significance level $\alpha = 0.05$, except the K-means. However, the p-values obtained from Ei-WG graph

CHAPTER 2. CLUSTERING TIME SERIES USING DIRECTED GRAPH

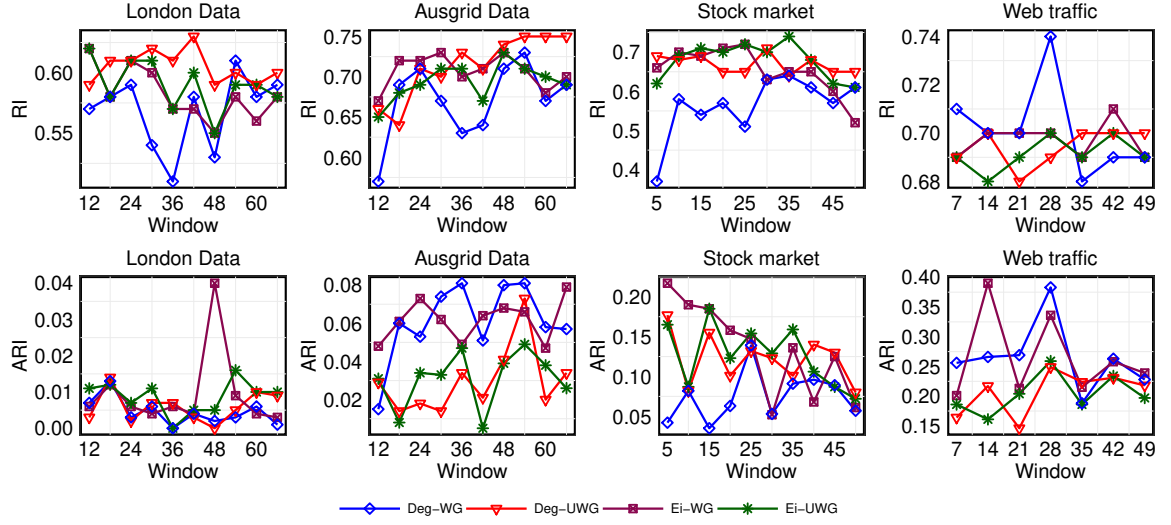


Figure 2.5: RI and ARI values obtained for the change in subsequence length in case of Deg-WG, Deg-UWG, Ei-WG and Ei-UWG graph representation on the proposed framework.

Table 2.4: The p-values of Wilcoxon signed rank test performed on the RI and ARI for the different clustering techniques. The columns corresponding to clustering on proposed framework is highlighted in bold.

	Deg-WG		Deg-UWG		Ei-WG		Ei-UWG	
	RI	ARI	RI	ARI	RI	ARI	RI	ARI
HC-C	1.2e-5	4.5e-4	9.5e-5	9.3e-5	1.3e-5	1.1e-5	1.3e-5	9.08e-6
HC-S	0.1	8.8e-7	1.4e-4	2.3e-6	2.8e-5	6.2e-7	4.2 e-5	1.8 e-6
K-shape	0.6	5.8e-6	3e-4	1.6e-5	0.003	1.6e-6	6.5e-4	7.1e-6
K-means	5.9e-6	0.1	0.1	0.03	0.01	0.02	0.3	0.18
SAX	0.1	6.3e-4	4.8e-6	0.01	1.6e-4	0.02	2.2e-4	4.7e-3
Walktrap	0.5	0.01	0.007	1.8e-4	0.02	0.01	0.01	3.7e-4
MST	5.9e-5	2.4e-6	9.5e-5	3e-6	1.29e-5	1.5e-6	1.3e-5	2.1e-6
MP	2.69e-6	1.5e-6	6.4e-5	3.9e-6	1.4e-8	1.1e-6	5.9e-6	4.7e-6

representation in case of *K*-means shows a significant difference. This justifies that the weighted path extraction from Ei-WG graph gives better clustering results than the other graph representation techniques.

2.5. RESULTS AND ANALYSIS

Table 2.5: The highest support values obtained from temporal patterns. The best support values are highlighted in **bold**.

Data	Deg-WG	Deg-UWG	Ei-WG	Ei-UWG
London	0.89	0.94	0.95	0.96
Ausgrid	0.95	0.96	0.96	0.93
Stock	0.96	0.95	0.96	0.95
Web traffic	0.79	0.76	0.93	0.67

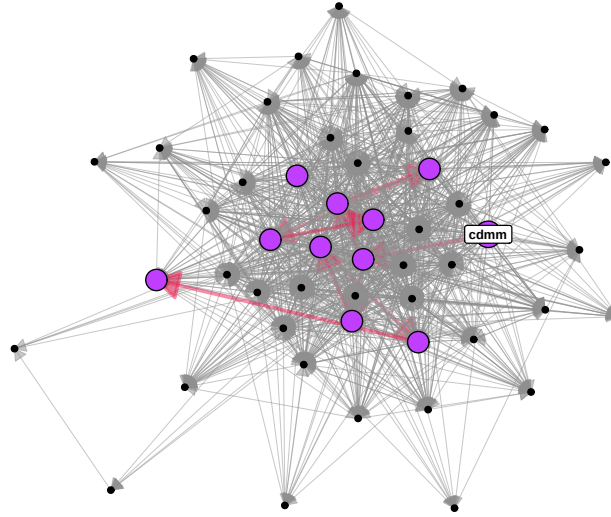
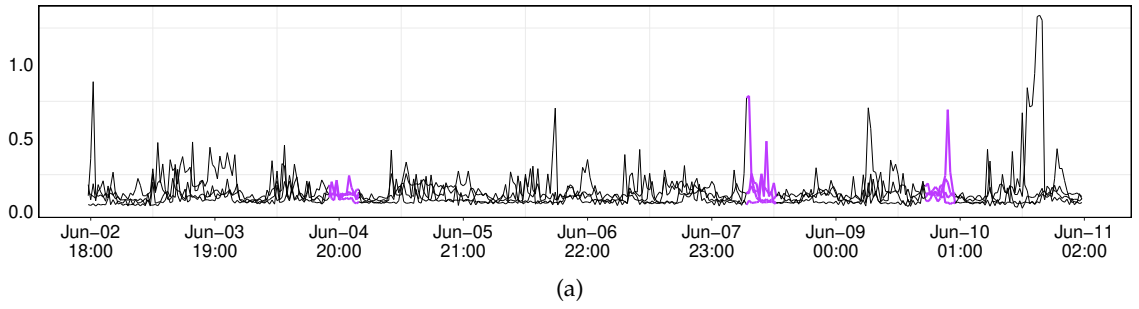


Figure 2.6: Subfigure (a) shows temporal subsequence patterns obtained in case of Ei-WG graph representation for London dataset and subfigure (b) shows the respective graph nodes for London dataset.

CHAPTER 2. CLUSTERING TIME SERIES USING DIRECTED GRAPH

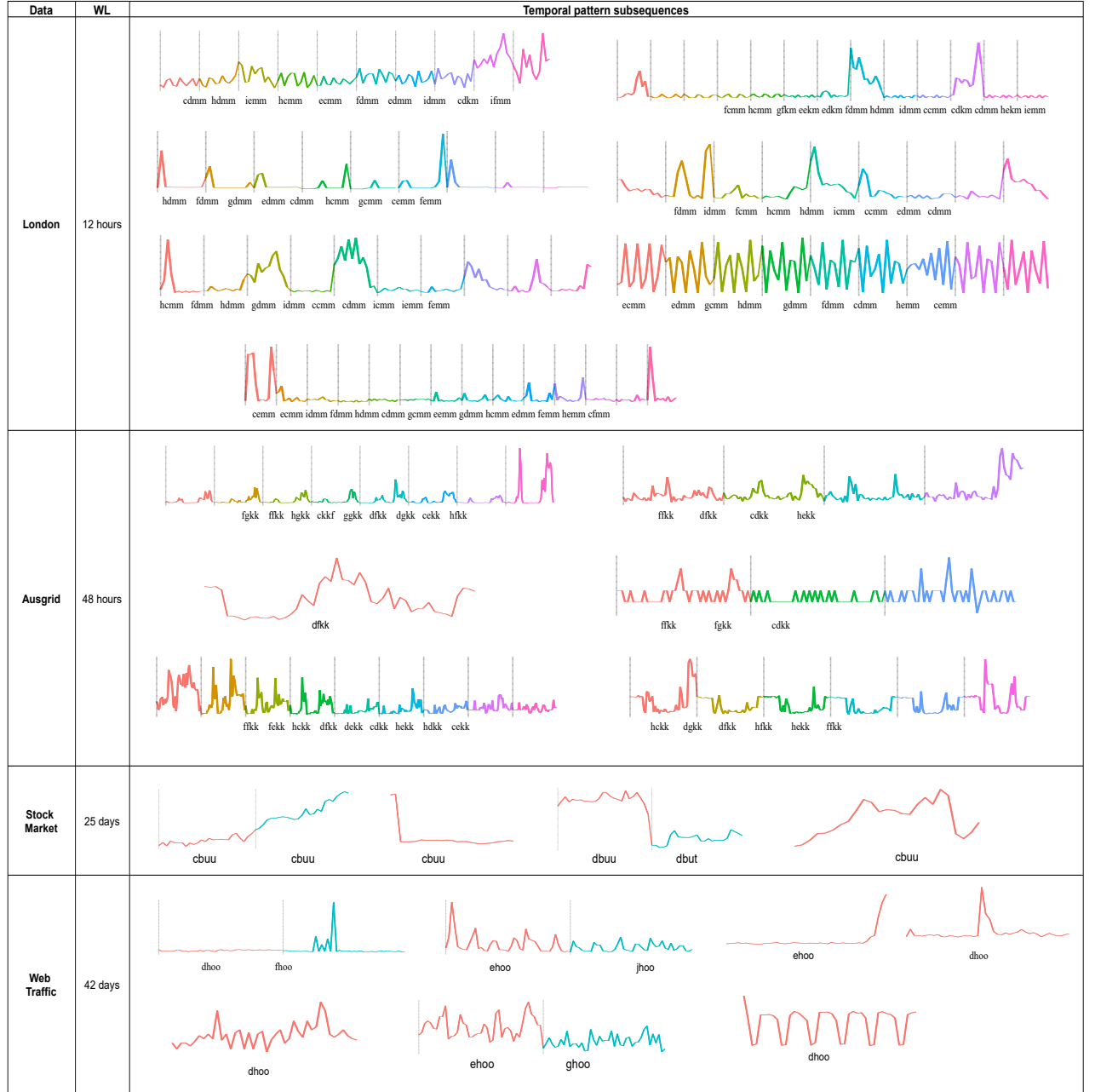


Figure 2.7: Temporal pattern and the respective symbols discovered from each cluster, shown in sequential order, in case of Ei-WG graph. For each dataset, the temporal patterns are illustrated using different colors in each window and is also marked by dotted lines. SL in the table is the subsequence length.

Temporal patterns

We report the support values obtained from the datasets in Table 2.5. In case of Web traffic data, the Ei-WG representation shows significantly high support value as that of others. Although the unweighted graph representation is better in terms of time complexity, but due to the higher support values obtained from Ei-WG, a weighted graph, we perform the rest of the experiments on it.

For each dataset, we show the occurrence of temporal patterns obtained from Ei-WG graph representation in as explained below. We show the date and time of occurrence of the temporal pattern and discuss how the temporal pattern discovery is useful with respect to the respective TS datasets. Temporal pattern for London dataset, cluster 1 is shown in Fig. 2.6 (a) starting with node *cdmm* and the length of the temporal pattern is 10. The colored nodes and edges of the graph forms the temporal pattern. The start node of the temporal pattern has been labeled by its symbol. The first pattern *cdmm* occurs during June 4th and 5th, second and third pattern occurs on June 8th and 9th respectively, as shown in Fig. 2.6 (b). This signifies that every building of cluster 1 has a similar consumption pattern at the given date and time, which can be the cause of increase in demand. This information will help the utilities to balance the demand supply needs by several demand-side management operations, one of which is the peak shifting. We obtain 7 different clusters from Ei-WG representation of London dataset. The temporal patterns and their respective symbols for other clusters is shown in Fig. 2.7 in a sequential order. The pattern in each window is colored for illustrative purpose. Let us discuss how we can characterize each cluster by the shape of temporal pattern occurring at a fixed date and time. Temporal patterns of cluster 1 has minor changes in consumption during the first 9 windows followed by a slow increase in the 10 window. Cluster 2 has pattern length of 13 and it can be characterized by alternate sequence of sharp rise and steady consumption pattern. Pattern length in cluster 3 is 9 and is characterized by a peak value occurring at the starting or the end of the day followed by a constant load consumption. The temporal pattern for cluster 6 has multimodal peak occurrence.

The temporal pattern for the Ausgrid dataset, cluster 1 is shown in Fig. 2.8 (a), that starts from the node *fgkk* and pattern length of 8. The first pattern *fgkk*, shown in graph in Fig. 2.8 (b), occurs on Aug 25th, followed by Aug 28th and Sep 1st respectively. This signifies that every house of cluster 1 generates the similar electricity patterns at the given date and time and this information can further help in designing accurate prediction models. The temporal patterns for the other clusters is illustrated in Fig. 2.7 with the respective symbols. Observing the temporal patterns, we can characterize the pattern of each cluster. In cluster 1, all the windows, except the last, shows an exponential increase in the load pattern. In cluster 2, the windows of temporal patterns forms sinusoidal shape due to the

CHAPTER 2. CLUSTERING TIME SERIES USING DIRECTED GRAPH

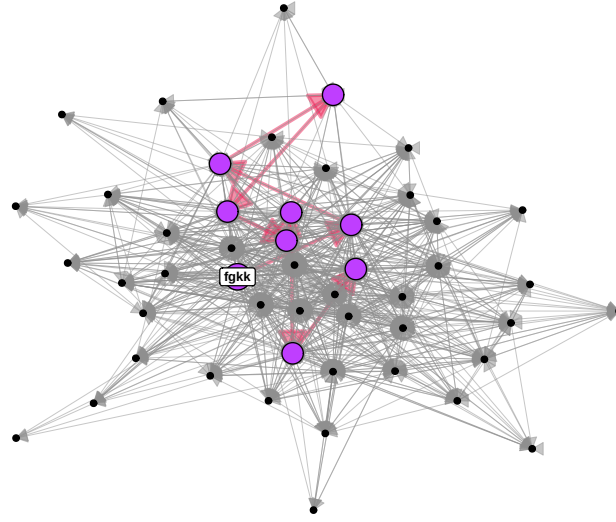
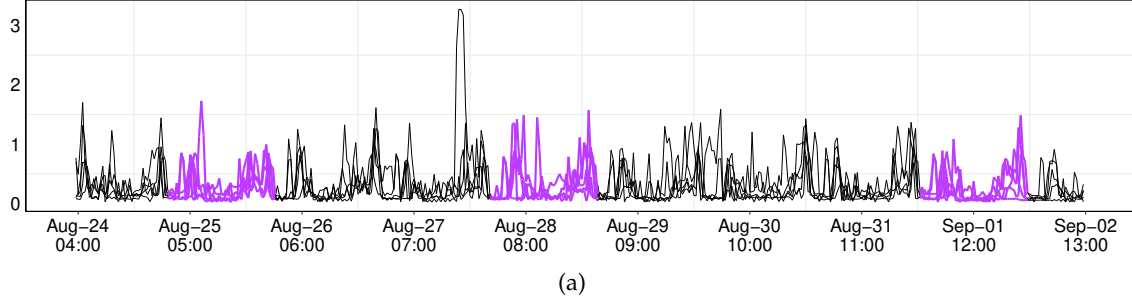


Figure 2.8: Subfigure (a) shows temporal subsequence patterns obtained in case of Ei-WG graph representation for Ausgrid dataset and subfigure (b) shows the respective graph nodes for Ausgrid dataset.

rise and falls in the pattern. The length of temporal pattern for cluster 3, is 1, which forms a sinusoidal pattern. The temporal pattern for cluster 4, has length 3, has triangular like patterns and the peak consumption is observed at the first and the last pattern. The length of temporal pattern in cluster 5 is 9 and shows slow decrease when moving towards the 9th pattern. The length of temporal pattern in cluster 6 is 9 out of which 8 of them have the rectangular shape patterns.

The temporal pattern for stock market data, cluster 1 is shown in Fig. 2.9 (a). The graph has a self loop as shown in Fig. 2.9 (b), which denotes that the pattern length is 2, starting and ending with *cbuu*. The occurrence of the first *cbuu* ranges from June 6th, 2017 to July 11th, 2017 and the occurrence of the second *cbuu* ranges

2.5. RESULTS AND ANALYSIS

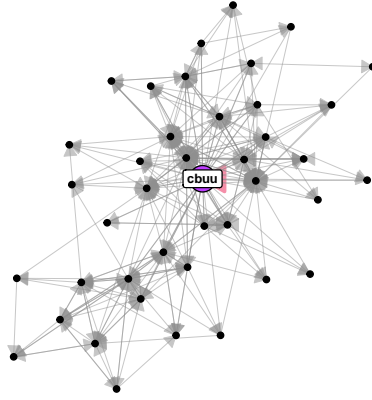
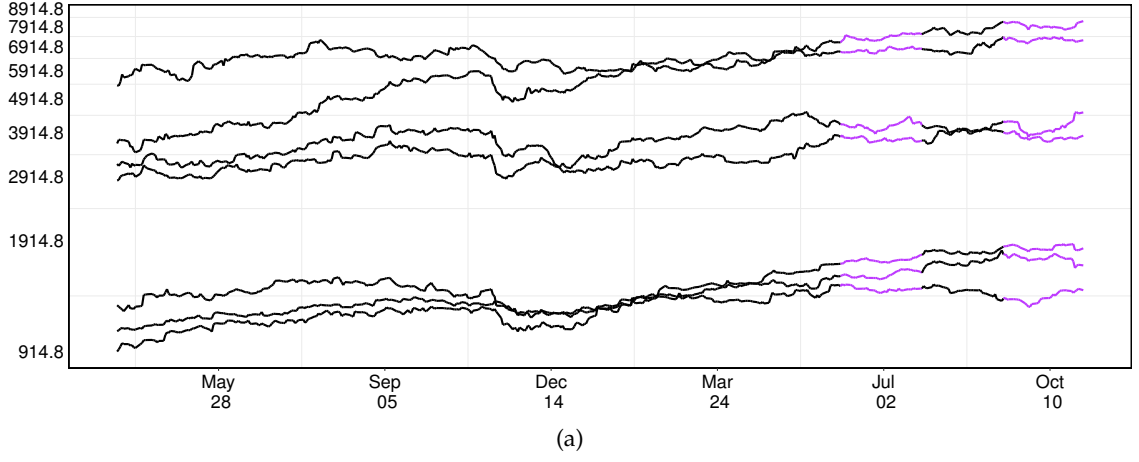


Figure 2.9: Subfigure (a) shows temporal subsequence patterns obtained in case of Ei-WG graph representation for Stock market dataset and subfigure (b) shows the respective graph nodes for Stock market dataset.

from Sep 12th, 2017 to Oct 17th, 2017. This signifies that every company of cluster 1 has similar closing price during the given date range, hence this information can help in building good recommender systems that can decide when to buy or sell the stocks. Temporal patterns for the other clusters are shown in Fig. 2.7 with the respective symbols. Cluster 1 has a linearly increasing trend, cluster 2 has a steep fall in the pattern, cluster 3 forms a staircase pattern with a slow fall, and cluster 4 shows sinusoidal shape. Although the symbols for cluster 1, 2 and 4 is same, the pattern shape differs.

The temporal pattern for web traffic data, cluster 1 is shown in Fig. 2.10 (a). The length of temporal pattern is 2, with *dhoo* as the starting node, shown in Fig 2.10 (b).

CHAPTER 2. CLUSTERING TIME SERIES USING DIRECTED GRAPH

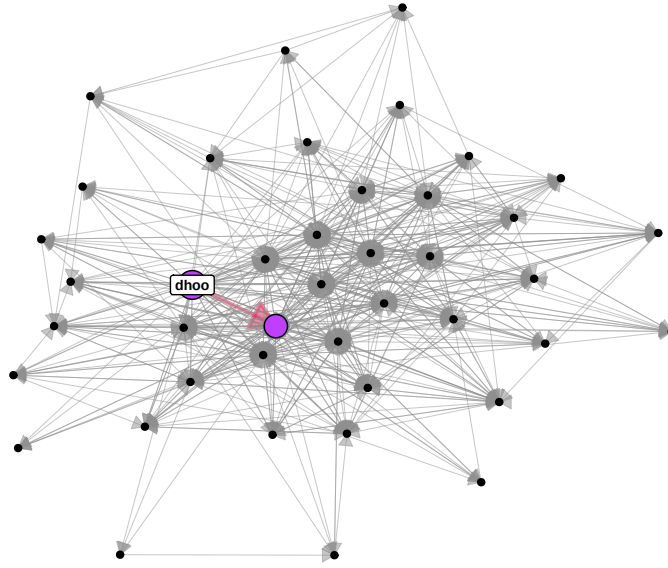
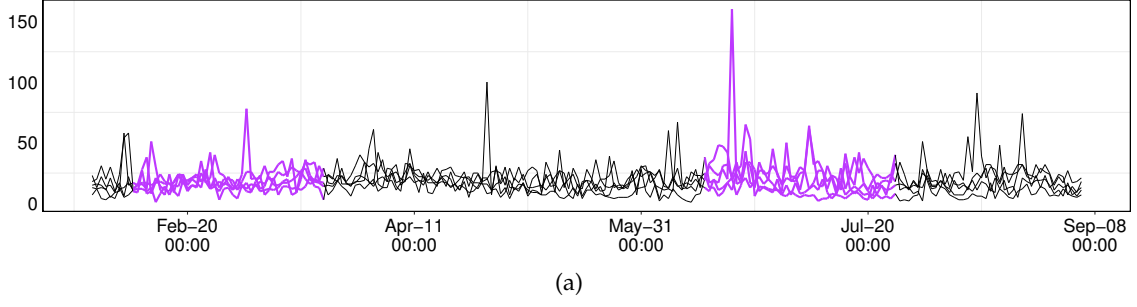


Figure 2.10: Subfigure (a) shows temporal subsequence patterns obtained in case of Ei-WG graph representation for Web traffic dataset and subfigure (b) shows the respective graph nodes for Web traffic dataset.

The time and date of occurrence of *dhoo* is from Feb 8th, 2017 to Mar 21st, 2017 and the occurrence of the next pattern, *fhoo* is from June 14th, 2017 to Jul 26th, 2017, shown in Fig. 2.10 (a). This signifies that all the web pages of cluster 1 have similar pattern of daily views during the above mentioned dates, and it can be the cause of peak in the web traffic. The temporal patterns for the other clusters are shown in Fig. 2.7 with their respective symbols, where cluster 1 shows no change in pattern during the first window and a peak in the second window. Cluster 2 shows a slow decrease in the number of visits from the first to the second window, cluster 3 shows a steep rise, cluster 4 shows a peak at the middle of the window, cluster 5

2.5. RESULTS AND ANALYSIS

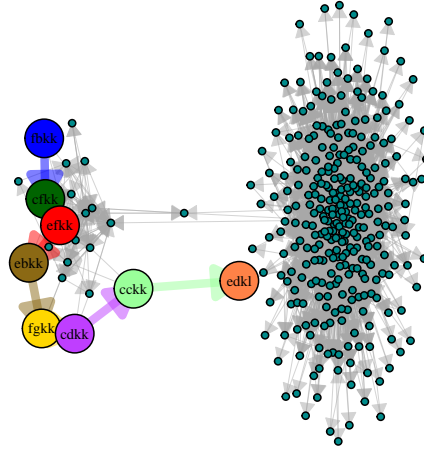
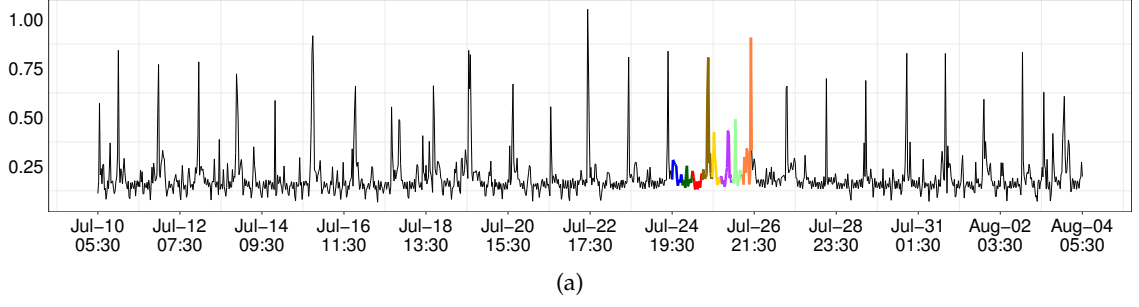


Figure 2.11: Subfigure (a) shows temporal dependent rare events obtained in case of Ei-WG graph representation for London dataset and subfigure (b) shows the respective graph nodes for London dataset.

shows a parabolic shape pattern, cluster 6 shows steady decrease from window 1 to window 2 and cluster 7 shows flat peak.

2.5.3 Rare event detection using graph component analysis

We illustrate the occurrence of rare event patterns and their graph structure obtained from Ei-WG graph representation and the respective patterns for all the datasets. Directed edges between the graph nodes in each case, denote the temporal order of occurrence of the rare events

The graph showing rare events for London dataset, in Fig. 2.11 (b) has 9 different component out of which 8 components are the rare events. The different components of the graph are shown using different colors and the rare events

CHAPTER 2. CLUSTERING TIME SERIES USING DIRECTED GRAPH

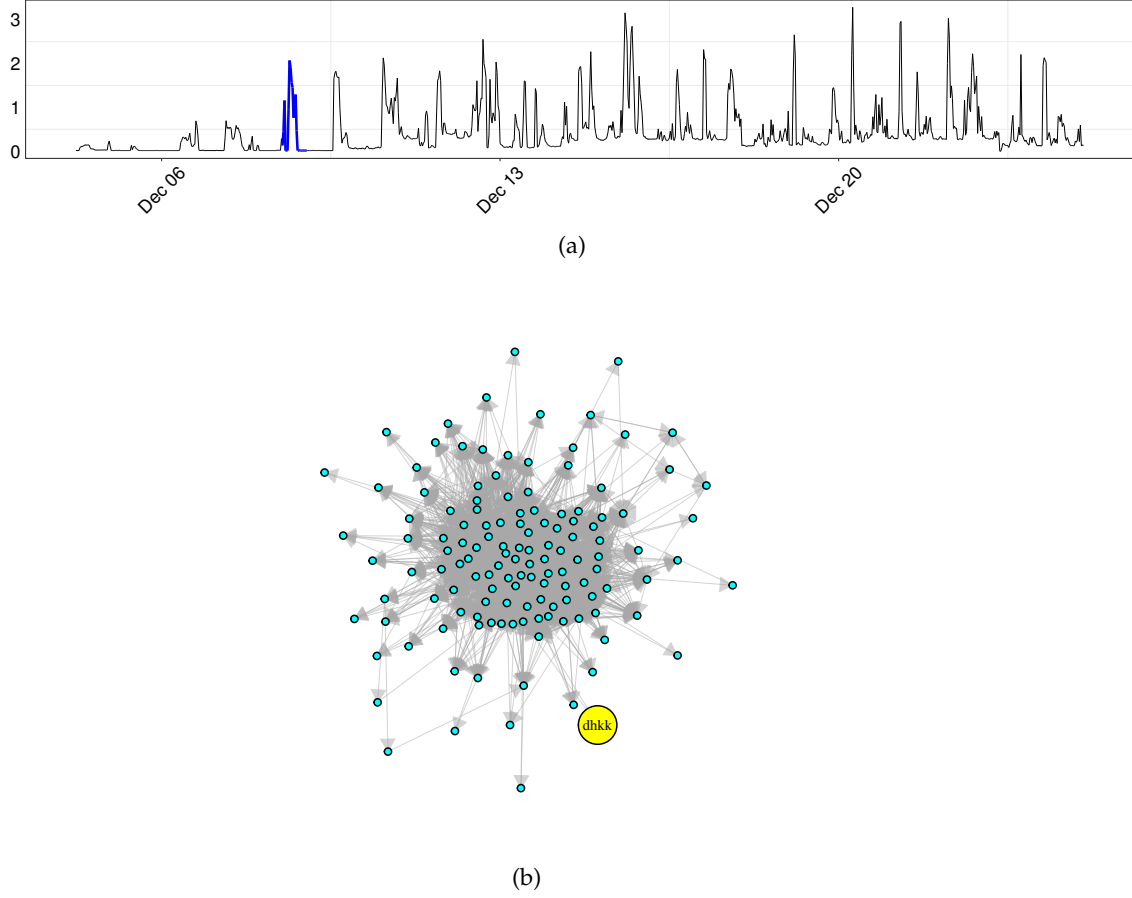


Figure 2.12: Subfigure (a) shows temporal dependent rare events obtained in case of Ei-WG graph representation for Ausgrid dataset and (b) shows the respective graph nodes for Ausgrid dataset.

are labeled by their symbols. Patterns for the rare events are highlighted in the plots for each dataset. Each of the rare events has $cs = 0.0001$, which is less than the threshold ct , while the $cs = 0.99$ s for the largest component. The nodes in the graph $fbkk \rightarrow cfkk \rightarrow efkk \rightarrow ebkk \rightarrow fgkk \rightarrow cdkk \rightarrow cckk \rightarrow edkl$ show the sequence of temporally dependent rare events. The pattern for the same is shown in Fig. 2.11 (a) and are colored same as the color of graph nodes.

Fig. 2.12 (a) and 2.12 (b) the rare event patterns and the graph for Ausgrid dataset. The graph shows two components. The $cs = 0.99$ for the largest component and the highlighted node symbolized as $dhkk$ is the rare event with $cs = 0.0004$, that is less than ct .

Fig. 2.13 (a) and 2.13 (b) shows the rare event patterns and the graph for the Stock market dataset. The $cs = 0.99$ for the largest component and the $cs = 0.002$ for rare event. The detected rare event is observed only in Eicher Motors Limited. The

2.5. RESULTS AND ANALYSIS

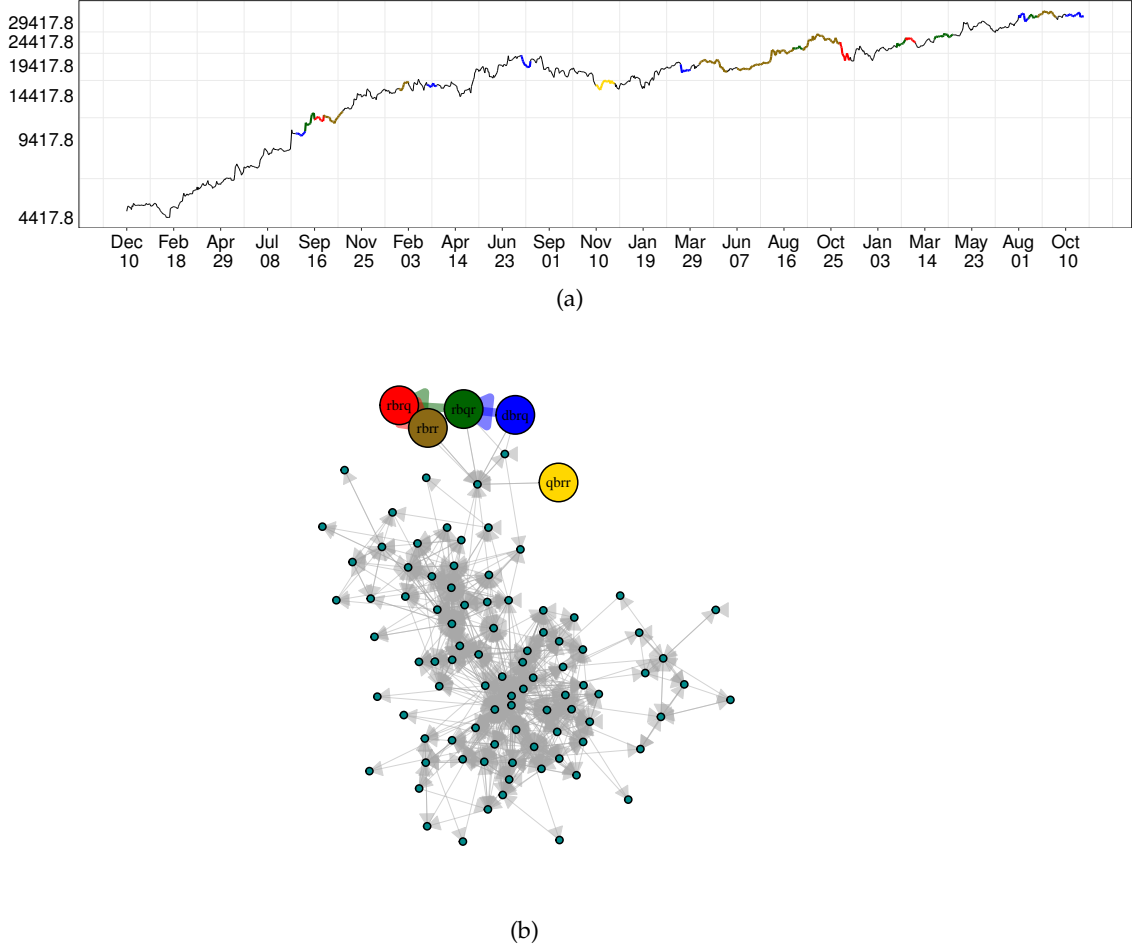


Figure 2.13: Subfigure (a) shows temporal dependent rare events obtained in case of Ei-WG graph representation for Stock market dataset and subfigure (b) shows the respective graph nodes for Stock market dataset.

nodes symbolized as $dbrq \rightarrow rbqr \rightarrow rbrq \rightarrow rbr$ are the temporally dependent rare events. The respective time and date of occurrence of the temporally dependent rare event starts from Mar 29th to Oct 25th as shown in the plot of Fig. 2.13 (a). Unlike the other companies, the closing price pattern for the Eicher motors show an increasing trend, which is the cause of rare event occurrence, as shown in Fig. 2.13 (a).

Fig. 2.14 (a) and 2.14 (b) shows the rare event patterns and the graph for a web page named Emanuel Macron in the Web traffic dataset. The rare event symbolized as $mhmo$ has the $cs = 0.0003$ and the largest sized component has $cs = 0.99$. The pattern corresponding to $mhmo$ shows the highest peak in the TS dataset and is detected as a rarely occurring event. The lowest cs and the pattern

CHAPTER 2. CLUSTERING TIME SERIES USING DIRECTED GRAPH

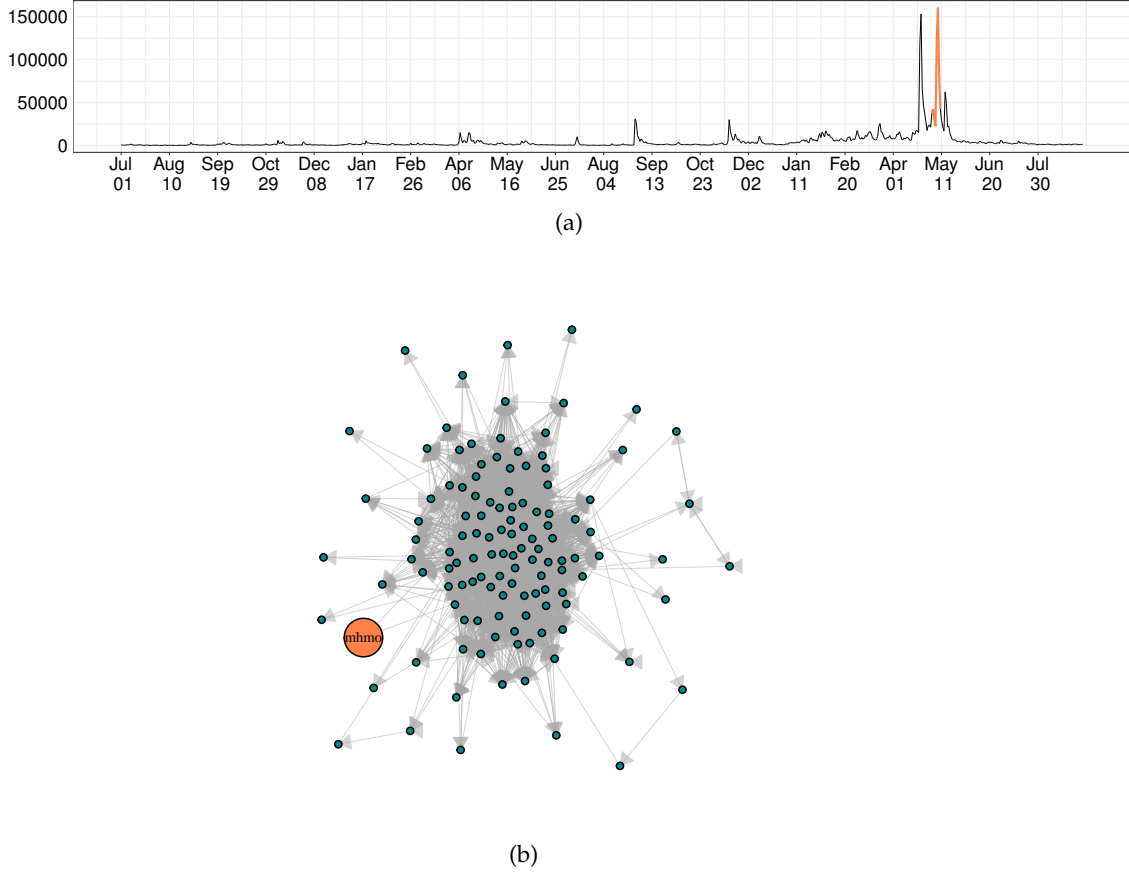


Figure 2.14: Subfigure (a) shows temporal dependent rare events obtained in case of Ei-WG graph representation for Web traffic dataset and subfigure (b) shows the respective graph nodes for Web traffic dataset.

of the rare event for the Web traffic dataset justify the rare events discovered from the proposed method.

