

UNIVERSIDADE FEDERAL FLUMINENSE

GUILHERME HENRIQUE APOSTOLO

**eSCIFI: An Energy Saving Mechanism for
WLANs Based on Machine Learning
Algorithms**

NITERÓI

2021

UNIVERSIDADE FEDERAL FLUMINENSE

GUILHERME HENRIQUE APOSTOLO

eSCIFI: An Energy Saving Mechanism for WLANs Based on Machine Learning Algorithms

Thesis presented to the Computing Graduate Program of the Universidade Federal Fluminense in partial fulfillment of the requirements for the degree of Master of Science in Computer Science. Area: Computer Science.

Advisors:

Débora Christina Muchaluat-Saade
Luiz Cláudio Schara Magalhães
Flávia Cristina Bernardini

NITERÓI

2021

Ficha catalográfica - SDC/BEE
Gerada com informações fornecidas pelo autor

A645e Apostolo, Guilherme Henrique
ESCIFI : an energy saving mechanism for WLANs based on
machine learning algorithms / Guilherme Henrique Apostolo ;
Débora Christina Muchaluat-Saade, orientadora ; Luiz Cláudio
Schara Magalhães, orientador ; Flávia Cristina Bernardini,
orientadora. Niterói, 2021.
121 f. : il.

Dissertação (mestrado)-Universidade Federal Fluminense,
Niterói, 2021.

DOI: <http://dx.doi.org/10.22409/PGC.2021.m.14829732784>

1. Rede sem fio. 2. Aprendizado de máquina. 3. Eficiência
energética. 4. Edifício inteligente. 5. Produção
intelectual. I. Muchaluat-Saade, Débora Christina,
orientador. II. Magalhães, Luiz Cláudio Schara,
orientador. III. Bernardini, Flávia Cristina , orientadora.
IV. Universidade Federal Fluminense. Instituto de Computação.
V. Título.

CDD -

Bibliotecário responsável: Rosiane Pedro do Nascimento - CRB7/6237

Guilherme Henrique Apostolo

eSCIFI: An Energy Saving Mechanism for WLANs Based on Machine Learning
Algorithm

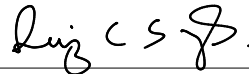
Thesis presented to the Computing Graduate Program of the Universidade Federal Fluminense in partial fulfillment of the requirements for the degree of Master of Science in Computer Science. Area: Computer Science.

Approved on 8 April 2021.

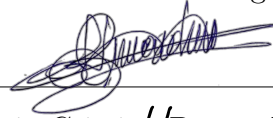
THESIS DEFENSE COMMITTEE



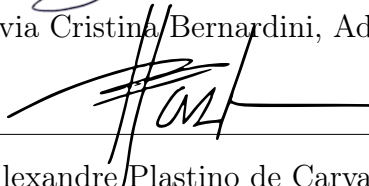
Prof. Débora Christina Muchaluat Saade, Advisor, UFF



Prof. Luiz Cláudio Schara Magalhães, Advisor, UFF



Prof. Flávia Cristina Bernardini, Advisor, UFF



Prof. Alexandre Plastino de Carvalho, UFF



Prof. José Ferreira de Rezende, UFRJ

Niterói

2021

*To my beloved mom, Maria Inês Henrique
Martins, for her unrestricted love and sup-
port, even in the toughest moments.*

Acknowledgements

First, I would like to thank my parents Maria Inês Henrique Martins and Alexandre Antonio Apostolo for their love, patience, encouragement and support during this journey. I owe them everything. I would like to thank my whole family and especially my grandmothers for always backing me up. None of this would have been possible without them.

I would like to thank my friends for always cheering me up with their smiles, kind gestures and motivational words. Especially to my friends Rafael Albino, Priscilla Pinheiro, Thatyana Magalhães, Romeu Vidal, Bianca Lopes, Amanda Costa, Diego Castilo, Juan Paz and Gabriel Guerra who brightened my days with their laughter and presence, even when physically apart.

I would like to thank all the Graduate Program at the Institute of Computing for all the lessons and help. Especially to my advisors Débora Christina Muchaluat-Saade, Luiz Cláudio Schara Magalhães and Flávia Cristina Bernardini who I look up to as a professional for all their patience, wisdom and help. Their vast knowledge and willingness to help were crucial to the completion of this work.

I also would like to thank Universidade Federal Fluminense, Laboratório Mídiacom and the SCIFI UFF Team for all the information and technical support given.

Finally, I would like to thank FAPERJ, CAPES and CNPq for the scholarships and financial support.

Resumo

Com o crescente tamanho das redes locais sem fio, em inglês *Wireless Local Area Networks (WLANs)*, para prover amplo acesso aos seus usuários, cresce também o seu consumo energético. Economia de energia em redes Wi-Fi de larga escala, sem impactar o serviço aos usuários, é indubitavelmente desejável. Este trabalho propõe e avalia o mecanismo de economia de energia para *WLANs* chamado de eSCIFI. O eSCIFI é um mecanismo de economia de energia que usa algoritmos de aprendizado de máquina como modelos de predição de demanda de ocupação. O eSCIFI foi desenvolvido para funcionar em um maior número de redes locais sem fio, o que inclui redes Wi-Fi como a rede SCIFI da Universidade Federal Fluminense (UFF). O eSCIFI pode funcionar em *WLANs* que não possam coletar dados em tempo real e/ou que possuam um poder de processamento limitado. O eSCIFI também inclui os algoritmos de agrupamento, o cSCIFI e o cSCIFI+, que auxiliam na garantia de cobertura da rede. O eSCIFI usa estes algoritmos de agrupamento e previsões fornecidas pelos modelos de aprendizado de máquina como entradas do seu algoritmo de decisão de estado de energia, que é o responsável por decidir que pontos de acesso devem ou não ser desligados durante o dia. Para avaliar o mecanismo e SCIFI para o cenário motivador, criaram-se dois *datasets* usando os dados coletados da rede SCIFI UFF no bloco H durante um período de 6 meses. Primeiro conduziu-se uma análise experimental usando a metodologia unificada proposta para determinar quais são os melhores modelos de aprendizado de máquina. Os resultados mostraram que o modelo COL/DT/SL/ADHDWD alcançou 86.69% de acurácia para as previsões de ocupação usando técnicas de classificação e um RMSPE (*Root Mean Squared Percentage Error*) de 0.29 para as previsões de ocupação usando técnicas de regressão. Por fim, foram conduzidas diversas simulações de rede comparando o mecanismo eSCIFI, utilizando ambos algoritmos de agrupamento, contra outros mecanismos presentes na literatura usando os traces de redes obtidos da rede UFF SCIFI. Os resultados mostram que o mecanismo eSCIFI usando o algoritmo de agrupamento cSCIFI+ obteve os melhores resultados e poderia economizar até 64.32% da energia consumida pela rede UFF SCIFI sem afetar a cobertura dos seus usuários.

Palavras-chave: Mecanismo de Economia de Energia para WLANs, Aprendizado de Máquina, Prédios Inteligentes, Redes Wi-Fi.

Abstract

As wireless local area networks grow in size to provide access to users, so does their power consumption. Power savings in a large-scale Wi-Fi network, with low impact to user service, is undoubtedly desired. In this work, we propose and evaluate the eSCIFI energy saving mechanism for WLANs. eSCIFI is an energy saving mechanism that uses machine learning algorithms as occupancy demand estimators. The eSCIFI mechanism was designed to cope with a broader range of WLANs, which includes Wi-Fi networks such as the Fluminense Federal University (UFF) SCIFI network. The eSCIFI can cope with WLANs that can not acquire data in a real time manner and/or possess a limited CPU power. The eSCIFI design also includes two clustering algorithms, named cSCIFI and cSCIFI+, that help to guarantee the network's coverage. eSCIFI uses those network clusters and machine learning predictions as input features to an energy state decision algorithm that then decides which Access Points (APs) can be switched off during the day. To evaluate the eSCIFI mechanism for our scenario, we created two dataset using the data collected from the Fluminense Federal University (UFF) SCIFI Wi-Fi network on the H building over a period of 6 months. We first conducted an experimental analysis using our proposed unified methodology to determine which machine learning models provide the best performance results. The results showed that the COL/DT/SL/ADHDWD model achieved 86.69% accuracy for occupancy prediction using classification techniques and RMSPE (Root Mean Squared Percentage Error) of 0.29 for occupancy count prediction using regression techniques. Later we conducted several trace-driven simulations comparing the eSCIFI mechanism using both clustering algorithms with other energy saving mechanism in the literature using the UFF SCIFI network traces. The results showed that eSCIFI mechanism using the cSCIFI+ clustering algorithm got the best results and that it could save up to 64.32% of the UFF SCIFI network without affecting the user's coverage.

Keywords: WLAN Energy Saving Mechanism, Machine Learning, Smart Buildings, Wi-Fi Networks.

List of Figures

2.1	Comparison between machine learning single-label and multi-label problems	8
2.2	Comparison between Binary Relevance (left) and Classifier/Regressor Chain (right) multi-label/multi-target methods	9
2.3	Multilayer Perceptron architecture	10
3.1	Average day occupancy comparing working days, weekends and holidays . .	23
3.2	Average week occupancy	23
3.3	Average busy state detection of the SCIFI network H's building APs during the time slots of a day comparing the differences between working days and holidays	25
3.4	Average busy detection state of the SCIFI network H's building APs during the time slots of a week	25
4.1	Our Proposed Methodology	28
4.2	Accuracy A_{tj} for several BR and CC ML methods and parameter configurations	34
4.3	Accuracy A_{tj} of ML and SL methods for several parameter configurations .	36
4.4	$RMSE_{tj}$ for several BR and RC MT methods and parameter configurations	39
4.5	$RMSP E_{tj}$ for several BR and RC MT methods and parameter configurations	40
4.6	$RMSP E_{tj}$ of MT and ST methods for several parameter configurations . .	40
5.1	eSCIFI architecture	46
5.2	Hybrid model result creation example	48
5.3	Hybrid model results compared with the real demand and the demand given by the regression results for the whole month of September	49
5.4	cSCIFI cluster formation algorithm	55

5.5	cSCIFI+ cluster formation algorithm	57
5.6	cSCIFI and cSCIFI+ cluster formation comparison	58
5.7	eSCIFI energy state decision algorithm flowchart	61
6.1	Normalized Energy saving factor for different special APs set sizes	67
6.2	Normalized Energy saving factor for different time window sizes	69
6.3	Coverage loss for different time window sizes	70
6.4	Normalized Energy saving factor for different T_{min} values	71
6.5	Coverage ratio for different T_{min} values	72
6.6	Normalized Energy saving factor for different T_{min} and tw values	73
6.7	Normalized Energy saving factor comparison between a holiday and a week-day	74
6.8	Coverage ratio loss comparison between a holiday and a weekday	75
6.9	Normalized Energy saving factor comparison between the eSCIFI mechanism using real traffic data and using model prediction demand estimations	77
6.10	Normalized Energy saving factor comparison between the SEAR and eSCIFI mechanism using both clustering algorithm with different neighborhood lists	78
A.1	H building ground floor blueprint showing the UFF SCIFI AP positions	90
A.2	H building second floor blueprint showing the UFF SCIFI AP positions	91
A.3	H building third floor blueprint showing the UFF SCIFI AP positions	92
A.4	H building fourth floor blueprint showing the UFF SCIFI AP positions	93
A.5	H building fifth floor blueprint showing the UFF SCIFI AP positions	94
C.1	H building UFF SCIFI network topology showing the neighbor APs	97

List of Tables

2.1	Occupancy prediction related work comparison.	14
2.2	RoD strategy mechanism related work comparison.	19
4.1	MLP ANN parameter values.	31
4.2	Classification performance results for BR and CC ML methods	35
4.3	Classification performance results for BR ML and SL methods.	37
4.4	DT and RF classifier’s mean number of leaves and depth size evaluation. .	38
4.5	Regression performance results for BR and RC MT methods.	41
4.6	Regression performance results for BR MT and ST methods.	42
4.7	DT and RF regressor’s mean number of leaves and depth size evaluation .	43
5.1	Mean Estimator and Hybrid models performance results.	50
6.1	AP’s consumed power and maximum power saving factor percentage	65
B.1	A part of the constructed ML dataset, showing the input attributes and the occupancy detection history for the APs	95
B.2	A part of the constructed SL dataset, showing the input attributes and the occupancy count history for the APs	96
D.1	Access Point statistics from April to August 2018.	99
E.1	Overall Rank of UFF SCIFI H building network	100
F.1	Cluster formation of UFF SCIFI H building network using cSCIFI algorithm	102
F.2	Cluster formation of UFF SCIFI H building network using cSCIFI+ algorithm	103

Acronyms

ALL	All Features	32
ANN	Artificial Neural Network	2
AP	Access Point	v
APHDWD	APid, holiday and weekday features	32
APid	AP Identification	29
BR	Binary Relevance	8
CC	Classifier Chain	9
Col	Collective	32
CPU	Central Processing Unit	3
DT	Decision Tree	31
HVAC	Heating, Ventilating and Air Conditioning	1
IEEE	Institute of Electrical and Electronics Engineers	14
Ind	Individual	32
K-NN	K-Nearest Neighbors	31
ML	Multi-Label	30
MLP	Multilayer Perceptron	9
MRTG	Multi Router Traffic Grapher	21
MT	Multi-Target	32
Numb. of Leaves	Number of Leaves	35
PoE	Power over Ethernet	50
QoS	Quality of Service	15
RC	Regressor Chain	9
RF	Random Forest	31
RoD	Resource On Demand	2

RSSI	Received Signal Strength Indication	51
SDN	Software Defined Networks	15
SGD	Stochastic Gradient Descent	32
SL	Single-Label	30
SNMP	Simple Network Management Protocol	21
ST	Single-Target	32
Std. Dev.	Standard Deviation	36
SVM	Support Vector Machine	32
UFF	Fluminense Federal University	v
WLAN	Wireless Local Area Network	iv

Nomenclature

S	Dataset
\mathcal{D}	Fixed and unknown distribution
\mathbf{x}_i	Feature vector of the i th instance
x_i^M	Value of the M th feature of the feature vector of the i th instance
\mathbf{Y}_i	Set of labels associated with the i th instance
L	Set of possible label values
l_q	q th label in the label set
$ L $	Label cardinality
t	Set of time slots
t_j	j th time slot in the time slot set
t_{max}	Maximum time slot value
$Y_i^{t_j}$	Label value in the j th time slot of the i th instance
\overline{Y}_{t_j}	Mean of the real association count values for a specific time slot t_j
S_{t_j}	Training set of the j th time slot
S'_{t_j}	Test set of the j th time slot
A_{t_j}	Accuracy of time slot t_j
$TP_i^{t_j}$	True positive value of the j th time slot of the i th instance
$FP_i^{t_j}$	False positive value of the j th time slot of the i th instance
$TN_i^{t_j}$	True negative value of the j th time slot of the i th instance
$FN_i^{t_j}$	False negative value of the j th time slot of the i th instance
P_{t_j}	Precision of time slot t_j
R_{t_j}	Recall of time slot t_j
$F1_{t_j}$	F1-score of time slot t_j
M	Set of metrics used
$RMSP E_{t_j}$	Root Mean Squared Percentage Error of time slot t_j
$RMSE_{t_j}$	Root Mean Square Error of time slot t_j
$MAPE_{t_j}$	Mean Absolute Percentage Error of time slot t_j
ESF	Energy Saving Percentage
ESF_{max}	Maximum power saving factor
P_{ext_on}	Access point external power when the wireless network interface is switched on
P_{ext_off}	Access point external power when the wireless network interface is switched off

t_{on}	Time that the access points stayed with their wireless interface switched on
t_{off}	Time that the access points stayed with their wireless interface switched off
CR	Coverage Ratio
U	Total number of clients in the network over a period of time
U_l	Total number of uncovered clients in the network over a period of time
SQ_{ij}	Signal quality of the measured AP j by the scanning AP i
WA_p	Weighted Average of the AP p
$NT_{j^{th}}^p$	Number of times that the AP p appears in the j^{th} position
V_i	Neighborhood set of AP i
C	Cluster set
C_i	Cluster formed starting from the AP i
tw	Time window size
T_{max}	Maximum user threshold for a time window
T_{min}	Minimum user threshold for a time window
CCA	Cluster maximum traffic capacity
DM_i	Traffic demand of AP i for a time window
CMR	Classification results matrix
RMR	Regression results matrix
HMR	Hadamard product matrix

Contents

1	Introduction	1
1.1	Goals and Contributions	3
1.2	Research Methodology	4
1.3	Text Outline	5
2	Background and Related Work	7
2.1	Background	7
2.1.1	Machine Learning Methods and Artificial Neural Networks	7
2.1.2	Machine Learning Metrics	10
2.2	Related Work	12
2.2.1	Occupancy Estimation based on Machine Learning Models	12
2.2.2	WLAN Energy Saving Mechanisms	14
3	SCIFI Network Data Collection and Analysis	20
3.1	Data Collection	21
3.2	Occupancy Analysis	22
4	Proposed Unified Methodology	27
4.1	Unified Methodology	27
4.1.1	Dataset Construction	29
4.1.2	Single-label and Multi-label Classification Analysis	30
4.1.3	Single-target and Multi-target Regression Analysis	31
4.2	Experimental Analysis	32

4.2.1	Classifier Analysis	32
4.2.1.1	Multi-label Methods	33
4.2.1.2	Multi-label and Single-label Evaluation	34
4.2.2	Regression Analysis	36
4.2.2.1	Multi-target Methods	38
4.2.2.2	Multi-target and Single-target Evaluation	39
4.3	Further Discussion on Our Methodology and Results	43
4.3.1	Seasonal Information	43
4.3.2	Individual and Collective Comparison	44
5	Proposed eSCIFI Mechanism	45
5.1	eScifi Mechanism	45
5.1.1	Unified Methodology and Model Selection	47
5.1.2	Hybrid Model	47
5.1.3	Heuristic Mechanism	50
5.1.3.1	Heuristic Cluster Formation: cSCIFI and cSCIFI+	50
5.1.3.2	Energy State Decision Algorithm	58
6	eSCIFI Evaluation	63
6.1	Number of Special APs Analysis	66
6.2	Time Window Size Analysis	68
6.3	Minimum Threshold Analysis	70
6.4	Weekday Versus Holiday Analysis	73
6.5	Real Data vs. Model Predictions	76
6.6	SEAR vs. eSCIFI Clustering Algorithms	77
7	Conclusion	79
7.1	Limitations	80

7.2 Future Work	82
References	84
Appendix A – UFF SCIFI AP Positions in the H Building	90
Appendix B – Datasets Example	95
Appendix C – UFF SCIFI Network Topology	97
Appendix D – APs Statistics	98
Appendix E – Overall APs Rank	100
Appendix F – Clusters Formed With cSFICI and cSCFI+	101

Chapter 1

Introduction

Buildings play an important role in our lives. People usually spend in average 20 hours per day inside buildings [61]. Also, the number of inhabitants in urban areas is quickly increasing [48]. Since buildings are heavily occupied, they require great amounts of energy and resources to operate. As a consequence, there are numerous studies about smart buildings [60, 21, 4], specially on the creation of low cost, efficient smart building management systems.