2.6 Conclusions

In this chapter, we propose a graph based TS analysis framework– *Graft*. The purpose of the proposed framework is to identify the frequently occurring longest temporal patterns and also identify the temporally dependent atypical patterns. To achieve the purpose, the TS dataset is represented by a weighted directed graph structure, that can store the information about the pattern similarity and the temporal pattern dependency. Clustering of the TS samples is performed based on the graph path extraction. Cluster evaluation metrics obtained from proposed method are compared with the existing traditional clustering methods

2.6. CONCLUSIONS

and the graph clustering techniques. Higher cluster evaluation metrics in case of the proposed path based clustering signify higher similarity of the predicted cluster labels with the target.

The higher support values of frequently occurring longest temporal patterns reported, denote that those patterns are observed in majority of the cluster samples in a certain cluster. To justify the discovery frequently occurring longest temporal patterns, the study has been experimented on some planted sinusoidal patterns in a sample TS dataset. The component score for a graph component denote the ratio of edges with respect to the total number of edges in the graph. Lower component score denote that the desired pattern occurrence is comparatively lesser observed in TS dataset. The importance of these clustering analysis is to build accurate forecasting load models, recommender systems for stock market, resource management for web servers. The atypical pattern will identify the possible theft or newer patterns detected, and find the cause for the same. A comparative analysis on time complexity show that the time complexity of clustering in the proposed framework is similar to that of the existing clustering techniques. The proposed method shows minimum runtime as compared to the other clustering techniques.

3

Clustering time series subsequences using directed graph

3.1 Introduction

Subsequence clustering aim to cluster the subsequences of a TS sample so that the similar patterns can be grouped into a cluster. Considering a TS dataset, the TS subsequence clustering aim to group the similar patterns of the several TS samples. Because the TS dataset has multiple TS samples, a pattern can be associated with multiple other co-occurring patterns. Discovery of the subsequently co-occurring patterns and clustering based on the discovered patterns is a challenging part.

In general, the feature extraction is applied either on raw TS or on the subsequence TS. The optimal set of features can be chosen using cluster quality scores. But the primary concern is to incorporate the information about the subsequently co-occurring patterns of the patterns into the feature extraction approach. Based on the length of the subsequences, the TS dataset has large number of subsequences but some of them can have redundant pattern information. Therefore, it is necessary to identify the subsequences that have distinct information, in order to reduce the space complexity.

Existing subsequence clustering approaches look for similarity between the subsequences but they do not consider the temporal dependency between the subsequences. The graph theoretic approaches have been successful in mining similar as well as dependent events across several applications.

In this work, we formulate the subsequence clustering on TS dataset, as a dense subgraph mining problem for a weighted directed graph. The novelty of proposed weighted directed graph is that it can capture the dependency between the subsequences using directed edges and the similarity between the subsequences using the edge weights. The dense sub-graphs formed from the subsequently co-occurring patterns from the weighted directed graph is defined using the concept of quasi-clique. For atypical pattern identification, a subsequence that is followed by minimum number of other unique patterns is considered atypical. Out of the

different application areas of the TS subsequence clustering, this chapter shows how to obtain the stable set of consumers from a residential smart grid system, that can be chosen for demand response (DR) [138] or Demand Side Management (DSM). The stable set of consumers denote the consumer set that have a consistent electricity consumption behavior. The historic TS with a consistent consumption behaviour is more likely to adjust their demand. This will help in persistent peak load shifting and reduced daily energy consumption through long term contractual agreement with the smart grid operators.

3.2 Literature

The TS subsequences are extracted using the sliding window, however the study in [139] shows that clustering all possible subsequences extracted using the sliding window yields meaningless results. Zhang et. al. [92] introduced the subsequences as shapelets, which are described as selective subsequence set that describe the unique shape based features in the TS data. Clustering the TS based on the learned shapelets has shown good results [92] but the shapelet searching from the entire set of all possible subsequences is an exhaustive process. Some other factors to obtain meaningful clusters of subsequences depend on window length and the distance measures used. There exists several data encoding techniques for TS subsequences that have given good clustering results. Pramod et. al. [140] extract the adaptive and robust extrema features from the subsequences that are used for subsequence matching. The TS data generated from each source is domain specific and so the clustering result depends on the extracted features. However, Masoumeh et.al [141] claim that the shift-invariant versions of the spherical clustering, principal component analysis (PCA), sparse coding and non-negative matrix factorization (NMF) are suitable for TS of all domains.

The applications of subsequence clustering extends– rule discovery [142], frequently occurring pattern discovery [143] and rare event discovery [63]. Rule discovery is extraction of the causal events that have multiple occurrences during a time frame. The frequently occurring patterns in the data are discovered by mining all the possible subsequences of TS [143]. In [44], the authors use K-medoid algorithm to cluster the discovered motifs. The clustering and motif detection techniques using SAX approach in [106, 144, 145] have shown promising results. Several subsequence TS clustering approaches in literature have used the pairwise similarity measure and achieved good results [146, 94, 147]. However, none of them discuss about extracting the subsequently co-occurring pattern based information.

Considering the graph structure for TS, it has been used for link prediction where the link denote communication between two entities. Authors in [148] claim

CHAPTER 3. CLUSTERING TIME SERIES SUBSEQUENCES USING DIRECTED GRAPH

that the graph based structure for link prediction can capture both the structural and temporal dependencies. The advantage of using graph structure for TS is that it provides multi-level analysis that reflect the co-occurring patterns present in the TS [6]. TS to graph based representation is shown in [63] for anomaly discovery, in [127] for whole TS clustering, in [53] for rule discovery.

In this chapter, the dimensionally reduced TS dataset is combined with the graph based structure. There are some properties in graph based structure, that assists in capturing the TS pattern 1. the vertices can be used to represent the unique patterns from TS 2. the weighted edges can be used to represent the similarity between paired vertices 3. the directionality of the edges can help in preserving the temporally dependent events in TS. The unweighted undirected graph clustering is faster and abundant in literature as compared to the weighted and directed graphs. There are some approaches, discussed in [149] to convert the directed graph into undirected and then apply the traditional community detection algorithms, but this conversion in case of TS will cause loss of temporal information.

3.3 Methodology

In this section, we discuss the proposed work and the details of the proposed clustering approach. A flowchart describing the control flow between different steps of the proposed framework is briefly described using Fig. 3.1. The input to the flowchart is the time series data, denoted as $T = \{t_1 \dots t_m\}$. The data pre processing steps for missing data imputation, duplicate removal and normalization are discussed in section 3.5.1. The first important step is segmentation and symbolic representation on T which is discussed in section 3.3.2. The symbolic data is used for graph representation as discussed in section 3.3.3. The details of the proposed quasi clique clustering technique on the graph is discussed in section 3.3.4. The cluster labels thus obtained (S_G) is finally used for DR programs as discussed in section 3.6.4.

3.3.1 Definitions and Notations

To describe the proposed framework we discuss the notations and define the underlying concepts.

Notation 1 *Time series dataset:* The TS dataset is denoted by a matrix, $T = \mathbb{R}^{m \times n}$, where, m is the number of TS samples and $|t_i| = n, \forall t_i \in T$.

Notation 2 *Time series subsequences:* A TS t_i , is split into equal length non-overlapping segments, called the subsequences of length l . Given the TS t_i , the subsequence of t_i starting at time instance j is $\{t_i^j \dots t_i^{(j+l-1)}\}$.

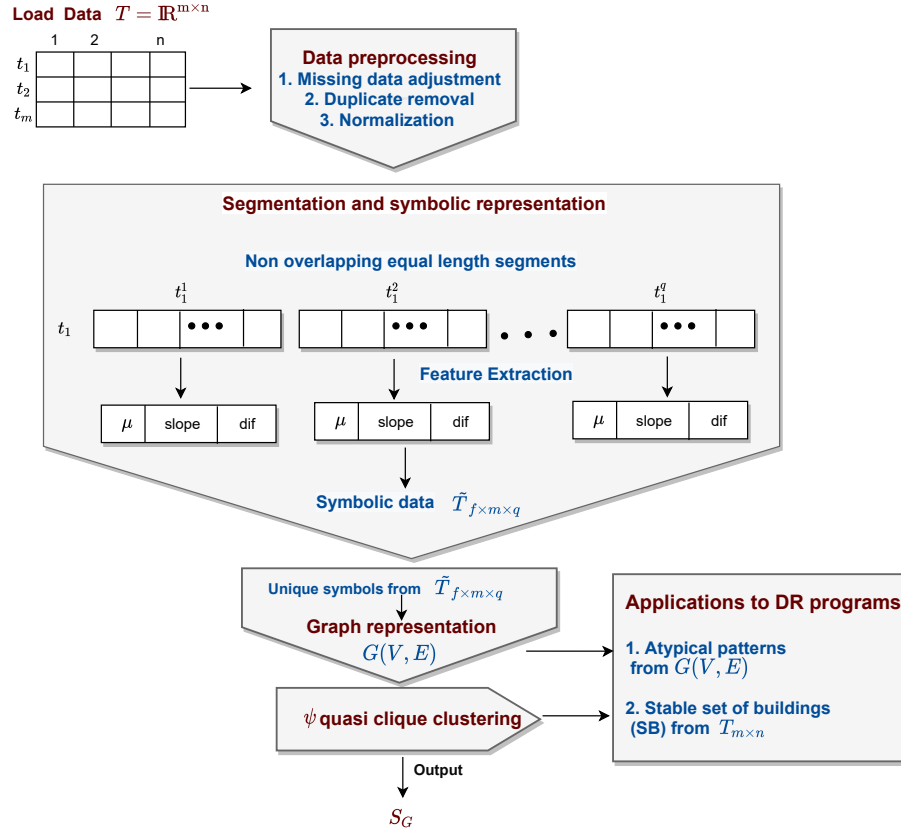


Figure 3.1: A flowchart of the proposed graph based subsequence clustering method with application to Demand Response (DR) program for identification of stable consumers and identification of atypical patterns.

Notation 3 *Graph*: A directed weighted graph is denoted as $G(V, E, W)$, where V is the set of vertices, E is the set of directed weighted edges and $W(e_{i,j})$ is the weight function that maps an edge $e_{i,j} \in E$ to a real value.

Notation 4 *Vertex weight*: Weight of a vertex $v_i \in V$ is the sum of weights of its outgoing edges given as: $W(v_i) \leftarrow \sum_j (W(e_{i,j}))$, where $e_{i,j} \in E$ is an outgoing edge from v_i to $v_j \in V$.

Definition 6 *Maximum Clique on a weighted graph*: Given a weighted graph, a clique is defined as a complete subgraph s such that for every pair of vertices $i, j \in s$ there exists an edge (i, j) in s and the summation of the edge weights of s is the maximum over all complete subgraphs in G .

Definition 7 *Clique on a directed graph*: Given a directed graph, a clique is defined as a

CHAPTER 3. CLUSTERING TIME SERIES SUBSEQUENCES USING DIRECTED GRAPH

subgraph s such that for every pair of vertices $i, j \in s$ there exists both the edges $i \rightarrow j$ and $j \rightarrow i$.

Definition 8 γ -quasi-clique: Given a unweighted undirected graph, a γ -quasi-clique is defined as a subgraph s if $\forall i \in s, \frac{\deg(i)}{(|s|-1)} \geq \gamma$, where $0 \leq \gamma \leq 1$.

3.3.2 Segmentation and symbolic representation

Algorithm 6 Symbolic representation

Input: T : TS dataset

Output: \tilde{T} : Symbolic representation

```

1: Initialize  $F_{f \times m \times q}$ 
2: for each TS  $t_i \in T$  do
3:   for each segment  $j$  in  $1 : q$  do
4:      $F_{1 \times i \times j} \leftarrow \mu(t_i^j)$ 
5:      $F_{2 \times i \times j} \leftarrow \text{slope}(t_i^j)$ 
6:      $F_{3 \times i \times j} \leftarrow \text{dif}(t_i^j)$ 
7:   end for
8: end for
9: For each  $f$  compute median
    $\text{med}[f] = \text{median}(F_{f \times m \times q})$ 
10: for each  $f$  do
11:   for  $a$  in  $3 : 20$  do
12:      $E[a] \leftarrow \min(|B^a - \text{med}[f]|)$ 
13:   end for
14:    $\alpha[f] \leftarrow \arg \min(E[a])$ 
15: end for
16:  $\alpha_{opt} \leftarrow \max(\alpha)$ 
17: Symbolic conversion of  $F_{f \times m \times q}$  using  $\alpha_{opt}$ 
18: Concatenate the symbols to obtain the symbolic representation  $\tilde{T}_{m \times q}$ 

```

Two subsequences, t_i^x and t_i^y are non-overlapping if $y \geq x + l - 1$, where x and y are the start times of the respective subsequences. The number of subsequences, for each t_i of length n is $q = n/l$, where l is the length of the subsequences. A higher value of l results in loss of information while a smaller value of l produces unique encodings that might fail to find the common patterns between the subsequences.

Segmentation of each subsequence $\forall t_i \in T$, is followed by symbolic representation. Following the extended SAX approach developed in [111], three different features are extracted from each subsequence which are then converted to symbols. The features are:

1. average or mean, $\mu(t_i^j)$

2. to capture the trend pattern of the subsequence, slope is used, $slope(t_i^j)$
3. to determine the range of values, the difference between maximum and minimum value in the subsequence is used, $dif(t_i^j)$.

Although other features may be considered, the above chosen feature set can be computed in linear time which is important for analysis of large datasets.

Let the PAA of normalized TS t_i be \tilde{t}_i . For a given alphabet $a > 2$ and $a = \{3, \dots, 20\}$, the sorted list of breakpoints $B^a = \{\beta_k\}_{k=1}^{a-1}$, is used to obtain the SAX symbols using the Equation 3.1,

$$\tilde{t}_i^j = \alpha_k, \text{ if } \beta_{k-1} \leq \tilde{t}_i^j \leq \beta_k \quad (3.1)$$

where $j \in \{1, \dots, q\}$ and α_k denote the k th element of the alphabet.

The pseudo code for symbolic representation is discussed in algorithm 6. $F_{f \times m \times q}$ is a tensor, where f denotes the number of features that represent a subsequence. The appropriate alphabet size for each feature $\{F_x\}_{x=1}^f$, is its median. For example, in case of mean represented as F_1 , we compute $\text{median}(F_{1 \times m \times q})$. In step 9, med is a vector which stores the median value of the features and E is the vector which stores the minimum value of $|B^a - med[f]|$. For the feature f , we obtain the alphabet size $\alpha[f]$ whose corresponding breakpoint is closest to the median value of the feature f , as shown in step 14. The maximum from vector α , that is, α_{opt} is chosen for symbolic representation of entire TS. Output of algorithm 6 is the symbolic representation of TS data T , given as $\tilde{T}_{m \times q}$.

3.3.3 Graph representation

The directed weighted graph G is formed from symbolically represented TS data \tilde{T} as discussed in notation 3. The aim is to extract groups of similar subsequences occurring in the TS dataset that also preserves the temporal order of their occurrence, therefore we propose the following approach for construction of G , that preserve the temporal order of occurrence of the subsequences. The subsequences with same symbols represent similar patterns, so we extract the unique symbols in \tilde{T} that form the graph vertices. Mapping of a unique symbol to a vertex of G is given by the mapping function $V : \tilde{t}_i^j \rightarrow v_i$. If $V(\tilde{t}_i^x) = v_i$ and $V(\tilde{t}_i^y) = v_j$, then the mapping function of edges $E(.,.)$ is given as, $E : (v_i, v_j) \rightarrow e_{ij}$, if $y = x + l - 1$. The weight of the edge is computed using a modified SAX distance given as $W(e_{ij}) = D(.,.)$. The SAX distance of two time series with SAX representation as: \tilde{t}_x, \tilde{t}_y , is computed using Equation 3.2, where, \tilde{t}_x^i is the i th element of the string \tilde{t}_x . In the Equation 3.2 the function $dist(.,.)$ can be implemented using a table lookup where the rows and columns of the table represent the letters of SAX alphabet and the value in the cell

CHAPTER 3. CLUSTERING TIME SERIES SUBSEQUENCES USING DIRECTED GRAPH

$$SAX(\tilde{t}_x, \tilde{t}_y) = \sqrt{\frac{n}{q}} \sqrt{\sum_{i=1}^q dist(\tilde{t}_x^i, \tilde{t}_y^i)^2} \quad (3.2)$$

$$dist(\alpha_r, \alpha_c) = \begin{cases} 0, & \text{if } |r - c| \leq 1 \\ \beta_{\max(r,c)-1} - \beta_{\min(r,c)} & \text{otherwise} \end{cases} \quad (3.3)$$

(r, c) is the distance between the r th and c th letter of the alphabet. The function $dist(\alpha_r, \alpha_c)$ is defined in the Equation 3.3, and β_k is as defined in Equation 3.1.

Limitation of SAX distance is that it ignores the distance between two successive symbols, causing information loss and hence the difference between two patterns could not be identified correctly. To reduce this discrepancy, we modify the existing SAX distance by introducing a gap function $gap(.,.)$ that capture the distance between two successive symbols. The modified SAX distance, given as $SD(F_x^i, F_y^i)$ is shown in Equation 3.4, where F_x^i is the SAX encoding of feature value representation of t_x^i . The $\max(\alpha_r)$ and $\min(\alpha_c)$ are maximum and minimum values of the regions corresponding to r th element of the string α_r and c th element of the string α_c . A working example of computing the modified SAX distance between two subsequences, is shown later in section 3.3.5 of the paper. The summation of $SD(.,.)$ over q subsequences may be used to get the modified SAX distance between two time series.

$$SD(F_x^i, F_y^i) = \sqrt{\frac{n}{q}} \sqrt{\sum_{k=1}^f dist(F_x^{i,k}, F_y^{i,k})^2 + gap(F_x^{i,k}, F_y^{i,k})} \quad (3.4)$$

$$gap(\alpha_r, \alpha_c) = \begin{cases} (\max(\alpha_r) - \min(\alpha_c))^2, & \text{if } (r - c) = 1 \\ 0, & \text{otherwise} \end{cases}$$

3.3.4 Clustering

The proposed quasi-clique based clustering approach on a directed weighted graph G , searches the groups of similar vertices that preserve the temporal ordering. Relaxing the existing definition 6 of clique on a weighted graph and definition 7 on a directed graph and extending definition 8 of a γ -quasi clique on a weighted graph, we propose definition of ψ -quasi clique on a directed weighted graph with normalized edge weight.

Definition 9 *Given a weighted directed graph G , a subgraph s is called a ψ -quasi clique if both the following conditions are satisfied*

Algorithm 7 Quasi- clique clustering

Input: G : directed weighted graph

Output: S_G : graph clusters

```

1: For each  $v_i \in G$ , compute  $W(v_i)$ 
2:  $Q \leftarrow$  Enqueue the vertices with  $W(v_i)$  as the associated priority
3: Initialize  $S_G = \emptyset$ 
4: while  $Q$  is not empty do
5:    $s_i \leftarrow$  Dequeue maximum element from  $Q$ 
6:   loop
7:      $s_j \leftarrow$  Dequeue maximum element from  $Q$  such that
        $s_j$  is a neighbour of  $s_i$ 
8:     if  $s_j = \emptyset$  then
9:       exit loop
10:    end if
11:     $new\_s_i \leftarrow subgraph(\{s_i \cup s_j\})$ 
12:    if  $WCC(new\_s_i) \ \& \ \psi(new\_s_i) \geq \psi_{min}$  then
13:       $s_i \leftarrow vertices(new\_s_i)$ 
14:       $new\_s_i = \emptyset$ 
15:      Continue
16:    else
17:       $R \leftarrow R \cup \{s_j\}$ 
18:    end if
19:  end loop
20:  Enqueue the vertices in  $R$  to  $Q$  with the
    associated priority
21:   $R = \emptyset$ 
22:   $S_G = S_G \cup \{s_i\}$ 
23: end while
24: return  $S_G$ 

```

1. s is a weakly connected component (WCCs), and

2. if $\forall i \in s, \frac{\min(w(i), deg^-(i))}{|s|-1} \geq \psi$, where $0 \leq \psi \leq 1$ and $deg^-(i)$ is the out-degree of vertex i .

Condition 1 ensures that the temporal order of occurrence of patterns represented by the vertices is preserved while condition 2 ensures that every γ -quasi clique is dense even for a disassortative graph where vertex weight and vertex degree are not correlated. A clique on a weighted directed graph with normalized edge weights is a special ψ -quasi-clique where every pair of vertices form a SCC that is for every pair of vertices (i, j) there exist both the directed edge $i \rightarrow j$ and $i \leftarrow j$ and $\psi = 1$.

CHAPTER 3. CLUSTERING TIME SERIES SUBSEQUENCES USING DIRECTED GRAPH

We propose the definition of maximal ψ -quasi clique on a directed weighted graph with normalized edge weight as below:

Definition 10 *Given a weighted directed graph G , a subgraph s is called a maximal ψ -quasi-clique if all the following conditions are met -*

1. s is a ψ -quasi clique on G , where $0 \leq \psi \leq 1$, and
2. for all $i \in V$ such that i is a neighbour to s , $s \cup \{i\}$ cease to remain a ψ -quasi clique on G .

The complexity of the naive problem of finding all maximal ψ -quasi clique from a given weighted directed graph G is exponential in the size of the input graph. It is known that there can be as much as $\Omega(3^{n/3})$ maximal cliques in a graph [150], and hence there can be at least as many maximal quasi-cliques, since each clique is a special ψ -quasi clique. Even it is shown that it is computationally hard to propose an approximation algorithm for this problem [151].

In this chapter we propose a bottom up greedy graph clustering algorithm such that every cluster is the maximal ψ -quasi clique in the given weighted directed graph G as mentioned in algorithm 7.

In algorithm 7 the vertices are first arranged in non-increasing order of their weight values $W(v_i)$ in the priority queue Q where the priority is the weight value of the vertices $W(v_i)$. Vertex with highest weight is considered as the central vertex of G because it is either connected to maximum number of other vertices of the graph or it is connected to other vertices which are most similar to it. According to the greedy strategy in the algorithm, inclusion of the vertex with higher weight to the cluster at an early stage, help to find the maximal ψ -quasi clique as compared to the strategy of including a vertex with lower weight.

The term clusters and subgraphs S_G are used interchangeably in algorithm 7. Initially, the vertex with highest weight forms an individual cluster $\{s_i\}$. At each iteration, the vertex with next highest weight that is also neighbour of s_i is chosen as s_j . The subgraphs s_i and s_j are merged if the following conditions are satisfied -

1. the new subgraph, new_s_i should form a WCC, that ensures a minimum connectivity between the vertices. The WCC are found by a breadth-first search algorithm.
2. the new subgraph, new_s_i should satisfy the condition $\psi(new_s_i) \geq \psi_{min}$, where,

$$\psi(new_s_i) = \min_{\forall v \in new_s_i} \frac{\min(W(v), deg^-(v))}{(z - 1)} \quad (3.5)$$

where, $z = |new_s_i|$, the number of vertices in subgraph new_s_i . The parameter ψ_{min} determines the quality of the clusters.

3.3. METHODOLOGY

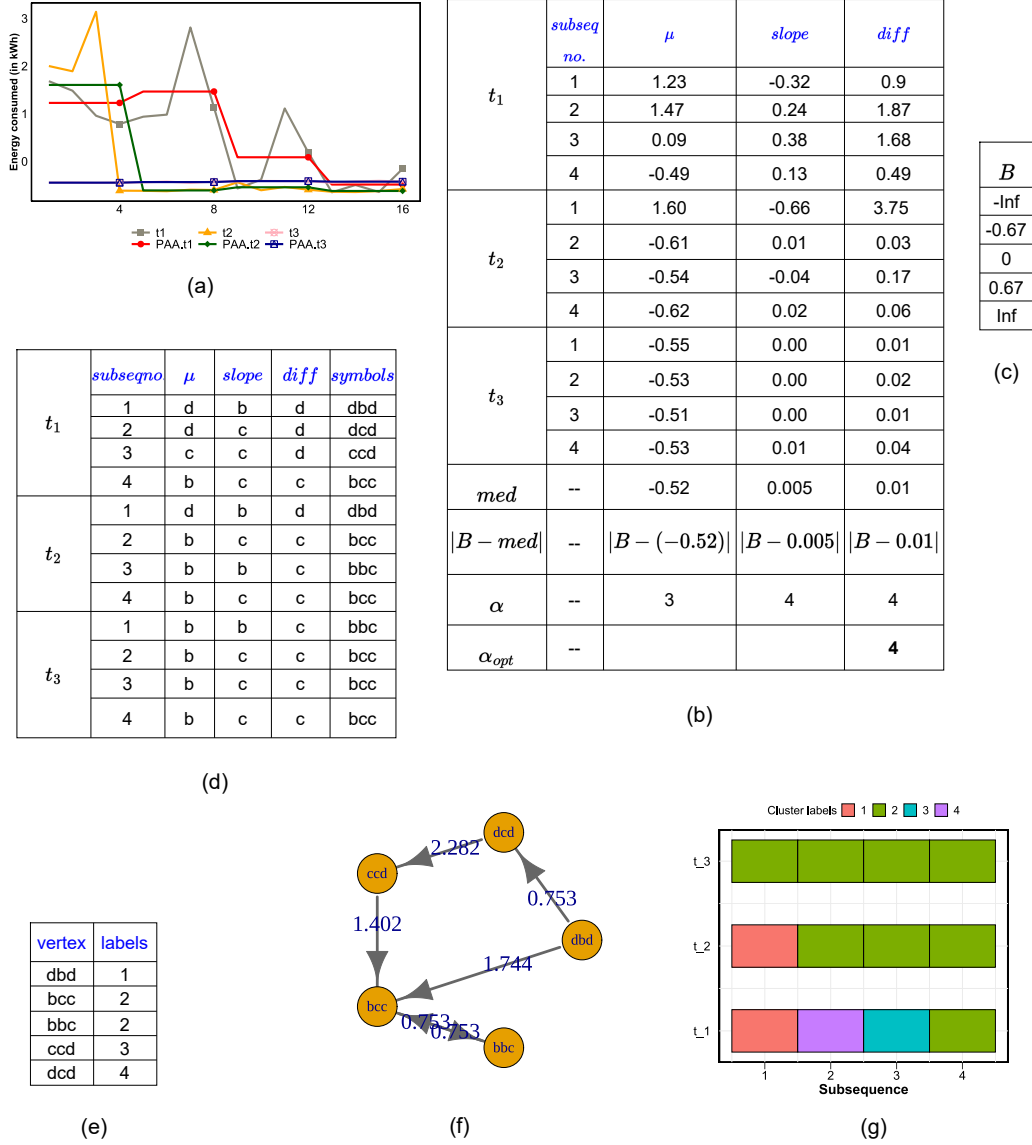


Figure 3.2: (a) A working example of the proposed TS subsequence clustering, shown for time series dataset T , with three time series t_1 and t_2 and t_3 , each with $n = 16$ and $l = 4$. The PAA is shown for the respective time series. The points in the lines denote the start and end of subsequences. x-axis is the time units and y-axis is the power consumption values. (b) The features obtained from each subsequence in T . α_{opt} is the alphabet size obtained, shown in **bold**. (c) The B^4 represents the breakpoints for $\alpha_{opt} = 4$ (d) Symbolic representation of the features \tilde{T} . (e) The graph representation obtained from \tilde{T} (f) The cluster labels obtained using the proposed quasi clique clustering approach, for $\psi_{min} = 0.3$. (g) Stability of t_1 , t_2 and t_3 . The cluster labels of the subsequences are illustrated with different colors.

CHAPTER 3. CLUSTERING TIME SERIES SUBSEQUENCES USING DIRECTED GRAPH

The iterations are repeated until no more merge is possible with s_i . s_i forms a maximal ψ -quasi clique. The above steps are repeated for all the vertices that do not belong to s_i . The algorithm returns a set of subgraph S_G which are the maximal ψ -quasi cliques of graph G .

3.3.5 Working Example

A working example of the proposed method is shown in Fig. 3.2. For simplicity, we assume three time series samples given as $T = \{t_1, t_2, t_3\}$ with $l = 4$ and $q = 4$. The 5 unique symbols obtained from Fig. 3.2 (d) is used for graph representation, as shown in Fig. 3.2 (e). The edge direction in $dbd \rightarrow dcd$ is based on the temporal order of occurrence and the edge weight is computed using Equation 3.4.

$$SD(\tilde{t}_x, \tilde{t}_y) = \sqrt{\frac{16}{4} \sqrt{(0+0) + (0 + gap(b, c)) + (0+0)}} \quad (3.6)$$

where $gap(b, c)$ is a non-zero value. According to the values of *slope*, given in Fig. 3.2 (b), we obtain $gap(b, c) = (0.38 - (-0.0006))^2 \approx 0.15$ and $SD(\tilde{t}_x, \tilde{t}_y) = 0.75$.

3.3.6 Time complexity

As shown in section 3.3, the proposed approach has three major steps- segmentation and symbolic representation, graph representation and clustering.

Time for obtaining a segment from a time series is $O(1)$. For time series t_i , the number of subsequences obtained is $q = n/l$. The time complexity for feature extraction from a subsequence of length l is $O(l)$. For the dataset with m time series samples each of length n , the worst case complexity for segmentation is $O(mn)$. In case $n \gg m$, the time complexity becomes, $O(n)$.

The symbolic representation require to compute the median value of features. For total $m \times n/l$ number of subsequences and $n \gg m$, the time complexity for median value computation in worst case is $O(n \times \log n)$. Time to obtain α_{opt} for the symbolic representation is $O(1)$. The symbolic representation is linearly dependent on number of features. For f different features, the worst case time complexity is $O(m \times (n/l) \times f)$, and $n \gg m$, it becomes $O(n)$. Therefore the worst case complexity for symbolic representation is $\max(O(n \log n), O(n)) = O(n \log n)$.

In the worst case, the total number of unique symbols extracted as graph vertices is approximately $m \times (n/l)$, therefore the time complexity to extract the unique symbols is $O(m \times (\frac{n}{l} \log \frac{n}{l}))$ and as $n \gg m$, it becomes $O(n \log n)$. Deciding the edge direction is based on sorting the subsequences in time order, which also require $O(n \log n)$ time. For f features, the time complexity to compute the edge weight using modified SAX distance is $O(f^2)$. In worst case, if we consider a complete graph with $m \times (n/l)$ number of vertices, the worst case time complexity for edge

3.4. APPLICATION TO DR PROGRAMS

weight computation is when $n \gg m$ is, $O(n^2)$. Therefore, the time complexity for graph representation is $\max(O(n), O(n \log n), O(n^2)) = O(n^2)$. Considering the directed weighted graph $G(V, E, W)$, the quasi-clique based clustering approach has three main steps-

1. Sorting vertices on vertex weights. The time complexity is $O(V \log V)$.
2. Two conditions are checked before each merge, as given in section 3.3.4. The time complexity to obtain WCC is $O(V + E)$ and the time to check whether ψ -quasi clique satisfies, is constant. In worst case, a merge occurs in each step, the time complexity becomes $O(V \times (V + E))$.

In worst case, the maximum number of vertices in G is approximately $m \times n/l$. For $n \gg m$, the total time complexity for the proposed quasi clique clustering approach is $O(n^2)$.

3.4 Application to DR programs

We discuss how the labeled subsequences can help in achieving energy efficiency through DR program. Firstly, we find a stable set of buildings which are suitable for the load shifting operations. Secondly, we identify the atypical patterns using graph degree analysis.

3.4.1 Measuring Stability

The buildings that have consistent load pattern, are considered as stable customers that are suitable for the DR as well as energy efficient (EE) programs [152, 153] focusing peak load shifting and peak as well as daily consumption reduction. The buildings showing sudden change in consumption patterns are less likely to meet the said objectives as sudden consumption change indicates the presence of uncertain factors that trigger the unpredictable load pattern.

Assuming a cluster set $S_G = \{s_i\}_{i=1}^C$, we use the information theory metric Shannon's entropy to measure the stability score of a building (t_j) that can be computed as [154]:

$$H(t_j) = - \sum_{i=1}^C \mathcal{S}(s_i, t_j) \times \log \mathcal{S}(s_i, t_j) \quad (3.7)$$

where the temporal variation function $\mathcal{S}(\cdot)$ intends to find the membership of the cluster label s_i in the building t_j . The temporal variation function is defined as: $\mathcal{S}(s_i, t_j) = \frac{1}{q} \sum_{p=1}^q \delta(s_i, t_j^p)$, where $\delta(s_i, t_j^p) = 1$, if the cluster label for the subsequence t_j^p is s_i and 0 otherwise. q is the total number of subsequences. The buildings showing lesser $H(\cdot)$ value are considered to have high stability score and are likely

CHAPTER 3. CLUSTERING TIME SERIES SUBSEQUENCES USING DIRECTED GRAPH

to have more consistent demand. If they are selected for DR programs, they will result in persistent participation to DR programs for longer period of time.

We compute the entropy of t_1 , t_2 and t_3 based on the cluster labels obtained, illustrated in Fig. 3.2 (g). In t_3 all the segments belong to cluster 2 and in t_2 , cluster label 2 is observed in three consecutive subsequences. t_1 has four different cluster labels. The entropy for t_3 is $-\left[\frac{4}{4} \times \log(\frac{4}{4})\right] = 0$. The entropy for t_2 is $-\left[\frac{1}{4} \times \log(\frac{1}{4}) + \frac{3}{4} \times \log(\frac{3}{4})\right] = 0.56$ and entropy for t_1 is $-\left[\frac{1}{4} \times \log(\frac{1}{4}) + \frac{1}{4} \times \log(\frac{1}{4}) + \frac{1}{4} \times \log(\frac{1}{4}) + \frac{1}{4} \times \log(\frac{1}{4})\right] = 1.38$. Therefore, as per the entropy measure, t_3 is most stable and t_1 is the most unstable.

3.4.2 Atypical pattern identification

It has been a tradition to identify atypical patterns using brute force method, which is rather computationally expensive in case of large time series data. We propose a graph based analytical approach to discover atypical patterns from entire time series dataset.

The key to identify atypical pattern is to examine the frequency of occurrence of the symbol and the degree of vertex in graph G . Degree of a vertex capture the number of temporally dependent unique patterns. The vertex with fewer degrees has only a few temporally dependent patterns while a higher degree vertex signify that it has many temporally dependent patterns. As a consequence, we isolate the lower degree vertices based on the threshold value vd and the representative patterns of the isolated vertices are considered as the atypical patterns. The value of vd is decided on the basis of degree distribution of vertices.

3.5 Experiments

In this section we discuss about the datasets used, the experimental design, the evaluation measures used for comparative analysis of clustering results obtained from proposed quasi clique technique and the baseline methods.

3.5.1 Data description

Two residential building dataset from smart grids are used in our experimental study- London household data discussed in section 1.1.1, Ausgrid solar homes electricity data discussed in section 1.1.1.

The data pre-processing part include- missing value adjustment, duplicate removal and normalization. Linear interpolation is used in the London households dataset to impute the missing values. The Ausgrid electricity data do not have missing values but it has duplicates which is replaced by the mean values. The real valued half-hourly data are z-score normalized before performing the experiments.

3.5.2 Experimental design

1. Parameter sensitivity: The subsequence length (WL) and ψ are the two graph parameters in the proposed graph representation. Total number of subsequences in a time series is inversely proportional to the subsequence length. Very long subsequence can cause information loss and very short length subsequences might fail to capture the common patterns leading to large number of clusters. To obtain the best WL (WL= 2l), we vary it as- 1,2,4,6,8,12 and 24 hours.

ψ regulates the edge density of a subgraph and its value ranges from 0 to 1. The edge weights are normalized from 0 to 1, where 0 signifies no similarity between the corresponding vertex pair and 1 signifies maximum similarity. Greater the value of ψ , more similar are the vertices of subgraph.

2. Planted quasi clique: To evaluate the reliability of proposed quasi clique clustering technique, we design a graph and implant maximal quasi cliques as subgraphs to it. The vertices of the subgraph form a single cluster which are assumed as the true labels. The proposed clustering technique is applied on the designed graph which then form the predicted labels. The normalized mutual information (NMI) is computed between the actual and the predicted labels, discussed in section.3.5.3.
3. Cluster quality: We apply the proposed quasi-clique clustering algorithm on the directed weighted graph obtained from each WL and determine the best cluster set using the cluster quality measures. At this stage, we compare the cluster quality obtained from the proposed clustering technique with the existing state-of-the-art methods. The cluster quality measures used for comparison with the state-of-the-art methods are: Davies Bouldin (DB) and Edge Density (ED) which is discussed in section 3.5.3. The state-of-the-art clustering techniques is discussed in section 3.5.4.

Statistical test: The Friedman test is carried out to check whether the proposed method or any of the state-of-art clustering technique perform better than others or not. As each clustering method is evaluated on multiple WL, the Friedman test is suitable for comparisons across multiple state-of-the-art clustering methods. The rejection of null-hypothesis in Friedman test signify that the cluster quality results obtained from all the clustering methods are not same. On rejection of null hypothesis of Friedman test, the post hoc Nemenyi test [155] is carried out that is useful for pairwise comparison of the proposed quasi clique clustering technique with the existing state-of-the art clustering.

4. Cluster characterization: The cluster characterization is performed on the

CHAPTER 3. CLUSTERING TIME SERIES SUBSEQUENCES USING DIRECTED GRAPH

cluster labels obtained from the best graph representation, decided on the cluster quality measures: DB index and ED, as discussed in section. 3.5.3. Firstly we categorize the clusters based on the similarity in their shapes. Cluster centroid of each cluster is studied, which is the mean of load data values of the cluster samples.

Secondly, we analyse the time of occurrences of the patterns, identified by the cluster shapes, using the heat map. Finally we obtain the consecutively occurring patterns in the time series that belong to same cluster.

5. Atypical pattern identification: The Wilcoxon signed rank test is performed to check if there exist a significant difference in the frequency of occurrence of the atypical patterns with that of the normal patterns. The number of atypical patterns in the time series dataset is significantly lower than the normal patterns and therefore Wilcoxon signed rank test is considered suitable for the lower number of samples. We randomly choose the samples from the frequency of occurrence of the normal patterns and pair with frequency of occurrence of the atypical patterns, to perform the Wilcoxon signed rank test.

3.5.3 Evaluation metrics

Below we discuss the evaluation metrics and the statistical tests used in the experiments.

Normalized mutual information (NMI) : This is the measure of normalized mutual dependencies between the ground truth information (A) and the predicted information (P), computed as

$$NMI(A, P) = \frac{2 \times I(A, P)}{H(A) + H(P)} \quad (3.8)$$

where, $H(\cdot)$ is the Shannon's entropy measure [154] and $I(\cdot)$ is the mutual information between the A and P .

Davies Bouldin index (DB) : DB index is the ratio of the intracluster to inter-cluster distances. It is given as

$$DB = \frac{1}{k} \sum_{i=1}^k \max_{j \neq i} \left(\frac{\text{intra}(s_i) + \text{intra}(s_j)}{\text{inter}(x_i, x_j)} \right) \quad (3.9)$$

where, $\text{intra}(s_i) = \frac{1}{n} \sum_{p_a \in s_i} (d(p_a, x_i))$, is the intracluster distance formed by the cluster s_i with x_i as the cluster centroid and $\text{inter}(x_i, x_j) = d(x_i, x_j)$, is the intercluster distance between the cluster centroids x_i and x_j . Lower values of DB index indicates

3.5. EXPERIMENTS

better clustering. The minimum value of DB index is 0.

Edge density (ED): Edge density of a graph is the mean of edge densities of subgraphs obtained.

$$ED(G) = \text{mean}(ed(s_i)) \forall s_i \in S_G \quad (3.10)$$

For e_{s_i} number of edges, $|v_{s_i}|$ number of vertices and, $|v_{s_i}| \times (|v_{s_i}| - 1)$ set of all possible edges in the subgraph s_i , the $ed(s_i) = \frac{|e_{s_i}|}{|v_{s_i}| \times (|v_{s_i}| - 1)}$. The ED ranges from 0 to 1. Higher value of ED denotes dense subgraph formation.

Accuracy: Accuracy[64] is a measure of closeness of the predicted labels to the true labels, computed as:

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn} \quad (3.11)$$

where tp is the number of true positives, tn is the number of true negatives, fp is the number of false positives and fn is the number of false negatives. The higher accuracy values imply that the predicted labels are closer to the true labels.

Statistical tests

Friedman test : This is a non-parametric statistical test which finds if the proposed clustering method or any of the state-of-the-art clustering techniques is better than that of the others [155]. It computes the chi-square test statistic as:

$$\chi_F^2 = \frac{12 \times |WL|}{u(u+1)} \left[\sum_{j=1}^u R_j^2 - \frac{u(u+1)^2}{4} \right] \quad (3.12)$$

where $|WL|$ is the number of different subsequence length used, u is the number of different clustering method used for evaluation and the average rank $R_j = \frac{1}{|WL|} \sum_{i=1}^{|WL|} r_i^j$ where r_i^j is rank of j th clustering technique for the i th WL. According to the null hypothesis, all the clustering techniques have the same performance if the p-value obtained from the corresponding χ_F^2 is no less than the assumed significance level.

On rejection of null hypothesis of Friedman test, we proceed with post hoc Nemenyi test. The test considers that the performance of proposed method is significantly different from the other state-of-the-art clustering methods if, $|R_1 - R_j| > \alpha / \sqrt{2} \times \sqrt{u(u+1)/(6 \times |WL|)}$, where α is the significance level [155], R_1 is the rank of proposed clustering and $\{R_j\}_{j=1}^{u-1}$ is the rank of other methods. The

CHAPTER 3. CLUSTERING TIME SERIES SUBSEQUENCES USING DIRECTED GRAPH

p-values for the above statistical test is obtained, which if less than α , denote that the proposed method significantly differ from the compared method [156].

Wilcoxon signed rank test : This is non-parametric test that finds if there exist a significance difference between the paired values [157]. According to the null hypothesis H_0 : no significant difference exist because, the median of differences is zero and the alternate hypothesis H_1 : the median of differences is positive at $\alpha = 0.05$. The test statistic for n paired samples is $W = \min(W^+, W^-)$. The W^+ is the sum of positive ranks, given as $W^+ = \sum_{i \in n} r_i$, where r_i is the rank assigned to i th pair which has positive difference value and W^- is the sum of negative ranks, given as $W^- = \sum_{j \in n} r_j$, where r_j is the rank assigned to j th pair which has negative difference value. The hypothesis H_0 is rejected if $W \leq W_{crit}$, where W_{crit} is the critical value of W obtained from the critical value table.

3.5.4 Comparative methods

Because this chapter proposes a graph based clustering technique, we choose to compare it with the commonly used existing graph clustering algorithms as discussed below.

1. Fast greedy (FG): This is a greedy approach [158] in which the edges are added to a complete disconnected graph to form the communities and each time the modularity score is computed. This process continues till all the communities are merged, resulting in just one community. The cluster labels with highest modularity score is returned.
2. Minimum spanning tree based vertex betweenness: We obtain the minimum spanning tree (MST) of the given graph and the vertices are removed from MST in order of their betweenness score and each time the modularity score is computed. The community formed from the cut vertex with highest modularity score is returned.
3. Girvan Newman edge betweenness (EB): The edge with highest edge betweenness score is removed from the given graph and the betweenness score is computed for the remaining edges. The process is repeated until no edge remains [159]. Cluster labels with the highest modularity score is returned.
4. Multilevel: At each iteration, the vertices are merged to form a community with maximum modularity score. The process is repeated until all the vertices find their communities or they form a community on its own [160].
5. Infomap: This clustering creates the community structure by minimizing the description length of a random walk trajectory [161].

3.6. RESULTS AND ANALYSIS

Table 3.1: Actual and predicted cluster labels of the vertices, in the planted quasi clique to a time series dataset.

Clusters	Actual	Predicted	
		$\psi = 0.2$	$\psi = 0.25$
1	bce, cce, dce, ebe, ece	bce, cce, dce, ebe, ece	bce, cce, dce, ebe, ece
2	ccc, ccd, dbce, dcd, ede	ccc, ccd, dbce, dcd, ede	ccc, ccd, dbce, dcd, ede
3	bcc, bcd, cbe, cde, dde	bcc, bcd, cbe, cde, dde	bcc, bcd, cbe, cde, dde
4	dcc, eae, ecc, ecd, eee	dcc, eae, ecc, ecd, eee	dcc, eae, ecc, ecd, eee
NMI	–	1	1

Table 3.2: Cluster quality results and the p-values obtained from post-hoc Nemenyi test, between pairwise comparisons of state-of-art clustering techniques with the proposed method. The best cluster quality and the paired comparisons which vary significantly are highlighted in bold.

Method	Cluster quality				Nemenyi Test			
	London		Ausgrid		London		Ausgrid	
	DB	ED	DB	ED	DB	ED	DB	ED
FG	3.78	0.83	3.5	0.87	0.1	0.001	0.1	0.001
MST-VB	2.48	0.81	2.45	0.92	0.98	0.0002	0.6	0.02
EB	1.89	0.92	1.73	0.96	0.9	0.4	0.98	0.03
Multilevel	2.9	0.85	5.17	0.9	0.02	0.007	0.004	0.09
Infomap	3.7	0.89	2.91	0.9	0.2	0.61	0.02	0.73
Proposed	1.6	0.99	1.06	1	–	–	–	–

3.6 Results and Analysis

In this section, we discuss the results corresponding to the experimental design discussed in section 3.5.2.

3.6.1 Planted quasi clique

The experiments are performed on a time series dataset T with $m = 30$ different time series samples extracted randomly from the Ausgrid dataset. For illustrative purpose, we assume smaller length of time series, $n = 80$ and $l = 8$. We obtain a graph representation from the time series dataset T , with $|v| = 20$. The graph consists of the manually implanted maximal quasi cliques each of cluster size 5. The graph vertices and the actual and predicted labels are shown in Table 3.1. We obtain the predicted labels using the proposed quasi-clique approach for $\psi = \{0.2, 0.25\}$.

CHAPTER 3. CLUSTERING TIME SERIES SUBSEQUENCES USING DIRECTED GRAPH

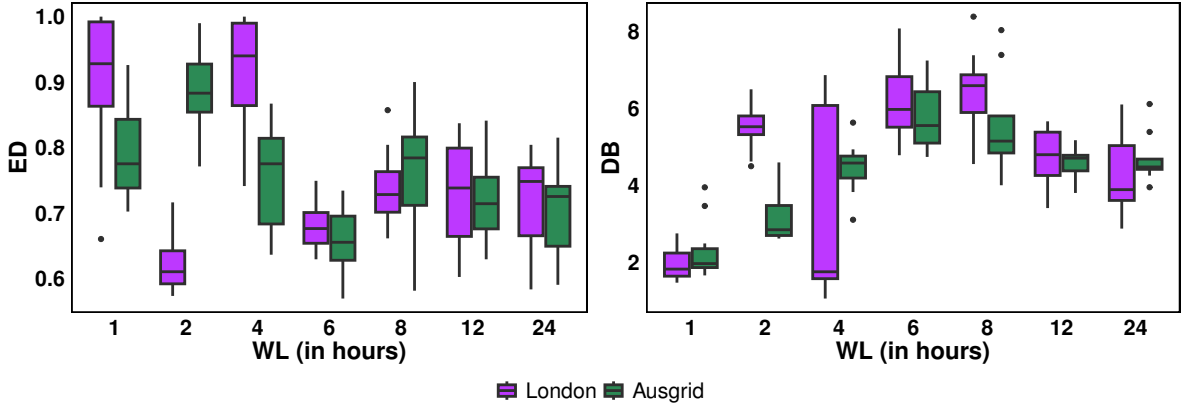


Figure 3.3: Cluster quality measures, DB index and the ED for London and Ausgrid datasets achieved using the proposed clustering method.

The obtained NMI value 1, indicate that the proposed quasi-clique successfully extract the maximal quasi cliques for the given values of ψ .

3.6.2 Cluster quality

In Table 3.2, we report the DB index and ED of the proposed clustering technique and other existing graph based clustering techniques- FG, MST-VB, EB, Multilevel and Infomap. In case of the proposed clustering technique, we report the lowest DB and highest ED, out of those obtained from varying WLs and ψ . While for the other graph based clustering approaches, we report the lowest DB and highest ED, out of those obtained from varying WLs. It is important to note that the proposed method shows a significant 15% and 38.7% improvement over the best performing baseline method EB when compared in terms of DB index of London and Ausgrid datasets respectively.

The graph clustering techniques like FG and Multilevel decide the cluster labels based on the best modularity score, hence they are unable to capture the similarity between the vertices resulting to a higher DB index. Also the high ED values in FG, MST-VB, Multilevel and Infomap, signify that although they can form denser subgraphs but the similarity between the vertices is poor.

The London dataset show highest ED of 0.99 because of the dense subgraphs obtained for $\psi = 0.45$ and WL=2 hours. Similarly, the Ausgrid dataset show highest ED of 1 for $\psi = 0.75$ and WL= 4 hours. Although the DB index for WL 1 and 2 is lower in both the datasets, it shows a poor cluster distribution which leads to large number of clusters.

The distribution of DB index and the ED values with changing ψ is depicted using boxplot in Fig. 3.3. To identify the best cluster set, we study the DB and ED

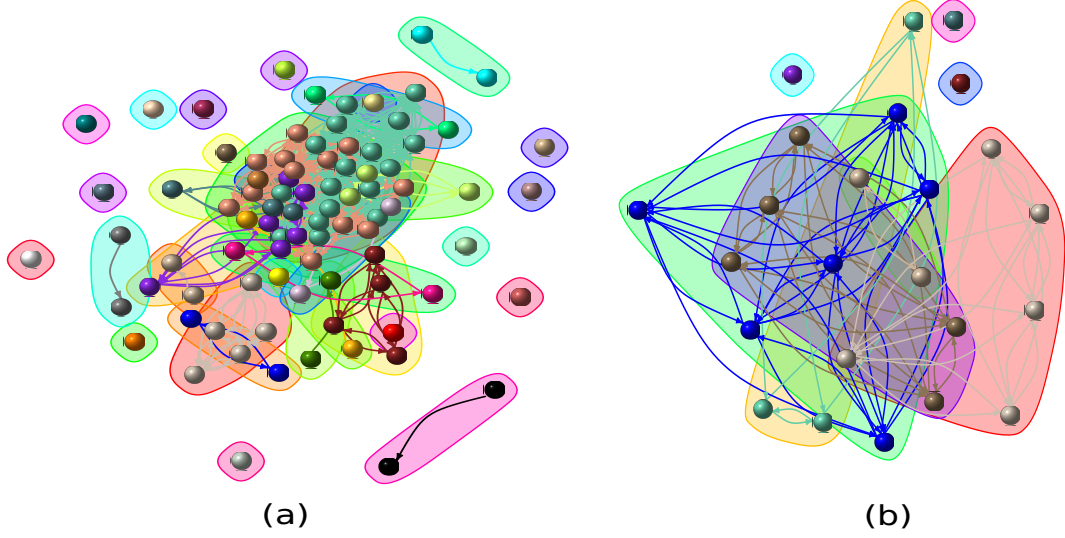


Figure 3.4: Graph structure obtained from London and Ausgrid dataset for (a) WL 12 hours and $\beta = 0.5$ and (b) WL 4 hours and $\beta = 0.65$. Clustered regions highlight the vertices and edges using different colors.

values and also study the cluster distribution.

For both the London and Ausgrid datasets, WL 6 show higher DB index. The DB index for WL 4 and 12 in case of London dataset, is similar and both of them show similar results. However the maximum DB index in case of WL 12 is less than that of WL 4. Based on this study, we obtain WL 12 and $\psi = 0.5$ as the best cluster distribution and best graph representation for London dataset.

In case of Ausgrid dataset in Fig. 3.3, the WLs 2, 6, 8 show higher DB index but WL 4 show lower DB index. Based on the DB index and cluster distribution, WL 4 and $\psi = 0.65$ is the best cluster set and best graph representation for the Ausgrid dataset.

The Friedman $\chi^2 = 16.5$, p-value = 0.005 for the DB index and Friedman $\chi^2 = 26.837$, p-value = $6.137e - 05$ for the ED in case of London dataset. Therefore both the p-values show significant difference at $p < 0.01$, rejecting the null hypothesis. Therefore, we proceed with the post-hoc Nemenyi test, and obtain the pair of clustering methods that differ significantly. In Table 3.2, we report the p-values obtained from post-hoc Nemenyi test. For the London dataset, the DB index of Multilevel clustering show p-value= 0.02 ($p < 0.05$). FG, MST-VB and Multilevel clustering differ significantly from the proposed method in terms of ED with p-value of 0.001, 0.0002 and 0.007 respectively ($p < 0.05$).

The Friedman $\chi^2 = 19.8$, p-value = 0.001 for the DB index and Friedman

CHAPTER 3. CLUSTERING TIME SERIES SUBSEQUENCES USING DIRECTED GRAPH

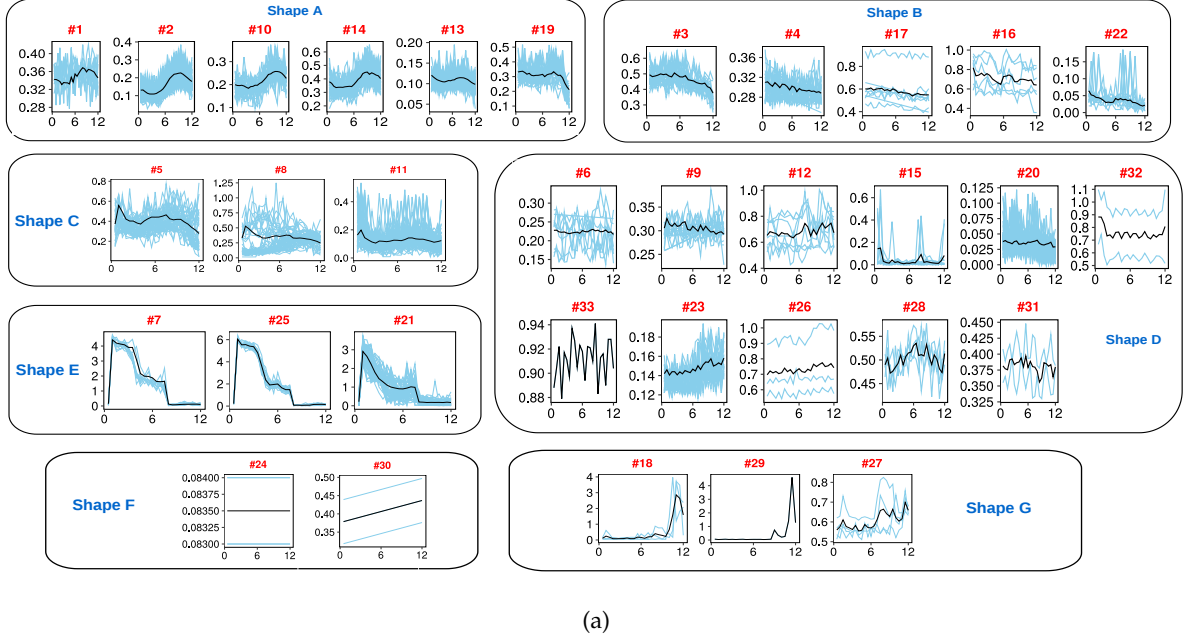


Figure 3.5: Subfigure (a) shows the clusters, grouped into shapes A through shape G for the London dataset with a cluster centroid (in black). x-axis shows the time units of the subsequence (in hours) and y-axis is the energy consumed (in kWh).

$\chi^2 = 19.3$, p-value = 0.001 for the ED in case of Ausgrid dataset. Therefore both the p-values show significant difference at $p < 0.01$, rejecting the null hypothesis. In Table 3.2, we report the p-values obtained from post-hoc Nemenyi test for the Ausgrid dataset. The DB index of Multilevel and Infomap clustering show p-values 0.004 and 0.02 respectively ($p < 0.05$). FG, MST-VB, and EB clustering differ significantly from the proposed method in terms of ED with p-value of 0.001, 0.02 and 0.03 respectively ($p < 0.05$).

As shown in Table. 3.2, for London dataset, the proposed method perform better than Multilevel in terms of both DB index and ED, and better than FG and MST-VB in terms of ED. In case of Ausgrid, the proposed method perform better than Multilevel and Infomap in terms of DB index and better than FG, MST-VB and EB in terms of ED.

3.6.3 Cluster characterization

In this section we characterize the clusters obtained using the proposed method for each data set.

1. London : Fig. 3.4 (a) shows the graph representation of London dataset. The

3.6. RESULTS AND ANALYSIS

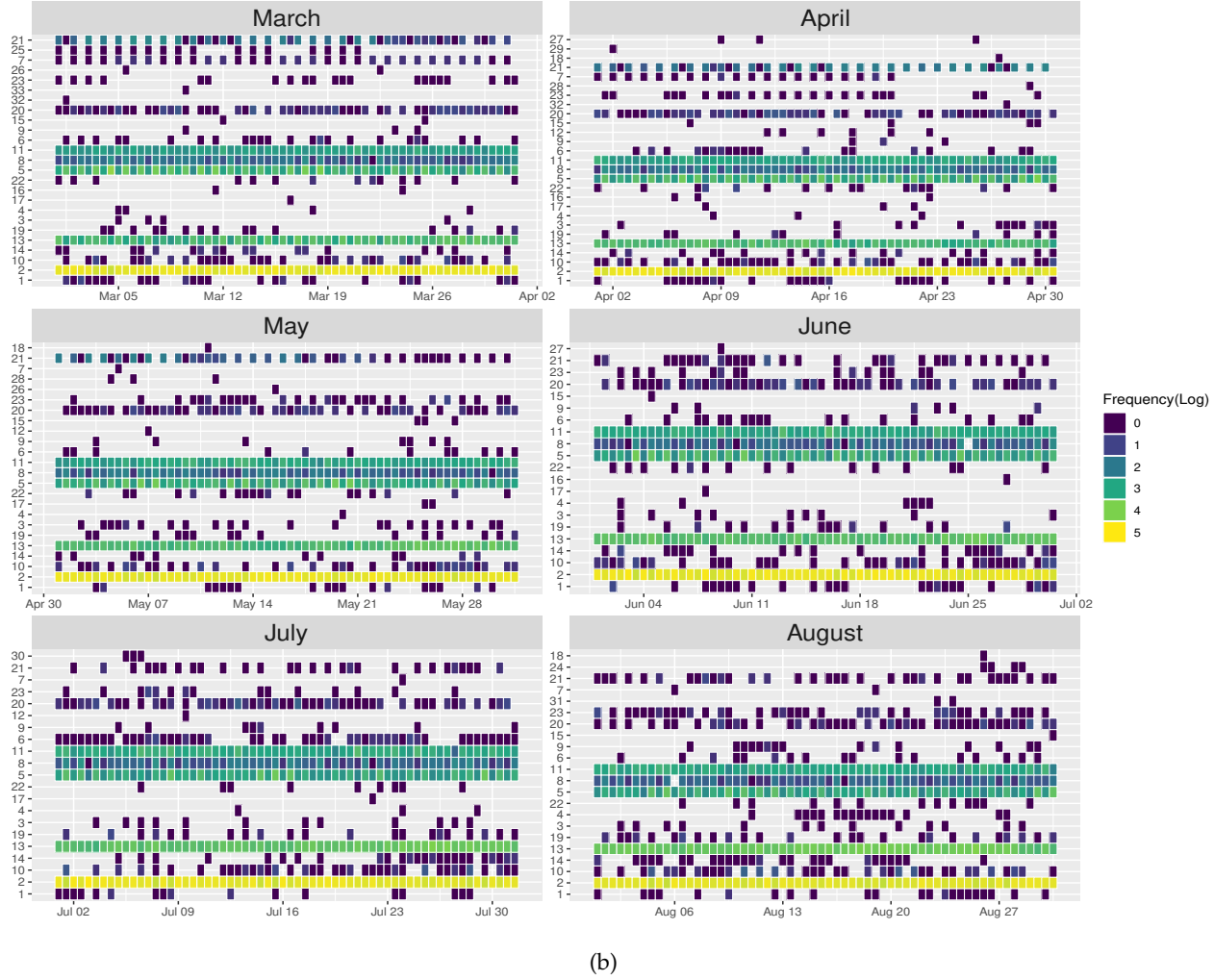


Figure 3.5: Subfigure (b) shows the month wise occurrence of cluster labels in London dataset using heatmaps where x-axis shows the time, y-axis shows cluster labels. Total number of subsequences corresponding to a given time and belonging to a particular cluster is denoted using different colors in log scale.

vertices of each cluster are marked with same colors. Out of all the cluster obtained, 17 are densely connected subgraphs of size 2 or more and 16 are found to be isolated vertices. All further experiments are carried out using the graph shown in Fig. 3.4 (a) for London dataset.

Fig. 3.5 (a) shows the cluster samples and cluster centroids of the respective clusters (in bold face). Based on the similarity in shape, the clusters are grouped as the shapes A-G. Each group may be considered as representation

CHAPTER 3. CLUSTERING TIME SERIES SUBSEQUENCES USING DIRECTED GRAPH

of a subsequence pattern.

- Shape A: The cluster centroids of #1, #2, #10, #14 are denoted as shape A. The cluster centroid shows single peak of sinusoidal shape. Cluster centroid of #13, #19 also shows sinusoidal like behavior however, a sharp fall is observed towards the end of the subsequence. 73% of the subsequences in #1 belongs to affluent category.
- Shape B: The cluster centroids of #3, #4, #17, #16, #22 are denoted as shape B. These cluster centroids show a downfall in consumption during the last 6 hours. Cluster #6 occurs in affluent and comfortable category of buildings while #4 is observed only in affluent category.
- Shape C: The cluster centroids of #5, #8, #11 are denoted as shape C. They show a sharp peak during first 2 – 3 hours of the window.
- Shape D : The cluster centroids of #6, #9, #12, #15, #20, #32, #33, #23, #26, #28, #31 are denoted as shape D. These clusters show multimodal peaks. No steep rise or fall in pattern is observed. Cluster #12 occurs in the affluent category buildings only while #31, #32 is observed in only one building of affluent category.
- Shape E: The cluster centroid profiles of #7, #25, #21 are denoted as shape E which have flat top peaks during the first 6 hours of window. All the subsequences in cluster #25 belong to only one building of affluent category.
- Shape F: The cluster centroids of #24, #30 are denoted as shape F which show a constant consumption pattern. All the cluster samples in #24 belong to a building in adversity category.
- Shape G : The cluster centroids of #18, #29, #27 are denoted as shape G which have a sharp peak at the end of the window. Cluster #27 occurs in affluent category buildings only.

We next analyze the occurrence of each of the patterns identified by the cluster shapes. In Fig. 3.5 (b) each heat map display the month wise distribution of the cluster labels. The columns of each heatmap show the 12 hour time segments while each row represent a cluster label. The cluster label which is not found during a particular month, is omitted from the heat map.

Clusters #2, #13 with shape A as shown in Fig. 3.5 (b) occur in every month in every time interval with maximum frequency. Clusters #1 with shape A is rather an infrequent pattern and does not occur regularly as compared to #2, #13, #5, #11, #8, except for month of May where cluster #1 is observed once in every 2 days. Other clusters with shape A, namely #10, #14 and #19

3.6. RESULTS AND ANALYSIS

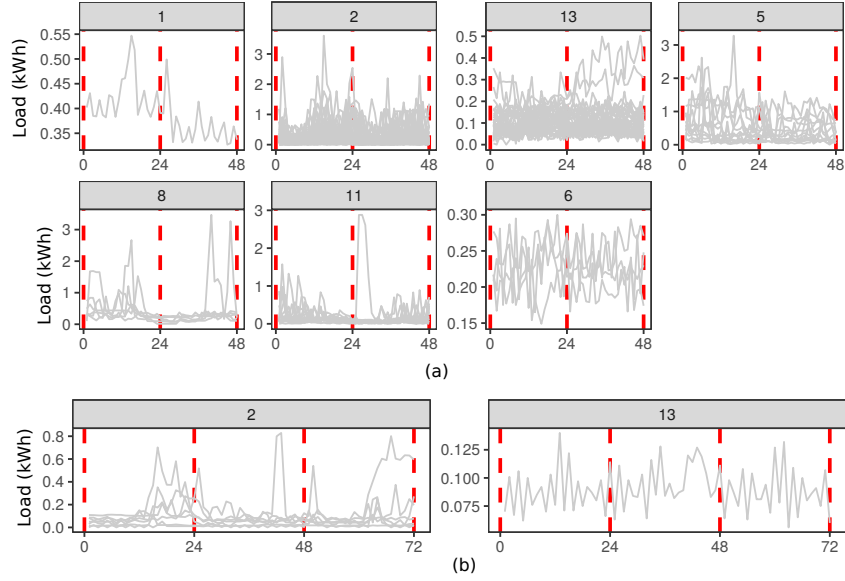


Figure 3.6: Subfigures show the consecutive occurrence of patterns that fall into the same cluster for (a) two subsequent time segments and (b) three subsequent time segments in London dataset. The red dashed lines denote the time segment boundaries. The cluster numbers are shown in the panel of each plot.

though occur in every month but no typical pattern of occurrence is observed. Majority of the subsequences in cluster #3 with shape B has occurrence during the month of May and none of the occurrences is observed during the first day of any month. Cluster #16 with shape B is observed during March, April and June. Cluster #17 with shape B occurs in all months except August and for all months it mostly occurs during the last four days of the week. Cluster #22 and #4 with shape B are observed to occur very infrequently in all the months of March through August. Cluster #5, #8, #11 with shape C occur in every month and in every subsequence. Cluster #9 with shape D mostly occurs in July and August. Cluster #12 with shape D occurs during April, May and July. No occurrence of cluster #15 with shape D is observed during July and only a single occurrence observed in August. Cluster #20 and #23 are observed to occur in majority of the days in every month while cluster #6 occurs rarely in a month. Clusters #26, #28, #31 #32, #33 are only rarely observed to occur in the month of March and May, in the month of April and May, in August, in the month of March and April, and in March respectively. They are not observed during June and July. Cluster #7 with shape E is observed in every alternate subsequence during March and the first 20 days in April. Cluster #21 is observed in every month while cluster

CHAPTER 3. CLUSTERING TIME SERIES SUBSEQUENCES USING DIRECTED GRAPH

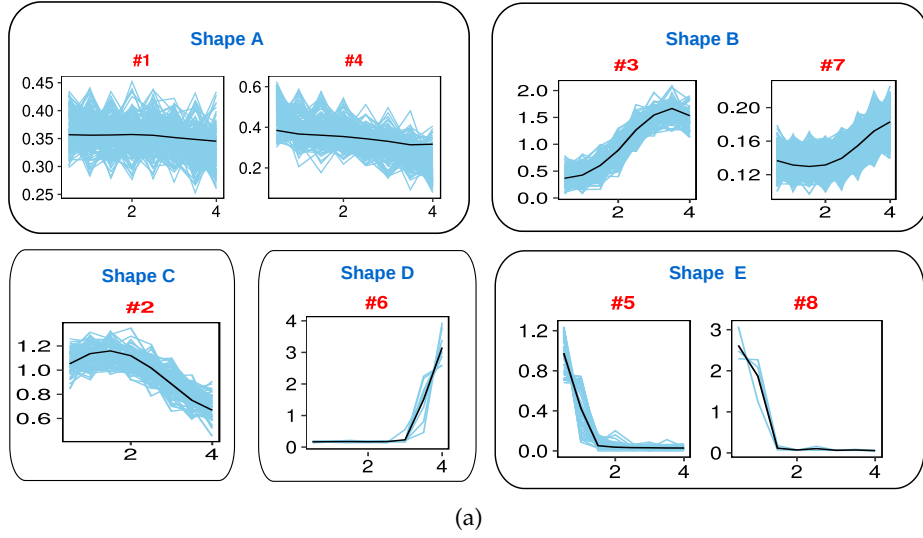


Figure 3.7: Subfigure (a) shows the respective cluster samples in *skyblue*, grouped into shapes A through shape E for Ausgrid dataset and the cluster centroid is shown in *black* lines. x-axis shows time units of subsequence (in hours) and y-axis is the energy consumed (in kWh).

#25 is observed only during March. Cluster #24, #30 with shape F occur only in July and August. Cluster #27 with shape G occurs on Monday, Tuesday and Thursday. #18 and #29 are observed only once in a month during April, May and August.

We finally analyze the consecutive occurrence of patterns that fall under same cluster for London dataset. Fig. 3.6 shows the occurrence of similar patterns that belong to same cluster in two and three successive time segments. As shown in the figure, the cluster shapes are clearly repeated in successive time segments for clusters #1, #2, #13 with shape A, #5, #8, #11 with shape C and #6 with shape D.

2. Ausgrid: Fig. 3.4 (b) shows the best graph representation obtained from the proposed clustering technique in case of Ausgrid dataset. Out of all the clusters obtained, 5 are densely connected subgraph of size 2 or more and 3 are found to be isolated vertices. All further experiments on Ausgrid dataset are carried out using the graph shown in Fig. 3.4 (b).

Fig. 3.7 (a) shows the cluster samples and cluster centroids of the respective clusters (in bold face). Based on the similarity of shape, the clusters are grouped as the shapes A-E.

- Shape A: Cluster centroid of #1, #4 is denoted as shape A. These cluster

3.6. RESULTS AND ANALYSIS

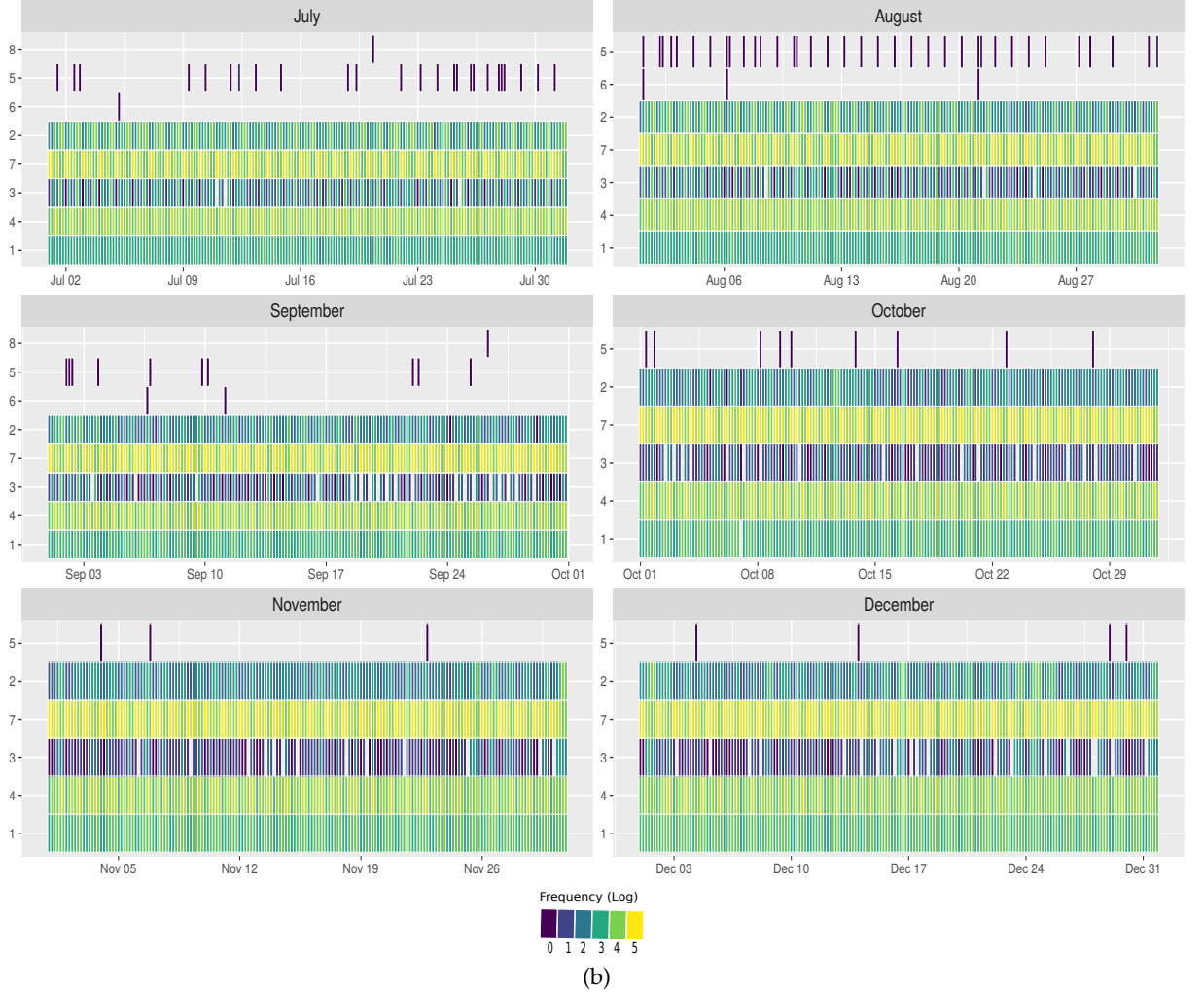


Figure 3.7: Subfigure (b) shows the month wise occurrences of cluster labels of Ausgrid dataset using heatmap where x-axis shows the time, y-axis is cluster label. The total number of subsequences corresponding to a given time and belonging to a particular cluster is denoted using different colors in log scale.

centroids show a decreasing trend. The rate of decrease in #4 is greater than #1.