The key concept behind smart building management systems is the preemptive control of building infrastructure in order to save resources such as lighting, Heating, Ventilating and Air Conditioning (HVAC), elevators and even network infrastructure [4, 60, 18]. Automatic control of the building elements creates an ambient intelligent building once it increases people's quality of life by saving resources through the introduction of technology [8]. Some building management systems do not require precise occupancy information to be functional and capable of saving energy, especially HVAC systems, by using fixed building control schedules [12]. Several studies have demonstrated that occupancy information could help to reduce energy consumption in buildings, specially in non-residential buildings [60, 54], which operate under more predictable schedules [60].

The presence of WLANs on shopping centers, conventions centers, commercial and universities buildings is increasing daily [15]. Because of that, the use of Wi-Fi networks has attracted a lot of attention for the provisioning of occupancy predictions for areas inside buildings. There are several papers on the literature that uses Wi-Fi infrastructure to gather information about the building areas occupancy history and current state in combination with different machine learning approaches to predict occupancy of building areas, offices and rooms [54, 60, 21, 4].

Wi-Fi networks can be used to conduct occupancy detection or occupancy counting for buildings. The ubiquity of large-scale Wi-Fi networks on non-residential buildings turns them into an excellent source of information with no additional cost [21, 45, 60]. There are several studies that use Wi-Fi infrastructure and machine learning techniques to create prediction models for smart building management. Some of them collect information on the building areas occupancy history to predict if they are occupied or not (occupancy detection) [54, 60, 4, 51]. Others use Wi-Fi information to predict the occupancy count of some building areas [61, 18, 21, 45]. In this scenario, several studies use Wi-Fi infrastructure combined with machine learning methods to predict occupancy of building areas, floors and rooms [54, 60, 21, 4, 61, 51, 27]. They do not necessarily use the association history information from the Wi-Fi network to build their dataset and create prediction systems, but rather other information such as channel utilization or bandwidth [54, 60, 21, 4, 61, 51, 27, 37, 62]. Those studies used single-label or multi-label machine learning classification models and Artificial Neural Networks (ANNs) to address the occupancy detection problem using Wi-Fi association history and developed mechanisms that decide whether an AP should be turned on or off [15, 61, 22, 29, 37, 18, 45]. There are others that use Wi-Fi association data to create single-target machine learning regression models to estimate the occupancy count that can also be used on Wi-Fi AP energy saving mechanisms [60, 36]. However, those models are mostly used to develop HVAC scheduling systems [4, 51, 53, 27]. On the other hand, none of them has used multi-target regression methods for occupancy count or compared and evaluated single-label and the multi-label methods to classification models to determine which would have greater accuracy on occupancy detection. Thus, we fill this gap with our work.

According to Cui *et al* [13], energy consumption in a Wi-Fi network is considerable. University wireless networks display a bimodal periodic behavior with daily and weekly cycles, and Wi-Fi APs may stay unused for extensive periods of time [18, 50, 19]. These long idle periods represent a considerable energy waste that presents an excellent optimization opportunity. That scenario allows the use of machine learning prediction models capable of delivering occupancy demand predictions for network APs throughout the day [18]. The Wi-Fi network controller can switch off the network interface of unused APs during idle time slots using Resource On Demand (RoD) strategy mechanisms based on those predictions.

Wi-Fi RoD strategy management systems, or simply RoD strategy mechanisms, implement algorithms and policies that decide which APs should be switched off to save energy and which APs must stay switched on to cope with the traffic demands [15]. Some

of those mechanism use real time data acquisition or sophisticated RoD strategies to create their energy saving mechanisms [56, 57]. However some wireless network controllers have limited Central Processing Unit (CPU) power, what makes it unfeasible to collect and predict occupancy in real time, therefore requiring these systems to make predictions based exclusively on past information. But even such networks could benefit from RoD strategy mechanisms and few to no adjustments would be required. Those mechanisms can aid both wireless network energy savings and also other building systems such as elevator scheduling. Therefore our scenario requires an analysis on how machine learning algorithms are capable of looking at the future based on previous information and giving accurate predictions about the Wi-Fi network demand in both occupancy detection and count methods.

1.1 Goals and Contributions

This work proposes eSCIFI, an energy saving mechanism for WLANs. eSCIFI uses machine learning models to predict the wireless network future demand, therefore it can work in wireless networks where the controller's CPU power does not allow real time data acquisition to estimate this demand. eSCIFI uses two RoD strategy algorithms to ensure client's association and the network minimum coverage: the AP clustering algorithm and the double threshold algorithm. The eSCIFI mechanism can determine which APs should be active or turned off during certain moments of the day in order to cope with the actual network demand and also save energy.

The main contributions of this dissertation are:

- design of an energy saving mechanism for WLANs that can work in scenarios where real data acquisition is not possible: eSCIFI;
- analysis of how can eSCIFI cope with the network demand while saving energy and the comparison of its results with other RoD strategy mechanism in the literature;
- proposal of a unified experimental methodology based on machine learning to evaluate classification and regression models about their capacity to accurately predict access point demands for energy-efficient smart buildings;
- an experimental analysis using our unified methodology to determine which models provide the best results or are the most suitable for an energy-efficient wireless network RoD strategy management system;

- construction of a dataset using real user data collected from a subset of the APs of the Fluminense Federal University (UFF) wireless network located in a specific building of the Engineering campus;

1.2 Research Methodology

The work presented in this dissertation required the research of distinct papers about several topics to be completed. We used academic digital libraries to find papers that are relevant to the topics presented in this dissertation. First, we searched for papers that present available occupancy history datasets and their creation process. This search gave us insights and helped on the creation process of our own dataset using the UFF SCIFI network. Later on, we searched for papers that used machine learning algorithms and methods to predict occupancy detection and count for smart buildings. Those papers gave us insights on how distinct algorithms and methods were used in several smart buildings scenarios. It also helped us to develop a methodology to evaluate and select them. Finally, we searched papers that develop energy saving mechanism for WLANs. Those papers were crucial to help us to understand the available energy saving mechanism and to develop the eSCIFI mechanism. We searched specially for those energy saving mechanisms for WLANs based on machine learning approaches.

The steps followed in our research methodology were:

1. Search occupancy history datasets creation process for smart building scenarios : We found some papers that present dataset construction process. In section 4.1.1 we present the papers that served as a basis for our construction process.
2. Construction of a dataset using real user data collected from the UFF SCIFI network: In section 3.1 we present our data collection process and in section 4.1.1 we present our dataset construction process.
3. Search distinct machine learning approaches and evaluations for smart building occupancy detection and occupancy count estimations: Those machine learning approaches and evaluations searched are presented and compared in section 2.2.1
4. Proposal of a unified experimental methodology : In section 4.1 we present a unified methodology to evaluate and select machine learning classification and regression models for smart buildings scenarios

5. An experimental analysis of our unified methodology: In section 4.2 we present an experimental analysis that can help to determine the most suitable machine learning model different for our scenario and other smart cities or network scenarios;
6. Search energy saving mechanism for WLANs, specially those energy saving mechanism based on machine learning approaches: In section 2.2.2 we present a taxonomy to compare those searched mechanism and to classify our proposed approach.
7. Proposal of the eSCIFI: In chapter 5 we present our proposed energy saving mechanism for WLANs, its architectural elements and its advantages.
8. A trace driven analysis: In chapter 6 we show of how the eSCIFI can cope with our experimental scenario and the comparison of its results with other RoD strategy mechanism in the literature.

1.3 Text Outline

The remainder of this dissertation is organized as follows. Chapter 2 presents the multi-label classification and multi-target regression methods and performance metrics that help understanding the proposed solution and evaluations and the related work to this dissertation. It presents the related work to occupancy detection and count based on machine learning models and to WLANs energy saving mechanisms.

Chapter 3 presents the UFF SCIFI network and discuss its advantages and limitations. It also present the data collection process and discuss the UFF SCIFI occupancy.

Chapter 4 explains our proposed methodology. It includes the detailed overview aspects of it, the dataset set creation process, the evaluation of our unified methodology for the UFF SCIFI scenario and discussions about the unified methodology results.

Chapter 5 describes the eSCIFI energy saving mechanism solution proposed in this work. It includes the architectural details of our solution such as the hybrid machine learning model and the heuristic algorithm.

Chapter 6 covers the evaluation of the proposed eSCIFI energy saving mechanism. It explain the details if our trace driven analysis and how different architectural components may affect the energy saving factor and coverage ratio loss.

Finally, Chapter 7 concludes this dissertation. It presents the conclusions of our work. It also discusses some enhancements and applications that might be explored in future

work.

Chapter 2

Background and Related Work

In this chapter we first discuss the background that helps on the understanding of our solutions and evaluation. Later, we discuss the related work to occupancy estimation based on machine learning models and then we will discuss the related work to energy saving mechanism.

2.1 Background

This work involves the use and assessment of distinct supervised machine learning methods. Those models are used for classification and regression problems, therefore distinct evaluation metrics will be needed for each scenario. It also involves the assessment of the energy saving factor and coverage ratio of the wireless network while working under certain network characteristics imposed by our energy saving mechanism. In this section, we define and explain the machine learning methods, the metrics used to evaluate their results for distinct scenarios and briefly explain what are ANNs.

2.1.1 Machine Learning Methods and Artificial Neural Networks

In a general supervised learning scenario, a dataset $S = \{(\mathbf{x}_1, Y_1), \dots, (\mathbf{x}_N, Y_N)\}$ is given to the learning method, with fixed and unknown distribution \mathcal{D} . Each instance \mathbf{x}_i is a vector of the form $\mathbf{x}_i = (x_i^1, \dots, x_i^M)$. Each value (x_i^1, \dots, x_i^M) is relative to each feature (X_1, \dots, X_M) . Y is a special feature called class. $Y_i, i = 1, \dots, N$, represents a set of labels associated to each instance \mathbf{x}_i . If all sets $Y_i, i = 1, \dots, N$, have only one value, the problem is called single-label. So, in single-label problems, machine learning algorithms have only one possible output prediction. However, some machine learning problems cannot be

treated as a single-label problem [24]. There are cases, such as movie classification, where a movie can be classified as action and fiction simultaneously [26]. Multi-label machine learning classification algorithms and methods are those capable of dealing with more than one exclusive output. In other words, if the sets Y_i contain one or more values, the problem is called multi-label classification. In a multi-label classification problem or simply multi-label problem, a set $L = \{l_1, \dots, l_q\}$ is given, such that all $Y_i \in L$. Figure 2.1 shows an example comparison between single-label and multi-label methods. In the single-label method there is only one possible output while on the multi-label method there are more than one possible output.

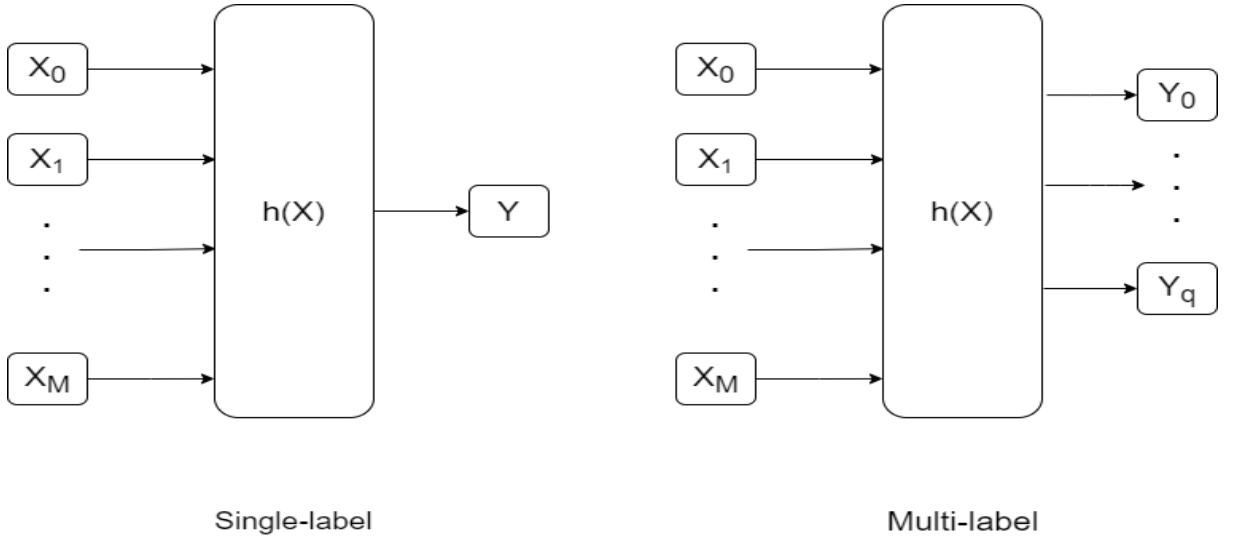


Figure 2.1: Comparison between machine learning single-label and multi-label problems

There are many distinct methods to tackle multi-label problems. Problem transformation is the simplest and the most often used, converting the multi-label problem with L labels into L single-label problems, *i.e.*, each label $l_q \in L$ is turned into a feature, composing a set of features $l_q, q = 1, \dots, Q$. The cardinality of L is denoted by $|L|$. Thus, each feature l_q is a class associated with the set of instances \mathbf{x}_i to be given to a single-label classification algorithm [23]. In our scenario, for modeling occupancy prediction as a multi-label problem, L represents the time slots for predicting occupancy during a day. For instance, considering a set of time slots $t = \{t_1, \dots, t_{max}\}$, if each time slot has 10 min, then $t_{max} = 144$ and $|L| = 144$. Therefore, to each instance \mathbf{x}_i and label (or time slot) l_q , we can associate a value $Y_i^{t_j}$ that represents a value of the set $\{0, 1\}$, indicating absence or presence of people in an AP for time slot t_j , defining a classification problem.

Binary Relevance (BR) is the simplest, and most used, problem transformation approach. BR approach uses the same input features in all the L prediction models, but each one is responsible for predicting one specific label l_q . Each model is completely indepen-

dent from others, which transform them in completely independent single-label models. The BR allows the classifiers to work on parallel since they are independent, however the classifiers can not benefit from correlations between the labels on the classification task.

Classifier Chain (CC) methods can be used, as they benefit from label correlations. It is expected that CC achieve more accurate results than Binary Relevance (BR) when there are dependencies among labels [23]. Like BR, CC also build a unique model for each label, but the models are sorted in a chain order. Each model input is composed by the domain features and the labels that precede the label being predicted by the model, forming a chain structure. Differently from the BR, the CC approach is serialized and the classifiers can not work independently since they have to provide their predictions to the next classifier on the chain in order to achieve a final prediction.

There are also regression cases, such as stock price estimation [47], where more than one single-target prediction using the same set of predictive variables is possible. Multi-target, also known as multi-output, regression algorithms and methods are those capable of dealing with more than one exclusive real value output. Therefore we can associate to each instance \mathbf{x}_i a value $Y_i^{t_j}$ that represents the number of people associated to an AP for time slot t_j , defining a regression problem. Those multi-target regression problems can also be tackled by problem transformation approaches. On the Binary Relevance method for multi-target problems, each regression model uses the same input features to predict a real output value $Y_i^{t_j}$. Similarly to the CC method, the Regressor Chain (RC) method builds a chain of regression models where each model input is composed by the domain features and the real target value that precede it. Figure 2.2 shows a comparison between the two problem transformation approaches used in this dissertation: BR and Classifier/Regressor Chain.



Figure 2.2: Comparison between Binary Relevance (left) and Classifier/Regressor Chain (right) multi-label/multi-target methods

Artificial Neural Network (ANN) models have proven to be successful in a number of prediction applications [43]. According to Gardner and Dorling [20], a Multilayer Perceptron (MLP) is an ANN where the neurons are interconnected and grouped into layers. Neuron connections are weighted and their output signal is an activation function of the sum of its weighted inputs [20]. MLP allows a single ANN to have a single or

multiple output targets easily turning the MLP into a multi-label/multi-target prediction model. Figure 2.3 shows the architecture of a MLP. The MLP represented in Figure 2.3 has only one hidden layer with t neurons inside it and q output targets.

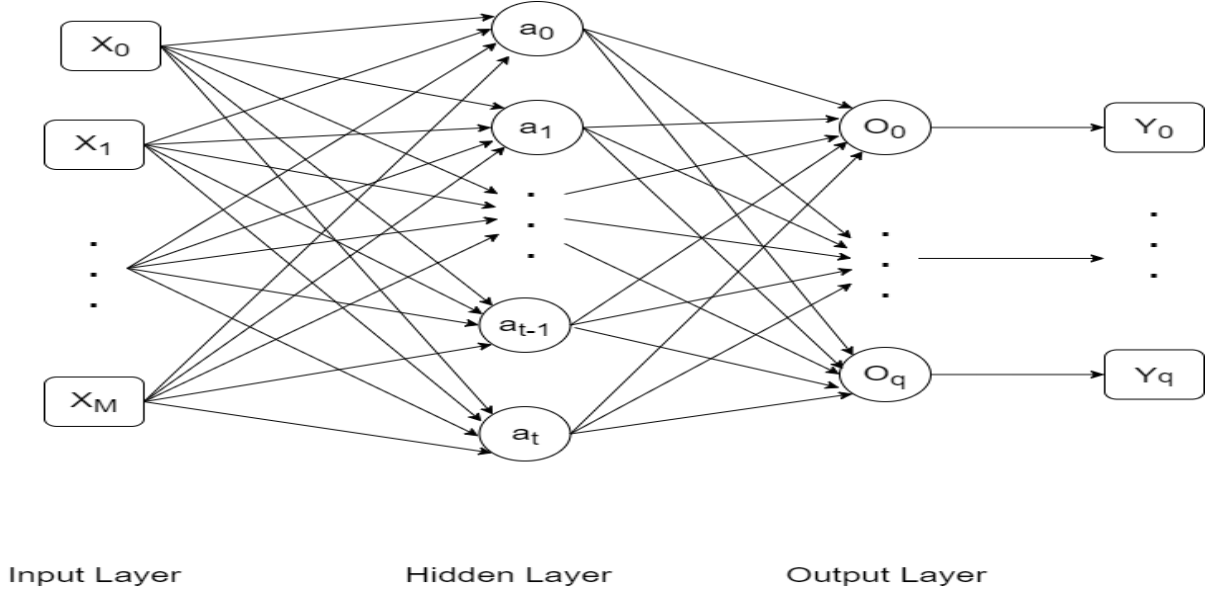


Figure 2.3: Multilayer Perceptron architecture

2.1.2 Machine Learning Metrics

Several metrics can be used for evaluating the classification results. In this work, we use specific label-based micro averaged metrics [26] for both single-label/single-target and multi-label/multi-target models. So, we evaluate occupancy predictions for each time slot and then average those results to get an overall view. Considering a training set $S_{t_j} = \{(\mathbf{x}_1, Y_1^{t_j}), \dots, (\mathbf{x}_N, Y_N^{t_j})\}$ collected in an interval of N days; a test set $S'_{t_j} = \{(\mathbf{x}'_1, Y_1'^{t_j}), \dots, (\mathbf{x}'_{N'}, Y_{N'}'^{t_j})\}$ collected in an interval of N' days after N days; time slots in a day $t_j \in t$ (if each time slot has 10 min then $t_{max} = 144$); and $\mathbf{h}(\mathbf{x}, t_j)$ a model constructed using S labeled using time stamp t_j , $t_j \in t$, and to be evaluated with S' also labeled using time stamp t_j , $t_j \in t$, we can define time slot accuracy A_{t_j} for each time slot $t_j \in t$ as shown in Equation 2.1, which calculates the accuracy of correctly predicting presence or absence detection in each time slot in a day, averaged by the number of N' days. It is worth mentioning that this measure is applicable for both single and multi-label models.