- Shape *B* and *C*: The cluster centroid of #3, #7 and #2 are denoted as shapes *B* and *C* respectively. These show sinusoidal patterns where cluster #2 has peak during the first 2 hours, cluster #3 has peak during the last 2 hours and cluster #7 has gradual increase during the last hour.
- Shape *D* and *E* : The cluster centroid of #6 and #5, #8 are denoted as shapes *D* and *E* respectively. While cluster #6 shows a flat consumption

CHAPTER 3. CLUSTERING TIME SERIES SUBSEQUENCES USING DIRECTED GRAPH

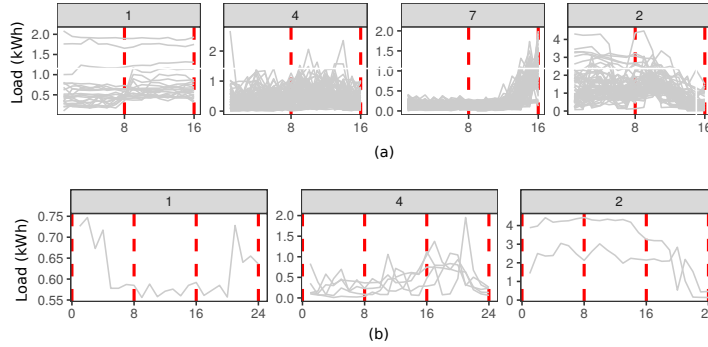


Figure 3.8: (a) Consecutive occurrence of patterns that fall into the same cluster for a) two subsequent time segments and b) three subsequent time segments in Ausgrid dataset. The red dashed lines denote the time segment boundaries. The cluster numbers are shown in the panel of each plot.

pattern at the initial hours followed by a sharp peak, clusters #5, and #8 shows a sharp peak at the initial hours followed by a flat consumption pattern.

We next analyze the occurrence of each of the patterns as identified by the cluster shapes. In Fig. 3.7 (b) each heat map displays the month wise distribution of the cluster labels. The Ausgrid dataset is experimented on dataset over an year, the heat maps for the months of July through December are shown. The columns of each heatmap show the 8 hour time interval while each row represents a cluster label. The cluster label which is not found during a particular month, is omitted from the heat map. Cluster #1 and #4 with shape A, #7 with shape B and #2 with shape C, as shown in Fig. 3.7 (b) occur in every month and in every time interval with clusters #7 and #4 showing high frequency. Cluster #3 with shape B is also observed to occur in majority of the time intervals every month. Clusters #6 with shape D and #5 and #8 with shape E are rarely observed and do not demonstrate any pattern of occurrence.

We finally analyze the consecutive occurrences of patterns that fall under same cluster in Ausgrid dataset. Fig. 3.8 shows the occurrences of similar patterns that belong to same cluster in two and three successive time segments. As shown in the figure, the cluster shapes are clearly repeated in successive time segments for clusters #1, #4 with shape A, #7 with shape B and #2 with shape C.

3.6.4 Application to DR programs

In this section we discuss the results obtained from the application areas as detailed in section 3.6.4.

3.6. RESULTS AND ANALYSIS

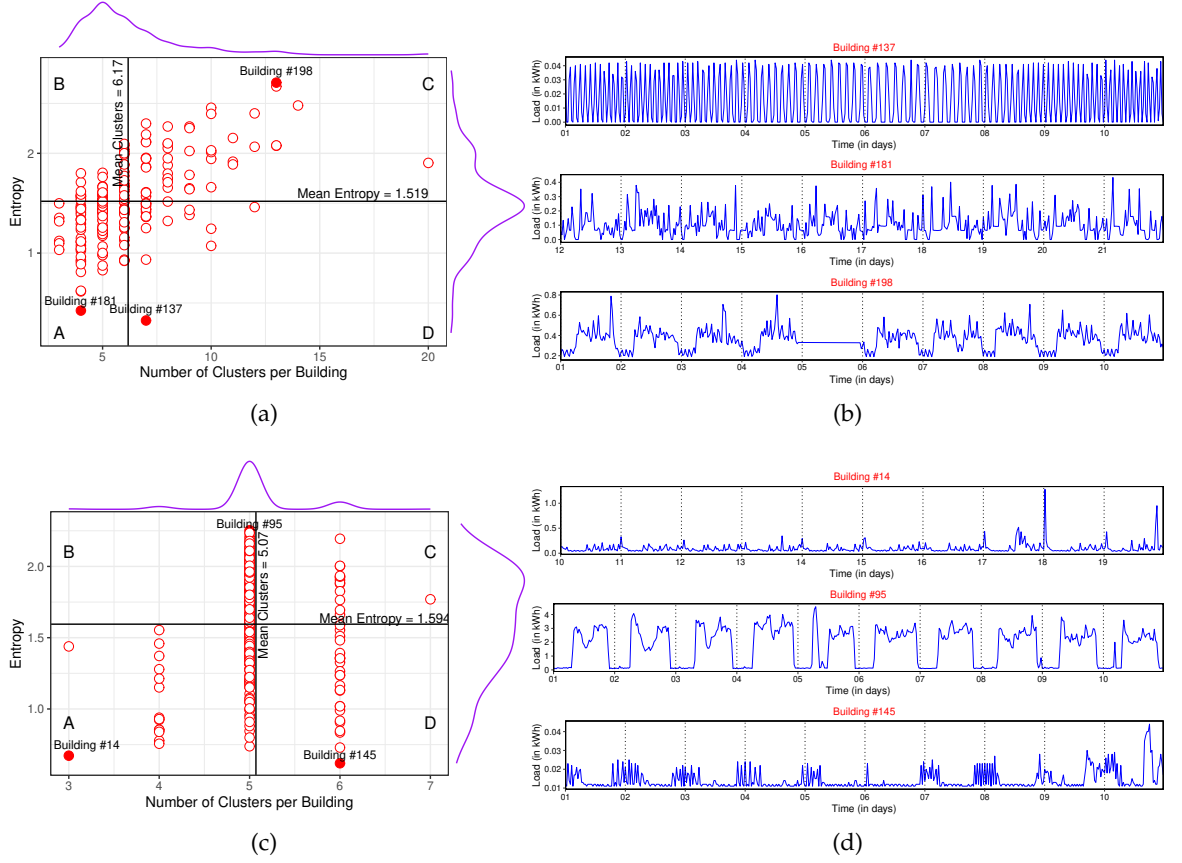


Figure 3.9: Subfigures (a) and (c) show the entropy values (along Y-axis) of the buildings against the number of clusters per building (along X-axis) for (a) London and (c) Ausgrid datasets. The density plots for entropy values and number of clusters per building are shown along the right and top axis respectively. The thick dotted lines show the mean entropy and mean number of clusters that break the plots into four regions: A) low entropy and low number of clusters per building, B) high entropy and low number of clusters per building, C) high entropy and high number of clusters per building, D) low entropy and high number of clusters per building. In subfigure (a) the buildings #137, #181 and #198 are highlighted for their typical positions while in subfigure (c) the buildings #14, #95, #145, are highlighted for their typical positions. The load data in subfigure (b) show the pattern for buildings #137, #181 and #198 of London grid dataset and in subfigure (d) show the pattern for buildings #14, #95, #145 of Ausgrid dataset.

CHAPTER 3. CLUSTERING TIME SERIES SUBSEQUENCES USING DIRECTED GRAPH

Stability

For application to DR programs we first identify the buildings appropriate for the DR program based on their stability. The stability score measured using entropy of each building is computed using Equation 3.7.

1. London: In Fig. 3.9 (a) we plot the entropy values of the buildings against the number of clusters per building. The right and top axes of Fig. 3.9 (a) shows the distribution of the entropy and number of clusters per building in the London dataset. While the second distribution shows a peak around the mean, the entropy shows a positively skewed distribution with mean as 1.519 and median as 1.495. The mean entropy and mean number of clusters partition the plot into four regions, where all the building have: A) below average entropy and below average number of clusters per building, B) above average entropy and below average number of clusters per building, C) above average entropy and above average number of clusters per building and D) below average entropy and above average number of clusters per building. Though the choice of how much stability the utilities need is application and data dependent, the regions A, D and C are of particular interest. The regions A and D may be considered as a safe choice for stability as they denote the region where the successively occurring subsequences have fewer or infrequent changes in patterns. On the contrary, the region C may be considered as volatile region where the successively occurring subsequences frequently change their patterns. It is observed that in the London dataset nearly 46% buildings belong to region A while 22% buildings belong to the region C. Further, considering all entropy values greater than equal to 1 as high, we choose to filter the buildings having lower than 1 entropy and below average number of clusters per building as stable. In such case, we observe that in the London data set only 8% buildings have entropy lower than 1 whereas 7% buildings can be labeled as stable.

In Fig. 3.9 (a) we have highlighted the buildings with globally minimum entropy (building #137), with globally maximum entropy (building #198) and with minimum entropy within region A (building #181) and have studied their load patterns in further details. In Fig. 3.9 (b) we report the consumption pattern for 10 days for the highlighted buildings with various stability. Building #198 with highest entropy value of 2.70 in Fig. 3.9 (a), depicts frequent change in patterns and mean values in temporally dependent subsequences. The subsequences of building #198 is distributed over 13 different clusters. On the other hand, building #137 with lowest entropy value of 0.32, depicts regular patterns during the 10 day period. The subsequences of building #137 is distributed over 7 clusters and 96% of the total subsequences belong

to cluster #2. Building #181 belongs to region A with entropy value of 0.423 while the subsequences of building #181 is distributed over only 4 clusters. In building #181 it is observed that 92.3% of the subsequences belong to the cluster #2 and remaining 7.2% belong to the cluster #11 with only exception of two subsequences. It is also noticed that in building #181 nearly 13% of time patterns belonging to clusters #2 and #11 occur in subsequent time steps. The majority of occurrences in a single cluster for building #137 and #181 signifies that the consumption patterns are stable and predictable in nature.

2. Ausgrid: Fig. 3.9 (c) shows the entropy values of the buildings against the number of clusters per building and the density plots for entropy and number of clusters per building. It is observed that in the Ausgrid dataset nearly 46% buildings belong to region A and D while 20% buildings belong to the region C. Further, we observe that in the Ausgrid data set only 4% buildings have entropy less than 1 and below average number of clusters per building that are labeled as stable. In Fig. 3.9 (d) we show the consumption pattern for 10 days for the buildings with globally minimum entropy (building #145), with globally maximum entropy (building #95) and with minimum entropy within region A (building #14) that are highlighted in Fig. 3.9 (c). As shown in Fig. 3.9 (c), building #95 shows highest entropy value of 2.25. Two highest peaks are observed upto day 6, but the pattern shapes differ each day. On the other plot, building #145 shows lowest entropy value of 0.61. The peaks are observed during the start and end of the day, except day 10 still shows some instability. 82% of the subsequences of building 145 belongs to cluster #7. Building #14 belongs to region A with entropy value as 0.672 while the subsequences of building #14 is distributed over only 3 clusters. Further, we found that over 84% of the subsequences of building #14 belong to cluster #7 and over 15% belong to cluster #4 with nearly 24% of time subsequent occurrences of patterns belonging to clusters #7 and #4 is observed. The majority of occurrences of subsequences in a single cluster for building #145 and #14 signifies that the consumption patterns are stable and predictable in nature.

Atypical pattern identification

1. London: Fig. 3.10 (a) shows some of the discovered atypical patterns when the assumed threshold value $vd = 4$. All the symbols shown in the figure belong to single sized cluster with maximum degree of 3. To highlight the uniqueness in the pattern, we have shown two subsequences before and after the occurrence of atypical pattern in Fig. 3.10 (a). Symbol dfe

CHAPTER 3. CLUSTERING TIME SERIES SUBSEQUENCES USING DIRECTED GRAPH

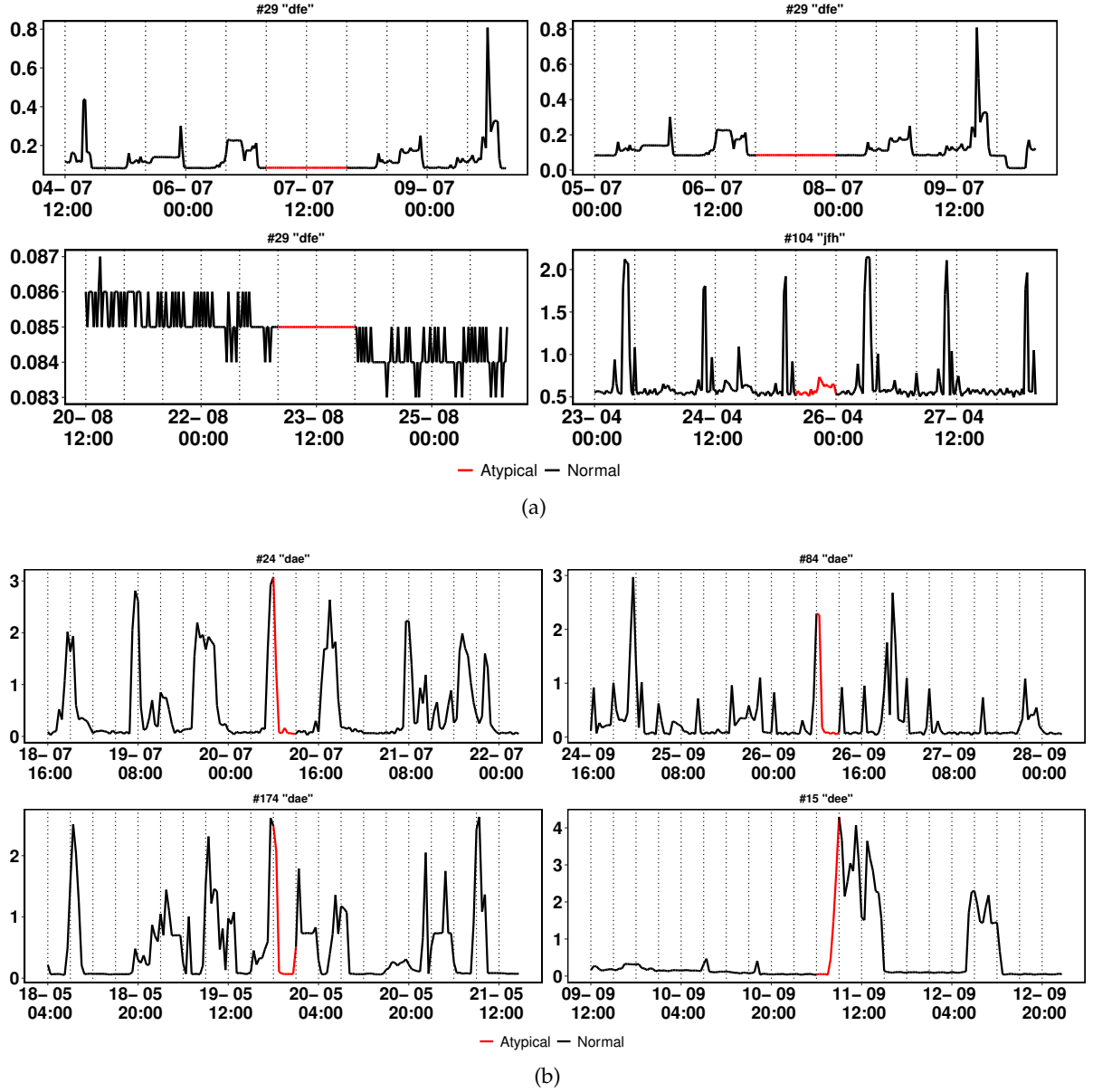


Figure 3.10: Atypical patterns from (a) London and (b) Ausgrid datasets. Each subsequence is bounded by dotted lines. The *black* represent normal subsequence and the *red* represent atypical pattern. x-axis shows date-month-time of occurrence and y-axis shows the load consumption values (in kWh). The building number and the symbol corresponding to the atypical pattern is shown at the top.

show a constant consumption and is observed in 4 different buildings with majority occurrences at 12 noon. The occurrence in building #29 is shown in Fig. 3.10 (a). Hence the discovery signify that those buildings have minimal consumption during 12 noon.

We perform the Wilcoxon signed rank test for $n = 30$ and the significance level $\alpha = 0.05$. From the critical value table, $W_{crit(n,\alpha)} = 137$. The test statistic W obtained is 19 which is significantly less than the W_{crit} and therefore we obtain a significant evidence that the frequency of occurrences of the atypical patterns is different from the frequency of occurrence of any randomly chosen normal pattern. Hence hypothesis H_0 is rejected in favor of H_1 . The p-value obtained is $9.3e - 10$.

2. Ausgrid: Fig. 3.10 (b) show the discovered atypical patterns when the assumed threshold value $vd = 8$. To highlight the uniqueness in the pattern, we have shown two subsequences before and after the occurrence of the atypical pattern. Cluster #8 has a vertex *dae*, with the lowest degree 6. Cluster #6 has vertex *dee* with degree 7. We observe that *dae* occur in three different buildings in the entire dataset. The atypical pattern *dee* show 8 occurrences and majority of the occurrences is observed at 4 a.m. The peak towards the end of the pattern in case of *dee* signify the occurrence of the atypical pattern during the early morning hours.

Since only two atypical patterns are detected from the Ausgrid dataset, we choose a smaller value of $n = 20$, for the Wilcoxon signed rank test. The significance level chosen is $\alpha = 0.05$. From the critical value table, $W_{crit(n,\alpha)} = 51$. The test statistic W obtained is 1 which is significantly less than the W_{crit} and therefore we obtain a significant evidence that the frequency of occurrences of the atypical patterns is different from the frequency of occurrence of normal patterns. Hence hypothesis H_0 is rejected in favor of H_1 . The p-value obtained is $9.5e - 7$.

Case study on labeled data

The buildings that follow consistent load patterns, are considered as stable customers that are suitable for the DR as well as energy efficient (EE) programs [152, 153]. The buildings showing sudden change in consumption patterns are less likely to meet the said objectives as sudden consumption change indicates the presence of uncertain factors that trigger the unpredictable load pattern. To show how the proposed framework can be applied for DR programs, we show a case study on labeled CER-IRISH dataset which is discussed in section 1.1.1. We consider the test and control groups of buildings as the true labels. The test set of

CHAPTER 3. CLUSTERING TIME SERIES SUBSEQUENCES USING DIRECTED GRAPH

buildings have a stable consumption behavior. We therefore use the proposed method to find the stable set of buildings.

We apply the proposed quasi-clique clustering approach with changing WLs to obtain the best cluster set using the cluster quality measures DB and ED discussed in section 3.5.3. Best cluster quality is observed for WL=1 hour, $\psi = 0.35$. We use the Shannon's entropy as the stability score to predict the buildings with highest k stability scores.

In Table 3.3, we report accuracy results for WL=1 hour where k varies from 10 to 35. For $k = 25$, the predicted results successfully match with that of the true labels with accuracy of 0.76. The highest accuracy is achieved for the minimum k value of 30. We compared the results with the k least stable buildings (last- k) that identify the building with maximum fluctuations as probable participants to DR programs.

We divide the dataset into 4 groups and evaluate the change in the consumption of test period over that of the benchmark period.

- Group 1: the actual buildings that belong to test group
- Group 2: the actual buildings that belong to control group
- Group 3: the predicted top-25 buildings
- Group 4: the buildings that belong to both actual and predicted test group in top-25.

In Fig. 3.11 (a), we illustrate the average day wise load consumption in the 4 groups. According to the CER-IRISH dataset, 5:00-7:00 p.m is considered as the peak time during which the DR operations are active ¹. As shown in Fig. 3.11 (a), Group 1 show a decrease in peak hour consumption, Group 2 do not show any significant decrease in the consumption during the peak hours of the test period.

¹<https://www.ucd.ie/issda/t4media/cer11080.pdf>

Table 3.3: The Table shows accuracy results for different values of k . The top- k signify the accuracy achieved for the buildings with highest k stability scores and the last- k signify the k least stability scores.

k	Hourly (WL=1 hour)		Daily		Weekly	
	top-k	last-k	top-k	last-k	top-k	last-k
10	0.58	0.30	0.57	0.32	0.57	0.32
15	0.58	0.23	0.58	0.23	0.58	0.23
20	0.69	0.16	0.67	0.14	0.67	0.14
25	0.76	0.16	0.76	0.16	0.76	0.16
30	0.83	0.21	0.8	0.23	0.82	0.21
35	0.83	0.30	0.82	0.32	0.82	0.30

3.6. RESULTS AND ANALYSIS

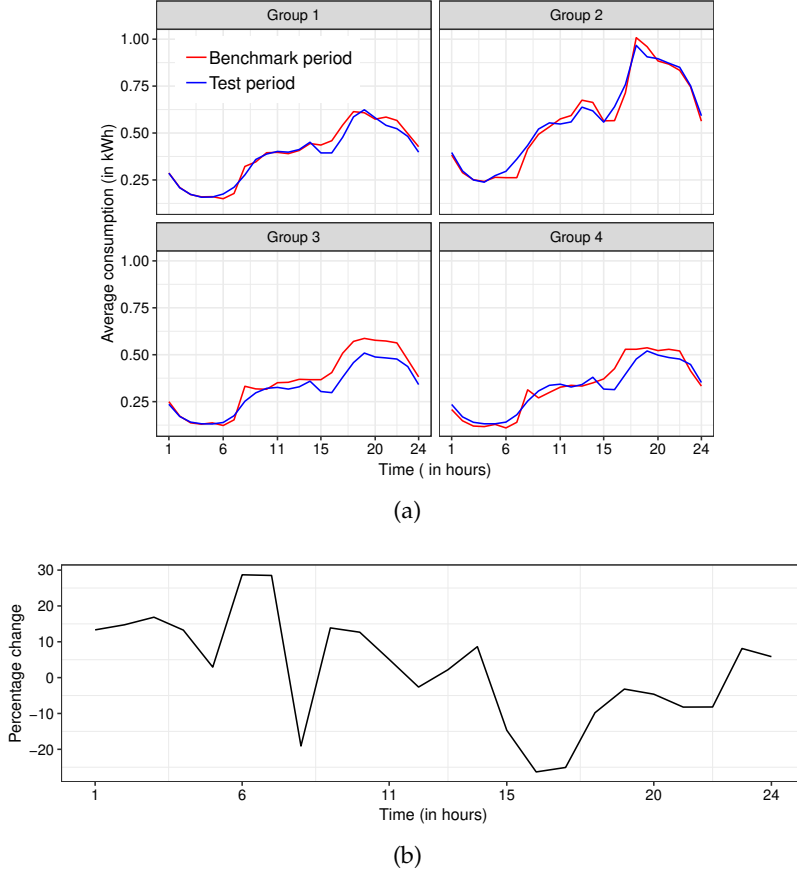


Figure 3.11: Subfigure (a) shows the average load consumption of the subsequences (WL=1 hour), of top-25 buildings during the benchmark and the test period in case of CER-IRISH dataset. Subfigure (b) shows percent change during test period to that of the benchmark period in case of the top-25 buildings in case of CER-IRISH dataset.

A significant decrease in consumption during the peak hours of test period is observed for Group 3 and 4. The percentage of change during the average daywise consumption of the predicted top-25 buildings, is illustrated in Fig. 3.11 (b) which shows a gradual decrease of upto 25% during 5 p.m and then a gradual increase when moving away from the peak hours.

Analysis on aggregate data: To study the stability of the residential buildings over the longer time period, we obtain the aggregate load values from the time series dataset T . Two different aggregation is obtained.

1. Daily aggregate: The daily aggregate is the sum of the daily loads $\forall t_i \in T$.
2. Weekly aggregate: The weekly aggregate is the sum of the weekly loads

CHAPTER 3. CLUSTERING TIME SERIES SUBSEQUENCES USING DIRECTED GRAPH

$$\forall t_i \in T.$$

For daily aggregate, we obtain the clusters of subsequences for $WL = \{2, 4, 8, 12, 16, 24, 48\}$ days. The maximum accuracy achieved for daily aggregate is 0.82 for the top-35 buildings, shown in Table 3.3. For the weekly aggregate, we obtain the clusters of subsequences for $WL = \{2, 4, 8, 12\}$ weeks. The maximum accuracy achieved for the weekly aggregate is 0.82 for both top-30 and top-35 buildings, shown in Table 3.3. The lower accuracy values in case of the last- k for both the daily and weekly aggregate, signify the validity of the proposed method. The top- k accuracy results for the aggregate data are similar to that of the hourly data, therefore instead of choosing the larger values of WLS, the proposed clustering can be applied on the aggregate data to study the DR programs for the longer time period.

3.7 Conclusions

In this chapter, we propose a time series subsequence clustering framework. The proposed framework aim to cluster the subsequently co-occurring patterns of the TS dataset, that actually form cohesive mesoscopic structures. To achieve the purpose, a weighted directed graph structure is obtained from the TS dataset and a novel quasi-clique based clustering technique is proposed. The graph vertices store information about the pattern mean, trend and range of values present in the subsequence. Therefore the graph structure store the pattern based information, similarity between patterns and also the sequential occurrence of the patterns. The labeled subsequences obtained through quasi- clique clustering help to obtain the regularity of patterns in TS samples.

Experiments are performed on multiple TS datasets and cluster quality measures are obtained for the proposed method and the other state-of-the-art graph based clustering techniques. The cluster samples are more closely located when clustered using the proposed method as compared to the other state-of-the-art graph based clustering techniques. The statistical test– Friedman Test establish that the proposed method is better than most of the other clustering techniques. AN elaborate study on cluster centroids gives insight about which of the patterns are dominant during each month, which of the patterns are observed during each month. The cluster centroid shapes differ during the weekdays and the weekends.

The atypical patterns are obtained from the degree based analysis that reveal the patterns that rarely occur in the TS. Wilcoxon test show that the frequency of occurrence of the atypical patterns is different from that of the normal patterns. The time complexity of the proposed method is not more than the state-of-the-art graph clustering techniques.

The proposed framework shows a societal impact when applied for DSM in case of smart grids. The proposed framework compute regularity of household

3.7. CONCLUSIONS

buildings and it is shown that when the predicted building set is actually chosen for DSM, it results to 25% decrease in electricity consumption during the peak hours.

4

Clustering time series subsequences using multiple directed graph

4.1 Introduction

Univariate TS data is an ordered sequence of real values data with a mature research in analysis, however, the existing techniques rarely discuss on change or the similarity of the temporal patterns in a TS dataset, at multiple time granularity. It is discussed in [162] that in majority of the research, the time factor as a function, is overlooked. The periodic patterns like trend, seasonality are ordered in nature and can be easily obtained [95]. In addition to periodic patterns, there exist a sequence of patterns that are not necessarily periodic and are obtained from different statistical or TS transformation. Such complex patterns have a repeating nature when analysed at different time granularities, also, show a dependency in their occurrence. This is generally observed in road traffic data [163], energy data [164]. Complex network have a high potential of encapsulating the pattern based information in several time granularities [165].

The recent efforts focus on summarizing large TS using small length patterns called the shapelets, that would also help in classification and clustering. The shapelets are space effective and computationally efficient when being used in clustering and classification problems. But, shapelet discovery at multiple time granularity could not help in discovery of similarity of the patterns at multiple time granularity.

Motivated by the success of complex network in knowledge representation, in this chapter, the TS dataset is converted into multiple weighted directed network and consensus network at different time granularity. A maximal quasi-clique based network clustering method is shown on multiple weighted directed network and consensus network. The maximal quasi-clique based graph clustering approach at multiple time granularity help to index the TS samples. The entropy based stability measure is used for indexing the TS samples at multiple time granularity. Application of the proposed method in case of appliance level dataset

uncover important insights to maintain the demand-supply balance.

4.2 Related work

Despite of several clustering techniques on univariate TS dataset, in literature, there is a constant need for state-of-the-art clustering because of the increasing data size and its different pattern types. In literature, the change point detection and rule discovery [78] is vastly studied for pattern mining. The pattern mining techniques aim for TS clustering operations such that the TS samples can be segregated based on the desired application domain.

Every TS sample show change in static properties– mean, median at certain time interval, or the change in periodic properties, which is called the change points. Detection of change points help in event detection that is further used for clustering [166, 3]. The change point detection techniques can be categorized into supervised and unsupervised methods, discussed in [167, 3]. Considering the subsequence length in case of change point detection, a fixed length [168] or a variable length [169] both has been used for clustering .

In contrast to change points, the rule discovery gives the idea of temporally dependent patterns in the time series. The association rule mining (ARM) strictly find the patterns in the form of $A \rightarrow B$, that is, if A and B are the two different patterns, then A precede B in the entire TS [78]. The time lag between patterns A and B can differ based on the application areas [170, 171]. ARM techniques are linguistic in general, which is popular in market analysis [172]. The development of technology has observed an increase in clustering using the rule discovery techniques [78, 173]. In case of rule discovery, a reduced representation of original TS is obtained that retain the characteristics of the original dataset and analysis could be done with less computational time [174].

Several applications in Internet of Things (IoT) rely on subsequent co-occurring pattern discovery in TS, specifically in smart homes as given in [175]. Cheng et.al [172] have proposed an PrefixSpan, apriori based algorithm to mine the sequential patterns in a cross marketing data and then cluster them based on the patterns extracted. The transaction data of the customers will help to find the appropriate partners for cross marketing and promote industry growth [172]. Rather than clustering on the extracted temporal patterns, Wang et. al [97] have computed Dynamic Time Warping (DTW) distance and applied clustering to extract the rainfall temporal patterns. But the DTW on entire TS is rather a computationally expensive process than applying the clustering on a reduced data representation. Kaustubh et. al [176] propose a unified framework that can extract frequent subsequences considering the multiple constraints like length, gap, span of the patterns from the text data, but extending the same framework for TS on multiple time gran-

CHAPTER 4. CLUSTERING TIME SERIES SUBSEQUENCES USING MULTIPLE DIRECTED GRAPH

ularity require a TS representation [26]. The symbolic representation is applied on TS in [121] to extract the deviant events using ARM algorithm. To describe the spatial relationship between the variables, [177] has constructed a complex network where, a shape based distance is applied on a reduced TS representation and the nearest neighbor approach to built the complex network. Instead of employing any symbolic representation technique, the temporal patterns can also be studied by computing the statistical features at multiple time granularity as shown in temporal self regulation (TSR) of energy demand [178].

The previous discussed works shows how to cluster the TS data on the basis of temporal patterns, however, discovery of subsequent co-occurring patterns across multiple time scales and study of the clusters obtained at the respective time scales is rarely observed in literature.

Clustering techniques in monolayer graph based networks are abundant, but dividing the monolayer into groups minimizing the information loss in an NP-hard problem [179]. The authors in [179] have shown spectral clustering in multilayer network (SC). Other clustering techniques in case of multilayer graph based networks include Non-negative matrix factorization using Frobenius (NMF-F) and Lee (NMF-L) norm [180]. Another approach in case of multilayer network is aggregation technique in which multiple graph based network merge into a monolayer using the aggregation functions and then apply the traditional clustering techniques [181], shown in multidimensional community (MC) [181] and the projection average (PA) [181] methods.

4.3 Method

The methodology for this study as shown in Fig. 4.1 involves the broad steps as discussed below. The terms– network and graph, and– graph layer and season, is used interchangeably throughout the chapter.

4.3.1 Data preparation

Assuming a TS dataset T , where $|T| = M$, a TS sample $t_i \in T$ is of length $|t_i| = N$, we explain the proposed method below.

To deal with the issues of missing value, duplicate values at a certain time in case of the TS dataset, we first apply the data cleaning step as shown in Fig 4.1, that involves missing value imputation using moving average technique and duplicate replacement by their average.

As shown in Fig. 4.1, the data cleaning is followed by segmentation. Each t_i is divided into overlapping subsequences of a predefined length L and a stride value \hat{s} . A subsequence of t_i at time instant a is denoted by t_i^a , therefore the subsequence set of t_i becomes $\{t_i^a\}_{a=1}^S$. The overlapping subsequences can capture information

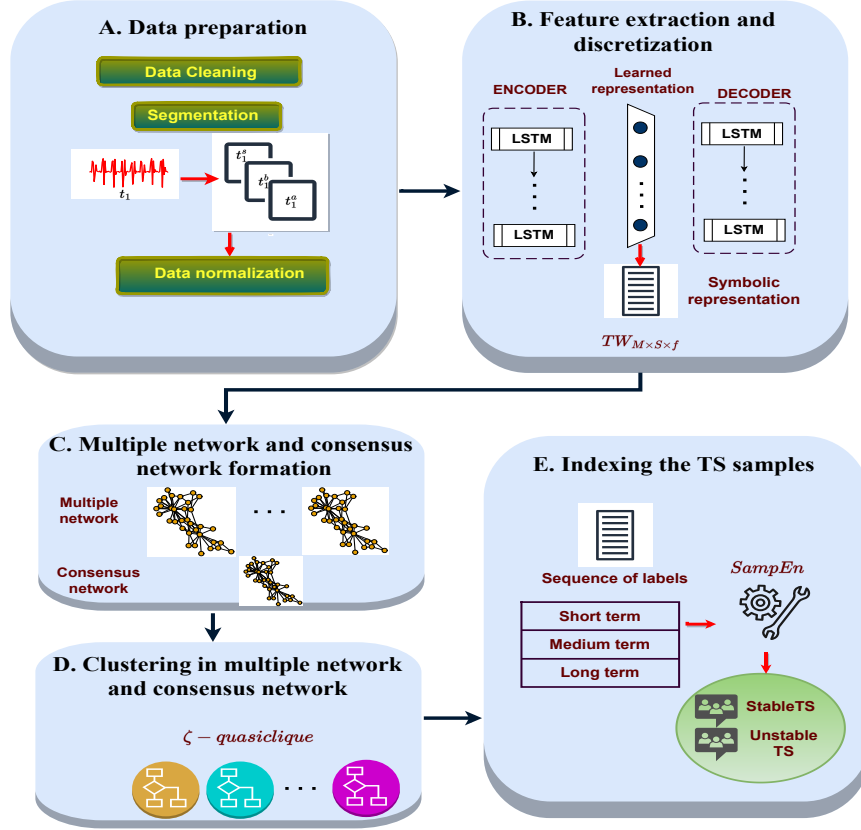


Figure 4.1: A flow chart of the proposed TS subsequence clustering using multiple directed graphs. Step A, is data cleaning, segmentation and normalization. Step B is the feature extraction from the LSTM-AE model. Step C is the multiple network and consensus network formation followed by the clustering of networks based on quasi-clique partitioning given in step D. Residential building indexing using *SampEn* and the stable set is obtained in step E.

about break points [182] unlike the disjoint subsequences. Two subsequent subsequence of t_i given as, t_i^a and t_i^b , are called overlapping if, $b \leq a + \hat{s}$, where \hat{s} is the stride value. To avoid the trivial match in two consecutive subsequences, the stride value follows the inequality: $\hat{s} > L/2$.

As shown in Fig. 4.1, the segmentation is followed by normalization. The subsequences are normalized using the inverse normal transformation (INT) [2]. The INT based normalization technique gives a normalized data representation from a highly skewed data unlike the other techniques like z-score and min-max normalization in which the normalized data is also skewed.

CHAPTER 4. CLUSTERING TIME SERIES SUBSEQUENCES USING MULTIPLE DIRECTED GRAPH

4.3.2 Feature extraction and discretization

Fig 4.1 shows the next step for the proposed framework is the feature extraction and discretization. The handcrafted feature set can cause information loss if not chosen appropriately. To overcome this, an automated feature extraction technique is shown in this work. The objectives of this step is: 1. capturing information about the current consumption pattern and autocorrelation present in a segment 2. dimensionality reduction. Motivated by the success of Long Short Term Memory- Autoencoder (LSTM-AE) models for relevant feature extraction and capturing the trend pattern from the sequential data[183, 184], the normalized subsequences are fed to the LSTM-AE model.

The LSTM-AE model consists two LSTM networks– encoder and the decoder in which the decoder is a reverse representation of the encoder, each with l hidden layers. Each input subsequence given as $\{t_i^1, t_i^2 \dots t_i^q\}$, is processed into a 2-d tensor of shape $lookback \times n_feat$, with n_feat as the number of features. The working principle of LSTM-AE for input t_i^q is given as:

$$Err = \frac{1}{2} \sum_q (t_i^q - \hat{t}_i^q)^2 \quad (4.1)$$

$$z_i^q = e_l(t_i^q) \quad (4.2)$$

$$\hat{t}_i^q = d_l(z_i^q) \quad (4.3)$$

$$e_l(t_i^q) = \sigma(W_e \cdot [e_{(l-1)}(t_i^q)] + b_e) \quad q \subset M \times S \quad (4.4)$$

$$d_l(z_i^q) = \sigma(W_d \cdot [d_{(l-1)}(z_i^q)] + b_d) \quad (4.5)$$

where, z_i^q, \hat{t}_i^q is the encoded and decoded representation of t_i^q which is computed using Equation. 4.4 and Equation. 4.5 respectively. The encoded representation z_i^q , is of length $|z_i^q| = f$. The functions $e_l(\cdot), d_l(\cdot)$ are encoder and decoder functions of the l^{th} layer, σ is the activation function of LSTM-AE. W_e, W_d are the weight matrices and b_e, b_d are bias vectors.

The Symbolic aggregate approximation (SAX) is a popular discretization technique for the TS that divide the normalized data into equiprobable regions, therefore retains the pattern based information of the data. As shown in Fig. 4.1, the discretized words are represented as a tensor $TW_{M \times S \times f}$. Each element of the tensor $TW_{M \times S \times f}$ is a vector τ_i^q .

4.3.3 Multiple network and consensus network formation

The next step, is the multiple and consensus network formation, shown in Fig. 4.1. The multiple network and consensus network formation can capture the change in TS patterns at multiple time granularity. The TS data is divided based on the seasons and a graph based structure is obtained from each season, that forms

the multiple network, which can also be used for daily analysis while for yearly analysis a consensus network is formulated. The definitions and construction of graphs are discussed below.

Multiple network: The multiple network is formally defined as a set $\mathcal{G} = \{G_s | s \in \text{layer}\}$ where each weighted directed network is given as $G_s = (\mathcal{V}_s, \mathcal{E}_s, \mathcal{W}_s)$, \mathcal{V}_s is a set of vertices, $\mathcal{E}_s \subseteq \mathcal{V}_s \times \mathcal{V}_s$ and the edge weight is \mathcal{W}_s . The multiple network is constructed as below: i) the *layer* corresponds to seasons the TS is divided into ii) unique words from the season s , given as, $T\bar{W} \subset TW_{M \times S \times f}$ form the set of vertices $v_i \in G_s$, such that each vertex represent a unique pattern iii) a directed edge $v_i \rightarrow v_j$, where v_i corresponds to τ_i^a is formed if $a < b \leq a + s$ iv) weight of the directed edge $\mathcal{W}(v_i \rightarrow v_j)$ is computed using SAX distance [185] between the words corresponding to v_i and v_j .

Consensus network: For a given multiple network \mathcal{G} , the consensus network is the weighted directed network given by a three tuple $\hat{G} = (V, E, \hat{W})$, where, $V = \bigcup_{\mathcal{V}_s} \mathcal{V}_s$ and $E = \bigcup_{\mathcal{E}_s} \mathcal{E}_s$. The edge weight $\hat{W}(v_i \rightarrow v_j)$ of the consensus network is the respective edge weights in \mathcal{G} .

Algorithm 8 Clustering

Input: $TW_{M \times S \times f}, \text{layer}$

Output: (C, \check{C})

- 1: Initialize the lists $V \leftarrow \emptyset, E \leftarrow \emptyset, \text{priority} \leftarrow \emptyset$
 - 2: **for** each $s \in \text{layer}$ **do**
 - 3: Construct G_s from $(\mathcal{V}_s, \mathcal{E}_s, \mathcal{W}_s)$
 - 4: $\text{priority} \leftarrow \text{Compute VW}(\cdot)$ for G_s using Eqn 4.6
 - 5: $C[s] \leftarrow \text{QC}(G_s, \text{priority})$
 - 6: $V \leftarrow V \cup \mathcal{V}_s$ and $E \leftarrow E \cup \mathcal{E}_s$
 - 7: **end for**
 - 8: Construct \hat{G} from (V, E)
 - 9: $\check{C} \leftarrow \text{CG}(\hat{G})$
 - 10: return (C, \check{C})
-

4.3.4 Clustering in multiple network and consensus network

The next step is the clustering as shown in Fig. 4.1. The clustering problem is formulated as a maximal quasi clique finding problem on a set of weighted directed network. We propose a greedy solution to find all possible maximal quasi clique structures from the multiple network and consensus network.

Initially the vertex set $\mathcal{V}_s \in G_s$ are ordered based on their non-increasing weighted degree measure (vertex priority) $VW(\cdot)$. According to the greedy choice rule, the vertex with highest priority is chosen so as to identify the larger maximal quasi clique structures before identifying the smaller quasi clique structures.

CHAPTER 4. CLUSTERING TIME SERIES SUBSEQUENCES USING MULTIPLE DIRECTED GRAPH

While the out degree of a vertex measure the number of subsequent patterns, the weighted degree measure indicate similarity between the subsequently occurring patterns. The weighted degree measure $VW(.)$ is given as:

$$VW(v_i) = \frac{d^-(v_i) \times \mathcal{W}(v_i)}{|\mathcal{V}_s|} \quad (4.6)$$

$$d^-(v_i) = \frac{\mathbf{d}^-(v_i) - \min_{\forall v_i \in G_s} (\mathbf{d}^-(v_i))}{\max_{\forall v_i \in G_s} (\mathbf{d}^-(v_i)) - \min_{\forall v_i \in G_s} (\mathbf{d}^-(v_i))} \text{ and} \quad (4.7)$$

$$\mathcal{W}(v_i) = \sum_{\forall v_j \in \mathcal{N}(v_i)} \mathcal{W}(v_i \rightarrow v_j) \quad (4.8)$$

where, $d^-(v_i)$ is the normalized out degree of v_i , while $\mathbf{d}(\mathbf{v}_i)$ is the out degree, $\mathcal{W}(v_i)$ is the weight of v_i , that is, the sum of weights of the outgoing edges of v_i , $\mathcal{N}(.)$, is the neighborhood. The value of $VW(v_i)$ ranges from 0 to 1 with 1 as the highest priority.

In general, clustering on networks aim to obtain dense subgraph structures. Definition of dense subgraph varies depending on the application. The quasi clique, a relaxed clique structure is one possible way to define the dense subgraphs [72]. According to the existing definition of quasi-clique, it is a subgraph such that every edge $(v_i \rightarrow v_j)$ between a vertex pair (v_i, v_j) implies the existence of a reverse edge $(v_j \rightarrow v_i)$ to it. For weighted directed graph, the ζ -quasi clique computation on a subgraph $g \in \mathcal{G}$ or $g \in \hat{\mathcal{G}}$ is based on the density function $\Delta(.)$ defined as below.

Definition 11 A subgraph g is said to form a ζ -quasi clique if- i) $\Delta(g) \geq \zeta$ such that $\zeta \in [0, 1]$ ii) g forms a strongly connected component, that is, $SCC(g) = 1$ and iii) $|g| \geq 3$ where,

$$\Delta(g) = \min_{\forall v_i \in g} \frac{\min(\mathcal{W}(v_i), d^-(v_i))}{|g|} \text{ and,} \quad (4.9)$$

where $d^-(.)$ and $\mathcal{W}(.)$ is defined in Equation. 4.7 and 4.8 respectively. Greater the value of ζ , more similar are the vertices indicating more coherent patterns. For a layer s , the maximal ζ -quasi clique is a subgraph formed from the vertex set $v \subset \mathcal{V}_s$, such that including any node from the neighborhood of v does not form a larger ζ -quasi clique.

The problem of partitioning a graph into the maximal quasi clique structure in a weighted directed multiple network is atleast as hard as that of unweighted undirected graph [74, 72], because for every directed edge between a vertex pair, there must be an edge in the reverse direction. We propose a greedy method that

iteratively finds all the possible maximal quasi clique structures from each layer of the multiple network \mathcal{G} followed by exploration of maximal quasi clique structures from the consensus network \hat{G} .

The algorithm 8 shows the steps of the proposed greedy strategy. It starts with arranging the vertices of multiple network in order of vertex priority. In step 4 of algorithm 8, *priority* queue stores the vertices of G_s arranged in the non-increasing order of the $VW(.)$ computed using Equation. 4.6. Final cluster set in G_s is $C[s] = \{c_i\}_{i=1}^k$, where k is the number of unique clusters and the final cluster set in \hat{G} is \hat{C} .

Algorithm 9 Quasi- clique: $QC(G_s, priority)$

Input: G_s : directed weighted graph, *priority*: priority queue of vertices based on $VW(.)$.

Output: C_s : cluster labels of G_s .

```

1:  $Q_{new}, \zeta \leftarrow threshold(priority, G_s)$ 
2:  $old \leftarrow 0, new \leftarrow length(Q_{new})$ 
3: while  $old \neq new$  do
4:   for each  $g \in Q_{new}$  do
5:      $g \leftarrow$  Dequeue subgraph from  $Q_{new}$  based on  $\Delta$ 
6:      $\mathcal{N}(g) \leftarrow$  Dequeue all subgraphs from  $Q_{new}$  based on
        $\Delta$  such that  $\mathcal{N}(g)$  is in neighborhood of  $g$ 
7:     for each  $\tilde{g} \in \mathcal{N}(g)$  do
8:        $g_{new} \leftarrow$  a subgraph formed from  $\{g \cup \tilde{g}\}$ 
9:       if  $SCC(g_{new}) \ \& \ \Delta(g_{new}) \geq \zeta$  then
10:        Enqueue  $g_{new}$  to  $Q_{new}$  with associated  $\Delta$ 
11:         $g_{new} = \emptyset$ 
12:        Exit loop
13:       end if
14:     end for
15:   end for
16:    $old \leftarrow new, new \leftarrow length(Q_{new})$ 
17: end while
18: return  $Q_{new}$ 

```

The algorithm 9 intends to obtain clusters from each layer G_s . The vertex with highest priority designates the maximum number of temporally dependent similar patterns connected to it. In step 1 we use a function $threshold(.,.)$ to optimize the value of ζ , which is discussed further in algorithm 10. The steps 3 to 17 allows to merge neighboring subgraphs so that the merged subgraph is a quasi-clique. The for loop in step 4 intends to merge ζ -quasi clique without violating the property as given in definition 11. The neighborhood of a subgraph g is, $\mathcal{N}(g) = \{\tilde{g} | \text{quasi clique and } \exists E_{v_i, v_j} \text{ where } v_i \in g \text{ and } v_j \in \tilde{g}\}$. The steps are repeated

CHAPTER 4. CLUSTERING TIME SERIES SUBSEQUENCES USING MULTIPLE DIRECTED GRAPH

until the number of clusters obtained after each iteration do not change.

Algorithm 10 Obtain ζ : $threshold(Q, G_s)$

Input: Q : priority queue of vertices

G_s : directed weighted graph G_s obtained from s^{th} season

Output: ζ : threshold value and

Q_{new} : priority queue of subgraphs

```

1: Initialize an empty priority queue  $Q_{new}$ 
2: Initialize  $\zeta = \emptyset$ 
3: while  $Q \neq \emptyset$  do
4:    $v_i \leftarrow \text{Dequeue } Q$ 
5:   for  $v_j \leftarrow \text{Dequeue } Q$  such that  $v_j \in \mathcal{N}(v_i)$  do
6:     for  $v_k \leftarrow \text{Dequeue } Q$  such that  $v_k \in \mathcal{N}(v_j)$  do
7:        $g \leftarrow \text{Subgraph}\{v_i, v_j, v_k\}$ 
8:       if  $\text{SCC}(g)$  then
9:         Enqueue  $g$  to  $Q_{new}$ , with associated  $\Delta(g)$ .
10:         $\zeta \leftarrow \zeta \cup \Delta(g)$ 
11:        Exit loop
12:       end if
13:     end for
14:   end for
15: end while
16:  $\zeta \leftarrow \min(\zeta)$ 
17:  $IV \leftarrow$  isolated vertices in  $Q_{new}$ 
18: for each  $iv \in IV$  do
19:    $g_{new} \leftarrow$  Add  $iv$  to the quasi-clique cluster  $g$  of  $Q_{new}$  such that the change in  $\Delta$ 
    is minimum
20:   Update  $\zeta \leftarrow \Delta(g_{new})$ 
21:   Update  $g_{new}$  in  $Q_{new}$  with the associated  $\Delta$ 
22: end for
23: return ( $Q_{new}, \zeta$ )

```

The algorithm 10 intends to discover the optimal value of ζ such that most of the vertices of graph G_s can find their quasi clique structure and are not left isolated. Initially the vertex with the highest priority is chosen for the merge. The merging is a two step process 1. identify two closest neighbors (v_j, v_k) of v_i in order of priority 2. merge them to form a quasi clique structure g . This is repeated in steps 3 to 15. The neighbors of v_i , form a quasi clique g with highest possible $\Delta(g)$ value. The vector ζ stores $\Delta(.)$ values of the quasi-clique obtained from the merging process and the optimal value is the minimum in ζ , given in step 16.

The vertices of lower priority for which no quasi-clique cluster could be obtained is denoted as IV in step 17. The set IV consists of the vertices that form lesser number of temporal patterns as compared to the other vertices. The steps 18

to 22 aim to merge $iv \in IV$ with an already formed quasi clique g , such that the change in $\Delta(g_{new})$ of the new subgraph is minimum. The output Q_{new} contain the maximal ζ -quasi-clique structures of G_s .

Algorithm 11 Consensus: $CG(\hat{G})$

Input: \hat{G}

Output: Q : priority queue in which each element forms a cluster

- 1: Initialize $c_priority \leftarrow NULL$
 - 2: Initialize a priority queue $Q \leftarrow NULL$
 - 3: **for** each unique $v_i \in V$ **do**
 - 4: $c_priority[v_i] \leftarrow mean(W_{ij})$
 - 5: **end for**
 - 6: $Q \leftarrow QC(\hat{G}, c_priority)$
 - 7: **return** (Q)
-

Algorithm 11 shows the steps to obtain the consensus network \hat{G} from the multiple network and the clusters in \hat{G} . The steps 3 to 5 is to obtain the priority of each vertex v_i . Priority of the vertices is stored in a vector $c_priority$, which is the mean of the out degree of edge weights given as $mean(W(v_i \rightarrow v_j))$. The mean of the out degree edge weights is used to capture the overall strength between two nodes present in all layers.

4.3.5 Indexing the TS samples

The next step is the indexing of TS samples, as shown in Fig. 4.1. The subsequences belonging to a cluster denote that they represent similar patterns and the change in cluster labels denote the change in patterns. Hence, lesser the change in cluster labels, higher is the temporal similarity and that denote a stable behavior of TS sample. Therefore, the indexing is done by measuring the stability of t_i based on how frequently the cluster label of its subsequences change with time. Cluster labels of subsequences obtained from the quasi-clique structures in \mathcal{G} and \hat{G} are used for indexing at multiple time granularity. The cluster label representation of subsequences is given by a sequence $tc_i = \{\check{c}_i^1, \check{c}_i^2, \dots, \check{c}_i^a, \dots, \check{c}_i^S\}$, where \check{c}_i^a is the cluster label for the subsequence $t_i^a \in \{t_i^a\}_{a=1}^S$. Below we discuss the indexing at multiple time granularity with respect to tc_i , where $tc_i \in C[s]$ denote the cluster labels from multiple network G_s and $tc_i \in \check{C}$, denote the cluster labels from the consensus network \hat{G} . The different granular levels are named as:

1. Short term granularity measure stability based on the pattern observed on day basis. In this case tc_i is broken to daywise sequence of labels.
2. Medium term granularity measure the stability based on pattern observed during the entire season. The cluster set $C[s]$, for layer s is used.
3. Long term granularity measure

CHAPTER 4. CLUSTERING TIME SERIES SUBSEQUENCES USING MULTIPLE DIRECTED GRAPH

the stability based on pattern observed in the entire yearly data. In this case the $tc_i \in \tilde{C}$ is considered.

A statistical measure called the sample entropy (*SampEn*), is used for the indexing both in multiple and the consensus network [186]. The *SampEn* for cluster label representation tc_i , is shown in Eqn 4.10.

$$\begin{aligned} \text{SampEn}(tc_i) &= -\log \frac{A}{B} \text{ where,} \\ A &= \text{number of pairs, where, } d[x_m(y), x_m(z)] < r \\ B &= \text{number of pairs, where, } d[x_{m+1}(y), x_{m+1}(z)] < r \end{aligned} \quad (4.10)$$

where $d(.,.)$ is the distance function, $x_m(y) = \{c_i^y, \dots, c_i^{y-m+1}\}$, m is the sliding window length and r is the tolerance level. The choice of parameters $d(.,.)$, m and r are discussed in section 4.4. *SampEn* quantify the regularity in the sequence of labels. Lower values of *SampEn* signify regularity and predictable nature of patterns in the TS whereas the higher values indicate irregular nature of patterns [186]. In case of the labeled dataset, we obtain the stable set by assuming, $\rho = \{0, 0.02, 0.04, \dots, 1\}$. The TS samples with $\text{SampEn} \leq \rho$ are considered as the stable set and vice versa. The optimal value of ρ at each granular level is decided on the accuracy scores. In case of unlabeled dataset, while the value ρ can be chosen as the one of mean, median, standard deviation of *SampEn* at each granular level, we choose the ρ as the median value of *SampEn*.

While the idea behind the use of *SampEn* is to estimate how many patterns are repeated throughout the dataset, it would also be interesting to measure what is the maximum percentage that a pattern is repeated. To estimate the latter we propose the definition of Temporal similarity (TPS). For the cluster label representation tc_i , at a given granularity the TPS is the percentage of sliding window sequence of cluster labels which are common, given in Equation 4.11. The higher value of TPS denotes the existence of a higher number of repeated pattern.

$$\begin{aligned} \text{TPS} &= \max \left[\frac{100}{S - m + 1} \sum_{z=1}^{S-m+1} \beta_z \right]; \\ \beta_z &= \begin{cases} 1, & \text{if } tc_i[z] = x_m(z) \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \quad (4.11)$$

where, S is the total number of subsequences obtained, m is the sliding window length.

4.3.6 Time complexity

The first step is to obtain the time complexity for the feature representation of TS. Extraction of overlapping subsequences from M different TS samples is $O(M)$. A standard LSTM unit has elementwise multiplication and addition operations over the input, previous hidden state and the cell state. Considering ω different weights in an LSTM unit, the time complexity of learning the weights per timestep using the Adam optimization technique is $O(\omega)$. Considering the input shape to LSTM-AE $q \times \text{lookback} \times n_{\text{feat}}$, where $q \subset M \times S$, lookback is the number of time steps to lookback and n_{feat} is the number of features to be extracted from LSTM-AE. The training time complexity is dominated by $O(q \times \text{lookback})$, with $q \gg \text{lookback}$, it is $O(q)$. Discretizing the features into symbols using the breakpoint table is $O(1)$ [187]. With $M \times S$ different subsequences and $M \gg S$, the time complexity for symbolic representation is $O(M)$.

Graph is formed by extracting the unique words from each s layer followed by finding \mathcal{E} and \mathcal{W} . For a layer s , the worst case time complexity to extract the unique words is $O(M \log M)$. The edge formation is $O(1)$ and \mathcal{W} uses SAX based distance which has the worst case time complexity of $O(M^2)$. Therefore the overall complexity is $O(M^2)$.

The time complexity to compute the vertex priority of set \mathcal{V} is $O(\mathcal{V})$, which is a constant. In worst case, time complexity to sort the vertices in order of priority is $O(\mathcal{V} \log \mathcal{V})$. The algorithm 9, calls the *threshold(.)* function to obtain the optimal ζ value. The steps of *threshold(.)* function algorithm 10 is enqueueing and dequeuing the elements and check the conditions as given in Definition 11, that is, check for $\text{SCC}(g)$, which takes $O(\mathcal{V} + \mathcal{E})$ time. On satisfying the given quasi clique condition, the newly formed ζ -quasi clique is inserted to the priority queue, this require $O(\mathcal{V})$ time. In worst case, time complexity is $\mathcal{E} = \mathcal{V}^2$. Therefore total time required in algorithm 10 is $O(\mathcal{V} \log \mathcal{V}) + O(\mathcal{V}^2) = O(\mathcal{V}^2)$.

The steps 3 to 18 in algorithm 9 is merging the already formed ζ -quasi clique structure. In worst case, if the merge occurs at each iteration in the priority queue, then time required is $O(\mathcal{V}(\mathcal{V} - 1)) = O(\mathcal{V}^2)$. Therefore the overall time complexity for algorithm 9 is $O(\mathcal{V}^2)$.

The algorithm 8 shows repeating the $\text{QC}(.,.)$ for multiple and consensus network, therefore the overall time complexity is $O(\mathcal{V}^2)$.

4.4 Experimental design

In this section, we discuss the datasets used, the comparative study, the validation measures used to validate the performance of the proposed approach and also discuss the parameter tuning of LSTM-AE and *SampEn* computation.

CHAPTER 4. CLUSTERING TIME SERIES SUBSEQUENCES USING MULTIPLE DIRECTED GRAPH

4.4.1 Data description

For the current study, the datasets used are- CER-IRISH dataset discussed in section 1.1.1, CU-BEMS dataset discussed in section 1.1.1 and the Stock market discussed in section 1.1.1.

For multiple graph analysis, the CER-IRISH dataset is further split into the winter (Dec-Feb), spring (Mar-May), summer (Jun-Aug) and autumn (Sep-Nov) seasons, based on the average climate in Ireland¹. Based on the monthly average climate in Bangkok, the CU-BEMS is divided into three seasons- winter (Nov-Feb), summer (Mar-Jun) and rainy (Jul-Oct) [188].

The cluster quality analysis is done on both the labeled and unlabeled datasets. The proposed method is validated on labeled dataset, that is, CER-IRISH, and, the study on stable TS samples is performed. Lastly in case of the unlabeled data- CU-BEMS, we predict the stable appliances at multiple granularity.

4.4.2 Comparative Study

For a fair comparative study of the proposed clustering applied on multiple and consensus network, we use the existing clustering techniques on the monolayer and the multi layer models as discussed below.

The multilayer community detection methods- NMF-F, NMF-L, PA, MC, SC, TSR which are discussed in section 4.2 are used for comparative study. In addition, because the proposed clustering approach is applied to discover the stable TS samples in case of smart grids, a comparative analysis is also done with [7], where, the authors extract features from cluster centroid to find the suitability of the residential building/appliance for DR.

To compare between the handcrafted features with the automatic feature extraction using the LSTM-autoencoder model as shown in this paper, we construct a monolayer network (MN) using the handcrafted feature set as shown in [173], denoted as (MN-F1) and also on the features obtained from the LSTM-autoencoder model, denoted as (MN-F2) [173]. The monolayer network is obtained from the entire TS dataset by choosing the unique symbols as vertices and connecting the subsequent vertices based on time order.

4.4.3 Validation measures

1. F1-score: Assuming tp : #true positives, tn : #true negatives, fp : #false positives and fn : #false negatives,

$$F1 - score = 2 \times \frac{precision \times recall}{precision + recall} \quad (4.12)$$

¹<https://www.met.ie/climate/climate-of-ireland>

where $precision = tp/(tp + fp)$, $recall = tp/(tp + fn)$.

2. Area under curve (AUC): This measure the performance under all possible classification thresholds.
3. Davies-Bouldin (DB) index: For a given cluster set C , the DB index is the measure of scatter within the cluster c_i and the separation between the clusters c_i and c_j .

$$DB = \frac{1}{k} \sum_{i=1}^k D_i \text{ where, } D_i \equiv \max_{j \neq i} J_{i,j} \text{ and, } J_{i,j} = \frac{\hat{S}_i + \hat{S}_j}{d(c_i, c_j)} \quad (4.13)$$

where k is the total number of clusters, \hat{S}_i is the average distance of each point in cluster c_i to its centroid and \hat{S}_j is the average distance of each point in cluster c_j to its centroid, $d(c_i, c_j)$ is the distance between cluster centroids of c_i and c_j .

4. Average scattering for clusters (*Scatt*): It is the mean of the norms of the variances in the clusters:

$$Scatt(C) = \frac{\frac{1}{k} \sum_{i=1}^k \|var(c_i)\|}{\|var(C)\|} \quad (4.14)$$

where the $var(c_i)$ is the variance of c_i and $var(C)$ is the variance of the entire cluster set.

5. Wilcoxon signed rank test: It has two research hypothesis, H_0 : the proposed method performance is same as that of the compared method and H_1 : there exist a significant improvement in the performance of the proposed method to that of the compared method. We obtain the paired difference in scores between the cluster validity measures of the proposed method to that of the existing methods and report the P-value for the test statistic. H_0 is rejected if the P-value is significantly smaller than the significance level $\alpha = 0.05$.

4.4.4 Parameter tuning

1. LSTM-AE parameters: The optimal parameter set for the model is chosen on the basis of mean squared error (MSE) score, that is, the parameter set with best MSE score is chosen for feature extraction. The number of units in each layer is varied from 8-32, the number of epochs is varied from 200-400 with *Relu* and *tanh* activation functions. Each model is trained using the Adam (Adaptive momentum) optimization rule with the learning rate of 0.001. The early stopping rule is used to prevent model overfitting. In

CHAPTER 4. CLUSTERING TIME SERIES SUBSEQUENCES USING MULTIPLE DIRECTED GRAPH

Table 4.1: Hyperparameters of the LSTM-AE model– number of layers, hidden units, number of epochs, batch size and activation functions. The best MSE scores are reported in bold.

Data	#Layers	([Units], epochs, batch-size, activation)	MSE
IRISH	2	[16, 4], 200, 16, tanh	0.07
	3	[32, 16, 4], 300, 32, relu	0.05
	4	[16, 8, 4, 4], 400, 16, relu	0.06
CU-BEMS	2	[16, 4], 200, 16, relu	0.0181
	3	[16, 8, 4], 200, 32, relu	0.016
	4	[16, 8, 4, 4], 400, 32, relu	0.0183

Table 4.1, we report the parameters with the MSE in each layer. The dataset is divided into 80% training and 20% test set. Best values are obtained for $lookback = 2$ hours.

2. *SampEn* parameters: Computation of *SampEn* depends on the parameters– $d(.,.), m, r$. The $d(.,.)$ capture whether the cluster labels are same or different, hence (i) $d(.,.) = 1$ if $x_m(y) \neq x_m(z)$, 0 otherwise (ii) Edit distance. The value r depends on the ED.

In order to avoid long hours of discomfort, DR is executed by the utilities for a shorter time period during a day. So the value m is chosen such that it can find the patterns that exist in a day. Hence the value m is varied as $\{2, 3, \dots, \text{\#subsequences in a day}\}$.

4.5 Results and Observations

To check the validity of the proposed method, we first index the stable buildings using *SampEn* at each granular level, as discussed in section 4.3.5 and compare the F1-score and AUC of the proposed method with the state-of-the-art methods on labeled CER-IRISH dataset.

We carry out a comparative study on cluster quality and report the characteristics of the clusters obtained from the proposed method.

We also measure the average percentage change in daywise consumption during the benchmark and the test period for the predicted stable building set, given as % *Change*. The decrease (\downarrow) in % *Change* denote the decrease in load during test period to that of the benchmark and vice versa, thereby proves that the identified stable set of buildings can create a sustainable grid.

4.5. RESULTS AND OBSERVATIONS

Table 4.2: Comparative results of cluster quality indices for the proposed and baseline methods. WT is the p-value for Wilcoxon signed rank test. The best values are highlighted in bold.

Method	CU-BEMS		IRISH		SM		WT
	DB	Scatt	DB	Scatt	DB	Scatt	
NMF-F	3.24	1.78	1.78	0.95	1.4	0.76	4.19e-3
NMF-L	4.84	1.98	3.13	1.46	2.71	0.55	7.62e-4
PA	2.54	2.54	2.43	2.23	1.76	0.82	1.52e-05
MC	3.04	1.15	3.01	0.92	1.76	3.02	1.52e-05
SC	5.15	2.47	1.97	1.97	3.47	1.74	1.06e-4
MN-F1	4.74	0.98	3.92	0.7	2.2	0.21	0.01
MN-F2	1.1	0.29	1.54	0.72	0.5	0.14	1.44e-4
Proposed	1.07	0.27	0.91	0.33	0.6	0.11	–

4.5.1 Cluster quality analysis

In Table 4.2, we report the cluster quality results and the P-value obtained from the Wilcoxon test for the proposed method and the state-of-the-art methods. The best values are highlighted in bold face. In case of IRISH, the proposed method shows 2.8% improvement in *DB* index, 6.8% improvement in *Scatt*. In case of CU-BEMS, the proposed method shows 40% improvement in *DB* index and 52% improvement in case of *Scatt*. The lower value of *DB* signify that the intracluster distance is less than the intercluster and the lower values of *Scatt* signify lower variance in the clusters, thus indicating the clusters obtained from the proposed method is more coherent as compared to the other state-of-the-art methods. In case of Stock Market dataset, the *Scatt* is minimum in case of the proposed method. *The better results in majority of the cases for the proposed method signify that the cluster samples are closer in case of the proposed method as compared to the other state-of-the-art.*

The pairwise Wilcoxon signed rank test establish that the proposed method is a significant improvement in terms of the measured cluster quality metrics over the state-of-the-art methods with P-value 0.01 or lower.

We obtain a normalized consumption pattern from the cluster samples of each cluster denoted as the representative TS profiles (RTP). Fig. 4.2 shows the RTP of each dataset during different seasons. The inset barplot in Fig 4.2 shows the percentage of samples belonging to different clusters. The clusters with number of samples below 2% of data set, are not shown in the barplot.

In case of CER-IRISH dataset, shown in Fig 4.2 (a), RTP of cluster 1 during winter, spring and autumn show downfall during the 2nd hour of each subsequence, followed by a rise. Cluster 2 of winter and autumn seasons show a rise in consumption pattern during the 2nd hour. *This signify that the RLPs of two different seasons belonging to a cluster can match, means that the normalized consumption behavior*

CHAPTER 4. CLUSTERING TIME SERIES SUBSEQUENCES USING MULTIPLE DIRECTED GRAPH

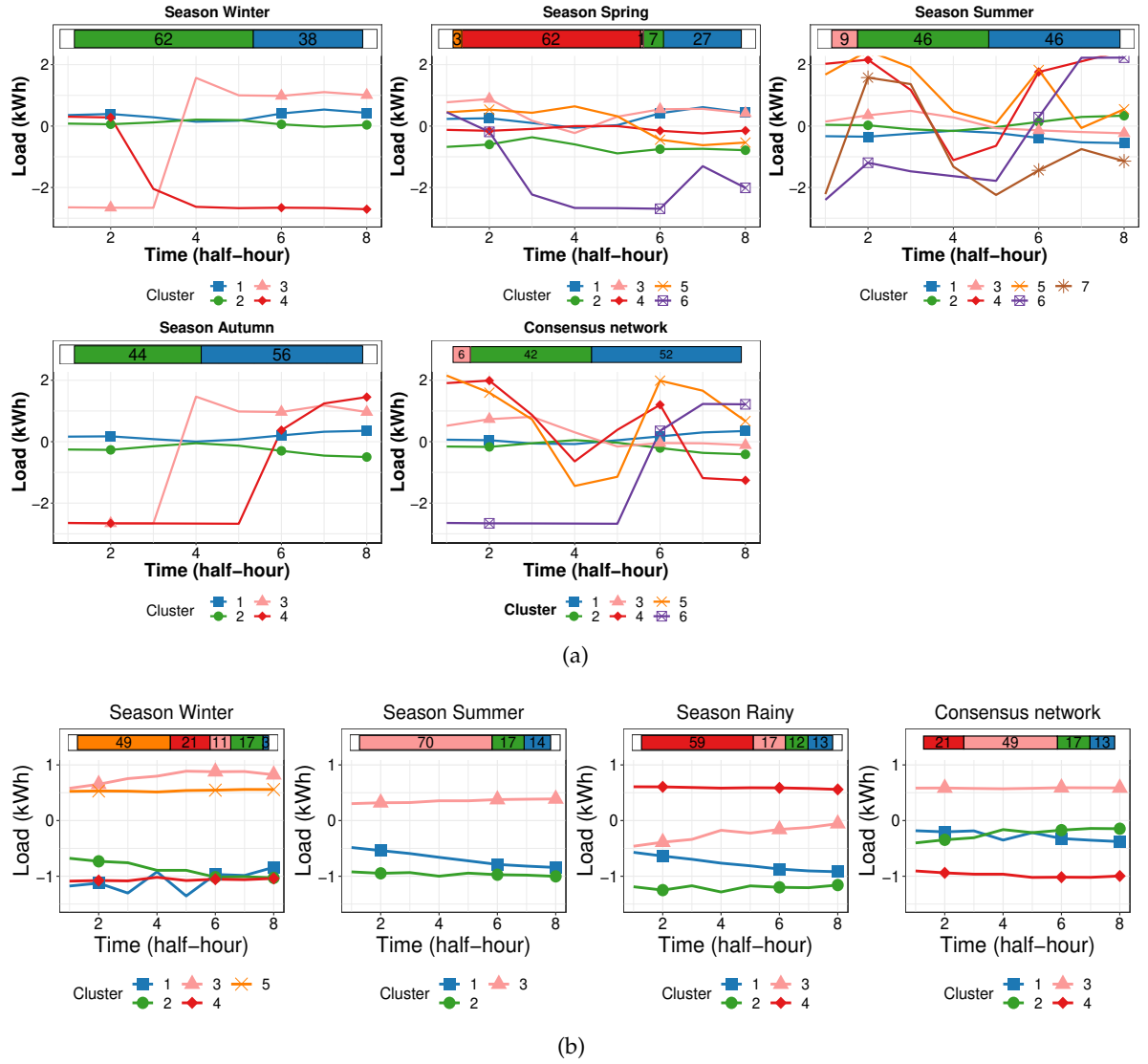


Figure 4.2: Normalized representative TS profiles (RTP) obtained from clusters in multiple network (respective seasons) and from the consensus network for (a) IRISH (b) CU-BEMS datasets

across the seasons can be similar.

In case of long term granularity, shown by the consensus network in Fig. 4.2 (a), the RTP of clusters 1 and 2 show a similarity with the RLPs of cluster 1 and 2 during winter, summer and autumn, while no similar consumption behavior is observed for summer.

In case of CU-BEMS dataset, shown in Fig. 4.2 (b), the similar RTP of cluster 1 and 2 in summer and rainy seasons signify that the appliances have similar normalized consumption behavior. In case of long term granularity, no such similar behavior is

4.5. RESULTS AND OBSERVATIONS

Table 4.3: Comparative study on accuracy measures (F1- score and AUC) for the proposed and state-of-the-art clustering methods. The *Change %* denote increase \uparrow or decrease \downarrow in electricity consumption of the stable set selected using the proposed and state-of-the-art methods. The best scores are highlighted in bold face. The *% improvement* in the last row highlight the % improvement in F1-score and AUC when compared with the best performing state-of-the-art clustering.

Method	Short Term		Medium term		Long Term		%Change
	F1	AUC	F1	AUC	F1	AUC	
NMF-F	0.73	0.63	0.64	0.47	0.64	0.47	8.51 (\uparrow)
NMF-L	0.74	0.63	0.64	0.47	0.64	0.47	8.51 (\uparrow)
PA	0.75	0.67	0.64	0.47	0.64	0.5	5.37(\uparrow)
MC	0.76	0.77	0.68	0.63	0.42	0.63	20.13 (\uparrow)
TSR	0.6	0.71	0.62	0.67	0.57	0.52	19.45 (\uparrow)
SC	0.73	0.63	0.54	0.6	0.52	0.57	4.33 (\uparrow)
MN-F1	0.78	0.8	0.62	0.63	0.68	0.53	19.13 (\uparrow)
MN-F2	0.64	0.73	0.61	0.57	0.67	0.53	37.93 (\uparrow)
2-level	0.32	0.64	0.57	0.6	0.26	0.6	3.21 (\uparrow)
Proposed	0.83	0.83	0.77	0.7	0.71	0.67	1.09 (\downarrow)
% improvement	6.4%	0.3%	13%	4%	4%	6%	–

observed with the different seasons.

Key observation: A decrease in DB index and *Scatt* cluster quality is observed for both the labeled and unlabeled datasets. The pairwise Wilcoxon test establish the significance of the clusters obtained from proposed method. The study on RLPs reveal the similarity and dissimilarity in the consumption behavior at the different temporal granularity.

4.5.2 Validating Proposed method

In Table 4.3 we show the F1-score and AUC at different granularity for the state-of-the-art methods discussed in section 3.2. The improvement of proposed method in the different scores as compared to best performing state-of-the-art is highlighted in the last row of table. The better F1-score and AUC in case of proposed method signify that the indexing approach can give more number of stable consumers as compared to that of the state-of-art techniques.

To estimate the improvement achieved due to inclusion of the steps (i) feature extraction and discretization (ii) multiple and consensus network formation, we

CHAPTER 4. CLUSTERING TIME SERIES SUBSEQUENCES USING MULTIPLE DIRECTED GRAPH

compare the proposed method with MN-F1 and MN-F2 respectively.

6.4% increase in F1-score, 0.3% increase in AUC, during short term, 13% increase in F1-score and 4% increase in AUC during medium term, 4% increase in F1-score and 6% increase in AUC during long term is obtained in comparison with the state-of-the-art methods, from Table 4.3.

An average % *Change* of 1.09% in case of proposed method in Table 4.3 denote that the predicted stable set of buildings show a decrease in load consumption during the test period by 1.09%, while all other methods show an increase in the load during the test period.

Key observation: The proposed method advances the state-of-the-art for the selection of consumers eligible for DR in all the temporal granularity. Consumers selected by the proposed method decreases the average load consumption by 1.09% as compared to that of the best performing state-of-the-art method.

4.5.3 Study on stable buildings

In this study, we show the variation of mean *SampEn*, temporal similarity (TPS) at different granular levels and the time of change in the temporal patterns, for both the stable and unstable set of buildings.

Variation of mean *SampEn* and TPS: Fig. 4.3 (a) shows the plot for daily mean *SampEn* values and TPS for each of the stable and unstable set. The average mean *SampEn* for the stable set is lower than that of the unstable set, clearly shown by *gap* between the average mean *SampEn* of the stable to that of the unstable set. The medium term granularity show lower mean *SampEn* for stable set during summer season as compared to that of other seasons. The mean *SampEn* in case of stable set for the long term granularity is 0.06 and that of the unstable set is 0.11.