$$A_{t_j} = \frac{1}{N'} \sum_{i=1}^{N'} \mathbf{h}(\mathbf{x}, t_j) = Y_i'^{t_j}, t_j \in t \quad (2.1)$$

Considering the true positive value $TP_i^{t_j}$ of an instance i for a time slot t_j as 1 if

$\mathbf{h}(\mathbf{x}, t_j) = Y_i'^{t_j}$ and $\mathbf{h}(\mathbf{x}, t_j) = 1$, or 0 otherwise; the false positive value $FP_i^{t_j}$ of an instance i for a time slot t_j as 1 if $\mathbf{h}(\mathbf{x}, t_j) \neq Y_i'^{t_j}$ and $\mathbf{h}(\mathbf{x}, t_j) = 1$, or 0 otherwise; true negative value $TN_i^{t_j}$ of an instance i for a time slot t_j as 1 if $\mathbf{h}(\mathbf{x}, t_j) = Y_i'^{t_j}$ and $\mathbf{h}(\mathbf{x}, t_j) = 0$, or 0 otherwise; and the false negative value $FN_i^{t_j}$ of an instance i for a time slot t_j as 1 if $\mathbf{h}(\mathbf{x}, t_j) \neq Y_i'^{t_j}$ and $\mathbf{h}(\mathbf{x}, t_j) = 0$, or 0 otherwise, we can define Precision P_{t_j} , Recall R_{t_j} and F1-score $F1_{t_j}$ metrics. Those metrics are calculated for each time slot t_j and defined respectively by Equations 2.2, 2.3 and 2.4.

$$P_{t_j} = \frac{\sum_{i=1}^{N'} TP_i^{t_j}}{\sum_{i=1}^{N'} TP_i^{t_j} + FP_i^{t_j}}, t_j \in t \quad (2.2)$$

$$R_{t_j} = \frac{\sum_{i=1}^{N'} TP_i^{t_j}}{\sum_{i=1}^{N'} TP_i^{t_j} + FN_i^{t_j}}, t_j \in t \quad (2.3)$$

$$F1_{t_j} = \frac{2 \times P_{t_j} \times R_{t_j}}{P_{t_j} + R_{t_j}}, t_j \in t \quad (2.4)$$

We also calculate an overall metric for each of these metrics (Equation 2.5), which is the mean of the corresponding metric considering all the set t of time slots. This allows an overview of \mathbf{h} prediction performance for the classification problem. Thus, M in Equation 2.5 can be either A , P , R or $F1$ metric.

$$\overline{M} = \frac{1}{t_{max}} \sum_{j=1}^{t_{max}} M_{t_j} \quad (2.5)$$

Several metrics can be used for evaluating regressors. Consider the same definitions described before, except that $Y_{i'}'^{t_j}, i' = 1, \dots, N'$ now represents the number of people associated to an AP in a time slot t_j ; and that \overline{Y}_{t_j} is the mean of the real association count values for a specific time slot t_j . So, we can use $RMSE_{t_j}$ (Root Mean Square Error), $RMSP_{t_j}$ (Root Mean Squared Percentage Error) and $MAPE_{t_j}$ (Mean Absolute Percentage Error) metrics, defined respectively by Equations 2.6, 2.7 and 2.8, calculated for each time slot t_j , where $\overline{Y}_{t_j} = \frac{1}{N'} \sum_{i'=1}^{N'} Y_{i'}'^{t_j}$.

$$RMSE_{t_j} = \sqrt{\frac{1}{N'} \sum_{i=1}^{N'} (Y_i'^{t_j} - h(x_i', t_j))^2} \quad (2.6)$$

$$RMSP_{t_j} = \frac{\sum_{i=1}^{N'} (Y_i'^{t_j} - h(x_i', t_j))^2}{\sum_{i=1}^{N'} (Y_i'^{t_j} - \overline{Y}_{t_j})^2} \quad (2.7)$$

$$MAPE_{t_j} = \frac{\sum_{i=1}^{N'} |Y_i'^{t_j} - h(x'_i, t_j)|}{\sum_{i=1}^{N'} |Y_i'^{t_j} - \bar{Y}_{t_j}'|^2} \quad (2.8)$$

\overline{RMSE} (Equation 2.9) is an overall metric, calculated by the mean of $RMSE_{t_j}$ using the entire set t . The overall metric for $MAPE$ or $RMSPE$ can also be calculated by Equation 2.5, where M can be \overline{MAPE} or \overline{RMSPE} .

$$\overline{RMSE} = \sqrt{\frac{1}{N' \times t_{max}} \sum_{i=1}^{N'} \sum_{j=1}^{t_{max}} (Y_i'^{t_j} - h(x'_i, t_j))^2} \quad (2.9)$$

2.2 Related Work

There are some papers that propose the use of machine learning occupancy estimation models using Wi-Fi data to estimate demand. However most of those papers develop RoD strategies and energy saving mechanisms for other building infrastructure such as HVAC systems and lighting control. Therefore our related work presents two discussions. First, we discuss how different papers on the literature constructed their occupancy estimation models using machine learning algorithms and Wi-Fi data. Later on we discuss several WLAN energy saving mechanisms proposed in the literature.

2.2.1 Occupancy Estimation based on Machine Learning Models

The information collected from Wi-Fi networks, used to build a dataset and create a prediction system, is not always the same, as can be observed in [54, 60, 21, 4, 61, 51, 27, 37, 62, 14, 16, 52]. However, the key concept behind those studies is collecting data about the Wi-Fi network to create a detection or counting system using machine learning algorithms. Those decision support systems provide information for an energy saving management mechanism that controls building infrastructure based on its demand, such as the Wi-Fi network itself or HVAC systems.

Both classifier and regression model are used on Wi-Fi management systems based on RoD strategies. Those RoD strategies are capable of controlling the energy state of APs and turn off the unnecessary APs during day periods based on the predicted occupation [30, 22, 29, 14]. Some studies used classification models to address the Wi-Fi occupancy detection problem and developed RoD strategy mechanisms [18, 37, 60]. Some other studies use single-label machine learning classification methods and ANNs using

Wi-Fi data to control building lights [4, 61]. The work presented in [45] used algorithm adaptation multi-label methods to deal with the classification problem for HVAC systems. Regression models using Wi-Fi data to give an estimated users count are mostly used in HVAC scheduling systems [51, 53, 27], but some studies have also used regression models to develop RoD strategy mechanisms [52, 16].

Another important difference between those studies is the prediction models construction. Some studies use collective models [4, 60, 61, 16, 51, 53], while others use individual models [18, 37, 45, 27]. Collective models are prediction models trained with information regarding all APs and responsible for predicting the occupancy detection of all APs. Individual models are prediction models trained only using information regarding one specific AP and responsible for that AP occupancy detection prediction. It is worth mentioning that while the study of Vallero *et al.* [52] use and compare both individual and collective models, it does not compare them using the same machine learning algorithms, but it rather compares collective and individual models using several machine learning algorithms.

Table 2.1 compares related work about how they build occupancy prediction models. We can see in the table that most of the occupancy detection studies use single-label classifiers and that none of the occupancy count studies use multi-target regressors, but only single-target ones. Also, those studies did not compare and evaluate single-label/single-target and multi-label/multi-target methods to determine which would give the best predictions, as our work does. Moreover, Table 2.1 shows that there was no consensus on whether to use collective or individual models to give predictions and that no other study compares them, as our work does.

Finally, there are also some studies where pieces of information related to weather and season of the year were added to the occupancy information, in order to help on decision support systems for smart buildings [43, 12, 53, 45]. None of these studies have developed a methodology where the significance of this information is evaluated though.

Our work presents a unified experimental methodology to evaluate classification models used for occupancy detection and regression models used for occupancy count where several machine learning methods, input configurations, types of model construction and machine learning algorithms are assessed. The main goal of the assessment is to determine which of these parameter combinations is the most suitable and precise to give occupancy predictions. Our methodology evaluates and compares multi-label/multi-target and single-label/single-target methods using several machine learning algorithms, collec-

Table 2.1: Occupancy prediction related work comparison.

Authors	Classification		Regression		Models	
	Multi-label	Single-label	Multi-target	Single-target	Collective	Individual
Balaji et al. [4]		X			X	
Zou et al. [60]		X			X	
Zou et al. [61]		X			X	
Fang et al. [18]		X				X
Lyu et al. [37]		X				X
Sangogboye et al. [45]	X					X
Donevski et al. [16]				X	X	
Trivedi et al. [51]				X	X	
Wang et al. [53]				X	X	
Hobson et al. [27]				X		X
Vallero et al. [52]				X	X	X
Our Unified Methodology	X	X	X	X	X	X

tive and individual model construction schemes and the significance of input parameters. Another major contribution of our experimental methodology and analysis is that it does not require real-time data acquisition for forecasts.

2.2.2 WLAN Energy Saving Mechanisms

Based on the work of Budzisz et al. [6], Jardosh el al. [29] and Lorincz et al [35] we developed a taxonomy that helped us to compare the distinct RoD strategy mechanisms for wireless local area networks presented in this section. Our taxonomy consists of seven non-overlapping categories, corresponding to the main characteristics of the analyzed related work: (1) network type used, (2) WLAN application scenario, (3) control scheme, (4) operation strategy, (5) metrics used, (6) type of the algorithm, and (7) evaluation method.

Most of the analyzed related work develop RoD strategy mechanisms for Wi-Fi (Institute of Electrical and Electronics Engineers (IEEE) 802.11) networks. However there are great contributions in the literature that developed RoD strategy mechanisms for mesh networks [9] and for cellular networks [14, 16, 34, 52]. Those wireless networks types have distinct characteristics but the strategies and algorithms used on their RoD

strategy mechanisms are interchangeable and sometimes even overlapping. It is important to notice that an RoD strategy mechanism developed and tested for a specific wireless network can be used in other wireless network types. Therefore the network type category does not mean any sort of limitation to the RoD strategy mechanism applicability, but only describes the type of wireless network used as the work motivation and experimental scenario.

Most of the RoD strategy mechanisms were developed for application scenarios where they depend on homogeneous WLANs to operate, such as [11, 18, 22, 46, 29]. In these cases the RoD strategy mechanism is implemented to fully cope with the implemented WLAN technology without depending on any other wireless networks that might work in that area to help to implement their energy saving strategies. However there are some RoD strategy mechanism that were designed to operate in heterogeneous WLAN scenarios such as [44, 49, 58]. In the heterogeneous WLAN application scenarios the WLAN can rely on other wireless technologies such as Bluetooth or in a separate wake-up radio transceiver to detect users activity while the WLAN infrastructure is turned off. The RoD strategy mechanism developed for heterogeneous WLAN application scenarios can usually achieve higher energy saving rates without affecting their users Quality of Service (QoS) since there is always a supportive wireless network to detect new users instantly. However homogeneous networks are less complex in terms of deployment, control and management, due to their independent WLAN nature.

The control scheme category expresses how the RoD strategy mechanism implements its energy saving strategy. The control scheme can be centralized or distributed. RoD strategy mechanisms with centralized control scheme uses a central controller to supervise the network and send the commands to rest of the network. Centralized control schemes are more common for large wireless networks since most of them already have a central controller and their APs usually are not powerful enough to implement the algorithms and calculations needed. However the centralized control scheme can be subdivided into two categories depending whether the central controller is designed for a Software Defined Networks (SDN) controller or not. Software defined networks (SDNs) separate the control and data plane by introducing a centralized controller that is responsible to resolve flows forwarding policies and to assign them to the switches' forwarding tables [10]. Some related work [56, 57, 46, 10] develop energy saving mechanisms for SDN based networks with a centralized SDN controller. The use of SDN controllers allows those energy saving mechanisms to use some network information collected them such as network topology and traffic usage easily. However not every large scale WLAN controller is based on the

SDN paradigm and therefore can not count with all advantages given by it. There are some proposed energy saving mechanisms in our related work that do not consider the controller to be SDN [9, 14, 37, 39]. Those energy saving mechanisms also work with a centralized control scheme, but with non-SDN controller which make them a feasible solution to WLANs where not all SDN advantages are present. On the other hand, in the distributed network, the WLAN elements are all responsible for controlling their energy state and deciding weather they can be turned off or not. However, it is important to highlight that a distributed control scheme does not necessarily mean that each WLAN AP works independently from the other. In [32, 31], the Wi-Fi APs implement an energy saving strategy without a central controller, but they use out of band communication to communicate between them and decide which APs can be turned off.

RoD strategy mechanism can be classified into two operation strategies: demand driven or schedule driven. Demand driven strategies collect real-time information from the WLAN resources to estimate user demand [29]. The advantage of these strategies is that they can generate an energy saving in the WLAN while satisfying the user demand. However demand driven strategies have a higher CPU power cost due to the overhead of assessing user demands continuously [28]. Demand-driven strategies are more suitable in scenarios where the user demand may unpredictably vary over time such as in stadiums [22]. On the other hand, schedule-driven strategies use predefined schedules to produce its energy saving. These schedules can be obtained with machine learning models trained with WLAN historical usage data [18, 52, 37] or can be based on the administrator's experience [46]. The advantage of using schedule-driven strategies is their low CPU power requirements. Schedule-driven strategies are only suitable for scenarios where user demand is predictable such as university networks [18, 37, 46].

The RoD strategy mechanisms can be divided into 4 metrics subsets according to the metrics they use to minimize the energy consumption. The most common and most intuitive metrics are the traffic metrics subset. The traffic metrics subset comprises any network traffic related metric such as number of associated users [39, 18, 37], throughput [14] or more sophisticated ones such as channel utilization [29]. Traffic metrics are usually used and measured in a network and therefore they are easily accessible, but it might not be enough to guarantee the QoS or coverage alone. Coverage metrics are used to ensure that the whole radio area network [28, 29, 46] and users [56, 57, 22] will be covered. Coverage implies that the RoD energy saving strategies will guarantee that all users can connect to at least one active radio. QoS metrics are usually used in works that try to minimize the most the impact on the user's service [34, 33], but they also

imply smaller savings or more complex algorithms to work. Energy metrics correspond to the work [36, 35] where the reducing energy quantitative is taking into the analysis for the switching on/off strategies. A clear implication is that the user's traffic or QoS constraints can not be met. One important thing to highlight is that every metric alone has its advantages and weaknesses, therefore most of the analyzed related work uses a combination of metrics to guarantee the user's demand will be met.

RoD strategy mechanisms can also be divided by the type of algorithms used to make the energy status decision for the WLAN resources based on the metrics available. Heuristic algorithms can rapidly determine a solution within reasonable time using reasonable resources [35]. As the name suggests, heuristic algorithms are based on heuristics solutions that are easier to implement and usually based on thresholds [52, 18, 28] or other metrics combination rules [49]. Heuristic algorithms are usually most suitable for WLAN scenarios where the CPU power and/or computational time required are low. On the other hand, optimization algorithms are based on different mathematical problems and solvers that guarantee the best possible solution to a specified problem [36]. Optimization algorithms require more time and resources to provide their solution and therefore are suitable for WLAN scenarios where the CPU power and/or computational time required are high. Our analyzed related work shows that optimization algorithms achieve better results when compare to heuristic ones [56, 57], however Lorincz et al. [35] conclude in their work that "heuristic algorithms can be valuable alternatives offering good solution in reasonable amount of time".

Lastly the analyzed related work can be divided according to the experimental test made to evaluate their RoD strategy mechanism performance. Simulation tests are those that make use of simulation software such as Matlab [19], Scenargie [41] or NS-3 [46] to recreate their WLAN scenarios and evaluate performance. Trace Driven tests are those that use network traces to reproduce a real network scenario comparing how their network would respond to the changes in that scenario using distinct energy saving mechanism [18, 52, 16, 37]. Testbed experiments are those where a real WLAN infrastructure is used but a limited set of users and their behavior are simulated [22, 15, 59]. There are authors in related work that refer to their test as real network scenario tests, however they do not analyze the real infrastructure in a regular usage scenario with undefined users or behaviors and therefore we classified them as testbed.

Table 2.2 compares the RoD strategy mechanism analyzed in related work and our proposed eSCIFI mechanism. It is important to highlight that eSCIFI can be a possible

solution to a wider range of WLAN networks than most of the mechanisms presented in related work that have a centralized control scheme. The eSCIFI network can cope with wireless networks that are not based on the SDN paradigm, does not have a high CPU power at the controller and can not collect data in a real time manner. Those characteristics make any energy saving mechanism that presents optimization algorithms or demand driven strategies unpractical. On the other hand, the opposite is not true and eSCIFI can work normally in wireless networks that present one or more of the above mentioned characteristics. However it is worth to mention that, in WLAN scenarios that present those characteristics, eSCIFI might not be the best practical solution since it might not take advantage of those characteristics.

The eSCIFI characteristics make it a feasible solution for our motivation and evaluation scenario once it allows the development of an energy saving mechanism that can cope with the UFF SCIFI pure Wi-Fi network characteristics. eSCIFI presents a centralized controlling scheme, a scheduling driven operation strategy based on machine learning, using heuristic algorithms, traffic and coverage metrics.

Table 2.2: RoD strategy mechanism related work comparison.

Authors	Network Type	WLAN Scenario	Control Scheme	Operating Strategy	Metrics	Type of Algorithm	Evaluation Method
Capone et al. [9]	Mesh Networks	Homogeneous	Centralized	Demand Driven	Coverage and QoS	Optimization	Simulation
Chen et al. [10]	IEEE 802.11	Homogeneous	Centralized	Demand Driven	QoS	Optimization	Testbed
Chin et al. [11]	IEEE 802.11	Homogeneous	Centralized	Demand Driven	Traffic and Coverage	Optimization	Simulation
Dalmasso et al. [14]	Cellular Networks	Homogeneous	Centralized	Demand Driven	Traffic	Heuristic	Trace Driven
Debele et al. [15]	IEEE 802.11	Homogeneous	Centralized	Demand Driven	Traffic	Heuristic	Testbed
Donevski et al. [16]	Cellular Networks	Homogeneous	Centralized	Schedule Driven	Traffic	Heuristic	Trace Driven
Fang et al. [18]	IEEE 802.11	Homogeneous	Centralized	Schedule Driven	Traffic	Heuristic	Trace Driven
Ganji et al. [19]	IEEE 802.11	Homogeneous	Centralized	Demand Driven	Coverage	Optimization	Simulation
Gomez et al. [22]	IEEE 802.11	Homogeneous	Centralized	Demand Driven	Traffic and QoS	Heuristic	Testbed
Jardosh et al. [28]	IEEE 802.11	Homogeneous	Centralized	Demand Driven	Traffic and Coverage	Heuristic	Simulation
Jardosh et al. [29]	IEEE 802.11	Homogeneous	Centralized	Demand Driven	Traffic and Coverage	Heuristic	Testbed
Kumazoe et al. [32, 31]	IEEE 802.11	Homogeneous	Distributed	Demand Driven	Traffic and QoS	Heuristic	Simulation
Lee et al. [33]	IEEE 802.11	Homogeneous	Centralized	Demand Driven	Coverage and QoS	Optimization	Simulation and Testbed
Liu et al. [34]	Cellular Networks	Homogeneous	Centralized	Demand Driven	QoS	Optimization	Simulation
Lorincz et al. [36]	IEEE 802.11	Homogeneous	Centralized	Demand Driven	Energy and Traffic	Optimization	Simulation
Lorincz et al. [35]	IEEE 802.11	Homogeneous	Centralized	Demand Driven	Energy and Traffic	Heuristic	Simulation
Lyu et al. [37]	IEEE 802.11	Homogeneous	Centralized	Schedule Driven	Traffic	Heuristic	Trace Driven
Marsan et al. [39]	IEEE 802.11	Homogeneous	Centralized	Schedule Driven	Traffic	Heuristic	Trace Driven
Nagareda et al. [41]	IEEE 802.11	Homogeneous	Centralized	Demand Driven	Traffic and Coverage	Heuristic	Simulation
Rossi et al. [44]	IEEE 802.11	Heterogeneous	Distributed	Demand Driven	Traffic and Coverage	Heuristic	Simulation and Testbed
Silva et al. [46]	IEEE 802.11	Homogeneous	Centralized	Schedule Driven	Traffic and Coverage	Heuristic	Simulation
Tanaka at al. [49]	IEEE 802.11	Heterogeneous	Centralized	Demand Driven	Traffic	Heuristic	Simulation
Vallero et al. [52]	Cellular Networks	Homogeneous	Centralized	Schedule Driven	Traffic	Heuristic	Trace Driven
Wu et al. [55]	IEEE 802.11	Homogeneous	Centralized	Demand Driven	Traffic	Optimization	Simulation
Xu et al. [56, 57]	IEEE 802.11	Homogeneous	Centralized	Demand Driven	Traffic	Optimization	Simulation
Yaodong Zhang et al. [59]	IEEE 802.11	Heterogeneous	Centralized	Demand Driven	Traffic	Optimization	Testbed
Yomo et al. [58]	IEEE 802.11	Heterogeneous	Distributed	Demand Driven	Traffic	Heuristic	Simulation
eSCIFI	IEEE 802.11	Homogeneous	Centralized	Schedule Driven	Traffic and Coverage	Heuristic	Trace Driven

Chapter 3

SCIFI Network Data Collection and Analysis

Here we present the main components of the UFF SCIFI wireless network, how we collected the data for our work and the occupancy analysis of that data [3, 1]. The UFF SCIFI wireless network is a large-scale wireless developed by UFF, financed by RNP (Brazilian National Research and Education Network). It was developed to be a low-cost open-source option to the deployment, configuration, operation and management of large-scale wireless network [38]. The SCIFI network is composed of a smart controller, also named SCIFI controller, and low cost APs, operating under the open source OpenWRT firmware [17]. SCIFI controller is a non SDN central management and monitoring unit of UFF's network. The SCIFI controller coordinates data gathering from system logs, channel selection and access point's transmission power level services [38]. SCIFI network allows an expressive reduction on a large scale wireless network deployment cost, which eases the deployment of bigger networks, with more APs. SCIFI network is used at UFF, Ouro Preto Federal University and Brazilian Navy, as well as it was used in many different events. It has been proven to be a stable, low-cost and easy-to-install solution for controlling wireless APs [38].

Solutions with a centralized controller for configuration, management and monitoring of wireless networks are often vendor lock-in, which limits the network hardware choice to equipment from the same vendor. This limitation has a direct impact on the wireless network flexibility and cost.

The SCIFI controller is extensible and allows the deployment of a great number of APs in the network. It provides algorithms for channel selection and transmission power control of the access points to maximize the spectral efficiency of the wireless network. The SCIFI

network allows an expressive reduction on a large scale wireless network deployment cost, which eases the deployment of bigger networks, with more APs, with the same budget. However the SCIFI controller has a limited CPU power when compared to enterprise controller solutions. This CPU power limitation makes the computational time to execute some complex tasks unfeasible, such as detailed traffic real-time acquisition or complex optimization algorithms.

3.1 Data Collection

We selected, for this study, all the APs located at one specific building from one of the UFF's campuses, called H building (see Appendix A for APs positioning in the building details). Differently from the other buildings on campus that have professor's offices, laboratories, student unions and other university administration rooms, the H building has only classrooms. So, its occupation mainly occurs through lectures and exam applications. SCIFI network has 28 APs distributed over the 5 floors inside H building. Our data was obtained from the APs event logs. These logs were collected and stored at the SCIFI controller. Each AP sends a text file with all the management and control events information from their physical and data link layers. These data are requested and stored weekly by the controller. We collected data from 6 months, between April and September 2018.

The SCIFI controller is also responsible for the storage of those gathered data and for the storage of the information needed by the whole system. It provides a graphical interface that allows the parameter configuration of the SCIFI network such as channel selection of a specific AP, addition of new APs and the network's monitoring through custom dashboards using Simple Network Management Protocol (SNMP) [18] (NAGIOS [5] and Multi Router Traffic Grapher (MRTG) [42]).

An association event log marks the beginning of the data transmission between the AP and the mobile station, while the disassociation event marks its end. As we were only interested on the event logs that show the beginning and end of connection between the mobile stations and the AP, we had to filter log files that contain only these events. We observed, however, that the disassociation message log did not always appear on the log data, although the deauthentication message always occurred in pair to the disassociation message. We also observed that whenever disassociation and deauthentication of mobile stations message appeared in the logs, both occurred in very close time intervals, with

approximately 1 second difference between them. Therefore, we used deauthentication messages as the end of a connection mark between a mobile station and an AP, when there was no registered disassociation messages. Only a deep analysis would show the real reasons behind the absence of those disassociation messages in the logs, but we can point the user movement to an area with no network coverage as a possible cause. After that analysis we filtered the text file to contain only information regarding the association, disassociation and deauthentication of mobile devices communicating with the access point since they mark the beginning and end of the data transmission between the AP e the mobile station.

3.2 Occupancy Analysis

Figures 3.1 and 3.2 show the average occupancy SCIFI network behavior in the H building from April to September 2018. It is possible to observe the daily and weekly average occupancy. Figure 3.1 shows that APs barely have users associated to it between 0 and 6AM. It also shows a slowly increasing occupation for time slots between 6 and 9AM. That slow growth can be explained by the lecture time schedules for the H building, which start at 7AM, but most of them start at 9AM, and the last lectures end at 10PM. Morning classes start at odd hours, and afternoon classes at even times, with an hour interval between 1 and 2PM. Figure 3.1 shows that AP's occupation during university weekdays is higher than the occupation on holidays and weekends. However we can still see users associated with the network on holidays and weekends. In fact the average occupancy for weekends and holidays is very similar. These results were unforeseen, but can be explained by the H building usage during student vacations for summer/winter courses or special activities and for exams or other special activities during weekends.

Figure 3.2 shows that the AP demand is higher during weekdays than during weekends. The average occupancy reaches its highest on Tuesdays, Wednesdays and Thursdays. Whilst smaller than the other weekdays, Saturday's average demand is relatively high when compared to Sunday. One explanation can be that the building is more used on Saturdays for exams and other special activities than Sundays. For a classroom building such as the H building, these results were expected. We noticed that some APs remain with a residual number of devices connected to it during closing hours. One possible explanation is that the H building still has appliances, such as computers, and university staff members, such as the campus security, that are still present in the building during closed hours and days.

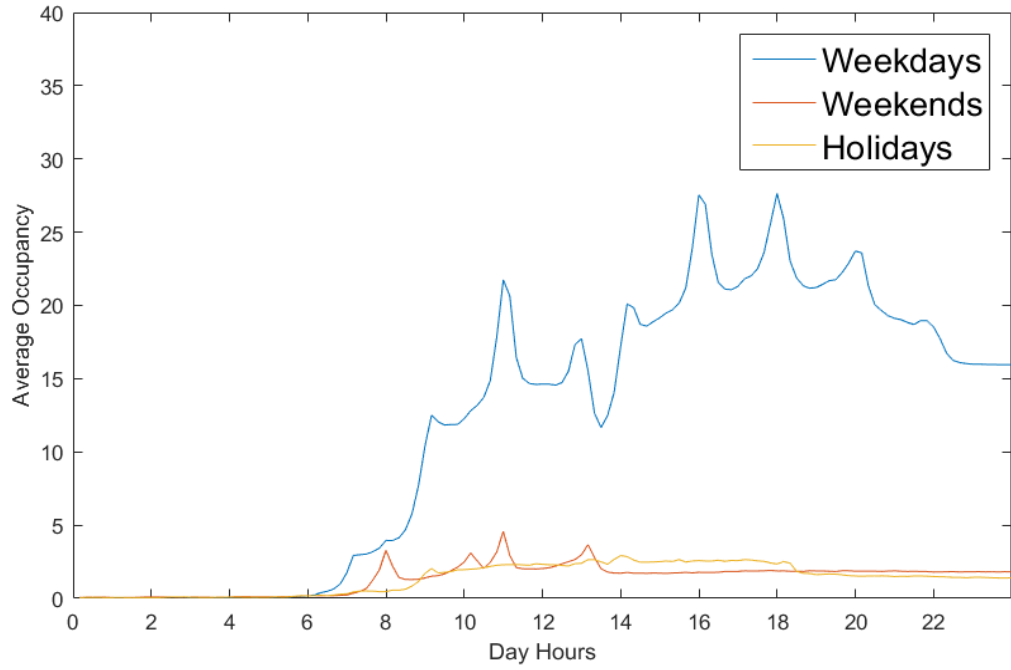


Figure 3.1: Average day occupancy comparing working days, weekends and holidays

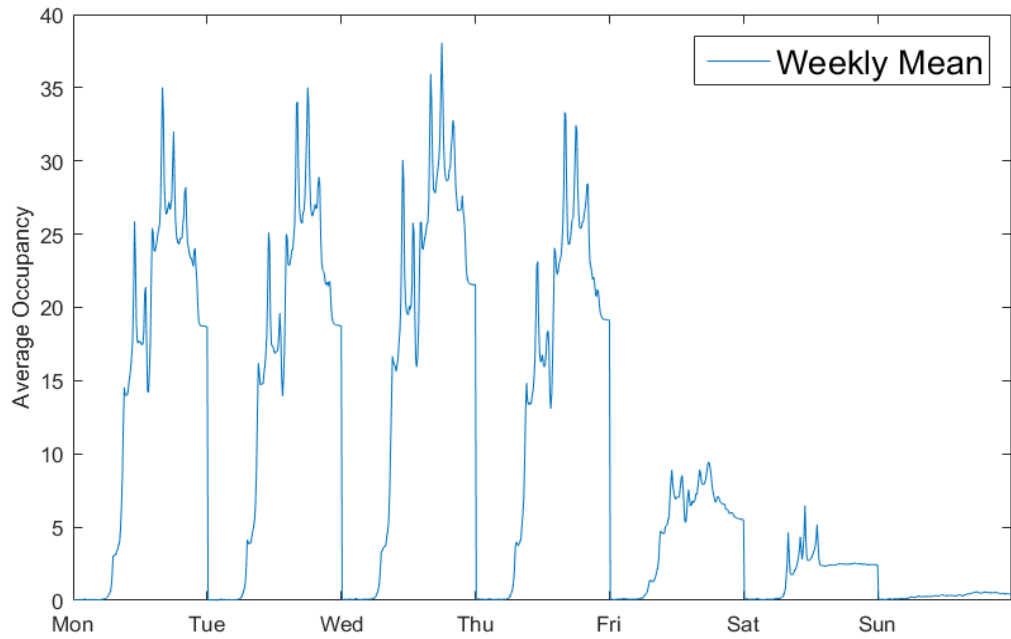


Figure 3.2: Average week occupancy

Figures 3.3 and 3.4 show the average daily and weekly AP's busy state detection for each 10 minutes time slots. A busy state detection for a specific time slot occurs when the AP had users associated with it, otherwise a idle state detection occurred. The visualizations show the total number of busy state detection events divided by the total

number of state detection events for a giving time slot, in order to allow a better scaling and visualization of the graph. So, the average busy state detection can vary between all real values ranging from 0 and 1. An average busy state detection value equals to 1 for a time slot means that for the whole observed period that specific 10 minutes time slot had all APs busy. On the other hand an average busy state detection value equals to 0 for a time slot means that for the whole observed period that specific 10 minutes time slot had all APs idle.

Based on Figure 3.1, it is possible to say that most of APs stay unused between 0AM and 6AM and mostly used for the rest of the day, after 6AM. It is also possible to notice a relative balance between the busy and idle state event for the time slots between 6AM and 9AM. That phenomenon can be explained by the lectures time tables for the H building. Figure 3.3 also shows that it is easier to have idle state periods during holidays and weekends than during the university working days. The busy state occurrence values showed for holidays are higher than those showed for weekends. These results when compared with the results showed in Figure 3.1 show that the slightly bigger average occupancy on holidays due to the summer/winter courses causes a bigger utilization of APs when compared to weekends. One explanation can be that exams and other special events that happened on weekends were more concentrated at some areas (having fewer APs being effectively used), while the summer/winter courses might happened throughout the building (causing a bigger number of APs to be used). This difference in the areas of the building being used might cause the APs usage differences.

Figure 3.4 also shows that the average busy state detection for the APs varies according to the day of the week. It is easier to have busy state periods during the weekdays than during the weekend. The average busy state reaches its highest on Tuesdays, Wednesdays and Thursdays. Although smaller than the other days of week, Saturday's average occurrence is relatively high when compared to Sunday. For the reasons above mentioned the results were expected since there is difference on the exam applications frequency between those days. Most of APs in the H building remain unused for long hours after 11PM until 6AM since the building opens at 7AM and closes at 11PM. However, we can still notice that some of the APs remain active during closing hours that is mainly because the H building still has appliances, such as computers, and university staff members, such as the campus security, that is still present in the building during closed hours and days. That explains why it is not possible to assume the idle state occurrence on the APs for those hours.

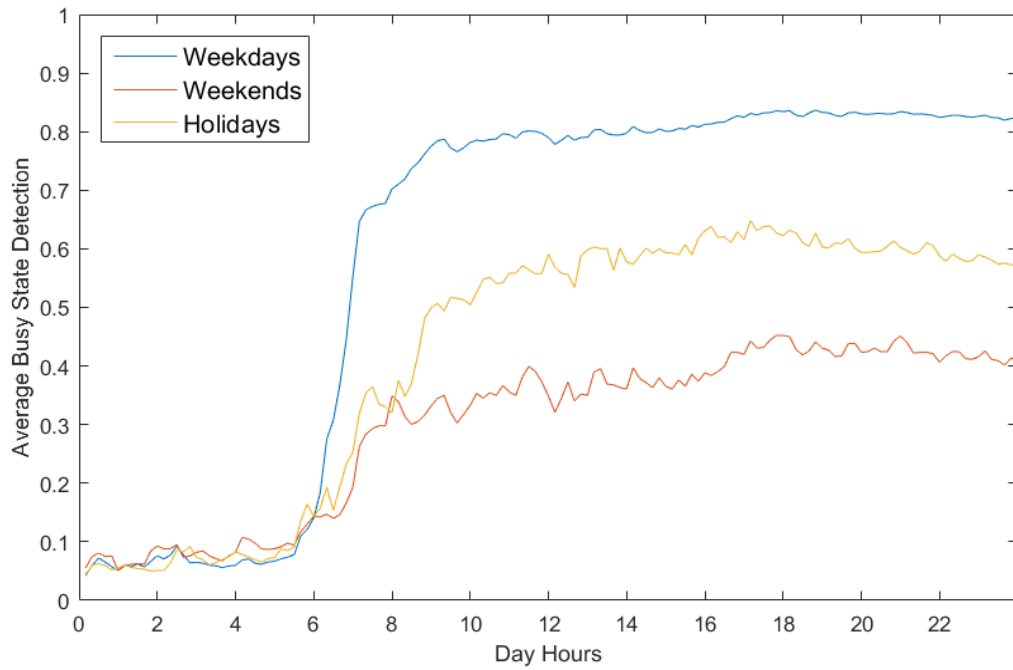


Figure 3.3: Average busy state detection of the SCIFI network H's building APs during the time slots of a day comparing the differences between working days and holidays

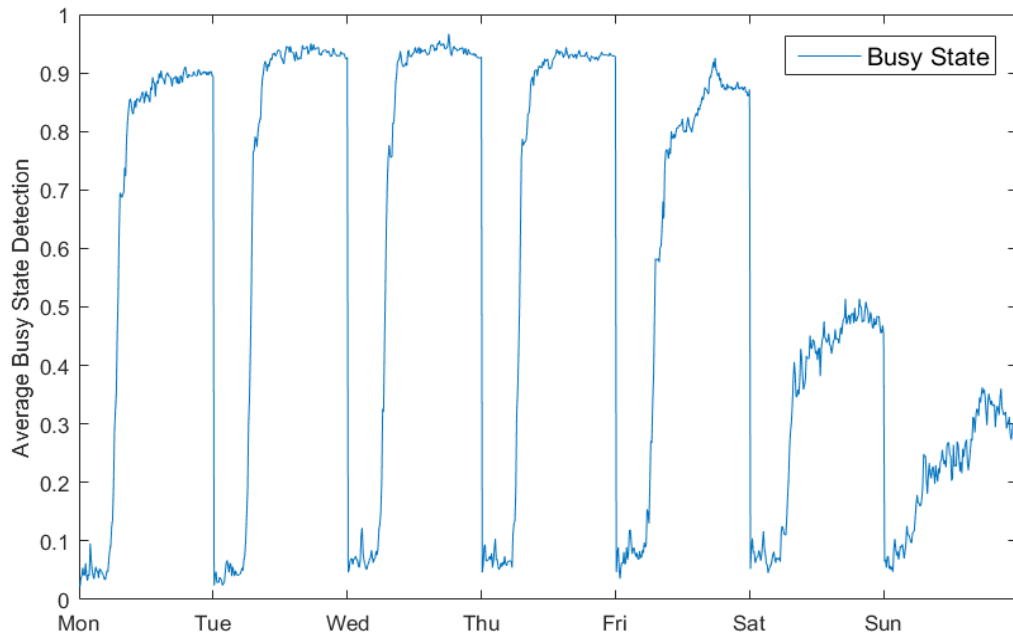


Figure 3.4: Average busy detection state of the SCIFI network H's building APs during the time slots of a week

The occupancy analysis shows a clear idleness between 0AM and 6:AM on weekdays and throughout the whole day on weekends or holidays. Those results indicate that it is

possible to reduce the number of active APs and reduce the energy consumption without causing any impact in the few network users connected to it during those periods. However the occupancy analysis also reveals that even between 6AM and 11PM on weekdays, where the average number of users is considerable, it is possible to reduce the number of active APs. Even in the busiest weekdays, there is an average of idle APs that ranges roughly between 20% and 10% of all APs in the building. Therefore even on busy hours and days there are APs being over provisioned for the demand. These analysis strongly suggests that the usage of RoD strategies can help to reduce the SCIFI network energy consumption.

Chapter 4

Proposed Unified Methodology

In this chapter we present our proposed unified experimental methodology [2] based on machine learning to evaluate classification and regression models about their capacity to accurately predict access point demands for energy-efficient smart buildings. Our proposed experimental methodology considers several machine learning algorithms and methods for constructing distinct classification and regression models using multiple input and output configurations.

First, in Section 4.1 we present the major steps and characteristics of our unified methodology. Section 4.2 presents the experimental analysis with UFF's SCIFI network data. In Section 4.3 further discussion on how our proposed unified methodology can help deciding the most suitable method to be used for several distinct smart buildings scenario is given.

4.1 Unified Methodology

Figure 4.1 shows a schema of our proposed unified methodology and its major steps, which are i) data acquisition and dataset construction; ii) input configuration; iii) regression and classification model configuration; iv) model selection.