The days of summer season, as shown in Fig. 4.3 (a) observes a fall in the mean *SampEn* without much increase in the TPS value. During the short term granularity, *SampEn* signify that though there exists a regularity in consumption pattern in the stable set, the percentage occurrence of the repeating patterns is similar to the rest of the days. The increasing order of average mean *SampEn* for the stable set observed from Fig. 4.3 (a) is the days of summer, spring, winter and autumn, which is also supported by the medium term analysis in Fig. 4.3 (b). The 94% of days in case of stable set shows TPS higher than 60%, while for unstable set, the TPS value is not more than 53%, which is also established from the long term analysis Fig. 4.3 (c). The increasing order of average TPS for the stable set observed from Fig. 4.3 (a) is the days of spring, summer, autumn and winter, which is also supported by the medium term analysis in Fig. 4.3 (b).

Therefore, this study reveals that the consumers with changing patterns can be appropriate for DR programs provided the time granularity at which the patterns are changing.

The stable set consumers obtained from long term granularity is a strict subset

4.5. RESULTS AND OBSERVATIONS

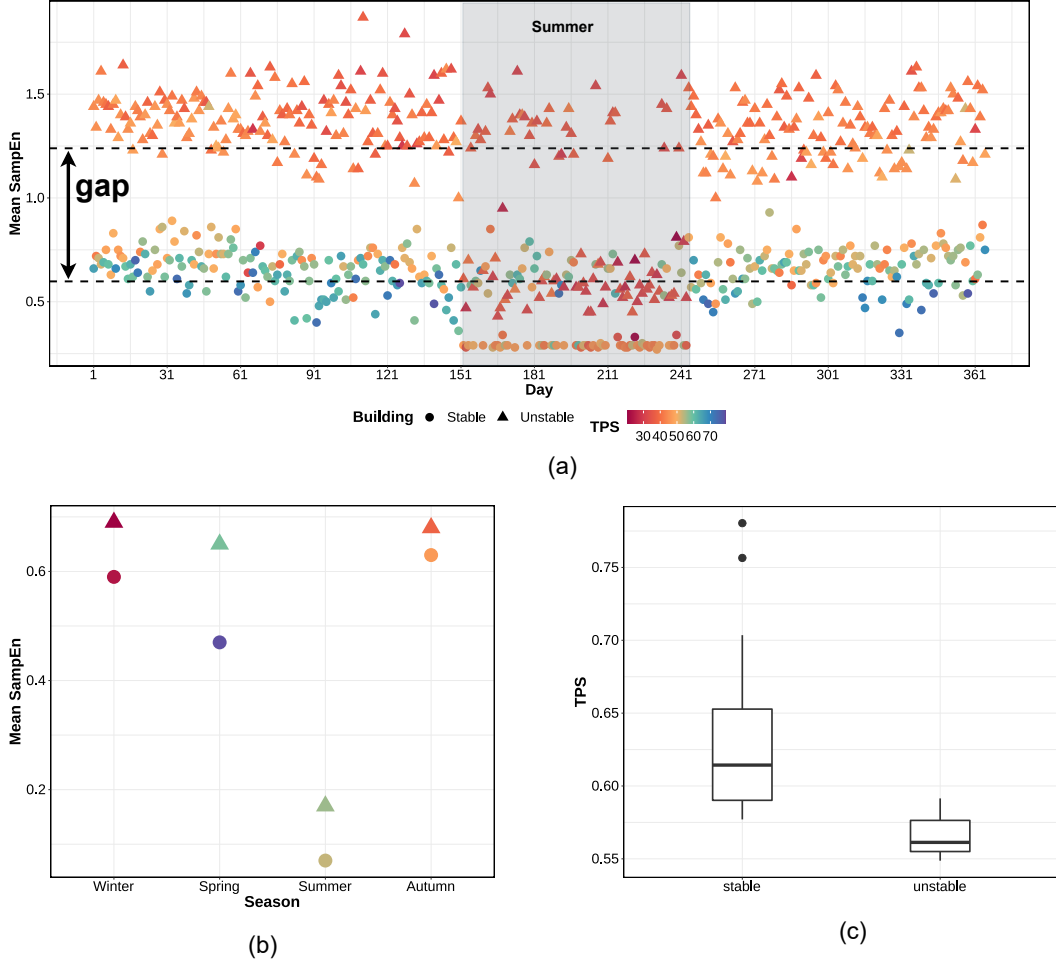


Figure 4.3: Subfigure (a) shows mean $SampEn$ and variation of TPS at short term granularity- the dotted horizontal lines denote the average of mean $SampEn$ for the stable and unstable set, the highlighted region in the plot shows the days of the summer season. Subfigure (b) shows mean $SampEn$ at medium term and subfigure (c) shows a boxplot for the TPS distribution in stable and unstable set at long term granularity, all for CER-IRISH data set.

of the medium term during spring season and, the same is true for the short term granularity. *Therefore, the consumers which are stable during both the short and the medium term granularity, are also found to be stable during the long term granularity.* The possible reason for this result could be the regularity in patterns observed during spring season as measured by low mean $SampEn$ and high TPS.

Key observation: A regularity in consumption pattern of the stable consumer set is observed during spring season of the short term, medium term and the long

CHAPTER 4. CLUSTERING TIME SERIES SUBSEQUENCES USING MULTIPLE DIRECTED GRAPH

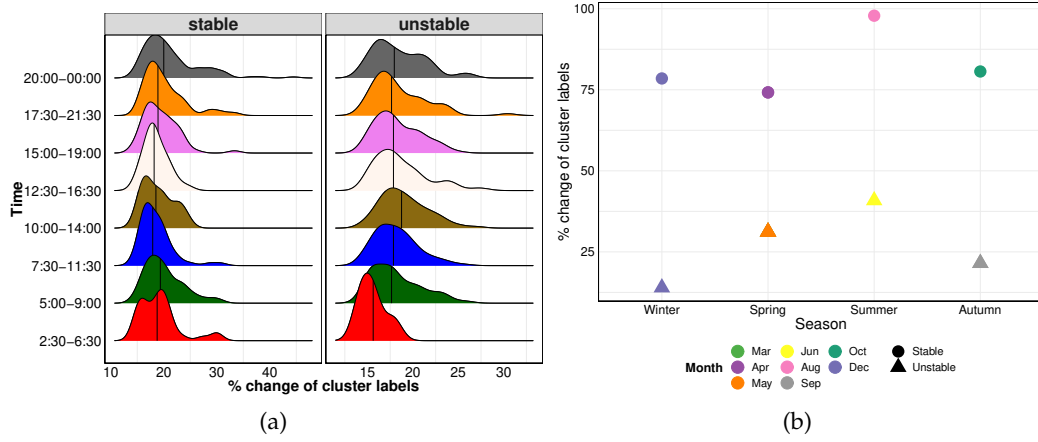


Figure 4.4: The % change in cluster labels in CER-IRISH data set out of all the observed changes during (a) short term granularity and (b) medium term granularity.

term granularity as measured by low mean *SampEn* and high TPS value.

Time of change: We next study the time span when maximum change in pattern happens. In case of short term, the maximum % of change for the stable set occurs during the time 17 : 30 – 00 : 00 in a day as shown in Fig. 4.4 (a). Similar result is obtained in long term analysis where the maximum change in pattern is observed during the late evening hours during the month May for the stable set while no such clear pattern is observed for the unstable set. In case of unstable set, the maximum change occur during the early morning hours (2 : 30 – 6 : 00) while all other time slots follow similar distribution of the values over the year, as shown in Fig. 4.4 (a). In case of medium term, the maximum % of change occurs during the summer season for both the stable and the unstable set as shown in Fig. 4.4 (b).

Key observation: The maximum % change in pattern in case of stable set occurs during the late evening hours observed from the short term and long term granularity study.

4.5.4 Study of appliances

To show how the proposed method can be used to identify the appliances with regular consumption behavior, we carryout the study on CU-BEMS dataset.

In Fig. 4.5 (a) we show the mean *SampEn* for the short term granularity for stable and the unstable appliances. It is observed that the mean *SampEn* for the stable set during the summer season, that is Day #60 to Day #180 is consistently similar, while the mean *SampEn* in case of the unstable set vary widely across the

4.5. RESULTS AND OBSERVATIONS

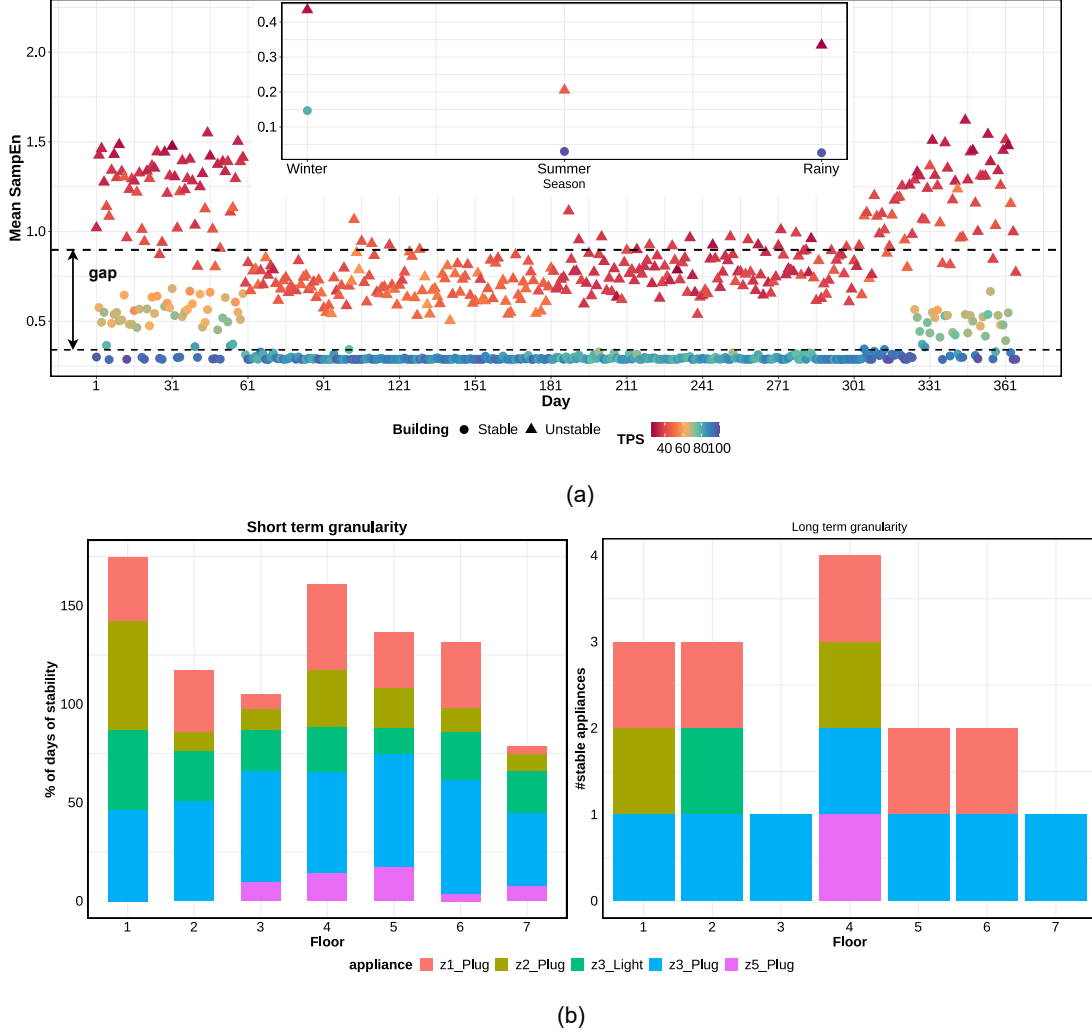


Figure 4.5: Subfigure (a) shows mean *SampEn* and variation of TPS for short term granularity analysis. The dotted horizontal lines denote the average of mean *SampEn* for the stable and unstable set. The inset plot shows the mean *SampEn* and TPS for the medium term granularity. Subfigure (b) shows barplot of % of days when an appliance is identified as stable in each floor during the short term granularity and for number of stable set appliances in each floor during the long term granularity, all in case of CU-BEMS data set.

CHAPTER 4. CLUSTERING TIME SERIES SUBSEQUENCES USING MULTIPLE DIRECTED GRAPH

summer days. The *gap* in Fig. 4.5 (a) shows a clear demarcation in the average of mean *SampEn* values of the stable and unstable set. The TPS for stable set for all the days during summer is greater than the days of the other seasons as shown in the inset plot. *This denotes that the change in patterns for the appliances during winter and rainy season is more as compared to that of the summer.* Similar result is observed in medium term granularity as shown in the inset plot in Fig. 4.5 (a).

Fig. 4.5 (b) shows that *z3_Plug* has the stable pattern for an average of 51% of the days across the different floors. On further analysis, it is also found that *z3_Plug* shows 100% TPS for 64% of the days in case of short term granularity. On the contrary, *z5_Plug* is found to be the unstable appliance for an average of 90% of the days in short term as well as long term granularity, as shown in Fig. 4.5 (b).

Key observation: The mean *SampEn* and TPS during summer season over all appliances is found to be better than the other seasons indicating higher regularity in consumption during summer season. The appliance *z3_Plug* is found to be stable at all granular levels and *z5_Plug* is found to be unstable at all granular levels however no such inference could be obtained for the other appliances.

4.6 Conclusions

In this chapter, we propose clustering on TS subsequences at multiple time granularity. The purpose of this chapter is to identify the subsequent co-occurring patterns at multiple time granularity and index the TS samples based on their pattern stability. The weighted graph structure is obtained at each granular level, where the graph vertices store pattern based information. Because the hand-crafted feature set can cause information loss, in this chapter an automatic feature extraction approach is shown. The state-of-the art encoder-decoder model extract features from each TS subsequence. The quasi-clique based clustering is applied on it to capture the subsequent co-occurring patterns at multiple time granularity.

The proposed technique is validated using accuracy scores on a labeled electricity based load consumption data. The cluster quality analysis is performed on multiple TS data, and lower values of cluster quality achieved from proposed method signify better tightness of clusters. The statistical test prove that the cluster quality measures are different from the other state-of-the-art clustering techniques. A stability measure is shown that can measure regularity based on whether the subsequent patterns have the same or different cluster labels. The stability measure perform indexing of TS samples and the lower value of stability signify that the patterns of a TS sample, is uniform in nature. An elaborate study on an office building is shown to identify which of the appliances used in building have a stable consumption behavior.

5

Classification of time series using graph convolutional network

5.1 Introduction

The aspect of TS classification is to assign a class label to some unseen TS, given the prior information about the class labels of other TS. The application of TS classification is seen in anomaly detection, medical diagnosis, smart grids, human activities, satellite images. Based on how are the features extracted from TS, the TS classification can be categorized into– feature based, distance based and model based methods [189].

Considering the feature based approach, it aim to extract the relevant information from the TS to reduce the dimensionality. The conventional classifiers like Markov models, Decision trees are then applied on the extracted feature set. Because of high granularity in TS, the classification on feature based approaches are easier. However, the reduced dimensionality can ignore the temporal patterns in historic data or the subsequently co-occurring patterns. The distance based approaches either compute the distance on entire TS data or on the reduced dimensionality space, which is followed by classification using the conventional or the state-of-the-art techniques. The choice of distance measures however affects the accuracy of classifier. Also, on a highly granular TS data, the pairwise distance computation is a computationally intensive process. The distance measures applied on raw TS cannot be useful for storing the temporal pattern based information or the subsequently co-occurring pattern based information. The model based approach automatically learn features from the available data, unlike the handcrafted features. Initially the models are trained on prior input-output labeled information. During the forward pass, the randomly initialized weights are trained to obtain a final output vector. The output vector actually store the probabilities of input belonging to different classes.

Considering the advantages of using the model based approach, this chapter propose a novel TS classification approach based on the learnt graph structure.

CHAPTER 5. CLASSIFICATION OF TIME SERIES USING GRAPH CONVOLUTIONAL NETWORK

The graph structure is learned for each TS sample using the state-of-the-art model. The undirected weighted graph formed for TS sample, is the input to the classifier model.

5.2 Literature

In this section we discuss the related work on feature based, distance based and model based classification methods.

In case of the feature based classification, features can be any of the temporal, statistical or the spatial. The temporal features are the time dependent features. The authors in [99] have extracted temporal features from the deep learning model and used SVM for classification of the EEG images. A greedy feature selection approach is shown in [190] that choose the features by optimizing the linear classification rate on the training data. The statistical features like the -mean, variance, standard deviation when extracted from multiple subsequences can capture the temporal data patterns. The subsequences can either be fixed or dynamic in nature. The statistical features from TS has been used for classification as an anomaly or not [98]. Several works in literature show that the CNNs and the LSTM models can be used for automatic temporal and spatial feature extraction [100], but it require large datasets which can result in poor classification results otherwise. The words extraction from TS in [191] is a novel classification technique which extract the words from the sliding window or a fixed length window subsequences.

There are some of the benchmark distance measures applied for TS pairwise distance computation– dynamic time warping distance, Euclidean distance, Manhattan distance, which are detailed in [192]. The distance based measures are mostly used for computing the similarity between whole TS. 1-KNN classifier model is the baseline to obtain any novel distance based classifier [99]. Because the distance measures between whole TS samples is a computationally expensive process, the authors in [104] convert the whole TS into shapelets and apply a distance measures on the shapelet to classify the TS using CNN model. The authors in [105] propose an ensemble of proximity-trees based classifier where the distance measure to be used is chosen at random. Several fast distance computation techniques are shown in literature, but, the distance based classification cannot capture the temporal nature of the patterns.

Considering the model based approach for classification, the authors have shown the performance of the basic NN models– the MLPs and the fully CNNs. Current state-of-the-art techniques include the CNN and LSTM models which have been applied on multiple TS domain like- disease classification in tomato leaf [8], classifying the limb movements from brain activity [4], fruit classification [101] and brain tumor classification using attention based LSTM model [193],

residual networks for medical image classification [102]. For a better feature approximation from TS, a two-level model based approach is suggested [194]– first, model for feature extraction of TS and the second model for the classification. The authors in [103] propose a novel TS classifier that uses large number of random convolutional kernels, to extract the temporal and the spatial features from large TS and a linear classifier at the end for classification.

Graph structure in case of TS is gaining importance during the recent years because it has the ability to represent temporal dependencies of patterns [195]. The graph structure include representing each time point by a vertex or the TS subsequences [37] as a vertex [196]. Given a pre-defined graph structure, the graph neural networks (GNNs) are capable of incorporating neighborhood information of the vertices and use the message passing technique to update the vertex information [163]. GNNs have been used in varying TS domains; Zhang et. al in [163] propose a GNN driven traffic forecasting model that fully capture the spatial correlation between roads and address the problem of sparse data and irregular road relations. In [197], the authors construct a graph structure from appliance dataset and use the graph spectral clustering for predicting the state of the appliances at the next time instant. GNNs furnish promising results in case of anomaly detection of TS when trained on the patterns [198, 199]. The GCN based models have shown success in the multivariate TS prediction where each variable is modeled as a graph vertex for exploring the correlations between vertices [200]. A combination of GCN with the gated recurrent unit (GRU) is shown in [65] for traffic forecasting, where the GCN capture the spatial relationship and the GRU capture the subsequent co-occurring pattern based information in the data.

5.3 Proposed method

This section explains the proposed Graph Convolutional Network (GCN) based classification framework. We use the TS dataset that consist of half hourly power consumption readings of \mathcal{Z} different residential homes, $\{H_1, H_2, \dots, H_j, \dots, H_{\mathcal{Z}}\}$. The power consumption for a residential home $H_j = \{p_1, p_2, \dots, p_q, \dots, p_l\}$, is broken down into overlapping subsequences of window length n , and the subsequence set obtained is represented as $\{T_t^{(H_j)}\}_{t=1}^{(l-n+1)}$, where each subsequence $|T_t^{(H_j)}| = n$, which is input to the classification model. Each H_j is assigned a label $tr(H_j)$ based on whether it has been chosen for DSM operation or not.

The proposed GCN framework consist of two different levels– Level 1 is used to obtain a graph structure for each house and Level 2 is the Graph convolution network (GCN) that predicts whether the residential home is suitable for DSM or not. The proposed framework predicts the output as $o(H_j) \in [0, 1]$ where 1 signify that the residential home has been selected for DSM and vice-versa. Fig. 5.1 shows

CHAPTER 5. CLASSIFICATION OF TIME SERIES USING GRAPH CONVOLUTIONAL NETWORK

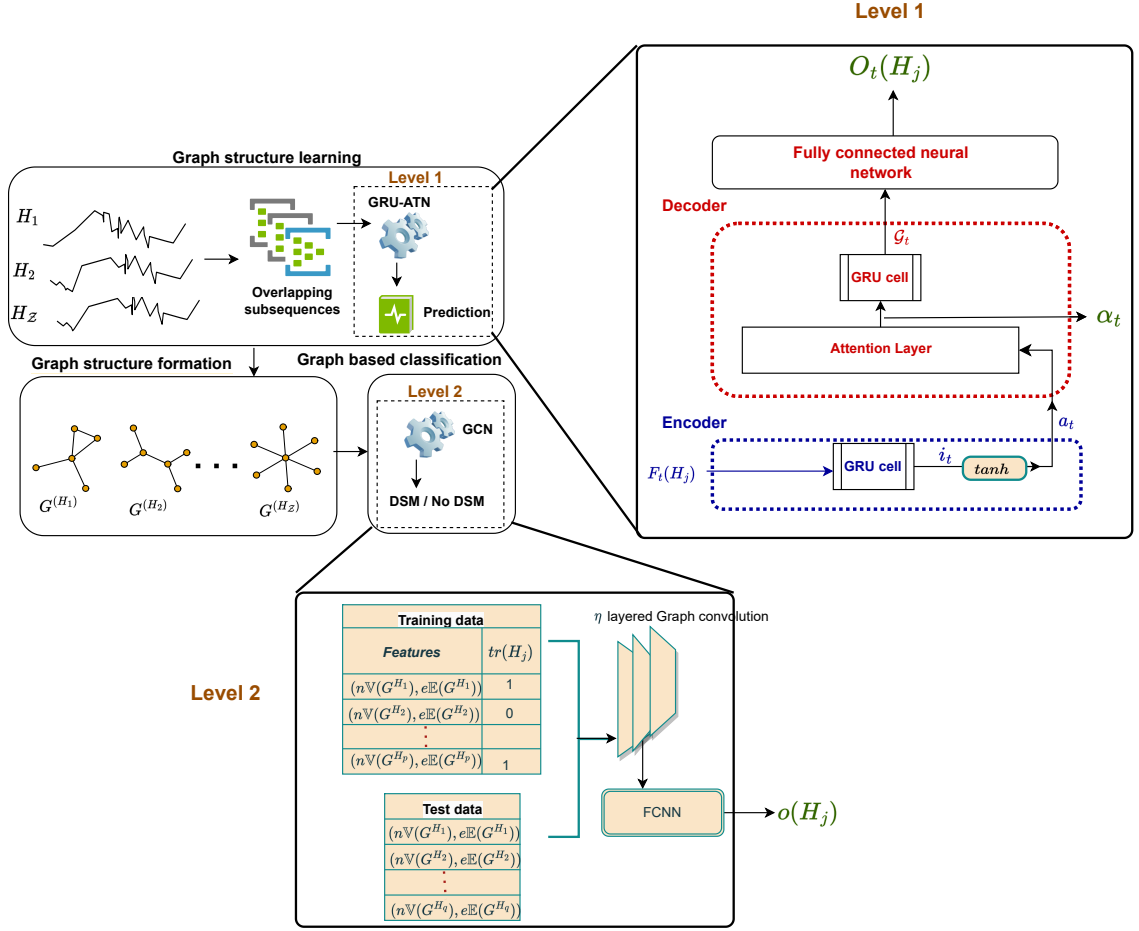


Figure 5.1: An overview of the proposed GCN framework. The framework is made of two levels– Level 1 (GRU-ATN) and Level 2 (GCN). An elaborate representation of Level 1 and Level 2 is shown.

an overview of the proposed GCN framework with the different levels as explained below.

5.3.1 Graph structure learning

The Level 1 of GCN consist of an attention based encoder-decoder model that aim to obtain a graph structure for each H_j . The encoder-decoder model consists of Gated Recurrent Unit (GRU) with attention (ATN) applied at the decoder end, referred as GRU-ATN model. The uniqueness of GRU-ATN is that it helps to obtain a graph structure when no prior information about the vertices or the edges is provided. The reason of forming a graph structure from each residential home

5.3. PROPOSED METHOD

is because it help to obtain the temporal information about the patterns formed from $\{T_t^{(H_j)}\}_{t=1}^{(l-n+1)}$, in the form of vertices and edges. The subsequences form unique pattern and every pattern has a temporal relationship with the subsequent patterns formed, and this relationship can be clearly represented in a graph structure. The GRU-ATN model is elaborated in Fig. 5.1.

To predict the subsequence at time instant t , given as T_t , the GRU-ATN model uses a feature set, $F_t^{H_j}$, given as below,

$$F_t(H_j) = [T_{t-\$, T_{t-\$+1}, \dots, T_{t+1}, T_{t+2}, \dots, T_{t+\$-1}] \quad (5.1)$$

where, $\$$ is the sliding window length applied on the subsequence set $T_t^{(H_j)}$, $F_t(H_j) \in \mathbb{R}^{2\$ \times n}$. The proposed graph structure can capture the subsequently co-occurring patterns within the sliding window length $\$$. Below we explain the working principle of GRU-ATN model.

Encoder The encoder side of the model consists of GRU which has only two gates– reset and update gate. The feature set $F_t^{H_j}$ is the input to GRU cell, as shown in Fig. 5.1. The working principle of GRU cell is explained in the following equations.

$$R_t = \sigma(W_R F_t(H_j) + U_R h_{(t-1)} + b_R) \quad (5.2)$$

$$Z_t = \sigma(W_Z F_t(H_j) + U_Z h_{(t-1)} + b_Z) \quad (5.3)$$

$$new_h = \tanh(W_n F_t(H_j) + R_t \star (U_n h_{(t-1)} + b_h)) \quad (5.4)$$

$$i_t = (1 - Z_t) \star new_h + Z_t \star h_{(t-1)} \quad (5.5)$$

where R_t, Z_t, new_h, i_t are the reset, update, new hidden state and the encoded feature representation respectively. Equation 5.2 and 5.3 are the weighted sum of input and the previous hidden state $h_{(t-1)}$, passed through the sigmoid (σ) activation. W_R, U_R are the weight matrices of reset gate and b_R is the bias of reset gate, W_Z, U_Z are the weight matrices of update gate and b_Z is the bias of update gate, and W_n, U_n are the weight matrices and b_h is the bias to generate the new hidden state. The new_h in Equation 5.4 is the weighted sum of the input and Hadamard product (\star) of R_t with the previous hidden state $h_{(t-1)}$, passed through the \tanh activation. All the weights and biases are initialized from uniform distribution with $\mathcal{U}(-\sqrt{k}, \sqrt{k})$, where $k = 1/\text{input size}$.

Attention decoder The decoder predicts the output subsequence at time instant t , of the house H_j , given as $O_t^{H_j}$. Because all the subsequences in F_t are not

CHAPTER 5. CLASSIFICATION OF TIME SERIES USING GRAPH CONVOLUTIONAL NETWORK

equally important for predicting the $O_t^{H_j}$, the attention layer focus on which of the subsequence is important for the prediction of $O_t^{H_j}$. Working of the attention decoder is explained below.

$$a_t = \tanh(W_a i_t) \quad (5.6)$$

$$\alpha_t(x, y) = \frac{\exp(a_t(x, y))}{\sum_{x=1, \dots, \$} \sum_{y=1, \dots, n} \exp(a_t(x, y))} \quad (5.7)$$

$$\mathcal{G}_t = \text{GRU}(\alpha_t) \quad (5.8)$$

$$O_t^{H_j} = \text{ReLu}(W\mathcal{G}_t + b) \quad (5.9)$$

The input to the attention layer is a_t , given in Equation 5.6, where, W_a form the learnable weights, which is initialized from the uniform distribution given as $\mathcal{U}(-\sqrt{k}, \sqrt{k})$, where $k = 1/\text{input size}$. The output from attention layer is α_t , that is, the normalization applied on a_t using the softmax activation across the sliding window length $\$$ as given in Equation 5.7.

The α_t is passed through the GRU cell, given in Equation 5.8. A non-linear activation ReLu is performed on \mathcal{G}_t , given in Equation 5.9 to obtain the final output $O_t^{H_j}$. Then to compute the overall attention coefficient $\mathcal{A}(\cdot)$ for each subsequence in $F_t^{H_j}$, we aggregate over the subsequence length n using the max function as given in Equation 5.10.

$$\mathcal{A}(F_t) = \max_{x=1, \dots, 2\$} (\alpha_t[x, n]) \quad (5.10)$$

The GRU-ATN is trained to minimize the L1 loss function, given in Equation 5.11.

$$l1(O_t^{H_j}, T_t^{H_j}) = |T_t^{H_j} - O_t^{H_j}|$$

$$L1 = \frac{\sum_{j=1, i=1}^{i=p, j=n} l1(O_t, T_t)[i, j]}{n \times p} \quad (5.11)$$

where, $l1(O_t^{H_j}, T_t^{H_j}) \in \mathbb{R}^{p \times n}$, p is the number of training samples, $T_t^{H_j}$ is the target subsequence at time instant t .

5.3.2 Graph structure formation

The learned attention coefficient $\mathcal{A}(\cdot)$ from GRU-ATN model in Level 1 is used to form weighted undirected graph structure for each residential home.

5.3. PROPOSED METHOD

The graph structure for a residential home H_j is given as $G^{H_j} = (v, e, w)$, where the unique subsequences in $\{T_t^{(H_j)}\}_{t=1}^{(l-n+1)}$ form the vertices, given by a mapping function $\mathbb{V} : v_p \rightarrow T_p$. To identify the edges connected to a vertex v_p , we use a threshold value δ on vector $\mathcal{A}(\cdot)$. The δ intends to capture the most significant vertices which are required to predict the desired subsequence at time instant t . An undirected edge between two arbitrary vertices v_p, v_q , is given by a mapping function $\mathbb{E} : \{e_{pq} | v_q \in F_p(H_j); \mathcal{A}(F_p)[q] \geq \delta\}$, where $F_p^{H_j}$ is the feature set used to predict $T_p^{H_j}$ as shown in Equation. 5.1. The graph structure obtained is illustrated in Fig. 5.1.

The edge weight w is given as $w(e_{pq}) = D(T_x^{H_j}, T_y^{H_j})$. The edge weight aim to capture the pointwise distances between subsequences, therefore $D(\cdot)$ is the Euclidean distance used in this work.

5.3.3 Graph based classification

The graph structure of each residential home, obtained from the previous stage is used as input to the Graph convolution network (GCN) [201], from which the vertex features and edge features are extracted. The vertex and edge features form a tuple $(n\mathbb{V}(G^{H_j}), e\mathbb{E}(G^{H_j}))$, where, $n\mathbb{V}(G^{H_j}) = \{T_t^{H_j}\}_{t=1}^{l-n+1}$, and $e\mathbb{E}(\cdot) = (e_{pq}, w(e_{pq}))$ is a tuple storing the edge and the edge weight. The vertex and the edge features obtained is illustrated in Fig. 5.1.

For an η layered GCN, the hidden state is computed as:

$$\mathcal{H}_i^{(\eta+1)} = \sigma \left(\sum_{q \in N(v_i)} \frac{1}{d(v_i, v_j)} \mathcal{H}_j^\eta W_\eta \right) \quad (5.12)$$

$$y(n) = \sigma(\mathcal{H}_i^{(\eta+1)} \mathbf{W}^T + b) \quad (5.13)$$

$$o(H_j) = \text{softmax}(y(n)) \quad (5.14)$$

where $N(v_i)$ is set of neighbors for vertex v_i , $d(v_i, v_j)$ is computed as $\sqrt{|N(v_i)| \times |N(v_j)|}$, W_η is the weight matrix for the η^{th} layer, σ is the differentiable non linear rectified linear unit (ReLU) activation function. The hidden state $\mathcal{H}_i^{(\eta+1)}$ is then fed to an n layered fully connected neural network (FCNN) with ReLU activation. The final output label of $o(H_j)$ is based on the *softmax* activation applied on $y(n)$. The weights of the GCN model, W_η , are initialized using Xavier initialization technique shown in [202]. Assuming $tr(H_j)$ as the true label, the model is trained to minimize the cross entropy loss function L_{CE} as given in Equation. 5.15.

$$L_{CE} = -\frac{1}{p} \sum_{j=1}^p tr(H_j) \log o(H_j) + [1 - tr(H_j)] \log[1 - o(H_j)] \quad (5.15)$$

CHAPTER 5. CLASSIFICATION OF TIME SERIES USING GRAPH CONVOLUTIONAL NETWORK

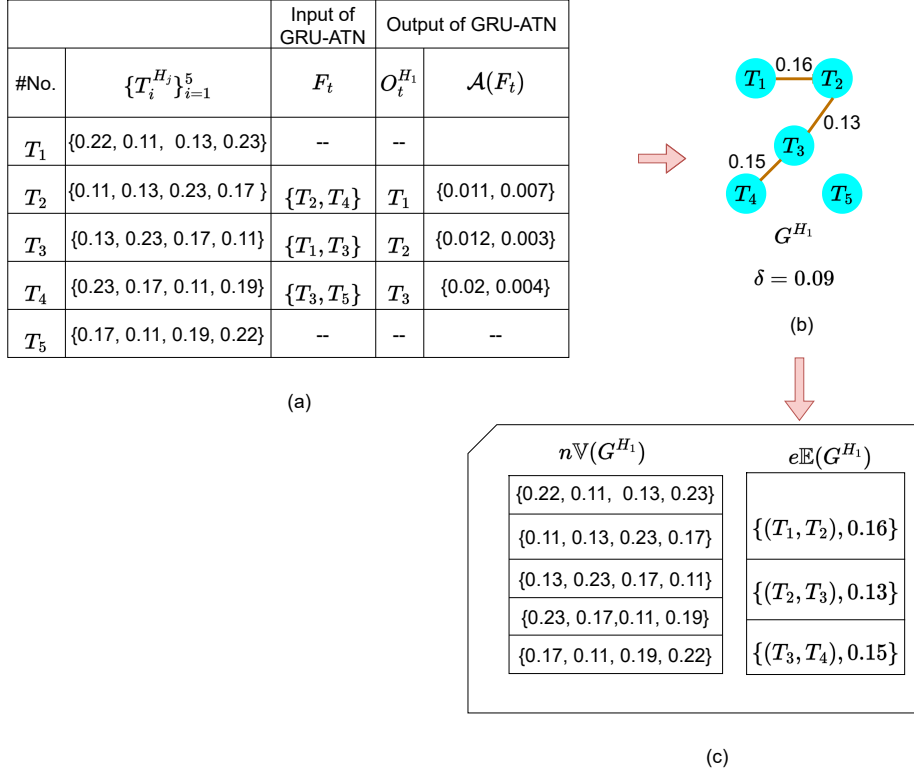


Figure 5.2: A toy example that shows the working principle of proposed GCN where, (a) shows the input subsequences obtained for a TS sample, the input and output for the GRU-ATN used at Level 1 (b) is the graph structure formed and (c) is the vertex and the edge features to be used as input to GCN.

5.3.4 Toy example

To show the working example using the proposed GCN framework, we assume a TS data sample, $H_1 = \{0.22, 0.11, 0.13, 0.23, 0.17, 0.11, 0.19, 0.22\}$, where, $|H_1| = 8$. Assuming $n = 4$, the number of subsequences obtained is 5, which is the subsequence set for H_1 , given as $\{T_i^{H_1}\}_{i=1}^5$ in Fig. 5.2. Assuming, $\$ = 1$, the feature set F_t , to predict T_2 is $\{T_1, T_3\}$, similarly, to predict T_3 , $F_t = \{T_2, T_4\}$, as shown in Fig. 5.1. The input and the output to GRU-ATN is shown in Fig. 5.2 (a).

To obtain the graph structure for the H_1 , we assume the mean value of $\mathcal{A}(F_t)$ as δ , that is, $\delta = 0.09$. Each unique subsequence corresponds to a graph vertex and for $\mathcal{A}(\cdot) \geq \delta$ is connected by edges. The graph structure obtained for $\delta = 0.09$ is shown in Fig.5.2 (b). The edge weights w is computed using the Euclidean distance between the vertex features $nV(G^{H_1})$. The vertex and edge features are obtained from G^{H_1} , as given in Fig.5.2 (c), which is used as training data for the GCN.

5.3.5 Time complexity

In case of GRU-ATN model in Level 1, the GRU unit consists of elementwise multiplication and addition during the forward and the backward passes and then the weight updates. The total time complexity for each iteration of GRU-ATN model with a single layer of GRU cell is approximately $O(2\$\times n)$. The same applies for the reset and update operations. The training time complexity depends on the data size, which is $O(\mathcal{Z} \times 2\$ \times n)$, $\mathcal{Z} \gg n \gg 2\$$, the total time complexity is $O(\mathcal{Z})$.

The unique overlapping subsequences are used to form the graph vertices. In worst case, the total number of subsequences in a TS of length n is $l - n + 1$, so the time complexity to search for unique subsequences is $O(l \log l)$ when $l \gg n$. The time complexity for vertex formation is $O(1)$. In worst case, the total possible number of graph edges is $(l - n + 1) \times 2\$$. Given $l \gg n \gg 2\$$, the time complexity to form the graph edges based on the threshold δ is $O(l \log l)$. The time complexity to compute the edge weights using Euclidean distance is $O(n)$. Therefore the total time complexity to form the graph edges is $O(l \log l)$.