The first step, shown in the upper part of the figure, is to prepare four datasets to be used for the evaluation of classification and regression prediction models. Then, in the second step, we use several input feature configurations, training set constructions, distinct single-label and multi-label machine learning methods to build our classifiers or distinct single-target and multi-target machine learning methods to build our regressors, in order to evaluate the significance of these characteristics for prediction models.

In the third step, we build single and multi-label classifiers capable of predicting the occupancy states for network APs and/or the construction of single and multi-target regressors capable of predicting the occupancy count for network APs. For multi-label classification, we propose using BR and CC problem transformation methods and Multi-layer Perceptron ANN to produce forecasts. For multi-target regression, we propose using BR and RC problem transformation methods and Multilayer Perceptron ANN to produce those predictions.

Finally, in the last step shown in Figure 4.1, an evaluation using multi-label/multi-target and single-label/single-target metrics helps the selection of a model that provides the best performance results and that can be used in smart building energy-efficient systems for several purposes.

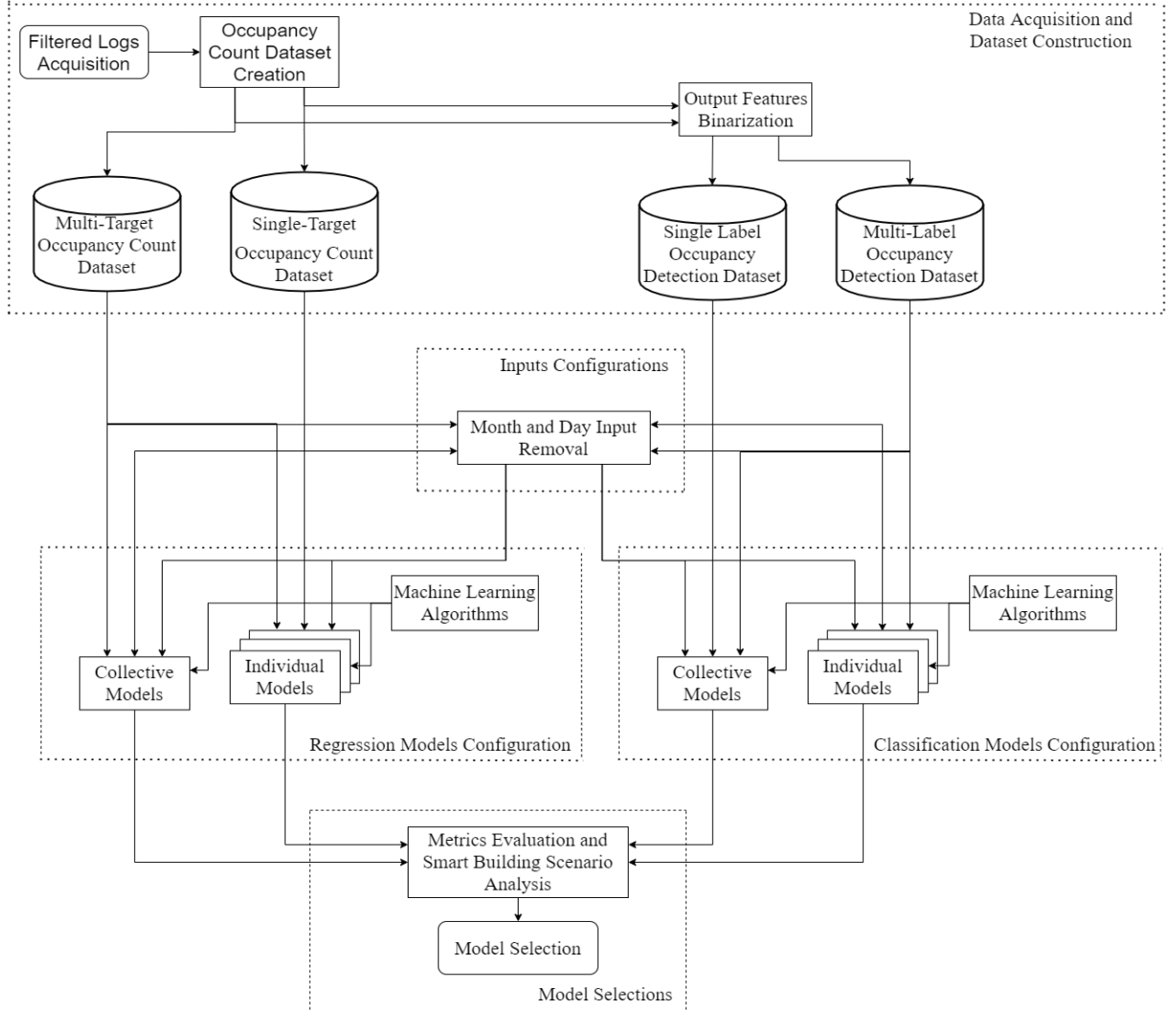


Figure 4.1: Our Proposed Methodology

Our experiments, dataset transformations, classification and regression model con-

struction and measurements were developed using Python scikit-learn API [7] and Pandas [40].

4.1.1 Dataset Construction

After filtering and preprocessing event logs, our methodology creates a dataset that compiles information related to a daily occupancy history for APs during fixed time slots. We firstly create our occupancy count dataset containing the number of associated devices (see Table B.2 in Appendix B for an example). The occupancy count dataset creation idea is based on Sangogboye et al. [45] and Balaji *et al* [4]. We divided a day into 144 (10 minutes) time slots, where each time t_j represents a specific time interval of the day. Time T_0 represents the time slot between 00:00 and 00:10 and the rest follows it in a crescent order, always adding 10 minutes more when compared to the feature before its own time window. The exception is the feature T_{143} , the last one, which has only nine minutes and ranges from 23:50 to 23:59. In this way, the number of mobile stations associated to an AP at a specific period of time t_x is the number of mobile devices that have been connected before time period x and have not disconnected, plus the number of mobile devices that have connected to the AP during the time period x . Mobile devices that have been connected and disconnected during the time period are also counted. Therefore any mobile device that had been connected to the AP is counted, even if it was just for a few seconds.

On the occupancy detection dataset (see Table B.1 in Appendix B for an example), we are only interested in binary classification (whether the AP has some associated station or not), so we applied a label binarization filter to our dataset outputs, in order to transform each numeric occupation count into a boolean output feature. To be classified as occupied (value 1) for a 10 minute time interval, the AP needs to have at least one mobile station associated to it. If no mobile station tries to associate to that AP during the whole duration of that time slot, the AP is considered unoccupied (value 0). The datasets¹ show occupancy count and detection for each AP over a period of 6 months, from April to September 2018.

In the single-label/single-target datasets, each instance has only one output feature representing a specific date and time interval occupation. The single-label/single-target dataset contains the following input features: Month, Day, Day of the Week, Holiday, AP Identification (APid), Hour, Minute. The multi-label/multi-target datasets have each

¹The datasets are available at <https://github.com/midiacom/UFF-SCIFI-Datasets>

instance representing one specific date and 144 output features representing the time intervals of a day occupation. The multi-label/multi-target dataset contains the following input features: Month, Day, Day of the Week, Holiday, APid.

Month and Day are numeric and show the instance date. Day of the week is categorical and indicates one of the 7 week days. Holiday is boolean and indicates if the day is a normal semester day with lectures (False) or a public holiday or university vacation day (True). APid carries the access point identification number and it informs to which specific AP the occupancy history belongs. Hour and Minutes are also numerical and are only present in the single-label/single-target datasets. The Hour input feature ranges from 0 to 23 representing day hours. The Minute feature ranges from 0 to 50 in 10 minutes steps. Although we could have combined Hour and Minute features to create a time interval feature ranging from 0 to 144, we decided to keep semantic information given by the hour/minute tuple.

4.1.2 Single-label and Multi-label Classification Analysis

We evaluated multiple types of classification model constructions, with varying training and testing sets. We trained collective models where only one classifier was trained with information regarding all APs and responsible for predicting the occupancy detection of all APs. We also trained individual classification models where multiple classifiers were trained only using information regarding one specific AP and responsible for that AP occupancy detection prediction. We built collective MLP ANN Multi-Label (ML) and Single-Label (SL) classifiers for our tests. Our goal with these distinct single and multi-label model construction was to evaluate if the occupancy detection of one AP could benefit from information from other APs, to determine if an AP individual information is capable of giving satisfactory detection predictions and which method has the best performance among those tested.

These collective and individual multi and single-label classifiers were also tested using multiple input feature configurations. We decided to evaluate if Month and Day features were significant to our model predictions. Month and Day features give date information to the classification models, which could benefit their predictions giving seasonal insights. On the other hand, more features can also represent more noise and increase the size of the classification data, which can consequently turn into waste of space and insignificant accuracy enhancement. Therefore, all classifiers were trained with and without Month and Day features.

Our label features are used respecting the time interval order for constructing the chain in the CC method. Therefore, our feature chain goes in crescent order from T_0 to T_{143} . Our time sequenced output features helped chain selection order in CC, because finding label order can be challenging [23]. We used Decision Tree (DT), K-Nearest Neighbors (K-NN) and Random Forest (RF) machine learning algorithms for our SL classification models, as they present the best single-label Wi-Fi occupancy detection results according to Fang et al [18]. Sangogboye, Imamovic and Kjærgaard [45] also stated that these algorithms were among the best algorithms in their ML method. We used default parameters values for DT and RF and we used $K = 5$ for K-NN.

We also built ANN MLPs. Table 4.1 shows the MLP hyper parameters selected for both SL and ML classification models after a search over a list of possible values for hyper parameters. We used the grid search algorithm GridSearchCV present in scikit-learn API [7]. Other non-listed parameters kept their default values.

Table 4.1: MLP ANN parameter values.

MLP Parameter	Best SL/ST Parameter	Best ML/MT Parameter
Hidden layer size	400	900
Alpha	0.0001	0.001
Learning rate	invscaling	invscaling
Activation	logistic	relu
Max iteration	1000	1000
Random state	1	1

To evaluate the performance of these models, we apply a train/test split on our datasets. The order of the collected data must be respected both for training and testing. So, dataset instances from April to August were used for training, and September dataset instances were used for testing the models. We used 4 metrics to evaluate our classification models: A_{t_j} , P_{t_j} , R_{t_j} and $F1_{t_j}$, as well as their overall versions, as discussed in Section 2.1.2.

4.1.3 Single-target and Multi-target Regression Analysis

For occupancy count, we also tested multiple types of regression model construction, with various training and testing sets. We trained collective and individual regressors using

distinct training sets. These collective and individual Multi-Target (MT) and Single-Target (ST) regressors also were tested having several input feature configurations. Consequently those MT and ST collective and individual regressors were trained with and without the Month and Day features. Those regression model constructions evaluate if the occupancy count system could benefit from information from other APs, determine if an AP individual information is capable of giving satisfactory results and evaluate if Month and Day features were significant for predictions.

The output label chain in RC methods is the same used in CC. We used DT, K-NN, RF and the XG optimized gradient boosting ST learning regression algorithms. Later on, we decided to construct collective MLP ANN, Support Vector Machine (SVM) and Stochastic Gradient Descent (SGD) ST and MT regressors. But since the occupancy count data presents a high variance, these regressors had their input and output data normalized. We also decided to test the K-NN algorithm with normalized input and output data. The MLP hyper parameters selected after an extensive search for both ST and MT regression models are the ones shown in Table 4.1.

Analogously to the classifier evaluation, we also applied a train/test split on our datasets. Dataset instances from April to August were used for training, and September dataset instances were used for testing the models. We used three metrics to evaluate our regression models: $RMSE_{t_j}$, $RMSPE_{t_j}$ and $MAPE_{t_j}$, as well as their overall versions, as discussed in Section 2.1.2.

4.2 Experimental Analysis

This section shows the results of our experimental analysis. We analyze which machine learning method, algorithm, model construction type and input combinations are more suitable to scenarios where Wi-Fi data can be used for smart building systems.

4.2.1 Classifier Analysis

In what follows, we show the experimental analysis for the occupancy detection problem. The models were constructed using a combination of four distinct parameters: the SL method and 2 distinct ML (BR and CC) machine learning methods; 2 distinct types of model construction, which can be Collective (Col) or Individual (Ind); 2 distinct input configurations, one composed by APid, holiday and weekday features (APHDWD) and other by All Features (ALL), including AP Id, holiday, weekday, day and month features;

and 3 distinct machine learning algorithms (RF, DT and K-NN) for constructing both SL models and the base classifiers of the ML methods. We also constructed 2 collective SL and 2 collective ML MLP ANNs, one using APHDWD features and other using ALL features. These combinations result in 40 distinct models. In order to guide our analysis, we firstly compare BR and CC ML methods. Then, we compare the best ML method against the SL method. We then evaluate types of model construction, algorithms and inputs. Finally, we evaluate if there is any observable advantage of one combination of parameters over the others.

4.2.1.1 Multi-label Methods

We selected the best results from the 40 evaluated models. Figure 4.2 depicts the accuracy A_{tj} of the best machine learning algorithm for each possible BR and CC ML classification model parameter combinations. We can see that BR models have better accuracy results than CC, as well as they drastically decrease from 6 to 8AM for both methods.

CC performance can be explained by the unpredictable AP occupancy from 6 to 8AM as seen in Figure 3.1. As the occupancy and idleness occurrence in those time slots are very alike and the states occur almost randomly, it is harder for classifiers to give a correct occupancy prediction for them, which leads to worse accuracy. That accuracy loss introduces a greater error on the label feature prediction and consequently affects the rest of the chain since the next time slots take the previous results into consideration. Because BR does not take the previous prediction into account, those prediction errors do not propagate.

Table 4.2 shows the overall metrics \overline{A} , \overline{P} , \overline{R} and $\overline{F1}$ for the best assessed models. From Table 4.2, it is clear that the BR method got better overall results than the CC method. Metric evaluation also shows that models using only APHDWD as input features present better results than using ALL features. Thus, this result indicates that, for our data, seasonal information is not a significant feature for ML classification models. Metric evaluation also shows that there is no significant difference between the types of model constructions (Col vs Ind), which indicates that both collective and individual models are equally valid model construction types for occupancy detection.

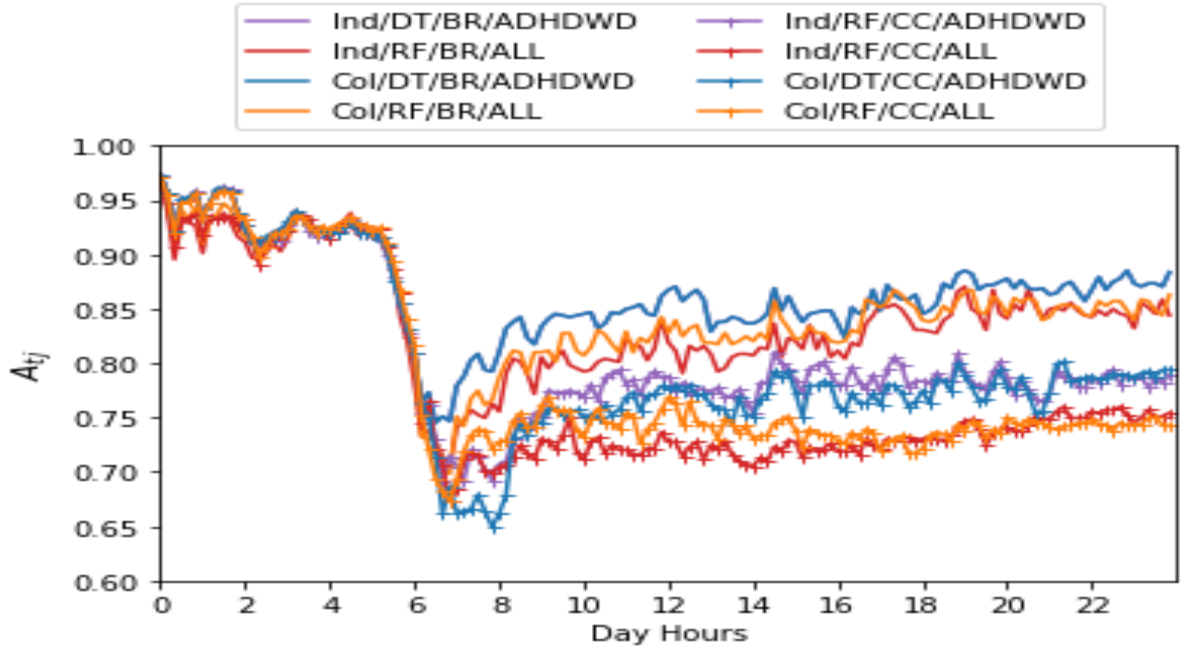


Figure 4.2: Accuracy A_{ij} for several BR and CC ML methods and parameter configurations

4.2.1.2 Multi-label and Single-label Evaluation

From the multiple combinations of parameters for constructing the SL and ML models, we chose at least one of the best results of 8 combinations for a deeper analysis. Figure 4.3 shows the A_{ij} accuracy of these 8 models, where we can notice that there is no significant difference between the ML and SL correspondent models. For instance, the A_{ij} curve of models Col/DT/BR/ADHDWD, Col/DT/SL/ADHDWD, Ind/RF/BR/ADHDWD and Ind/DT/SL/ADHDWD are quite similar. Also, we could observe that models using only APHDWD features had better results than models using all features (ALL).

Table 4.3 shows the overall metrics \bar{A} , \bar{P} , \bar{R} and $\bar{F1}$ for the best ML and SL models. It also shows the results for MLP ANN models. Table 4.3 demonstrates that the seasonal information do not improve the model predictions. Models using only the APHDWD features had better overall results, which suggest that day and month features carry no significant information about our occupancy data. Our results and the results in [51] comprise the same seasons and yet they showed distinct conclusions about seasonal information. Results reported in [51] showed that seasonal information carries relevant information about the occupancy data. One explanation for that difference can be the low influence of tropical climate at latitude -22.9, where UFF is located.

Table 4.3 shows that there is no significant difference between ML and SL methods.

Table 4.2: Classification performance results for BR and CC ML methods

Constructed Models	\bar{A}	\bar{P}	\bar{R}	$\bar{F1}$
Col/DT/BR/APHDWD	0.8669	0.8662	0.8960	0.8808
Col/DT/CC/APHDWD	0.8025	0.8683	0.7548	0.8076
Col/RF/BR/APHDWD	0.8631	0.8570	0.9010	0.8784
Col/RF/CC/APHDWD	0.8201	0.8536	0.8115	0.8320
Col/DT/BR/ALL	0.8268	0.8261	0.8671	0.8461
Col/DT/CC/ALL	0.7664	0.7693	0.8207	0.7942
Col/RF/BR/ALL	0.8495	0.8508	0.8804	0.8653
Col/RF/CC/ALL	0.7863	0.9161	0.6724	0.7756
Ind/DT/BR/APHDWD	0.8669	0.8662	0.8960	0.8808
Ind/DT/CC/APHDWD	0.8025	0.8683	0.7548	0.8076
Ind/RF/BR/APHDWD	0.8631	0.8566	0.9015	0.8785
Ind/RF/CC/APHDWD	0.8113	0.8469	0.8013	0.8235
Ind/DT/BR/ALL	0.8155	0.8155	0.8581	0.8363
Ind/DT/CC/ALL	0.7782	0.7797	0.8309	0.8045
Ind/RF/BR/ALL	0.8412	0.8412	0.8762	0.8583
Ind/RF/CC/ALL	0.7770	0.8755	0.6924	0.7733

From Table 4.3, we can also notice that there is no significant difference between collective and individual models. These conclusions make both machine learning methods and both model construction types equally valid. It is also possible to observe from Table 4.3 that DT and RF algorithms were the most suited for the occupancy detection problem. Finally Table 4.3 shows that the ML MLP ANN fails to have comparable results, however the Col/MLP/SL/APHDWD ANN got comparable results to the Col/DT/SL/APHDWD model.

We found that DT and RF machine learning algorithms were the most suited for occupancy detection. Since there was no noticeable difference on the evaluation metrics for ML and SL individual and collective models using the RF and DT algorithms, we decided to evaluate their model sizes in order to compare them. Smaller models are not only simpler to understand, but they also require less memory space to be stored and are also faster to traverse, which leads to a faster result and smaller CPU requirements to run them. Table 4.4 shows the mean Number of Leaves (Numb. of Leaves), depth and their

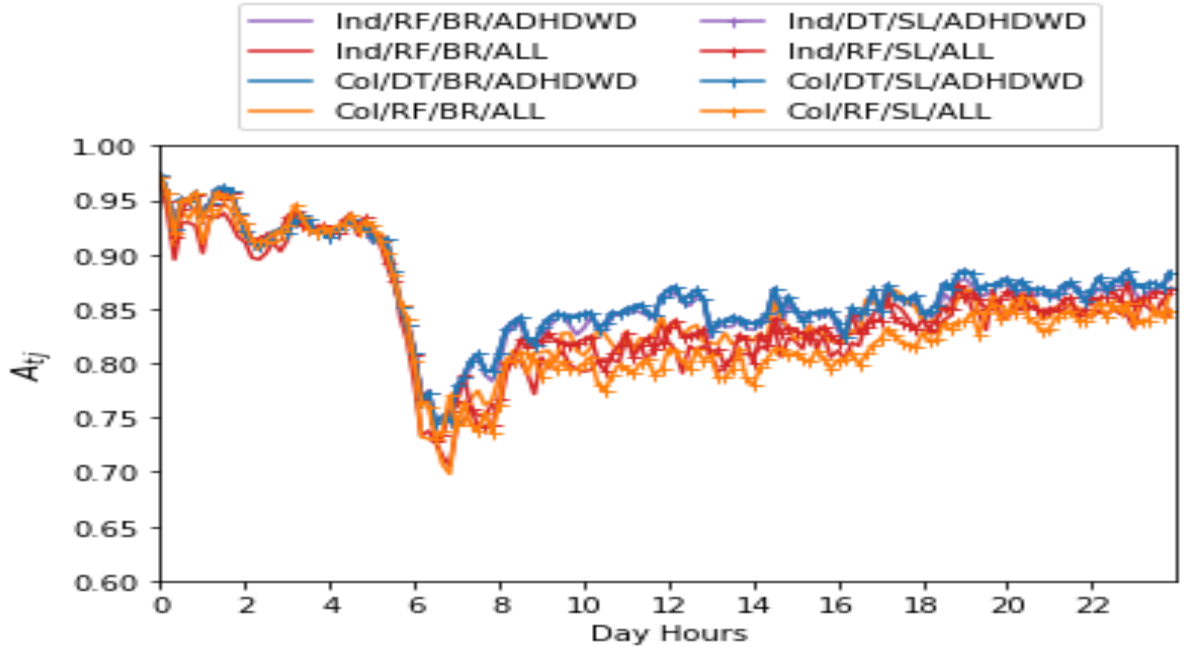


Figure 4.3: Accuracy A_{ij} of ML and SL methods for several parameter configurations

respective Standard Deviation (Std. Dev.) for all model possible combinations using only APHDWD input features. In this table, we can observe that SL models have a smaller size when compared to ML models. This was expected because the ML BR method consists of a group of individual SL models, each for one specific label. The second conclusion is that DT algorithms are significantly smaller when compared to RF algorithms. This result was also expected since random forests are a collection of decision trees. Finally, we can notice that collective models are larger than individual models. Since individual models train over a smaller part of the dataset they also present smaller sizes. SL and DT algorithms form the best combination to be used in scenarios using our data, because they are simpler and smaller. However, the same cannot be said about individual models over collective models. Individual models are smaller but they only give information about one AP. Depending on the scenario characteristics, the collective model can actually be a better option, such as in our motivation scenario where a central unit is responsible for the management of the whole AP network.

4.2.2 Regression Analysis

This section shows the experimental analysis for the occupancy count problem. We evaluated several regression models using ST and MT machine learning methods. 48 models were built using a combination of four distinct parameters: the ST method and 2 distinct MT (BR and RC) methods; 2 distinct types of model construction, which can be Col or

Table 4.3: Classification performance results for BR ML and SL methods.

Constructed Models	\bar{A}	\bar{P}	\bar{R}	$\bar{F1}$
Col/DT/BR/APHDWD	0.8669	0.8662	0.8960	0.8808
Col/DT/SL/APHDWD	0.8669	0.8662	0.8960	0.8808
Col/RF/BR/APHDWD	0.8631	0.8570	0.9010	0.8784
Col/RF/SL/APHDWD	0.8634	0.8567	0.9021	0.8788
Col/MLP/ML/APHDWD	0.8201	0.8536	0.8115	0.8320
Col/MLP/SL/APHDWD	0.8669	0.8662	0.8960	0.8808
Col/DT/BR/ALL	0.8268	0.8261	0.8671	0.8461
Col/DT/SL/ALL	0.8277	0.8878	0.8878	0.8498
Col/RF/BR/ALL	0.8495	0.8508	0.8804	0.8653
Col/RF/SL/ALL	0.8388	0.8241	0.8981	0.8595
Col/MLP/ML/ALL	0.7737	0.7359	0.9170	0.8165
Col/MLP/SL/ALL	0.8510	0.8880	0.8339	0.8601
Ind/DT/BR/APHDWD	0.8669	0.8662	0.8960	0.8808
Ind/DT/SL/APHDWD	0.8669	0.8662	0.8960	0.8808
Ind/RF/BR/APHDWD	0.8631	0.8566	0.9015	0.8785
Ind/RF/SL/APHDWD	0.8633	0.8572	0.9011	0.8786
Ind/DT/BR/ALL	0.8155	0.8155	0.8581	0.8363
Ind/DT/SL/ALL	0.8268	0.8214	0.8747	0.8472
Ind/RF/BR/ALL	0.8412	0.8412	0.8762	0.8583
Ind/RF/SL/ALL	0.8511	0.8445	0.8934	0.8683

Ind; 2 distinct input configurations, one composed by APHDWD features and other by ALL; and 4 distinct machine learning algorithms (RF, DT, K-NN, XG) for constructing both ST models and the base regressors of the MT methods. We also constructed 2 collective ST MLP ANNs and 2 collective MT MLP ANNs, using APHDWD features and using ALL features. Additionally, we constructed 12 more collective regression models using a combination of three distinct parameters: 3 distinct machine learning algorithms (SVM, SGD, K-NN); 2 distinct normalized input configurations, one composed by APHDWD normalized features and other by all normalized features (ALL); and 2 machine learning methods (ST and BR). These combinations result in 64 distinct models.

We firstly evaluate BR and RC MT methods. Then, we compare the best MT method

Table 4.4: DT and RF classifier’s mean number of leaves and depth size evaluation.

Constructed Models	Mean Numb.	Numb. of Leaves	Mean	Depth
	of Leaves	Std. Dev.	Depth	Std. Dev.
Col/DT/BR/APHDWD	43409	-	2393	-
Col/DT/SL/APHDWD	37756	-	33	-
Col/RF/BR/APHDWD	1918721	-	117475	-
Col/RF/SL/APHDWD	1818221	-	1604	-
Ind/DT/BR/APHDWD	1587	141	601	63
Ind/DT/SL/APHDWD	1346	172	20	2
Ind/RF/BR/APHDWD	71903	6690	30805	3490
Ind/RF/SL/APHDWD	62345	8219	932	22

against ST methods. We evaluate which model construction type, algorithms and inputs give the best results. Lastly, we evaluate if there is any observable advantage of one method over the others.

4.2.2.1 Multi-target Methods

As we tested 64 distinct models, the results shown here are the compilation of the best results found. Figure 4.4 shows the $RMSE_{tj}$ of the best machine learning algorithm for each possible BR and RC MT regression model parameter combinations. Figure 4.4 shows that the BR method models have lower $RMSE_{tj}$ values than the RC models and that the $RMSE_{tj}$ results start to significantly increase after 6AM for both methods.

Another interesting observation when comparing Figures 4.4 and 3.1 is that $RMSE_{tj}$ increasing behavior is very similar to the occupancy behavior. This means that heavily occupied hours have higher $RMSE_{tj}$ errors. Therefore, $RMSE_{tj}$ is a numerical error metric that alone cannot be enough to evaluate how good the occupancy count predictions are for each time slot individually. Figure 4.5 shows $RMSPE_{tj}$. We can observe that the BR method got better results than the RC method. BR better performance over RC can be explained by the same reasons we have discussed in Section 4.2.1.1.

Comparing Figures 4.4 and 4.5, we can also notice that, even though the $RMSE_{tj}$ values are higher for predictions after 9AM, their $RMSPE_{tj}$ values are smaller. Even though the absolute occupancy count error of these time intervals are higher, they are comparatively smaller than the data variance and therefore we can conclude that model

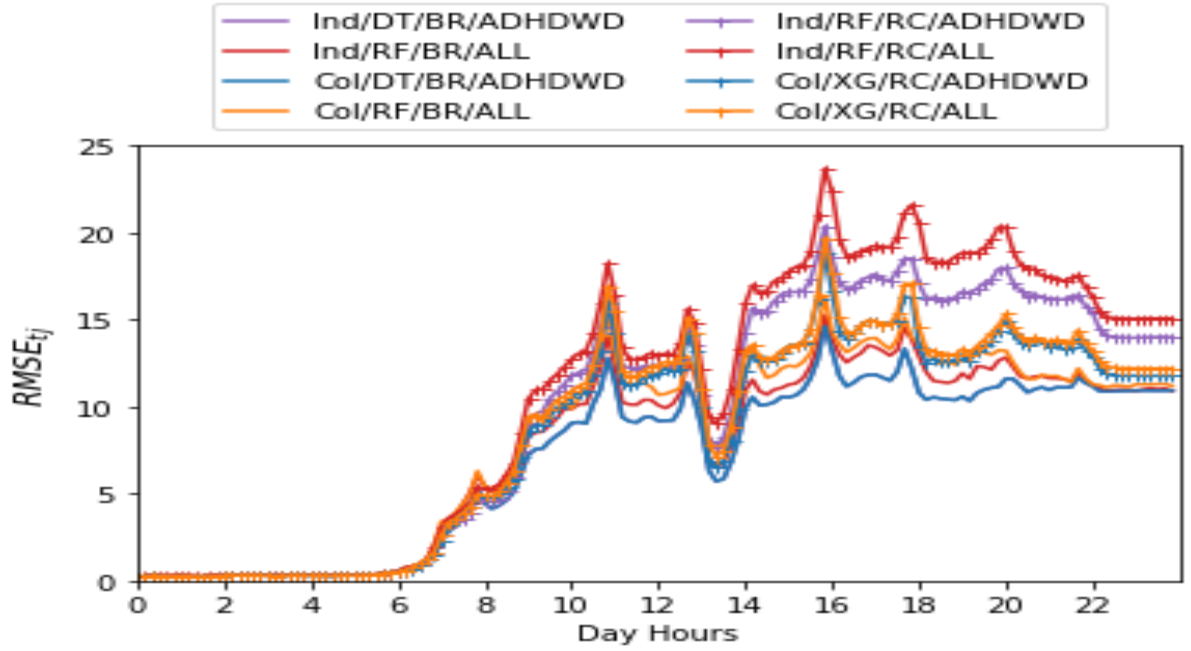


Figure 4.4: $RMSE_{tj}$ for several BR and RC MT methods and parameter configurations

predictions are acceptable. The $RMSPE_{tj}$ values presented before 9AM are relatively higher, being almost equal or superior to the variance itself. This happens because these hours real occupancy is low and presents a small variance. Therefore for late-night and early-morning hours, $RMSE_{tj}$ values are comparatively higher than the data variance. However since these hours correspond mostly to closing hours, we can not say that an occupancy count model would not be applicable. Even if we might be doubling the occupancy count values due to prediction errors, the total occupancy count would still be low. So, depending on the scenario and systems, these errors can be easily overcome.

Table 4.5 shows the overall metrics \overline{RMSE} , \overline{RMSPE} and \overline{MAPE} for the best models. Metric evaluation for the regression problem shows that models using only APHDWD input features had better results than the models that used ALL features, which indicates that seasonal information is also not a significant feature for MT regression models. Metric evaluation also shows that there is no significant difference between the model construction types, indicating that both models are equally valid for occupancy count prediction.

4.2.2.2 Multi-target and Single-target Evaluation

Figure 4.6 compares $RMSPE_{tj}$ among the best machine learning algorithms for MT and ST regression model construction combinations. It shows that there is no significant difference between MT and ST correspondent models. However it is possible to notice

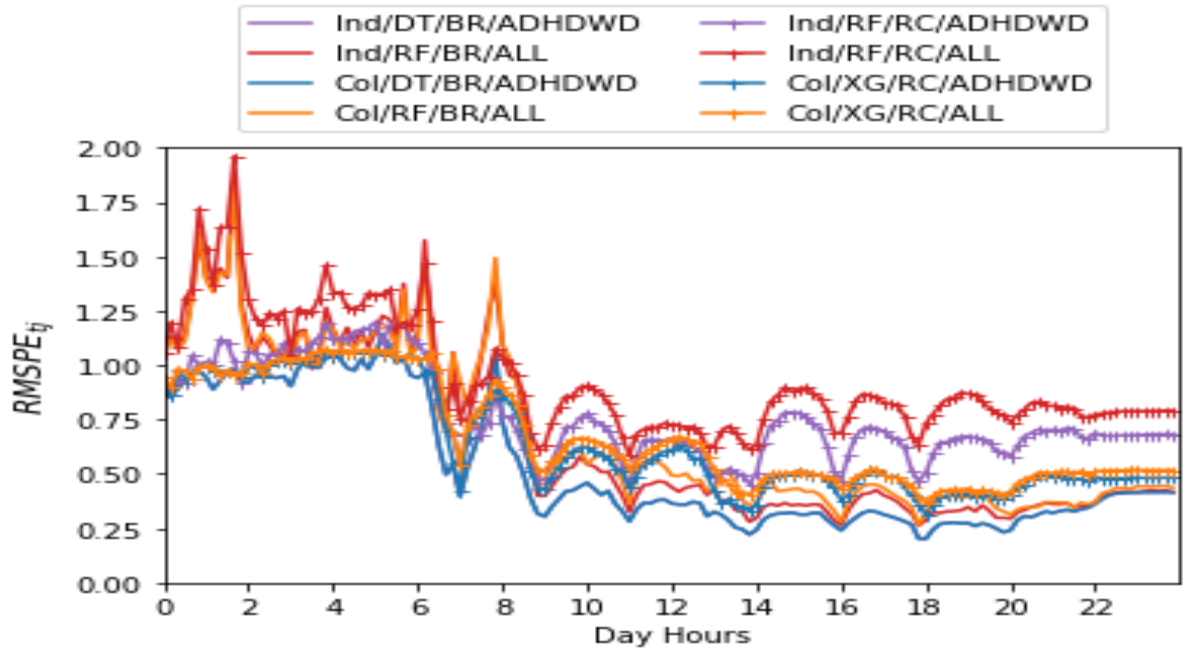


Figure 4.5: $RMSPE_{tj}$ for several BR and RC MT methods and parameter configurations

that models using only the APHDWD features had better results than the models that used ALL features.

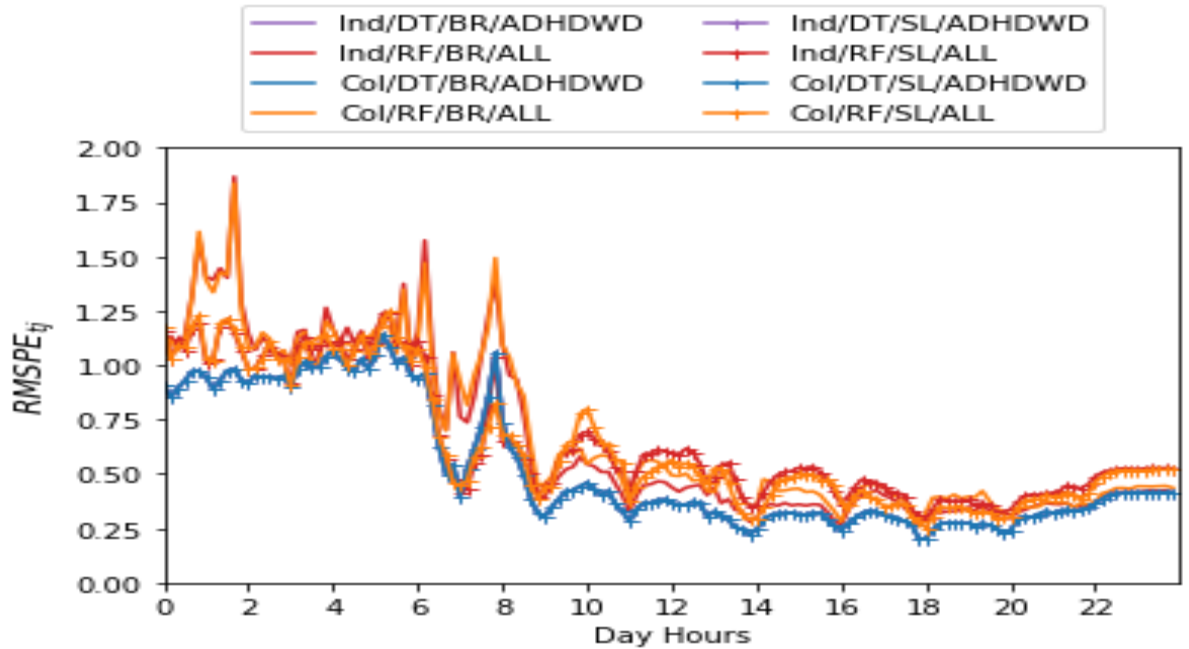


Figure 4.6: $RMSPE_{tj}$ of MT and ST methods for several parameter configurations

Table 4.6 shows the overall metrics \overline{RMSE} , \overline{RMSPE} and \overline{MAPE} for the best assessed models. It also shows the results for the MLP ANN models. Table 4.6 shows that regression models using only the APHDWD features had better overall results, which sug-

Table 4.5: Regression performance results for BR and RC MT methods.