Given $(l - n + 1)$ total vertices, the time complexity to update the weights of GCN model is $O(l \times n)$. For a η -layered GCN, the time complexity to train the GCN model is $O(\eta \times l \times n)$, that is, $O(l \times n)$.

5.4 Experiments

In this section, we discuss the validation measures and the statistical test, the baseline methods employed for comparative study. The CER-IRISH dataset discussed in section 1.1.1 is used in this study.

5.4.1 Validation measures

We validate the proposed GCN at two different levels and also perform the statistical test, which are discussed below.

- (a) **Performance of GRU-ATN:** The GRU-ATN model of the proposed GCN used at Level 1 is trained for each H_j . The overall performance is measured using the following.

- Symmetric mean absolute percentage error (SMAPE): Assuming q as the number of test samples for each H_j , the SMAPE is computed using Equation. 5.16. The mean SMAPE of the residential homes is computed

CHAPTER 5. CLASSIFICATION OF TIME SERIES USING GRAPH CONVOLUTIONAL NETWORK

as $mSMAPE(H_j) = \frac{1}{q} \sum_{j=1}^q SMAPE(T_q^{H_j})$.

$$SMAPE(T_t^{H_j}) = \frac{100}{n} \sum_{x=1}^n \frac{|O_t^{H_j}[x] - T_t^{H_j}[x]|}{(|O_t^{H_j}[x]| + |T_t^{H_j}[x]|)/2} \quad (5.16)$$

$$MAE(T_t^{H_j}) = \frac{\sum_{x=1}^n |O_t^{H_j}[x] - T_t^{H_j}[x]|}{n} \quad (5.17)$$

- Mean absolute error (MAE): Assuming q as the number of test samples, the MAE for a residential home is computed using Equation. 5.17. The mean MAE for H_j is computed as $mMAE(H_j) = \frac{1}{q} \sum_{j=1}^q MAE(T_t^{H_j})$.

(b) **Performance of GCN:** The performance of the GCN classifier used at Level 2 of the GCN is measured using the values of true positives, true negative, false positive and false negatives given as tp, tn, fp, fn respectively.

- Accuracy: The formula for computing accuracy is shown in Equation. 5.18.

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn} \quad (5.18)$$

$$F1 - score = 2 \frac{(precision * recall)}{(precision + recall)} \quad (5.19)$$

- F1-score: The formula for computing F1-score is given in Equation. 5.19, where, $precision = tp/(tp + fp)$ and $recall = tp/(tp + fn)$.

(c) Statistical test: Wilcoxon signed rank test (WSRT) is performed to prove the performance of the proposed GCN framework is significantly different than the others. According to \mathcal{H}_0 hypothesis of WSRT, the performance of proposed GCN framework is same as that of the other baseline techniques, while the \mathcal{H}_1 hypothesis of WSRT state that the performance of proposed GCN framework is significantly different than the others. The WSRT test rejects the \mathcal{H}_0 hypothesis in favour of \mathcal{H}_1 , if the P-value is less that the significance level $\alpha = 0.05$.

(d) Percentage change in power consumption (P_{change}): We compute the percentage change in power consumption for the predicted residential homes chosen for DSM, during the test period to that of the benchmark period as shown in Equation. 5.20.

$$P_{change} = \frac{p_t - p_b}{p_b} * 100 \quad (5.20)$$

where p_t is the mean power consumption during the test period and p_b is the mean power consumption during the benchmark period.

5.4.2 Baseline methods

Because the proposed GCN framework shows a graph learning approach using an encoder-decoder model, followed by classification model on the learned graph structure, we choose the baseline methods to distinguish that how is the graph based structure important for classification task as compared to the classification applied on the raw TS data of each residential home.

- Learning shapelets (Shapelet): Shapelets are the TS subsequences that best represents TS based on the minimum distance [203]. The brute-force shapelet discovery is a time consuming process and therefore, the shapelet learning model can efficiently reduce the search time and help in classification of TS [203].
- Symbolic aggregate approximation in vector space model (SAX-VSM): The subsequences extracted from the TS data using sliding window, are converted to symbols [204]. Next by using the term frequency-inverse document frequency (TF-IDF), the symbols are converted into weight vectors, which in turn, are used in classification.
- Principal component analysis (PCA): PCA is the dimensionality reduction approach in which the components can best explain the variance in the data are chosen in the lower dimension space [205]. The components extracted from PCA is fed to the fully connected neural network classifier.
- K-nearest neighbour (KNN): The KNN classifier [206] is applied on the historic TS data of the residential homes based on the computed Euclidean distance, where the value of $K=1$.
- Convolution neural network (CNN): CNN are the automatic feature extractors which consists of combination of the convolution and the pooling layers [207]. The features extracted from CNN is fed to the fully connected neural network classifier to identify the residential homes suitable for DSM programs.
- Long short term memory (LSTM): The LSTM models have the ability to learn from the sequential TS data [208], hence we train the raw TS on a 2-layer LSTM model followed by a fully connected neural network to identify the residential homes suitable for DSM programs.

CHAPTER 5. CLASSIFICATION OF TIME SERIES USING GRAPH CONVOLUTIONAL NETWORK

Table 5.1: Accuracy results of the proposed framework with the change in parameters like– Layers, Batch size (BS), number of hidden units (HU) and the optimizer in Level 1 and Level 2 (*Level 1-Level 2*). The best accuracy result is highlighted in bold.

Layers	BS	HU	Accuracy (Level 1-Level 2)			
			AdamW-Adam	Adam-Adam	AdamW-Adam	AdamW-AdamW
1	8	32	0.51	0.55	0.51	0.53
		64	0.58	0.5	0.67	0.53
	16	32	0.55	0.52	0.59	0.53
		64	0.59	0.54	0.56	0.54
	32	32	0.68	0.58	0.52	0.55
		64	0.68	0.56	0.58	0.58
	64	32	0.32	0.61	0.56	0.52
		64	0.58	0.612	0.55	0.57
2	8	32	0.53	0.62	0.58	0.59
		64	0.51	0.66	0.6	0.59
	16	32	0.53	0.63	0.65	0.54
		64	0.53	0.6	0.58	0.53
	32	32	0.73	0.62	0.6	0.57
		64	0.77	0.65	0.58	0.55
	64	32	0.72	0.67	0.6	0.5
		64	0.72	0.66	0.59	0.56

- Weighted directed graph (WDG): The DSM approach shown in [173] is used as a DSM based model here, for comparison. The unique symbols obtained from the symbolic representation of TS subsequences of each residential home is used to construct a directed weighted graph representation. The quasi clique clusters are obtained from the graph and the cluster labels obtained are used for the stability measure of the residential homes. Choice of residential homes for DSM operation is based on the entropy, computed on the cluster labels of each residential home [173].
- Graph attention network (GAN): The GAN is used as a DSM based model here, for comparison. The same graph structure obtained from GRU-ATN model of the proposed framework is used but at Level 2, the GAN model is applied for classification. Based on the existing graph representation, the GAN learns by assigning different weights to different neighbouring vertices as given in [209].

5.5 Results and analysis

In this section, we discuss the parameters of the proposed GCN framework. A parameter analysis is performed on the models used at Level 1 and Level 2 and the best results are reported. The results obtained from validation measures discussed in section 5.4.1 is shown. A comparative study of validation measures on the proposed GCN and the other baseline methods being discussed in section 5.4.2, is shown. Also, we analyse the energy consumption during the benchmark verses the test period for the residential homes chosen for DSM predicted using the proposed framework.

5.5.1 Model Parameters

Assuming the DSM operations will be operated for longer hours, the overlapping window length $n = 12$ hours in our experiments. Both the Level 1 and 2 of the GCN framework is trained using Adam and AdamW [210] optimizer each with a learning rate 0.001 and the best results are reported. The early stopping rule is applied with threshold of 0.05, that prevents the model degradation. GCN at both the Level 1 and 2 is trained for 100 epochs. The power consumption data for each H_j is split into training, validation and test sets in the ratio of 80 : 10 : 10.

The model accuracy obtained with varying number of parameters are reported in Table 5.1. The best accuracy value is obtained for a 2-layered GCN classifier with AdamW optimizer in Level 1 and Adam optimizer in Level 2, batch size of 32 and 64 hidden units. In other cases, no specific trend of increase or decrease is observed with the change in parameters of the model.

5.5.2 Graph structure results

The $mSMAPE$ value obtained from the GRU-ATN of Level 1 is 20.85% and the $mMAE$ value is 0.01. The minimum value of $SMAPE$ obtained is 0.79% and the MAE for the same residential home is 0.007. In Fig. 5.3, we show the predicted versus the actual results for a subsequence obtained from the building with minimum $SMAPE$ and MAE value.

We show a graph structure for a residential home with minimum $SMAPE$ and MAE value in Fig. 5.4 (a). For illustrative purpose, we show some of the vertices of the graph structure and the edges they are connected to, based on the threshold δ . Each subsequence of H_j is marked by its timestamp t . Considering subsequence #49 shown in Fig. 5.4, for which $\delta = 0.0104$, we find that #49 is connected by an edge to #29, #30, #31, #34, #38, which shows that the respective subsequences are more important for predicting #49 as compared to the others. On the other hand, for prediction of subsequence number #50 the subsequences #30, #31, #32, #35, #39 are more important. We show the patterns of respective subsequences which are

CHAPTER 5. CLASSIFICATION OF TIME SERIES USING GRAPH CONVOLUTIONAL NETWORK

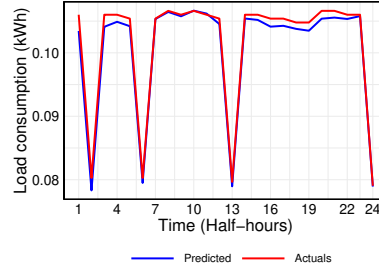


Figure 5.3: Predicted versus the actual results of a subsequence for the residential home that shows lowest $SMAPE$ and MAE value in case of the proposed method.

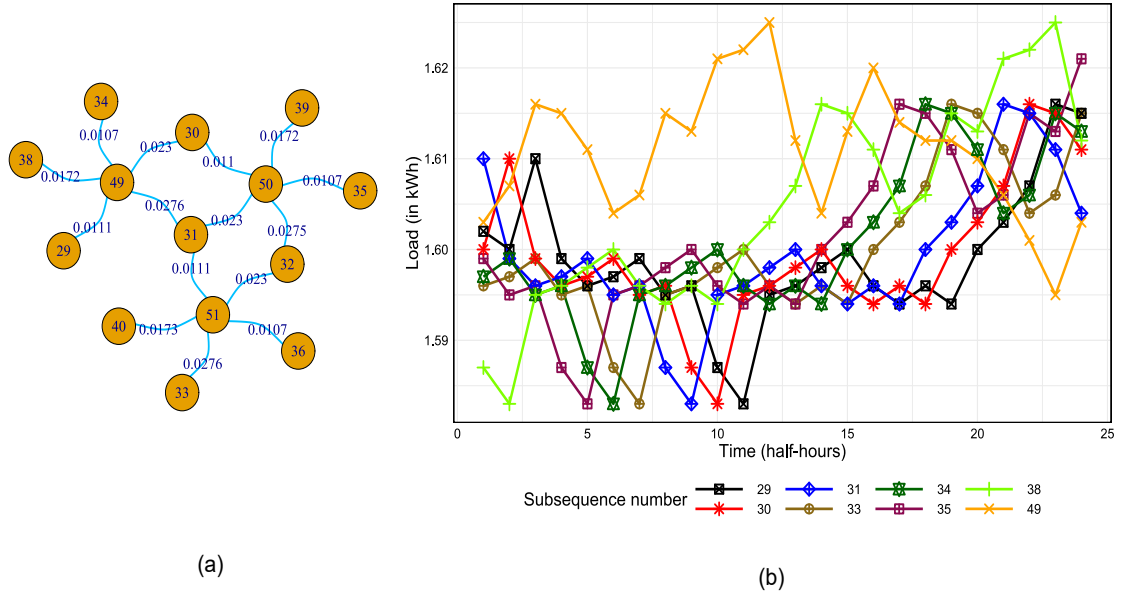


Figure 5.4: (a) A graph structure obtained from the proposed framework for the residential home with lowest $SMAPE$ and MAE value. The undirected edges show attention coefficient values (b) the subsequence patterns of the respective vertices of the graph shown.

connected to subsequence #49, in Fig. 5.4 (b). It is observed that subsequences #29, #30, #31, #34, #38, are similar patterns because of the overlapping nature. In contrast, we show the pattern of subsequence #33 and #38, both of which also follow the same pattern but is still not connected to #49 because of the lower attention coefficient. This shows the significance of the attention coefficient obtained from the GRU-ATN.

Table 5.2: Comparative study of the proposed framework in terms of accuracy and F1-score. The best values obtained are highlighted in bold.

Method	Metrics	
	Accuracy	F1-score
Shapelet	0.53	0.53
SAX-VSM	0.5	0.66
PCA	0.51	0.44
1-KNN	0.5	0.56
CNN	0.56	0.53
LSTM	0.57	0.54
WDG	0.68	0.8
GAN	0.75	0.27
GCN_ablation	0.62	0.61
GCN	0.77	0.74

5.5.3 Comparative study

In Table 5.2, we show the accuracy and F1-score obtained from the respective models. The performance of the proposed GCN is highlighted in bold. The comparative study is done at following different aspects:

1. Comparison with baseline methods: In Table 5.2, we report the accuracy and the F1-scores of the proposed GCN framework and the other baseline models. It is observed that the GCN shows an improvement in accuracy of upto 27% and improvement in F1-score of upto 24% as compared to-Shapelet, SAX-VSM, PCA, 1-KNN, CNN and LSTM methods.
2. Comparison with other DSM based models: In Table 5.2, we report the accuracy and the F1-scores of the proposed GCN framework and the other DSM based models- WDG and GAN. An improvement of 9% in accuracy score is observed in case of GCN as compared to WDG. Although the F1-score is best in case of WDG method, the WSRT gives the P-value of 0.002, thus rejecting the \mathcal{H}_0 hypothesis, which proves that the performance of the proposed GCN is significantly different than the others.
The accuracy in case of proposed GCN is 2% better than that of GAN model, but, there is a significant improvement in F1-score.
3. Comparison with a GCN model without attention (GCN_ablation): To study

CHAPTER 5. CLASSIFICATION OF TIME SERIES USING GRAPH CONVOLUTIONAL NETWORK

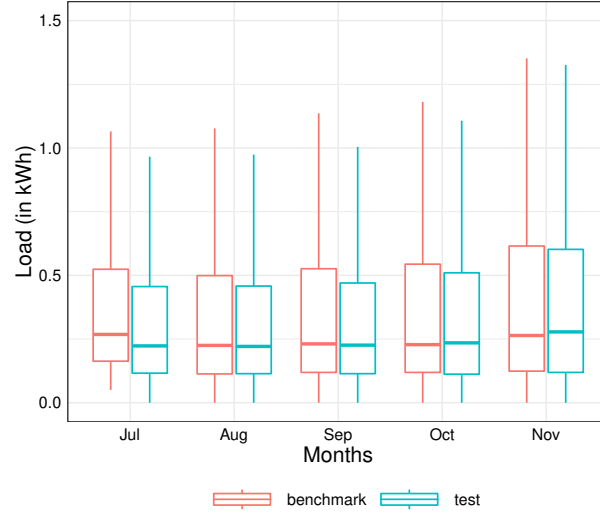


Figure 5.5: The power consumed by the residential homes predicted using the proposed GCN framework, during the test period- Jul' 2010 to Nov'2010 over that of the benchmark period- Jul' 2009 to Nov'2009.

the importance of attention coefficients at Level 1 of the proposed GCN framework, we apply a GRU-ATN without the attention layer on it. With no attention coefficients provided, we create a complete graph for each residential home and train it on the GCN classifier and report the results. This model is referred as GCN_ablation given in Table 5.2.

A significant improvement in accuracy and F1-score in the proposed GCN framework over that of GCN_ablation prove the effectiveness of attention used, as given in Table 5.2.

5.5.4 Energy consumption during benchmark vs test period

We illustrate the amount of power consumed during the test period- Jul' 2010 to Nov'2010 over that of the benchmark period- Jul'2009 to Nov'2009 by the residential homes, in Fig. 5.5.

An overall decrease in P_{change} by 10%, 2.2%, 4.4%, 2.1% and 1.9% is observed during Jul, Aug, Sep and Oct and Nov of the test period as compared to that of the benchmark period, respectively. In order to show the impact of DSM operation on the residential grid, we compute the load factor of the predicted residential set suitable for the DSM operation. The daywise load factor of a residential home is the ratio of average demand of the day to that of maximum demand during the day. The higher value of load factor signify that the maximum demand during the day is low and the consumer has a consistent electricity consumption whereas

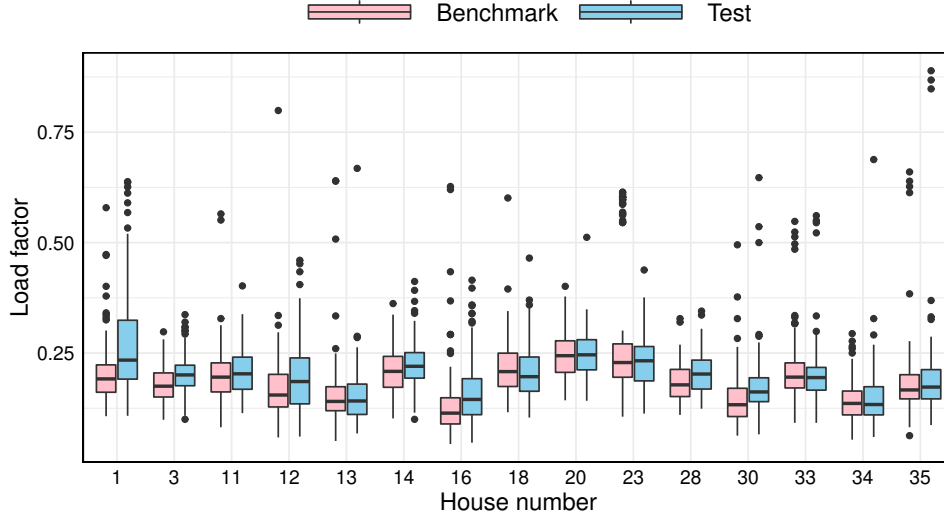


Figure 5.6: Load factor of the predicted set of residential homes for DSM operation during the benchmark and the test period. The x-axis denote the residential house number (*House number*) while the y-axis denote the load factor.

the low value of load factor imply that the maximum demand is high during the day. The Fig. 5.6 shows the load factor for some of the residential homes predicted as suitable for DSM. The mean load factor observed in the boxplot during the test period is greater than that of the benchmark period, which imply that the predicted residential homes maintain a consistent electricity consumption.

5.6 Conclusions

In this chapter, we propose a Graph Convolutional Network based (GCN) framework that classify the TS samples based on their regularity. An automatic weighted undirected graph structure is formed for each TS sample using an attention based encoder-decoder model. The graph structure can store the subsequent co-occurring pattern based information of a TS sample. The vertices of the graph is formed from the unique overlapping subsequences of the historic TS and edges are based on the attention values. The model is evaluated on a labeled residential grid dataset each with half hourly power consumption values over an year. A comparative analysis with some of the baseline classifiers as well as the existing DSM models, show improvement in accuracy results in case of the proposed framework. The results show a decrease in the overall load consumption when the predicted residential homes from the proposed GCN framework are actually chosen for demand side management operations.

Conclusions and Future Scope of Research

Time series analysis (TSA) is a century old research field that interpret information from the historic data points accumulated over time. Time series modeling and interpretation of patterns started with the development of statistical tools like the autoregressive models, named AR [21]. Since then, several statistical tools are developed for TSA. However, scalability of the statistical tools to handle large complex data is limited. To capture the nonlinear relationship from historic data and to achieve higher accuracy results, development of ML techniques in parallel to the statistical tools are observed in literature. The key advantages of ML techniques are 1. automatic feature extraction from historic TS 2. ability to handle large volumes of data across TS domains. Some of the recent ML techniques for TS modeling and information interpretation include the deep neural networks and the sequential models.

Amongst the several TSA problems studied in literature since last century, analysis of co-occurring patterns is rather less explored. In this thesis, we aim to answer the following research questions,

1. obtain the frequently occurring longest temporal patterns from the TS dataset.
2. extract the atypical patterns from TS dataset
3. identify the co-occurring patterns at multiple time granularity
4. identify the regular TS samples

. In order to address the research questions, it is important to obtain a TS representation suitable for multiple TSA. In conventional symbolic aggregate approximation (SAX) based TS representation, the subsequences are represented using the mean values. As a result, the SAX techniques are not adequate to capture enough information about the underlying patterns or pattern shift. Several variations of SAX are proposed by adding other handcrafted features alongwith the mean [111, 204, 121], that overcome the limitations of SAX. But adding too many feature values for each of the overlapping subsequences, can cause redundancy

with increased time complexity. Particularly for the problems of analysing the co-occurring patterns, it is challenging to have a TS representation that preserve the temporal and the structural similarity between the subsequences.

In this scope of research, we exploit graph structures to represent TS dataset so that the temporal and structural similarity based information is well preserved.

The first contributory work of this thesis propose a framework that facilitate pattern mining, atypical pattern discovery from the TS dataset, by using a graph based TS representation. This chapter addresses 1. how to represent the TS dataset so that the pattern information is retained 2. how to identify the frequently occurring longest temporal patterns and the atypical patterns. A weighted directed graph that uses the graph properties like vertex degree, Eigen vector centrality is used to represent the TS dataset. The directed minimum length paths are extracted from a single source in graph structure for subsequent path based clustering to successfully extract the frequently occurring longest temporal patterns. The path discovery help to find the different temporal patterns that exist in the TS dataset. The possible variants of the proposed graph structure are considered and the best graph structure is determined based on statistical test. The graph component score is used to obtain the temporal dependent rare events in the TS dataset. The component score for each of the graph component determine what ratio of edges is present in the desired graph component out of the total number of edges in the graph. A threshold value is applied on the graph component scores, to obtain the atypical patterns. It is observed that the discovered atypical patterns are rare in the TS dataset.

The proposed framework is experimented on multiple TS dataset. The cluster quality and the statistical tests show that the predicted labels using proposed method, are more similar to the target labels as compared to existing clustering graph based approaches. In case of the electricity consumption datasets– London and Ausgrid, the proposed method uncovers the frequently occurring longest electricity usage patterns in the dataset. Knowing such patterns can help the utility providers to build policies to mitigate sudden rise/fall in demand. On the contrary, the identified atypical patterns of electricity usage in the datasets are indicative of possibly faulty meter or devices. The discovery of temporal patterns from the clusters in case of Stock market data show group of companies that have similar closing price during a certain range of date. This information can be used to design recommender systems that can predict when to buy or sell the stocks. The atypical pattern discovery in case of Stock market data can be used as a precursor identification that is, a previous alert of atypical pattern in the near future. The temporal pattern discovery in case of Web traffic dataset will help to assign resources to the web pages, based on their frequency of occurrence. The atypical pattern results in Web traffic dataset, show peak values, which means,

CHAPTER 6. CONCLUSIONS AND FUTURE SCOPE OF RESEARCH

designing a recommender system can help to know how to assign the resources in case of sudden peak.

Therefore, the proposed framework help to uncover useful information about the temporal patterns, with assumptions that TS dataset demonstrate more than one unique subsequence pattern. In case there exists only one unique pattern in the entire TS dataset, the proposed method identifies a single cluster. On the contrary, if all symbols in TS dataset are unique, the proposed method finds each TS sample as a singleton cluster. However both the scenarios are rarely observed in real life TS datasets and hence such trivial cases of clustering can be ignored.

The dense coherent substructures found in the graph imply similar co-occurring subsequences of the underlying TS dataset. With this hypothesis, the second contributory work of the thesis proposes a quasi-clique clustering approach of TS subsequences on a weighted directed graph. While the unique symbols encode the structural patterns of the TS dataset, the edge represent the similarity between the patterns. The problem of finding all maximal quasi clique structure from a weighted directed graph is a computationally hard problem and therefore a greedy bottom-up approach is proposed such that every cluster forms a maximal quasi-clique. The chapter addresses how to identify the subsequently occurring patterns and the atypical patterns. To overcome the limitations of the SAX distance measure, a modified SAX distance is used for edge weight computation. An extensive analysis on multiple TS dataset is shown. Higher cluster quality measures in case of proposed method signify that the cluster samples are more tightly located as compared to other state-of-the-art graph clustering techniques. The chapter proposes a method to determine the regularity of TS samples using change in cluster labels. The cluster representative patterns reveal the peak and the fall occurrences during the different time of the day in case of load consumption dataset.

The proposed work can help the utility providers in balancing the demand with the supply under generation uncertainty caused due to renewable energy integration in the grid. The stable buildings identified based on regularity measure, are likely to participate in DR programs through long term contractual agreement. Introducing promotional offers to the stable consumer set who can convince their neighbors towards energy management will further help in load reduction. A case study on CER-IRISH smart grids and buildings dataset confirm a reduction in overall load consumption as a result when the buildings are chosen for DSM operation. The choice of a threshold value on entropy measure is subjective and can be automated. The atypical patterns are extracted by examining the frequency of occurrence of the symbols and the degree of the graph vertices. The discovery of atypical patterns in case of the London and Ausgrid datasets reveal that some of the buildings in case of the London and Ausgrid dataset show unusual behavioral pattern. The statistical test reveal that frequency of occurrences of the atypical

patterns is different from the frequency of occurrence of any randomly chosen normal pattern.

The third chapter of this thesis addresses how to identify the subsequently co-occurring patterns at the different time granularity. Mining the subsequently co-occurring patterns at multiple time granularity can help to obtain the patterns that are common across the different time granularity and the unique co-occurring patterns observed at different time granularity. Assuming each TS sample to be located at a designated place which has similar weather conditions, the TS samples are divided into seasons. Multiple graph networks is obtained from each season and a consensus network is formed. Advantage of forming multiple graph is that the cluster labels obtained from each graph structure help to compute the regularity of TS samples at each time granularity and the consensus graph help to obtain regularity based on yearly analysis. Unlike the handcrafted feature set as used in the previous chapters, this chapter shows an automated feature extraction approach from TS subsequences. The proposed method uses an LSTM autoencoder based model. The SAX is applied on the features that form multiple graph structure. The weakly connected component considered for quasi-clique discovery can cause loss of important patterns. Therefore with an improvement to strongly connected components of the weighted directed graph, a greedy quasi-clique clustering technique is shown on each graph structure, that represent dense coherent substructures at each time granularity. In the experimental study, the statistical test show that the cluster samples in case of the proposed method are closer as compared to the other state-of-the-art graph clustering techniques. The statistical test also confirm that the cluster quality results obtained from the previous chapters' clustering on graph structure is different from this proposed method.

The cluster representative profiles are extracted for each TS dataset, which show that the load profiles will not necessarily differ during each season. Therefore the rise/fall in patterns for the clusters of different seasons can be similar too. The similar and dissimilar co-occurring patterns are extracted at each time granularity from the smart grid and building datasets. The similar patterns will further help to identify the appliances of a building that are used subsequently, or the appliances that are dependent. The dissimilar patterns give the idea of change in appliance consumption during the different time granularities or identifying a newer pattern observed, which can be the cause of faulty occurrence. The experimental study also discover which of the appliances are stable on daily, seasonal and yearly basis in case of a seven storeyed office building. The predicted appliance set can be chosen for DSM operation. Therefore the study can successfully obtain the clusters provided that the quasi-clique structures are discovered from the multiple weighted directed graph structures. In case if all the TS load profiles show periodic patterns, then the proposed method places each vertex to a single cluster. But this

CHAPTER 6. CONCLUSIONS AND FUTURE SCOPE OF RESEARCH

scenario is rarely observed in case of electricity consumption dataset.

Classifying the TS samples based on their consistency in TS patterns help to automatically identify the TS samples that show a regularity in the patterns. An application where this automation is required is the smart grids and buildings. An automated system that can identify the consumers with regular consumption behavior will help the utilities in choosing the appropriate consumers for DSM operations. In this regard, the fourth contributory work of this thesis propose a TS classification framework using the Graph convolutional networks. Considering that the non-overlapping subsequences can cause pattern loss, in this work overlapping subsequences are obtained, with a predefined degree of overlap. The proposed method automatically obtain a directed unweighted graph structure from each TS sample using a state-of-the-art encoder-decoder model. The graph edges are formed based on the attention values generated from state-of-the-art encoder-decoder model. A threshold value is assumed on the attention values, for the edge connectivity. While the vertices represent unique patterns from the overlapping set of subsequences, edges denote the importance of previous patterns when predicting the future and, the edge weights denote similarity between the patterns. The buildings from CER-IRISH dataset, that have a regular consumption behavior, is identified. The identified set show a significant decrease in load consumption when chosen for DSM operations. A comparative analysis on accuracy and F1-score with the other traditional and state-of-the-art classification models show better results in case of the proposed framework. In case if most of the TS samples have similar patterns, that will result to similar graph structure formation, causing redundant information in training data.

There is a scope of automating the proposed work in terms of the following parameters: 1. the subsequence length, path discovery length can be automated. 2. the threshold assumption applied for atypical pattern extraction from the TS datasets 3. the threshold assumption for the identification of regular TS samples 4. automating the edge formation technique in case of graph learning rather than using the attention values. The additional features of the subsequences, for example, capturing the peak value in a subsequence will reveal additional insights. In regard to atypical pattern analysis, many other graph based measures can be used namely the Eigen vector centrality that consider the importance of a node with respect to its neighbors. Identifying the differences in anomaly when using the graph components verses the Eigen vector centrality will help further in mining. In case of classification of TS datasets, reducing the number of graph structure formed based on a similarity rule between the TS samples can effectively reduce the space complexity.

The future research directions in general, are listed below. 1. the proposed graph theoretical and ML techniques can be improved by creating dynamic graphs

that can identify the changing patterns or the patterns that do not change. 2. what are the results obtained using other path discovery algorithms in case of weighted and unweighted graph and what are the differences in the clustering results. 3. how to ensure the discovery of subsequently co-occurring patterns in case of streaming TS sample or the TS dataset with streaming samples? 4. how to create a framework that can obtain the important events in TS dataset in addition to the atypical patterns. The definition of important events can be TS domain specific. For example, an exceptionally high internet usage during each day. 5. how to identify the patterns that cause irregularity in TS samples ? 6. how to represent the multivariate TS dataset using a graph structure. 7. how to measure the regularity in case of multivariate TS dataset ? 8. what are the different TS domain where the proposed graph structure and the measure of regularity is of importance.

Bibliography

- [1] E. D. Feigelson, G. J. Babu, and G. A. Caceres, "Autoregressive times series methods for time domain astronomy," *Frontiers in Physics*, vol. 6, 2018. [Online]. Available: <https://www.frontiersin.org/articles/10.3389/fphy.2018.00080>
- [2] Z. R. McCaw, J. M. Lane, R. Saxena, S. Redline, and X. Lin, "Operating characteristics of the rank-based inverse normal transformation for quantitative trait analysis in genome-wide association studies," *Biometrics*, vol. 76, no. 4, pp. 1262–1272, 2020.
- [3] F. Jiang, Z. Zhao, and X. Shao, "Time series analysis of covid-19 infection curve: A change-point perspective," *Journal of Econometrics*, vol. 232, no. 1, pp. 1–17, 2023.
- [4] G. Zhang, V. Davoodnia, A. Sepas-Moghaddam, Y. Zhang, and A. Etemad, "Classification of hand movements from eeg using a deep attention-based lstm network," *IEEE Sensors Journal*, vol. 20, no. 6, pp. 3113–3122, 2020.
- [5] L. Anghinoni, L. Zhao, D. Ji, and H. Pan, "Time series trend detection and forecasting using complex network topology analysis," *Neural Networks*, vol. 117, pp. 295–306, 2019.
- [6] Y. Zhang, Y. Wang, and F. Ma, "Forecasting us stock market volatility: How to use international volatility information," *Journal of Forecasting*, vol. 40, no. 5, pp. 733–768, 2021.
- [7] E. Lee, J. Kim, and D. Jang, "Load profile segmentation for effective residential demand response program: Method and evidence from korean pilot study," *Energies*, vol. 13, no. 6, 2020.
- [8] A. Bhujel, N.-E. Kim, E. Arulmozhi, J. K. Basak, and H.-T. Kim, "A lightweight attention-based convolutional neural networks for tomato leaf disease classification," *Agriculture*, vol. 12, no. 2, 2022. [Online]. Available: <https://www.mdpi.com/2077-0472/12/2/228>
- [9] H.-H. Jo, T. Hiraoka, and M. Kivelä, "Burst-tree decomposition of time series reveals the structure of temporal correlations," *Scientific reports*, vol. 10, no. 1, p. 12202, 2020.
- [10] ISSDA, "CER smart metering project electricity customer behaviour trial accessed via the irish social science data," www.ucd.ie/issda, 2012.
- [11] LondonDatastore, "Smartmeter energy consumption data in london households," <https://data.london.gov.uk/dataset/smartmeter-energy-use-data-in-london-households>, 2015.

- [12] Ausgrid, "Solar home electricity data," <https://www.ausgrid.com.au/Industry/Our-Research/Data-to-share/Solar-home-electricity-data>, 2020.
- [13] V. K. W. C. Maggie, Oren Anava, "Web traffic time series forecasting," 2017. [Online]. Available: <https://kaggle.com/competitions/web-traffic-time-series-forecasting>
- [14] M. Pipattanasomporn, G. Chitalia, J. Songsiri, C. Aswakul, W. Pora, S. Suwankawin, K. Audomvongseree, and N. Hoonchareon, "CU-BEMS, smart building electricity consumption and indoor environmental sensor datasets," *Scientific Data*, vol. 7, no. 1, pp. 1–14, 2020.
- [15] D. C. Montgomery, C. L. Jennings, and M. Kulahci, *Introduction to time series analysis and forecasting*. John Wiley & Sons, 2015.
- [16] G. Zhang and M. Qi, "Neural network forecasting for seasonal and trend time series," *European Journal of Operational Research*, vol. 160, no. 2, pp. 501–514, 2005, decision Support Systems in the Internet Age.
- [17] H. S. Wauchope, T. Amano, J. Geldmann, A. Johnston, B. I. Simmons, W. J. Sutherland, and J. P. Jones, "Evaluating impact using time-series data," *Trends in Ecology & Evolution*, vol. 36, no. 3, pp. 196–205, 2021.
- [18] D. V. Glass, "John graunt and his natural and political observations," *Notes and Records of the Royal Society of London*, vol. 19, no. 1, pp. 63–100, 1964. [Online]. Available: <https://royalsocietypublishing.org/doi/abs/10.1098/rsnr.1964.0006>
- [19] J. L. Klein, *Statistical Visions in Time*, ser. Cambridge Books. Cambridge University Press, December 2005, no. 9780521023177. [Online]. Available: <https://ideas.repec.org/b/cup/cbooks/9780521023177.html>
- [20] G. U. Yule, "On a method of investigating periodicities disturbed series, with special reference to wolfer's sunspot numbers," *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, vol. 226, no. 636–646, pp. 267–298, 1927.
- [21] R. S. Tsay, "Time series and forecasting: Brief history and future research," *Journal of the American Statistical Association*, vol. 95, no. 450, pp. 638–643, 2000.
- [22] C. Chatfield and H. Xing, *The analysis of time series: an introduction with R*. CRC press, 2019.
- [23] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*. OTexts, 2018.

BIBLIOGRAPHY

- [24] F. Grassi, A. Loukas, N. Perraudin, and B. Ricaud, "A time-vertex signal processing framework: Scalable processing and meaningful representations for time-series on graphs," *IEEE Transactions on Signal Processing*, vol. 66, no. 3, pp. 817–829, 2017.
- [25] D. Li, Y. Zhao, and Y. Li, "Time-series representation and clustering approaches for sharing bike usage mining," *IEEE Access*, vol. 7, pp. 177 856–177 863, 2019.
- [26] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A symbolic representation of time series, with implications for streaming algorithms," in *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, 2003, pp. 2–11.
- [27] E. J. Keogh and M. J. Pazzani, "Scaling up dynamic time warping for datamining applications," in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2000, pp. 285–289.
- [28] J. Lin, E. Keogh, L. Wei, and S. Lonardi, "Experiencing sax: a novel symbolic representation of time series," *Data Mining and knowledge discovery*, vol. 15, pp. 107–144, 2007.
- [29] M. Linardi, Y. Zhu, T. Palpanas, and E. Keogh, "Matrix profile goes mad: variable-length motif and discord discovery in data series," *Data Mining and Knowledge Discovery*, vol. 34, pp. 1022–1071, 2020.
- [30] S. Basu, K. Mishra, and U. Maulik, "Analyzing load profiles in commercial buildings using smart meter data," *Towards Energy Smart Homes: Algorithms, Technologies, and Applications*, pp. 463–487, 2021.
- [31] A. Shifaz, C. Pelletier, F. Petitjean, and G. I. Webb, "Ts-chief: a scalable and accurate forest algorithm for time series classification," *Data Mining and Knowledge Discovery*, vol. 34, no. 3, pp. 742–775, 2020.
- [32] B. Bai, G. Li, S. Wang, Z. Wu, and W. Yan, "Time series classification based on multi-feature dictionary representation and ensemble learning," *Expert Systems with Applications*, vol. 169, p. 114162, 2021.
- [33] L. Lacasa, B. Luque, F. Ballesteros, J. Luque, and J. C. Nuño, "From time series to complex networks: The visibility graph," *Proceedings of the National Academy of Sciences*, vol. 105, no. 13, pp. 4972–4975, 2008.
- [34] M. Stephen, C. Gu, and H. Yang, "Visibility graph based time series analysis," *PloS one*, vol. 10, no. 11, p. e0143015, 2015.