Constructed Models	\overline{RMSE}	$\overline{RMSP\bar{E}}$	\overline{MAPE}
Col/DT/BR/APHDWD	8.4161	0.2977	0.4189
Col/DT/RC/APHDWD	13.4306	0.7843	0.6723
Col/RF/BR/APHDWD	8.4223	0.2980	0.4191
Col/RF/RC/APHDWD	11.4110	0.5690	0.5676
Col/DT/BR/ALL	11.2734	0.5379	0.5532
Col/DT/RC/ALL	16.2836	1.1478	0.8887
Col/RF/BR/ALL	9.6314	0.3880	0.4821
Col/RF/RC/ALL	11.6615	0.5896	0.5989
Ind/DT/BR/APHDWD	8.4161	0.2977	0.4189
Ind/DT/RC/APHDWD	12.8850	0.7168	0.6375
Ind/XG/BR/APHDWD	8.4174	0.2978	0.4189
Ind/XG/RC/APHDWD	11.9943	0.6255	0.5803
Ind/DT/BR/ALL	11.0656	0.5151	0.5522
Ind/DT/RC/ALL	15.3127	1.0207	0.8293
Ind/XG/BR/ALL	10.3054	0.4528	0.5841
Ind/XG/RC/ALL	13.2117	0.7667	0.7024

gest that day and month features carry no significant information about our occupancy data for the regression problem too. Table 4.6 overall metric evaluation shows that there is no significant difference between MT and ST methods and that there is no significant difference between collective and individual models, which make both machine learning methods and both model construction equally possible considering performance. It is also possible to observe in Table 4.6 that DT and RF algorithms were the best machine learning algorithms for occupancy count prediction. Table 4.6 shows that MLP ANN fails to have comparable results. However it is worth mentioning that the Ind/XG/BR/APHDWD model got comparable results to the DT collective ST model using APHDWD features.

DT and RF machine learning algorithms had better results for occupancy count than the others. Since there was no noticeable difference on the evaluation metrics for MT and ST individual and collective models using these algorithms, we also decided to evaluate their model sizes. The model size impacts on memory space and CPU requirements. Table 4.7 shows the mean number of leaves, depth and the standard deviation for all

Table 4.6: Regression performance results for BR MT and ST methods.

Constructed Models	\overline{RMSE}	$\overline{RMSP\bar{E}}$	\overline{MAPE}
Col/DT/BR/APHDWD	8.4161	0.2977	0.4189
Col/DT/ST/APHDWD	8.4161	0.2977	0.4189
Col/RF/BR/APHDWD	8.4223	0.2980	0.4191
Col/RF/ST/APHDWD	8.4221	0.2981	0.4192
Col/MLP/MT/APHDWD	10.6320	0.4875	0.6037
Col/MLP/ST/APHDWD	11.2536	0.5489	0.6403
Col/DT/BR/ALL	11.2734	0.5379	0.5532
Col/DT/ST/ALL	10.2350	0.4450	0.4994
Col/RF/BR/ALL	9.6314	0.3880	0.4821
Col/RF/ST/ALL	9.7373	0.3994	0.4718
Col/MLP/MT/ALL	14.3063	0.8955	0.8926
Col/MLP/ST/ALL	14.0249	0.8791	0.8336
Ind/DT/BR/APHDWD	8.4161	0.2977	0.4189
Ind/DT/ST/APHDWD	8.4161	0.2977	0.4189
Ind/XG/BR/APHDWD	8.4174	0.2978	0.4189
Ind/XG/ST/APHDWD	9.0894	0.3625	0.4732
Ind/DT/BR/ALL	11.0656	0.5151	0.5522
Ind/DT/ST/ALL	10.7165	0.4900	0.5231
Ind/XG/BR/ALL	10.3054	0.4528	0.5841
Ind/XG/ST/ALL	10.2894	0.4571	0.5777

model combinations using APHDWD features. This table shows that ST models are smaller when compared to MT models and that the DT algorithm is significantly smaller when compared to the RF algorithm. We can also notice that collective models are bigger than individual models. The reason why these results are expected are the same ones we have discussed in Section 4.2.1.2. ST method and DT algorithm are a better combination to be used in our scenario once they are simpler and smaller than MT methods and the RF algorithm. However, the same cannot be said about individual models over collective models. As we have discussed in Section 4.2.1.2, the collective model can actually be a better option depending on the scenario characteristics.

Table 4.7: DT and RF regressor’s mean number of leaves and depth size evaluation

Constructed Models	Mean Numb.	Numb. of Leaves	Mean	Depth
	of Leaves	Std. Dev.	Depth	Std. Dev.
Col/DT/BR/APHDWD	48066	-	-	2355
Col/DT/ST/APHDWD	46136	-	-	29
Col/RF/BR/APHDWD	2300688	-	-	115052
Col/RF/ST/APHDWD	2219991	-	-	1465
Ind/DT/BR/APHDWD	1735	170	633	69
Ind/DT/ST/APHDWD	1647	205	18	1
Ind/RF/BR/APHDWD	83215	8995	30949	3556
Ind/RF/ST/APHDWD	887	10593	887	17

4.3 Further Discussion on Our Methodology and Results

While other authors have analyzed how multiple machine learning algorithms may change the model prediction results, all studies we have seen in literature did that using only a specific ML or SL/ST method with a specific model construction type and input configuration, as discussed in Section 2.2.1. Therefore, they were able to evaluate which machine learning algorithm they should chose for their model. However, our experimental analysis showed that the model construction type, machine learning method and input configuration shall also be taken into consideration depending on the scenario. As we have seen in our experimental analysis, our proposed methodology allowed us to draw numerous conclusions about the types of model constructions, input configurations, machine learning methods and algorithms and helped on the decision of a best combination choice for our experimental scenario.

However, this analysis also shows that not always the best combination will remain the same for all possible scenarios. In this section we discuss how distinct scenarios may affect the model best combination choice.

4.3.1 Seasonal Information

In our scenario, where we used Wi-Fi association information to build a wireless network energy efficient management system without real time data acquisition, month and day

input features should not be used once these features showed no enhancement on the prediction model results. On the other hand, although data used in [51] and [45] present the same seasons of our experimental analysis, they showed seasonal information as a relevant input feature. Those studies were made in northern hemisphere countries in temperate regions, such as the ones found in Europe and North America, while our data were collected in a tropical country in South America. Therefore, we can conclude that seasonal information must be analyzed in these types of systems since is not always significant and depending on your building's location it should or should not be used as an input.

4.3.2 Individual and Collective Comparison

Another important question to answer is which type of model construction, individual or collective, should be used. Individual and collective models can have distinct results as they are trained with distinct dataset information. Our experimental analysis showed that there was no difference between the individual and collective models except for their sizes, where individual models were much smaller than the collective ones. However, it is not always true that information regarding various sensors can benefit other sensor's predictions. Also, further examination based on the scenario is required since model sizes can be relative. In our motivation scenario, for example, individual models would be actually bigger, once the collection of individual models stored at the central unit would be bigger than one single collective model capable of giving predictions for all APs. In scenarios where each individual model is deployed in its respective sensor or actuator, they would be smaller than the collective model.

Chapter 5

Proposed eSCIFI Mechanism

As previously discussed in Section 3.2, we have seen that most of UFF SCIFI network APs at the H building are switched on despite being idle. Those active idleness causes an unnecessary waste of energy. Therefore an energy saving WLAN mechanism based on RoD strategies, or simply RoD strategy mechanisms, that effectively control WLAN resources can help to prevent those energy waste while coping with the users demand. This dissertation proposes the eSCIFI energy saving mechanism for WLANs. eSCIFI uses machine learning prediction models and other RoD strategies to create an energy saving mechanism. The eSCIFI mechanism can also work with non SDN large wireless networks and/or large wireless networks where real-time data acquisition is not possible. Those possibilities make the eSCIFI a feasible solution for a greater number of wireless networks in use, especially university networks, such as the UFF's SCIFI network, which was used for evaluating our proposal.

5.1 eScifi Mechanism

Figure 5.1 shows eSCIFI main architectural components and its major steps, which are i) the unified methodology ; ii) the hybrid model; iii) heuristic algorithm.

The first step, shown in the left upper part of the figure, is to use our unified methodology to create the datasets and select the best regression and classification model configuration parameters. Later on, in the hybrid model, we combine the best trained regression and classification models selected in our unified methodology to give the future access points (APs) occupancy estimation. Those occupancy estimation is used by our heuristic algorithm to define which APs should be turned on or off.

In the heuristic mechanism, we first extract the APs statistics from the generated dataset. Later on, the heuristic network clusters formation uses the APs neighborhood list and the APs statistics to create the network clusters that can guarantee a minimum coverage to the network. Finally, the energy state decision algorithm uses the defined network clusters and the APs occupancy estimation to decide which APs should be switched on/off to cope with the network users demand. At the end of this process, our heuristic mechanism provides an energy scheduling of all APs in the network for an entire day that can guarantee a minimum coverage to the network while coping with the network users demand. That way the eSCIFI mechanism needs to run only once in the day to generate the energy scheduling of all APs in the network. Therefore the eSCIFI mechanism can run at any moment of low activity in the network such as late night hours after midnight in our case. This functioning scheme guarantees that the eSCIFI can run at any network controller without burdening its processing capacity.

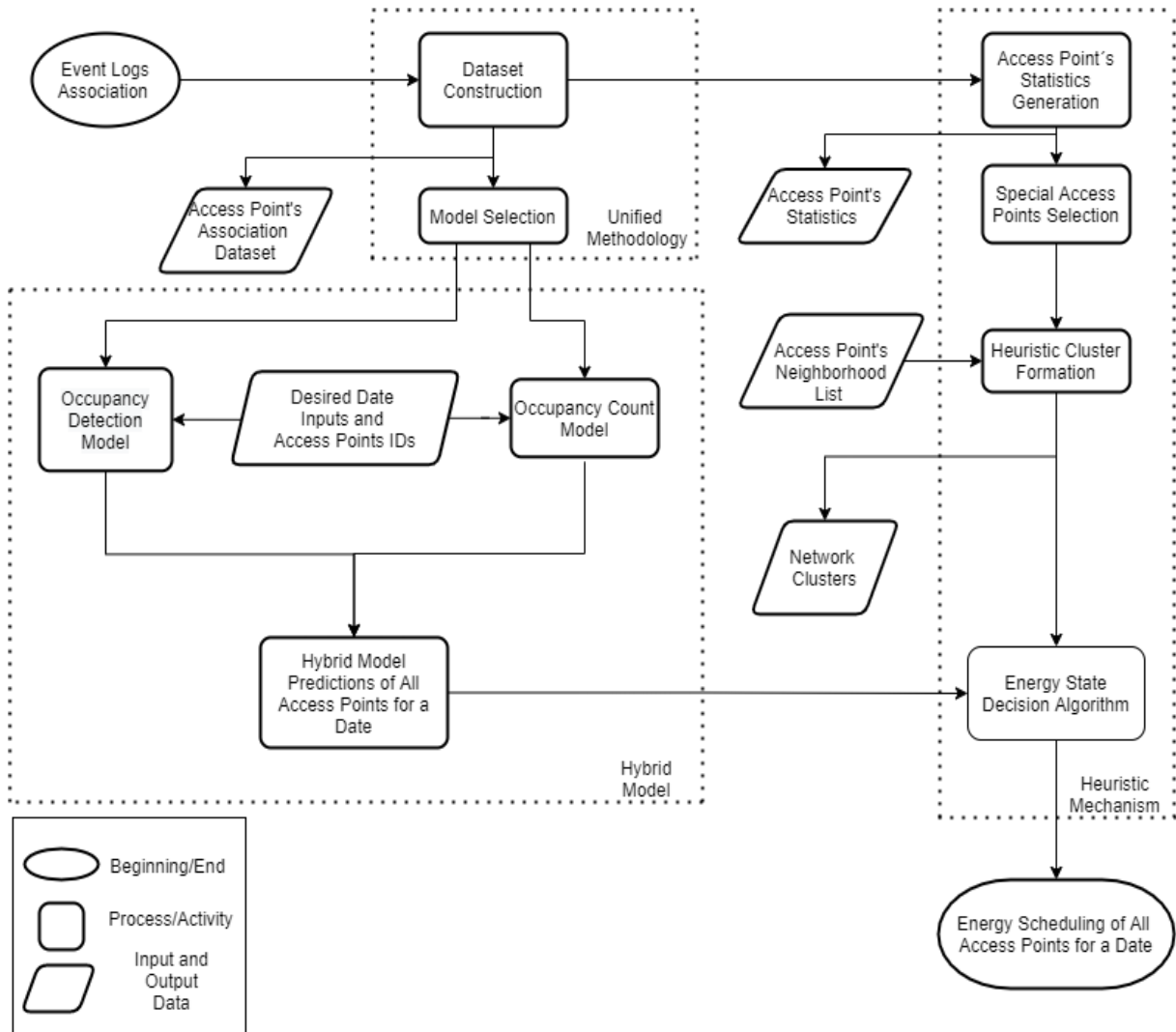


Figure 5.1: eSCIFI architecture

5.1.1 Unified Methodology and Model Selection

The unified methodology proposed in Chapter 4 explains how the occupancy count and occupancy detection dataset were created. Those datasets are crucial to extract the APs statistics and later on to select the special APs that are necessary for the network clusters formation. The model selection process in our unified methodology compares several model configuration and hyper parameters in order to determine the best classification and regression models for our evaluation scenario. Results show that the Col/DT/SL/APHDWD machine learning model is the best classification model while Col/DT/ST/APHDWD is the best regression model for our scenario. Therefore the Col/DT/SL/APHDWD machine learning classification and Col/DT/ST/APHDWD regression model will be used on the hybrid model to provide future usage predictions for the H building UFF SCIFI Wi-Fi network.

5.1.2 Hybrid Model

Observing Figure 4.6, it is possible to notice that even the best regression model has significant $RMSP E_{t_j}$ values during night and morning time slots, but the $RMSP E_{t_j}$ values for time slots after midday decrease. Figure 4.3, on the other hand, shows that A_{t_j} values for night and morning time slots are relatively higher than the time slots for the rest of the day. Therefore we propose a hybrid model. The hybrid model combines the accuracy results given by the classification models with the regression results given by the regression models in order to create a better occupancy count estimation. Considering CMR as the classification results matrix that shows the occupancy detection estimations provided by the classifier for the APs and RMR as the regression results matrix that shows the occupancy count estimations provides by the regressor, we can define that the hybrid model estimation HMR is the Hadamard product result between both CMR and RMR matrices. Equation 5.1 shows the Hadamard product that produce the hybrid model results matrix that is used as the demand estimation by our mechanisms. Figure 5.2 also shows an example of how the hybrid model result is created. The hybrid model results use the occupancy detection given by the classification model to determine whether or not a time slot prediction has users associate with the AP and the occupancy count given by the regression model to estimate the number of users associated to the AP for a time slot that has been classified as occupied.

$$HMR = CMR \circ RMR \quad (5.1)$$

Classification Result						Regression Result				
AP1	0	0	...	1		AP1	0	1	...	20
AP2	0	1	...	1		AP2	0.5	3.7	...	15.3
AP3	1	0	...	1		AP3	5.1	5.5	...	10.1

⊗

||

Hybrid Result				
AP1	0	0	...	20
AP2	0	3.7	...	15.3
AP3	5.1	0	...	10.1

Figure 5.2: Hybrid model result creation example

Figure 5.3 shows how the hybrid model demand prediction results are closer to the real demand than the regression model demand predictions for the month of September 2018. In fact, Figure 5.3 shows that the hybrid model results can reduce the over demand prediction that happened on the weekends (September 1,2,8,9,15,16,22,23,29,30) and on the Brazil's Independence day public holiday (September 7). It is important to highlight that the difference between the results is not significant enough to prove that the hybrid model is a better regression prediction model than the pure regression model for all scenarios. Depending on the scenario, the pure regression model can be a better option and used without imposing any change to the eSCIFI operation, but since it has shown better results in our case scenario, we decided to use the hybrid model instead.

The Hybrid model created only uses the APid, day of the week and holiday attributes as input features. Consequently there are only 14 possible demand estimations for a specific AP (one for each regular day of the week and one for each holiday on these days). Therefore we decided to compare the results of our hybrid model with a mean estimator. The occupancy count prediction provided by the mean estimator for a specific set of input features (APid, day of the week and holiday) is the average occupancy count of that specific set of input features in the association history. We compared the results of this mean estimator with the results of our hybrid model. Table 5.1 shows that the hybrid model had better \overline{RMSE} , \overline{RMSPE} and \overline{MAPE} results when compared to the mean estimator model. Those better results shown in Table 5.1 can be explained by the fact that the hybrid model results have reduced the error predictions that happened on weekends and on public holidays when compared to the mean estimation model results. Those reduced demands on weekend and on public holidays were more significant than the

errors caused in night time slots by the hybrid models results and therefore the \overline{RMSE} , \overline{RMSEP} and \overline{MAPE} results were lower in the hybrid model results.

As we have already previously explained in 4.2.2.1, those worse results during night time slots can be easily overcome because the occupancy prediction would still be low and would not impact much on the AP's energy state decision made by our heuristic algorithm. Therefore we decided to use the hybrid model results once it has reduced the overall \overline{RMSE} , \overline{RMSEP} and \overline{MAPE} metrics.

The mean estimator results in our case scenario are very close to those achieved by the hybrid model. However those results achieved by the mean estimator for our case scenario were only possible due the H building occupancy characteristics. The H building has only classrooms, so its occupation mainly occurs through lectures and exam applications. The lecture's schedule did not change drastically throughout the entire dataset which makes the occupancy behavior periodical and well behaved in our case. This behavior might not be common for other buildings in the university that have other rooms inside them (such as professor's offices or laboratories) or even other scenarios (such as parks or shopping centers). Therefore those results are not significant enough to prove that the hybrid model is not a better prediction model than the mean estimator for all scenarios. On other scenarios similar to ours, the mean estimator can be a viable option due to its

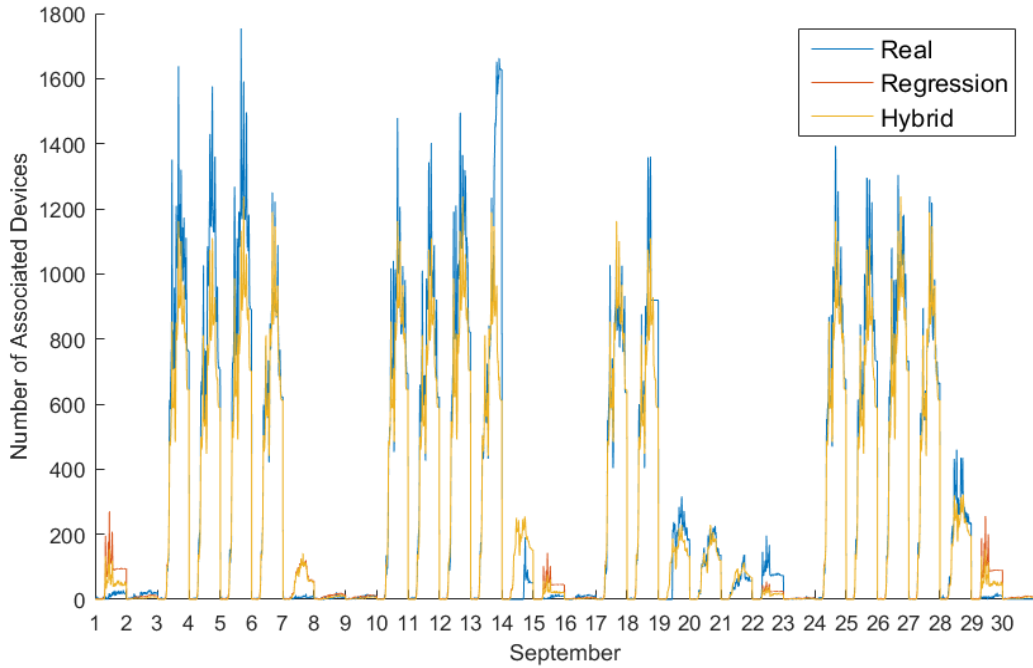


Figure 5.3: Hybrid model results compared with the real demand and the demand given by the regression results for the whole month of September

simplicity. The use of the mean estimator does not impose any change to the eSCIFI operation. However we decided to use the hybrid model since it has shown better results in our case scenario, specifically on weekends and holidays.