- [35] Y. Yang and H. Yang, "Complex network-based time series analysis," *Physica A: Statistical Mechanics and its Applications*, vol. 387, no. 5, pp. 1381–1386, 2008.
- [36] Y. Zhou, S. Zhang, L. Wu, and Y. Tian, "Predicting sectoral electricity consumption based on complex network analysis," *Applied Energy*, vol. 255, p. 113790, 2019.
- [37] Y. Zou, R. V. Donner, N. Marwan, J. F. Donges, and J. Kurths, "Complex network approaches to nonlinear time series analysis," *Physics Reports*, vol. 787, pp. 1–97, 2019.
- [38] P. R. Winters, "Forecasting sales by exponentially weighted moving averages," *Management Science*, vol. 6, no. 3, pp. 324–342, 1960. [Online]. Available: <https://doi.org/10.1287/mnsc.6.3.324>
- [39] J. G. De Gooijer and R. J. Hyndman, "25 years of time series forecasting," *International Journal of Forecasting*, vol. 22, no. 3, pp. 443–473, 2006.
- [40] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 03 1960. [Online]. Available: <https://doi.org/10.1115/1.3662552>
- [41] J. Ord, "Future developments in forecasting: The time series connexion," *International Journal of Forecasting*, vol. 4, no. 3, pp. 389–401, 1988.
- [42] K. Mishra, S. Basu, and U. Maulik, "Danse: A dilated causal convolutional network based model for load forecasting," in *Pattern Recognition and Machine Intelligence*, B. Deka, P. Maji, S. Mitra, D. K. Bhattacharyya, P. K. Bora, and S. K. Pal, Eds. Cham: Springer International Publishing, 2019, pp. 234–241.
- [43] A. F. Faisal, A. Rahman, M. T. M. Habib, A. H. Siddique, M. Hasan, and M. M. Khan, "Neural networks based multivariate time series forecasting of solar radiation using meteorological data of different cities of bangladesh," *Results in Engineering*, vol. 13, p. 100365, 2022.
- [44] N. A. Funde, M. M. Dhabu, A. Paramasivam, and P. S. Deshpande, "Motif-based association rule mining and clustering technique for determining energy usage patterns for smart meter data," *Sustainable Cities and Society*, vol. 46, p. 101415, 2019.
- [45] C.-C. M. Yeh, Y. Zhu, L. Ulanova, N. Begum, Y. Ding, H. A. Dau, D. F. Silva, A. Mueen, and E. Keogh, "Matrix profile i: All pairs similarity joins for time series: A unifying view that includes motifs, discords and shapelets,"

BIBLIOGRAPHY

- in *2016 IEEE 16th International Conference on Data Mining (ICDM)*, 2016, pp. 1317–1322.
- [46] H. Tang and S. S. Liao, “Discovering original motifs with different lengths from time series,” *Knowledge-Based Systems*, vol. 21, no. 7, pp. 666–671, 2008.
- [47] Y. Gao and J. Lin, “Hime: discovering variable-length motifs in large-scale time series,” *Knowledge and Information Systems*, vol. 61, pp. 513–542, 2019.
- [48] B. Zhu, Y. Jiang, M. Gu, and Y. Deng, “A gpu acceleration framework for motif and discord based pattern mining,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 8, pp. 1987–2004, 2021.
- [49] S. Buffett, “Discretized sequential pattern mining for behaviour classification,” *Granular Computing*, vol. 6, pp. 853–866, 2021.
- [50] Y. Wu, D. Weng, Z. Deng, J. Bao, M. Xu, Z. Wang, Y. Zheng, Z. Ding, and W. Chen, “Towards better detection and analysis of massive spatiotemporal co-occurrence patterns,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3387–3402, 2021.
- [51] J. Li, S. Chen, K. Zhang, G. Andrienko, and N. Andrienko, “Cope: Interactive exploration of co-occurrence patterns in spatial time series,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 8, pp. 2554–2567, 2019.
- [52] C. Panagiotakis, K. Papoutsakis, and A. Argyros, “A graph-based approach for detecting common actions in motion capture data and videos,” *Pattern Recognition*, vol. 79, pp. 1–11, 2018.
- [53] C. Fan, F. Xiao, M. Song, and J. Wang, “A graph mining-based methodology for discovering and visualizing high-level knowledge for building energy management,” *Applied Energy*, vol. 251, p. 113395, 2019.
- [54] A. Blázquez-García, A. Conde, U. Mori, and J. A. Lozano, “A review on outlier/anomaly detection in time series data,” *ACM Comput. Surv.*, vol. 54, no. 3, apr 2021.
- [55] S. Schmidl, P. Wenig, and T. Papenbrock, “Anomaly detection in time series: A comprehensive evaluation,” *Proc. VLDB Endow.*, vol. 15, no. 9, p. 1779–1797, may 2022. [Online]. Available: <https://doi.org/10.14778/3538598.3538602>
- [56] E. Keogh, J. Lin, and A. Fu, “Hot sax: efficiently finding the most unusual time series subsequence,” in *Fifth IEEE International Conference on Data Mining (ICDM’05)*, 2005, pp. 8 pp.–.

- [57] N. Laptev, S. Amizadeh, and I. Flint, "Generic and scalable framework for automated time-series anomaly detection," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2015, p. 1939–1947.
- [58] C. Wang, K. Viswanathan, L. Choudur, V. Talwar, W. Satterfield, and K. Schwan, "Statistical techniques for online anomaly detection in data centers," in *12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops*, 2011, pp. 385–392.
- [59] S. Muthukrishnan, R. Shah, and J. Vitter, "Mining deviants in time series data streams," in *Proceedings. 16th International Conference on Scientific and Statistical Database Management, 2004.*, 2004, pp. 41–50.
- [60] K. Mishra, S. Basu, and U. Maulik, "Mining representative load profiles in commercial buildings," in *Proceedings of the 7th International Conference on Advances in Energy Research*. Springer Singapore, 2021, pp. 1025–1036.
- [61] R. J. Hyndman, E. Wang, and N. Laptev, "Large-scale unusual time series detection," in *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, 2015, pp. 1616–1619.
- [62] S.-E. Benkabou, K. Benabdeslem, and B. Canitia, "Unsupervised outlier detection for time series by entropy and dynamic time warping," *Knowledge and Information Systems*, vol. 54, no. 2, pp. 463–486, 2018.
- [63] P. Boniol and T. Palpanas, "Series2graph: Graph-based subsequence anomaly detection for time series," *Proceedings of the VLDB Endowment*, vol. 13, no. 12, pp. 1821–1834, 2020.
- [64] F. Yang, K. Fan, D. Song, and H. Lin, "Graph-based prediction of protein-protein interactions with attributed signed graph embedding," *BMC bioinformatics*, vol. 21, no. 1, pp. 1–16, 2020.
- [65] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-gcn: A temporal graph convolutional network for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3848–3858, 2019.
- [66] C. Zhou, L. Ding, Y. Zhou, and M. J. Skibniewski, "Visibility graph analysis on time series of shield tunneling parameters based on complex network theory," *Tunnelling and Underground Space Technology*, vol. 89, pp. 10–24, 2019.

BIBLIOGRAPHY

- [67] D. He, Z. Tang, Q. Chen, Z. Han, D. Zhao, and F. Sun, "A two-stage deep graph clustering method for identifying the evolutionary patterns of the time series of animation view counts," *Information Sciences*, vol. 642, p. 119155, 2023.
- [68] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-gcn: A temporal graph convolutional network for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3848–3858, 2020.
- [69] W. Li, R. Bao, K. Harimoto, D. Chen, J. Xu, and Q. Su, "Modeling the stock relation with graph network for overnight stock movement prediction," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, C. Bessiere, Ed. International Joint Conferences on Artificial Intelligence Organization, 7 2020, pp. 4541–4547.
- [70] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1003–1012.
- [71] L. Anghinoni, L. Zhao, D. Ji, and H. Pan, "Time series trend detection and forecasting using complex network topology analysis," *Neural Networks*, vol. 117, pp. 295–306, 2019.
- [72] S.-V. Sanei-Mehri, A. Das, H. Hashemi, and S. Tirthapura, "Mining largest maximal quasi-cliques," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 15, no. 5, pp. 1–21, 2021.
- [73] C. Tsourakakis, F. Bonchi, A. Gionis, F. Gullo, and M. Tsiarli, "Denser than the densest subgraph: Extracting optimal quasi-cliques with quality guarantees." Association for Computing Machinery, 2013.
- [74] P. Conde-Cespedes, "Approaching the optimal solution of the maximal α -quasi-clique local community problem," *Electronics*, vol. 9, no. 9, p. 1438, 2020.
- [75] P. Bindu, P. S. Thilagam, and D. Ahuja, "Discovering suspicious behavior in multilayer social networks," *Computers in Human Behavior*, vol. 73, pp. 568–582, 2017.
- [76] K. Mishra, S. Basu, and U. Maulik, "A segmentation based similarity measure for time series data," in *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*, ser. CoDS COMAD 2020. New York, NY,

- USA: Association for Computing Machinery, 2020, p. 351–352. [Online]. Available: <https://doi.org/10.1145/3371158.3371221>
- [77] A. Satre-Meloy, M. Diakonova, and P. Grünewald, “Cluster analysis and prediction of residential peak demand profiles using occupant activity data,” *Applied Energy*, vol. 260, p. 114246, 2020.
- [78] X. Liu, H. Sun, S. Han, S. Han, S. Niu, W. Qin, P. Sun, and D. Song, “A data mining research on office building energy pattern based on time-series energy consumption data,” *Energy and Buildings*, vol. 259, p. 111888, 2022.
- [79] R. Zhang, J. Zhang, Q. Wang, and H. Zhang, “Doids: An intrusion detection scheme based on dbscan for opportunistic routing in underwater wireless sensor networks,” *Sensors*, vol. 23, no. 4, 2023.
- [80] D. Fischer, E. B. Klerman, and A. J. K. Phillips, “Measuring sleep regularity: theoretical properties and practical usage of existing metrics,” *Sleep*, vol. 44, no. 10, p. zsab103, 04 2021. [Online]. Available: <https://doi.org/10.1093/sleep/zsab103>
- [81] A. Rajabi, M. Eskandari, M. Jabbari Ghadi, S. Ghavidel, L. Li, J. Zhang, and P. Siano, “A pattern recognition methodology for analyzing residential customers load data and targeting demand response applications,” *Energy and Buildings*, vol. 203, p. 109455, 2019.
- [82] I. Revin, V. A. Potemkin, N. R. Balabanov, and N. O. Nikitin, “Automated machine learning approach for time series classification pipelines using evolutionary optimization,” *Knowledge-Based Systems*, vol. 268, p. 110483, 2023.
- [83] N. Sarafian Ben Ari and R. Moskovitch, “Predictive temporal patterns discovery,” *Expert Systems with Applications*, vol. 226, p. 119974, 2023.
- [84] “Visibility graph analysis on quarterly macroeconomic series of china based on complex network theory,” *Physica A: Statistical Mechanics and its Applications*, vol. 391, no. 24, pp. 6543–6555, 2012.
- [85] T.-T. Zhou, N.-D. Jin, Z.-K. Gao, and Y.-B. Luo, “Limited penetrable visibility graph for establishing complex network from time series,” *Acta Physica Sinica*, vol. 61, no. 3, p. 030506, 2012.
- [86] M. Small, “Complex networks from time series: Capturing dynamics,” in *2013 IEEE International Symposium on Circuits and Systems (ISCAS2013)*. IEEE, 2013, pp. 2509–2512.

BIBLIOGRAPHY

- [87] Z.-K. Gao, Q. Cai, Y.-X. Yang, N. Dong, and S.-S. Zhang, "Visibility graph from adaptive optimal kernel time-frequency representation for classification of epileptiform eeg," *International Journal of Neural Systems*, vol. 27, no. 04, p. 1750005, 2017.
- [88] Y. Yan, S. Zhang, J. Tang, and X. Wang, "Understanding characteristics in multivariate traffic flow time series from complex network structure," *Physica A: Statistical Mechanics and its Applications*, vol. 477, pp. 149–160, 2017.
- [89] M. Zheng, S. Domanskyi, C. Piermarocchi, and G. I. Mias, "Visibility graph based temporal community detection with applications in biological time series," *Scientific Reports*, vol. 11, no. 1, pp. 1–12, 2021.
- [90] "Clustering analysis of residential electricity demand profiles," *Applied Energy*, vol. 135, pp. 461–471, 2014.
- [91] S. Lin, F. Li, E. Tian, Y. Fu, and D. Li, "Clustering load profiles for demand response applications," *IEEE Transactions on Smart Grid*, vol. 10, no. 2, pp. 1599–1607, 2019.
- [92] Q. Zhang, J. Wu, P. Zhang, G. Long, and C. Zhang, "Salient subsequence learning for time series clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 9, pp. 2193–2207, 2019.
- [93] P. Boniol and T. Palpanas, "Series2graph: Graph-based subsequence anomaly detection for time series," *Proc. VLDB Endow.*, vol. 13, no. 12, p. 1821–1834, jul 2020. [Online]. Available: <https://doi.org/10.14778/3407790.3407792>
- [94] J. Duan and L. Guo, "Variable-length subsequence clustering in time series," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2020.
- [95] Q. Wen, J. Gao, X. Song, L. Sun, H. Xu, and S. Zhu, "Robuststl: A robust seasonal-trend decomposition algorithm for long time series," vol. 33, pp. 5409–5416, Jul. 2019.
- [96] J. Mallick, S. Talukdar, M. K. Almesfer, M. Alsubih, M. Ahmed, and A. R. M. T. Islam, "Identification of rainfall homogenous regions in saudi arabia for experimenting and improving trend detection techniques," *Environmental Science and Pollution Research*, vol. 29, no. 17, pp. 25 112–25 137, 2022.
- [97] F. Wang, "Temporal pattern analysis of local rainstorm events in china during the flood season based on time series clustering," *Water*, vol. 12, no. 3, 2020.

- [98] V. Kindl, B. Skala, R. Pechanek, V. Kus, and J. Hornak, "Low-pass filter for hv partial discharge testing," *Sensors*, vol. 18, no. 2, p. 482, 2018.
- [99] S. Raghu, N. Sriraam, Y. Temel, S. V. Rao, and P. L. Kubben, "Eeg based multi-class seizure type classification using convolutional neural network and transfer learning," *Neural Networks*, vol. 124, pp. 202–212, 2020.
- [100] Y. Liu, S. Garg, J. Nie, Y. Zhang, Z. Xiong, J. Kang, and M. S. Hossain, "Deep anomaly detection for time-series data in industrial iot: A communication-efficient on-device federated learning approach," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6348–6358, 2021.
- [101] T. B. Shahi, C. Sitaula, A. Neupane, and W. Guo, "Fruit classification using attention-based mobilenetv2 for industrial applications," *Plos one*, vol. 17, no. 2, p. e0264586, 2022.
- [102] J. Cheng, S. Tian, L. Yu, C. Gao, X. Kang, X. Ma, W. Wu, S. Liu, and H. Lu, "Resganet: Residual group attention network for medical image classification and segmentation," *Medical Image Analysis*, vol. 76, p. 102313, 2022.
- [103] A. Dempster, F. Petitjean, and G. I. Webb, "Rocket: exceptionally fast and accurate time series classification using random convolutional kernels," *Data Mining and Knowledge Discovery*, vol. 34, no. 5, pp. 1454–1495, 2020.
- [104] C. Ji, Y. Hu, S. Liu, L. Pan, B. Li, and X. Zheng, "Fully convolutional networks with shapelet features for time series classification," *Information Sciences*, vol. 612, pp. 835–847, 2022.
- [105] B. Lucas, A. Shifaz, C. Pelletier, L. O'Neill, N. Zaidi, B. Goethals, F. Petitjean, and G. I. Webb, "Proximity forest: an effective and scalable distance-based classifier for time series," *Data Mining and Knowledge Discovery*, vol. 33, no. 3, pp. 607–635, 2019.
- [106] A. Rajabi, M. Eskandari, M. J. Ghadi, L. Li, J. Zhang, and P. Siano, "A comparative study of clustering techniques for electrical load pattern segmentation," *Renewable and Sustainable Energy Reviews*, vol. 120, p. 109628, 2020.
- [107] A. Abanda, U. Mori, and J. A. Lozano, "A review on distance based time series classification," *Data Mining and Knowledge Discovery*, vol. 33, no. 2, pp. 378–412, 2019.
- [108] A. Alqahtani, M. Ali, X. Xie, and M. W. Jones, "Deep time-series clustering: A review," *Electronics*, vol. 10, no. 23, 2021. [Online]. Available: <https://www.mdpi.com/2079-9292/10/23/3001>

BIBLIOGRAPHY

- [109] J. Paparrizos and L. Gravano, "Fast and accurate time-series clustering," *ACM Trans. Database Syst.*, vol. 42, no. 2, 2017.
- [110] C. T. Zan and H. Yamana, "An improved symbolic aggregate approximation distance measure based on its statistical features," in *Proceedings of the 18th International Conference on Information Integration and Web-based Applications and Services*, 2016, pp. 72–80.
- [111] B. Lkhagva, Y. Suzuki, and K. Kawagoe, "Extended sax: Extension of symbolic aggregate approximation for financial time series data representation," *DEWS2006 4A-i8*, vol. 7, 2006.
- [112] N. Kumar and H. Kumar, "A novel hybrid fuzzy time series model for prediction of covid-19 infected cases and deaths in india," *ISA Transactions*, vol. 124, pp. 69–81, 2022.
- [113] K. Bandara, C. Bergmeir, and S. Smyl, "Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach," *Expert Systems with Applications*, vol. 140, p. 112896, 2020.
- [114] C.-C. M. Yeh, Y. Zhu, L. Ulanova, N. Begum, Y. Ding, H. A. Dau, D. F. Silva, A. Mueen, and E. Keogh, "Matrix profile i: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets," in *2016 IEEE 16th international conference on data mining (ICDM)*. Ieee, 2016, pp. 1317–1322.
- [115] Z. Zimmerman, K. Kamgar, N. S. Senobari, B. Crites, G. Funning, P. Brisk, and E. Keogh, "Matrix profile xiv: Scaling time series motif discovery with gpus to break a quintillion pairwise comparisons a day and beyond," in *Proceedings of the ACM Symposium on Cloud Computing*. New York, NY, USA: Association for Computing Machinery, 2019, p. 74–86.
- [116] Y. Zhu, A. Mueen, and E. Keogh, "Matrix profile ix: Admissible time series motif discovery with missing data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 6, pp. 2616–2626, 2021.
- [117] C.-C. M. Yeh, Y. Zhu, L. Ulanova, N. Begum, Y. Ding, H. A. Dau, Z. Zimmerman, D. F. Silva, A. Mueen, and E. Keogh, "Time series joins, motifs, discords and shapelets: a unifying view that exploits the matrix profile," *Data Mining and Knowledge Discovery*, vol. 32, no. 1, pp. 83–123, 2018.
- [118] S. Gharghabi, S. Imani, A. Bagnall, A. Darvishzadeh, and E. Keogh, "Matrix profile xii: Mpdist: a novel time series distance measure to allow data mining

- in more challenging scenarios," in *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2018, pp. 965–970.
- [119] S. Singh and A. Yassine, "Big data mining of energy time series for behavioral analytics and energy consumption forecasting," *Energies*, vol. 11, no. 2, p. 452, 2018.
- [120] I. Matloob, S. A. Khan, and H. U. Rahman, "Sequence mining and prediction-based healthcare fraud detection methodology," *IEEE Access*, vol. 8, pp. 143 256–143 273, 2020.
- [121] H. Park and J.-Y. Jung, "Sax-arm: Deviant event pattern discovery from multivariate time series using symbolic aggregate approximation and association rule mining," *Expert Systems with Applications*, vol. 141, p. 112950, 2020.
- [122] J. Zhang and M. Small, "Complex network from pseudoperiodic time series: Topology versus dynamics," *Physical review letters*, vol. 96, no. 23, p. 238701, 2006.
- [123] J. Zhang, J. Sun, X. Luo, K. Zhang, T. Nakamura, and M. Small, "Characterizing pseudoperiodic time series through the complex network approach," *Physica D: Nonlinear Phenomena*, vol. 237, no. 22, pp. 2856–2865, 2008.
- [124] P. Xu, R. Zhang, and Y. Deng, "A novel visibility graph transformation of time series into weighted networks," *Chaos, Solitons & Fractals*, vol. 117, pp. 201–208, 2018.
- [125] Y. Zhou, S. Zhang, L. Wu, and Y. Tian, "Predicting sectoral electricity consumption based on complex network analysis," *Applied Energy*, vol. 255, p. 113790, 2019.
- [126] Y. Keneshloo, J. Cadena, G. Korkmaz, and N. Ramakrishnan, "Detecting and forecasting domestic political crises: A graph-based approach," in *Proceedings of the 2014 ACM conference on Web science*, 2014, pp. 192–196.
- [127] L. N. Ferreira and L. Zhao, "Time series clustering via community detection in networks," *Information Sciences*, vol. 326, pp. 227–242, 2016.
- [128] B. V. Cherkassky, A. V. Goldberg, and T. Radzik, "Shortest paths algorithms: Theory and experimental evaluation," *Mathematical programming*, vol. 73, no. 2, pp. 129–174, 1996.
- [129] R. Tarjan, "Depth-first search and linear graph algorithms," *SIAM journal on computing*, vol. 1, no. 2, pp. 146–160, 1972.

BIBLIOGRAPHY

- [130] J. Zakaria, A. Mueen, and E. Keogh, "Clustering time series using unsupervised-shapelets," in *2012 IEEE 12th International Conference on Data Mining*. IEEE, 2012, pp. 785–794.
- [131] Z. Liu and M. Barahona, "Graph-based data clustering via multiscale community detection," *Applied Network Science*, vol. 5, no. 1, p. 3, 2020.
- [132] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.
- [133] J. Paparrizos and L. Gravano, "k-shape: Efficient and accurate clustering of time series," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, 2015, pp. 1855–1870.
- [134] L. Kaufman and P. J. Rousseeuw, *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons, 2009, vol. 344.
- [135] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical association*, vol. 66, no. 336, pp. 846–850, 1971.
- [136] J. M. Santos and M. Embrechts, "On the use of the adjusted rand index as a metric for evaluating supervised classification," in *International conference on artificial neural networks*. Springer, 2009, pp. 175–184.
- [137] F. Wilcoxon, "Individual comparisons by ranking methods," in *Breakthroughs in statistics*. Springer, 1992, pp. 196–202.
- [138] L. Gelazanskas and K. A. Gamage, "Demand side management in smart grid: A review and proposals for future direction," *Sustainable Cities and Society*, vol. 11, pp. 22–30, 2014.
- [139] E. Keogh and J. Lin, "Clustering of time-series subsequences is meaningless: implications for previous and future research," *Knowledge and information systems*, vol. 8, pp. 154–177, 2005.
- [140] P. K. Vemulapalli, V. Monga, and S. N. Brennan, "Robust extrema features for time-series data analysis," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 6, pp. 1464–1479, 2012.
- [141] M. Heidari Kapourchali and B. Banerjee, "Unsupervised feature learning from time-series data using linear models," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3918–3926, 2018.
- [142] G. Das, K.-I. Lin, H. Mannila, G. Renganathan, and P. Smyth, "Rule discovery from time series," ser. KDD'98. AAAI Press, 1998, p. 16–22.

- [143] M. Shokoohi-Yekta, Y. Chen, B. Campana, B. Hu, J. Zakaria, and E. Keogh, "Discovery of meaningful rules in time series," in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 2015, pp. 1085–1094.
- [144] P. Senin, J. Lin, X. Wang, T. Oates, S. Gandhi, A. P. Boedihardjo, C. Chen, and S. Frankenstein, "Grammarviz 3.0: Interactive discovery of variable-length time series patterns," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 12, no. 1, pp. 1–28, 2018.
- [145] X. Wang, J. Lin, N. Patel, and M. Braun, "Exact variable-length anomaly detection algorithm for univariate and multivariate time series," *Data Mining and Knowledge Discovery*, vol. 32, no. 6, pp. 1806–1844, 2018.
- [146] J. Gui, Z. Zheng, Z. Qin, D. Jia, Y. Gao, Z. Liu, and Q. Yao, "An approach to extract state information from multivariate time series," *Journal of Computers*, vol. 31, no. 6, pp. 1–11, 2020.
- [147] L. Wang, P. Xu, and Q. Ma, "Incremental fuzzy clustering of time series," *Fuzzy Sets and Systems*, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0165011421000130>
- [148] Z. Huang and D. K. J. Lin, "The time-series link prediction problem with applications in communication surveillance," *INFORMS Journal on Computing*, vol. 21, no. 2, pp. 286–303, 2009.
- [149] V. Satuluri and S. Parthasarathy, "Symmetrizations for clustering directed graphs," in *Proceedings of the 14th International Conference on Extending Database Technology*, ser. EDBT/ICDT '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 343–354.
- [150] S.-V. Sanei-Mehri, A. Das, and S. Tirthapura, "Enumerating top-k quasi-cliques," in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 1107–1112.
- [151] F. Pallonetto, M. De Rosa, F. D'Ettorre, and D. P. Finn, "On the assessment and control optimisation of demand response programs in residential buildings," *Renewable and Sustainable Energy Reviews*, vol. 127, p. 109861, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1364032120301544>
- [152] J. Wong and R. Rajagopal, "A simple way to use interval data to segment residential customers for energy efficiency and demand response program targeting," in *ACEEE Proceedings*, 2012.

BIBLIOGRAPHY

- [153] M. F. Tahir, C. Haoyong, I. I. Idris, N. A. Larik, and S. ullah Adnan, "Demand response programs significance, challenges and worldwide scope in maintaining power system stability," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 6, 2018. [Online]. Available: <http://dx.doi.org/10.14569/IJACSA.2018.090618>
- [154] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.
- [155] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, p. 1–30, dec 2006.
- [156] T. Pohlert, "The pairwise multiple comparison of mean ranks package (pmmr)," *R package*, vol. 27, no. 2019, p. 9, 2014.
- [157] R. Woolson, "Wilcoxon signed-rank test," *Wiley encyclopedia of clinical trials*, pp. 1–3, 2007.
- [158] A. Clauset, M. E. Newman, and C. Moore, "Finding community structure in very large networks," *Physical review E*, vol. 70, no. 6, p. 066111, 2004.
- [159] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical review E*, vol. 69, no. 2, p. 026113, 2004.
- [160] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, oct 2008. [Online]. Available: <https://doi.org/10.1088/1742-5468/2008/10/p10008>
- [161] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," vol. 105, no. 4, pp. 1118–1123, 2008.
- [162] A. T. Jebb, L. Tay, W. Wang, and Q. Huang, "Time series analysis for psychological research: examining and forecasting change," *Frontiers in psychology*, vol. 6, p. 727, 2015.
- [163] Q. Zhang, K. Yu, Z. Guo, S. Garg, J. J. P. C. Rodrigues, M. M. Hassan, and M. Guizani, "Graph neural network-driven traffic forecasting for the connected internet of vehicles," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 5, pp. 3015–3027, 2022.
- [164] M. Afzalan and F. Jazizadeh, "Residential loads flexibility potential for demand response using energy consumption patterns and user segments," *Applied Energy*, vol. 254, p. 113693, 2019.

- [165] V. F. Silva, M. E. Silva, P. Ribeiro, and F. Silva, "Time series analysis via network science: Concepts and algorithms," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 11, no. 3, p. e1404, 2021.
- [166] M. Corneli, P. Latouche, and F. Rossi, "Multiple change points detection and clustering in dynamic networks," *Statistics and Computing*, vol. 28, pp. 989–1007, 2018.
- [167] S. Aminikhanghahi and D. J. Cook, "A survey of methods for time series change point detection," *Knowledge and information systems*, vol. 51, no. 2, pp. 339–367, 2017.
- [168] C. Truong, L. Oudre, and N. Vayatis, "Selective review of offline change point detection methods," *Signal Processing*, vol. 167, p. 107299, 2020.
- [169] Z. Shi and A. Chehade, "A dual-lstm framework combining change point detection and remaining useful life prediction," *Reliability Engineering & System Safety*, vol. 205, p. 107257, 2021.
- [170] Q. Zhao, Q. Li, D. Yu, and Y. Han, "Tsarm-udp: An efficient time series association rules mining algorithm based on up-to-date patterns," *Entropy*, vol. 23, no. 3, 2021. [Online]. Available: <https://www.mdpi.com/1099-4300/23/3/365>
- [171] K. Mishra, S. Basu, and U. Maulik, "Graft: A graph based time series data mining framework," *Engineering Applications of Artificial Intelligence*, vol. 110, p. 104695, 2022.
- [172] Y. Cheng, M. Cheng, T. Pang, and S. Liu, "Using clustering analysis and association rule technology in cross-marketing," *Complexity*, vol. 2021, pp. 1–11, 2021.
- [173] K. Mishra, S. Basu, and U. Maulik, "Load profile mining using directed weighted graphs with application towards demand response management," *Applied Energy*, vol. 311, p. 118578, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0306261922000599>
- [174] H. Teichgraeber and A. R. Brandt, "Time-series aggregation for the optimization of energy systems: Goals, challenges, approaches, and opportunities," *Renewable and Sustainable Energy Reviews*, vol. 157, p. 111984, 2022.
- [175] S. Aljawarneh, V. Radhakrishna, P. V. Kumar, and V. Janaki, "A similarity measure for temporal pattern discovery in time series data generated by iot," in *2016 International Conference on Engineering & MIS (ICEMIS)*, 2016, pp. 1–4.

BIBLIOGRAPHY

- [176] K. Beedkar, R. Gemulla, and W. Martens, "A unified framework for frequent sequence mining with subsequence constraints," *ACM Trans. Database Syst.*, vol. 44, no. 3, 2019.
- [177] H. Li, T. Du, and X. Wan, "Time series clustering based on relationship network and community detection," *Expert Systems with Applications*, vol. 216, p. 119481, 2023.
- [178] S. N. A. U. Nambi, E. Pournaras, and R. Venkatesha Prasad, "Temporal self-regulation of energy demand," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 3, pp. 1196–1205, 2016.
- [179] D. R. DeFord and S. D. Pauls, "Spectral clustering methods for multiplex networks," *Physica A: Statistical Mechanics and its Applications*, vol. 533, p. 121949, 2019.
- [180] V. Gligorijević, Y. Panagakis, and S. Zafeiriou, "Non-negative matrix factorizations for multiplex network analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 4, pp. 928–940, 2019.
- [181] M. Berlingerio, M. Coscia, and F. Giannotti, "Finding and characterizing communities in multidimensional networks," in *2011 International Conference on Advances in Social Networks Analysis and Mining*, 2011, pp. 490–494.
- [182] P. Esling and C. Agon, "Time-series data mining," *ACM Computing Surveys (CSUR)*, vol. 45, no. 1, pp. 1–34, 2012.
- [183] L. Jiang, X. Wang, W. Li, L. Wang, X. Yin, and L. Jia, "Hybrid multitask multi-information fusion deep learning for household short-term load forecasting," *IEEE Transactions on Smart Grid*, vol. 12, no. 6, pp. 5362–5372, 2021.
- [184] X. Bampoula, G. Siaterlis, N. Nikolakis, and K. Alexopoulos, "A deep learning model for predictive maintenance in cyber-physical production systems using lstm autoencoders," *Sensors*, vol. 21, no. 3, 2021.
- [185] S. Aghabozorgi and T. Y. Wah, "Clustering of large time series datasets," *Intelligent Data Analysis*, vol. 18, no. 5, pp. 793–817, 2014.
- [186] A. Delgado-Bonal and A. Marshak, "Approximate entropy and sample entropy: A comprehensive tutorial," *Entropy*, vol. 21, no. 6, 2019.
- [187] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

- [188] M. Kiguchi, K. Takata, N. Hanasaki, B. Archevarahuprok, A. Champathong, E. Ikoma, C. Jaikaeo, S. Kaewrueng, S. Kanae, S. Kazama, K. Kuraji, K. Matsumoto, S. Nakamura, D. Nguyen-Le, K. Noda, N. Piamsa-Nga, M. Raksapatcharawong, P. Rangsiwanichpong, S. Ritphring, H. Shirakawa, C. Somphong, M. Srisutham, D. Suanburi, W. Suanpaga, T. Tebakari, Y. Trisurat, K. Udo, S. Wongsu, T. Yamada, K. Yoshida, T. Kiatiwat, and T. Oki, "A review of climate-change impact and adaptation studies for the water sector in thailand," *Environmental Research Letters*, vol. 16, no. 2, p. 023004, feb 2021.
- [189] A. Abanda, U. Mori, and J. A. Lozano, "A review on distance based time series classification," *Data Mining and Knowledge Discovery*, vol. 33, no. 2, pp. 378–412, 2019.
- [190] B. D. Fulcher and N. S. Jones, "Highly comparative feature-based time-series classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 12, pp. 3026–3037, 2014.
- [191] P. Schäfer and U. Leser, "Fast and accurate time series classification with weasel," ser. CIKM '17. Association for Computing Machinery, 2017, p. 637–646.
- [192] H. A. Abu Alfeilat, A. B. Hassanat, O. Lasassmeh, A. S. Tarawneh, M. B. Alhasanat, H. S. Eyal Salman, and V. S. Prasath, "Effects of distance measure choice on k-nearest neighbor classifier performance: a review," *Big data*, vol. 7, no. 4, pp. 221–248, 2019.
- [193] N. S. Shaik and T. K. Cherukuri, "Multi-level attention network: application to brain tumor classification," *Signal, Image and Video Processing*, vol. 16, no. 3, pp. 817–824, 2022.
- [194] Z. Xiao, X. Xu, H. Xing, S. Luo, P. Dai, and D. Zhan, "Rtfn: A robust temporal feature network for time series classification," *Information Sciences*, vol. 571, pp. 65–86, 2021.
- [195] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang, "Connecting the dots: Multivariate time series forecasting with graph neural networks," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 753–763.
- [196] Z. Cheng, Y. Yang, S. Jiang, W. Hu, Z. Ying, Z. Chai, and C. Wang, "Time2graph+: Bridging time series and graph representation learning via multiple attentions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 2, pp. 2078–2090, 2023.

BIBLIOGRAPHY

- [197] C. Dinesh, S. Makonin, and I. V. Bajić, “Residential power forecasting using load identification and graph spectral clustering,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 66, no. 11, pp. 1900–1904, 2019.
- [198] T. Zhao, T. Jiang, N. Shah, and M. Jiang, “A synergistic approach for graph anomaly detection with pattern mining and feature learning,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 6, pp. 2393–2405, 2022.
- [199] J. Liu, F. Xia, X. Feng, J. Ren, and H. Liu, “Deep graph learning for anomalous citation detection,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 6, pp. 2543–2557, 2022.
- [200] Y. Chen, F. Ding, and L. Zhai, “Multi-scale temporal features extraction based graph convolutional network with attention for multivariate time series prediction,” *Expert Systems with Applications*, vol. 200, p. 117011, 2022.
- [201] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *International Conference on Learning Representations*, 2017. [Online]. Available: <https://openreview.net/forum?id=SJU4ayYgl>
- [202] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, Y. W. Teh and M. Titterton, Eds., vol. 9. PMLR, 2010, pp. 249–256.
- [203] J. Grabocka, N. Schilling, M. Wistuba, and L. Schmidt-Thieme, “Learning time-series shapelets,” in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’14. Association for Computing Machinery, 2014, p. 392–401.
- [204] P. Senin and S. Malinchik, “Sax-vsm: Interpretable time series classification using sax and vector space model,” in *2013 IEEE 13th International Conference on Data Mining*, 2013, pp. 1175–1180.
- [205] E. Odhiambo Omuya, G. Onyango Okeyo, and M. Waema Kimwele, “Feature selection for classification using principal component analysis and information gain,” *Expert Systems with Applications*, vol. 174, p. 114765, 2021.
- [206] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2009, vol. 2.

BIBLIOGRAPHY

- [207] T. Navamani, "Chapter 7 - efficient deep learning approaches for health informatics," in *Deep Learning and Parallel Computing Environment for Bioengineering Systems*, A. K. Sangaiah, Ed. Academic Press, 2019, pp. 123–137.
- [208] A. Fathalla, A. Salah, K. Li, K. Li, and P. Francesco, "Deep end-to-end learning for price prediction of second-hand items," *Knowledge and Information Systems*, vol. 62, pp. 4541–4568, 2020.
- [209] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, "Deep learning-based text classification: A comprehensive review," *ACM Comput. Surv.*, vol. 54, no. 3, apr 2021. [Online]. Available: <https://doi.org/10.1145/3439726>
- [210] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2019. [Online]. Available: <https://openreview.net/forum?id=Bkg6RiCqY7>

Kakuli Nishra
Nishra