Table 5.1: Mean Estimator and Hybrid models performance results.

Metrics	Mean	Hybrid
\overline{RMSE}	8.4161	8.3996
\overline{RMSPE}	0.2977	0.2968
\overline{MAPE}	0.4189	0.4096

5.1.3 Heuristic Mechanism

The heuristic mechanism is responsible for providing the SCIFI APs energy state (on or off) schedule for a date. It is important to highlight that we only control the APs wireless interface energy state due to UFF SCIFFI existing infrastructure that only allows us to control its energy state. However in WLANs where the APs are connect to Power over Ethernet (PoE) switches, eSCIFI could normally control the energy state of the AP and not only its wireless interface.

Our heuristic mechanism has two main components: the heuristic cluster formation algorithm and the energy state decision algorithm. The clustering algorithm creates the AP clusters based on their neighborhood in order to guarantee the network coverage area to the clients. The energy state decision algorithm provides the energy state of all APs for a specific time slot and date based on the machine learning occupancy predictions and clusters. In next sections, we detail the heuristic cluster formation algorithm and the energy state decision algorithm and its challenges.

5.1.3.1 Heuristic Cluster Formation: cSCIFI and cSCIFI+

Jardsoh et al. [29] proposed a clustering algorithm called green clustering. The idea behind the green clustering algorithm is to create clusters of APs that are in proximity of each other. Several APs in large wireless network have overlapping coverage areas in order to cope with higher users demand. Those APs are in a spatially neighboring condition that allows one of them to provide coverage to the users of all APs in its vicinity. Therefore it is possible to create clusters of neighboring APs where any user within the cluster coverage

is able to connect to the network as long as at least one AP in the cluster is turned on. We proposed two heuristic cluster formation algorithms, cSCIFI (cluster SCIFI) and cSCIFI+ (cluster SCIFI +). Those clustering algorithms are based on the green clustering algorithm of Jardsoh et al. [29]. However we introduced some basic changes to improve the cSCIFI and cSCIFI+ clustering formation process such as the special AP set.

Our clustering algorithms need two input features to work: the neighborhood list and the special AP set. To create a neighborhood list, we need to define the vicinity criteria. Only APs that are considered neighbors can belong to the same cluster. Jardsoh et al. [28, 29] have used the spatial distance between APs and the median number of beacon messages and the median signal strength of the beacons as vicinity criteria. In our cSCIFI and cSCIFI+ algorithms, we are going to use the APs' signal quality scan to define our vicinity criterion. The SCIFI network periodically runs a signal quality scan that informs the different signal quality values received from the other APs that a certain AP has scanned. The signal quality is a measurement that takes into consideration the Received Signal Strength Indication (RSSI) and other network parameters. We considered APs with a measured signal quality above 50 to be neighbors. The available quality scan for the H building was incomplete and only half of the APs were scanned. For this limited set of APs present in the signal quality scan and their location on the building (H building blueprints with APs' location in Appendix A) we could observe that could be considered neighbors of an AP: (i) the APs on the same side of the building and floor; (ii) the APs that are directly above and below that AP. Therefore we extended this condition to all other APs in the network and defined that for one specific AP their neighbors will be all the APs in the same side and floor of the building as well as the APs that are directly above and below it in adjacent floors. Figure C.1 in Appendix C shows the UFF SCIFI network topology and the APs' neighbors using our defined vicinity criteria.

With the established vicinity criteria, we can determine which APs are neighbors and create a neighborhood set list for each AP. Another important input feature of our clustering algorithms is the special AP set. The special AP set comprises a set of APs that show some traffic statistics that differentiate them from the rest of the APs. Special APs show higher traffic statistics and therefore they are usually busier or they present a higher traffic demand than the rest of the APs in the network. In our case, we are only using the number of association as our traffic metric to calculate the user demand. Therefore, for the special AP selection, we defined the following AP statistics to be taken into consideration:

- Month Association Average
- Day Association Average
- Hour Association Average
- Time slot Association Average
- Total Number of Associations
- Maximum Number of Associations

We created an overall rank considering the APs position in each of those statistics using a weighted average. Equation 5.2 shows the weighted average WA_p of a specific AP p where $NT_{j^{th}}^p$ represent the number of times that a specific AP p appears in the j^{th} position in the statistics and n represents the number of APs in the network. Our work has only used 6 traffic statistics but more or distinct ones can be used in other eSCIFI implementation depending on the traffic metric used and network scenario. The special APs are then selected from the best positioned APs in the overall rank. Later on Section 6.1 we will discuss how the special AP set and its size can affect the cluster formation.

$$WA_p = \frac{\sum_{j=1}^n NT_{j^{th}}^p (n + 1 - j)}{\sum_{i=1}^n i} \quad (5.2)$$

Now that we have the neighborhood set and the special AP set, we can describe our cluster formation algorithms. Figure 5.4(a) shows a wireless network example where the APs carry their number of neighbors and the dashed lines between two APs represent that those APs are present in each other neighborhood sets. Consider V_i as the neighborhood set of AP i , C as our cluster set, and C_i as the cluster formed starting from the AP i . In our cSCFIFI clustering algorithm, we first start by selecting the special APs. Special APs have the highest traffic demand rates when compared to the rest of the APs and therefore are a better starting point for our clusters formation. For that reason those APs should not be put together in the same cluster. If special APs are neighbors, putting them together in a same cluster could mean a potential waste in the clustering formation, since they are very likely to always be turned on. For that reason, special APs should not be put together in a cluster but instead be separated from each other in order to guarantee that each special AP will form a cluster.

Figure 5.4(b) shows that each special AP initiates a cluster and it is added to it. When the special APs are added to their clusters, the cSCIFI algorithm also removes those APs from all other APs neighborhood sets and update their number of neighbors. Now our cSCIFI clustering algorithm starts the cluster formation by selecting the cluster where the special AP with the biggest neighborhood set is. Once the special AP s with the highest number of neighbors is chosen, the algorithm steps through all the APs in its V_s neighborhood set and adds the AP h that has the biggest neighborhood set as long as every new AP h added to C_s is in the neighborhood set of all other APs in C_s . We call this the neighboring condition. As long as the AP s has APs on its neighborhood set V_s that APs satisfy the neighboring condition, those APs are added to the cluster C_s and removed from the other APs neighborhood sets, as shown in Figure 5.4(c).

When there are no more APs in the AP s neighborhood set or there are no more APs that satisfies the neighboring condition, the algorithm steps to the next special AP with the biggest neighbor set and continues the cluster formation as shown in Figure 5.4(d). When there are no more special APs, the cSCIFI algorithm steps to the next normal AP with the biggest neighborhood set until there are no more APs left and the cluster set C is finished as shown in Figure 5.4(e).

Algorithm 1 shows the cSCIFI implementation where we can see that the cSCIFI guarantees that every AP will be only in one cluster and that every AP is on the vicinity of all other APs inside its cluster. The neighboring condition (line 4) allows any user in the cluster coverage area to connect to any of the powered on APs, since they are all each other's neighbors.

The cSCIFI+ is a simpler and more aggressive clustering strategy than cSCIFI. Figure 5.5 shows that the cSCIFI+ clustering algorithm works like the cSCIFI, but now the APs added to a certain cluster C_i do not need to cope with the neighboring condition. As we can see in Figure 5.5(c), all neighbors in the AP A neighborhood AP set V_A are added to cluster C_A .

As shown in Figure 5.5(e), cSCIFI+ guarantees that the size of the cluster set C will be the smallest possible. However users from a switched off AP in the cluster can only connect to the AP A that initiated that cluster. Considering the clusters formed with the cSCIFI, users from any AP can connect with other APs in the cluster which might balance the load between the switched on APs. Algorithm 2 shows the cSCIFI+ implementation where we can see that considering the clusters formed with cSCIFI+, only the AP that initiated the cluster formation can assure connection to all users from switched off APs

Algorithm 1 *cSCIFI*

```

1 : function Create_Cluster(Cluster_Head, Cluster_head_list_of_neighbors):
2 :   Cluster_auxiliary_list = [ Cluster_Head ]
3 :   for AP in Cluster_head_list_of_neighbors:
4 :     if AP in neighborhood list of all Cluster_auxiliary_list elements:
5 :       add AP to Cluster_auxiliary_list
6 :       remove AP from neighborhood list of all APs
7 :       remove AP from Regular APs list
8 :   return Cluster_auxiliary_list
9 : function Cluster_Formation(Position_list):
10 :   Cluster_Set = [ ]
11 :   while Special APs in Special APs list do:
12 :     chead = from Special APs with highest number of neighbors select the one in Position
13 :     add Create_Cluster (chead, chead neighborhood list) to Cluster_Set
14 :     remove chead from Special APs list
15 :   while APs in Regular APs list do:
16 :     regular_thead = from APs with highest number of neighbors select the one in Position
17 :     add Create_Cluster(regular_thead, regular_thead neighborhood list) to Cluster_Set
18 :     remove regular_thead from Regular Aps list
19 :   return Cluster_Set
20 : Possible_Sets = [ ]
21 : for all possible ties in cluster heads selection do:
22 :   selection = create a possible and unused position list for cluster head selection ties
23 :   add Cluster_Formation(selection) to Possible_Sets
24 : Cluster_Set_Selected = cluster set with the smallest number of clusters present in Possible_Sets

```

which may cause congestion.

As we can see in Figure 5.4 and Figure 5.5, the cSCIFI and cSCIFI+ greedy algorithms alone can not guarantee that the best cluster set is formed in cases where there is a tie between APs. A solution would be creating all cluster possibilities, choosing each one of the tied APs as the first choice. After creating all possible sets C , we would select the one that has the minimum number of clusters. Those multiple cluster sets

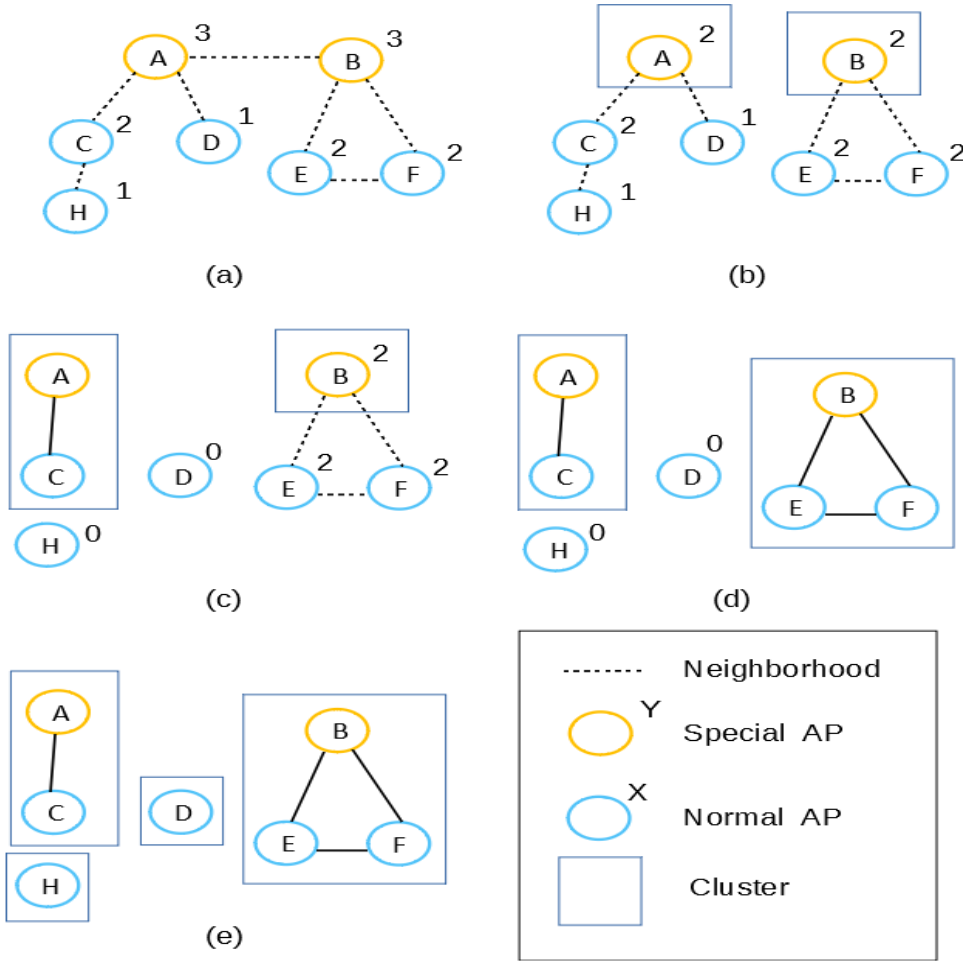


Figure 5.4: cSCIFI cluster formation algorithm

creation can cause an exponential growth in the execution time. Trying to minimize those problems, we simplified the cSCIFI and cSCIFI+ selection in cases of ties. The cSCIFI and cSCIFI+ will only create multiple cluster sets when there are ties between APs that will be selected to initiate a cluster formation. This selection criteria will guarantee that only different clusters initiation will be taken into relevance and not all possible cluster internal formations, which will minimize the possible solution set.

In the cSCIFI algorithm, we also added another selection criterion for cases where there are ties between APs to be added to cluster C_i where an AP i has already initiated it. In those cases, the AP j with the highest number of neighbors in the AP i neighborhood set V_i is selected. APs with the same number of neighbors in their sets can generate different clusters, since some of their neighbors might not be in the AP i neighborhood set V_i . Therefore in cases of ties, it is the best option to select the AP j that has the biggest number of matching neighbors to the APs in the neighborhood set V_i . This change on the internal cluster formation process guarantees that the next APs to be added will

Algorithm 2 *cSCIFI+*

```

1 : function Create_Cluster(Cluster_Head, Cluster_head_list_of_neighbors):
2 :   Cluster_auxiliary_list = [ Cluster_Head ]
3 :   for AP in Cluster_head_list_of_neighbors:
4 :     add AP to Cluster_auxiliary_list
5 :     remove AP from neighborhood list of all APs
6 :     remove AP from Regular APs list
7 :   return Cluster_auxiliary_list

8 : function Cluster_Formation(Position_list):
9 :   Cluster_Set = [ ]
10:  while Special APs in Special APs list do:
11:    chead = from Special APs with highest number of neighbors select the one in Position
12:    add Create_Cluster (chead, chead neighborhood list) to Cluster_Set
13:    remove chead from Special APs list
14:  while APs in Regular APs list do:
15:    regular_thead = from APs with highest number of neighbors select the one in Position
16:    add Create_Cluster(regular_thead, regular_thead neighborhood list) to Cluster_Set
17:    remove regular_thead from Regular Aps list
18:  return Cluster_Set

19: Possible_Sets = [ ]
20: for all possible ties in cluster heads selection do:
21:   selection = create a possible and unused position list for cluster head selection ties
22:   add Cluster_Formation(selection) to Possible_Sets
23: Cluster_Set_Selected = cluster set with the smallest number of clusters present in Possible_Sets

```

be the one that will contribute to a bigger cluster size.

Those characteristics cited previously minimizes the execution time and guarantee that a possible cluster set C will be selected independent of their appearances on the clusters neighborhood list. This is an important advantage to our clustering algorithms when compared to the green clustering algorithm proposed by Jardosh et al. [29] since we do not need to worry about the APs order of appearance in neighborhood list construction process. Figure 5.6 compares the cSCIFI and cSCIFI+ cluster set formed after those

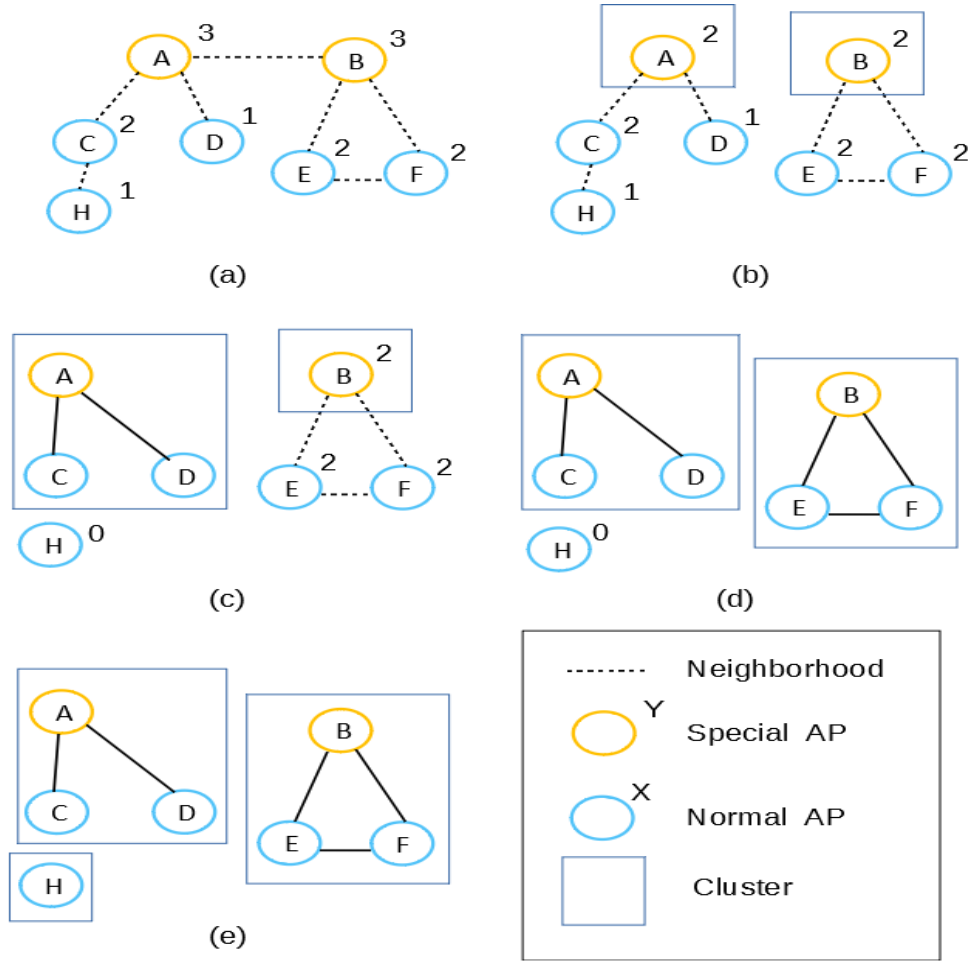


Figure 5.5: cSCIFI+ cluster formation algorithm

selection criterion have been implemented. From Figure 5.6 it is possible to notice that those changes applied to the cSCIFI clustering algorithm had generated a better cluster formation than the previous one made by the algorithm where those selection criterion were not implemented. Figure 5.6 also shows that the cSCIFI had achieved a cluster set with the same number of clusters of the cSCIFI+ for this example scenario. In this case, cSCIFI might be a better solution than the cSCIFI+, because the cSCIFI cluster is better balanced and may provide a better load balance for the reasons discussed previously.

The last characteristic of our clustering algorithms is the cluster head election. The cluster head is the AP that will be always switched on and will be responsible for guaranteeing the cluster coverage area. In cSCIFI+, the cluster head will always be the one that initiated the cluster formation. This AP is the only AP that can be the cluster head, since this is the only AP that has a guaranteed neighboring condition to all other APs in the cluster. On the other hand, the election of the clusters head in the cSCIFI algorithm can be more sophisticated since all APs in the clusters obey the neighboring condition.

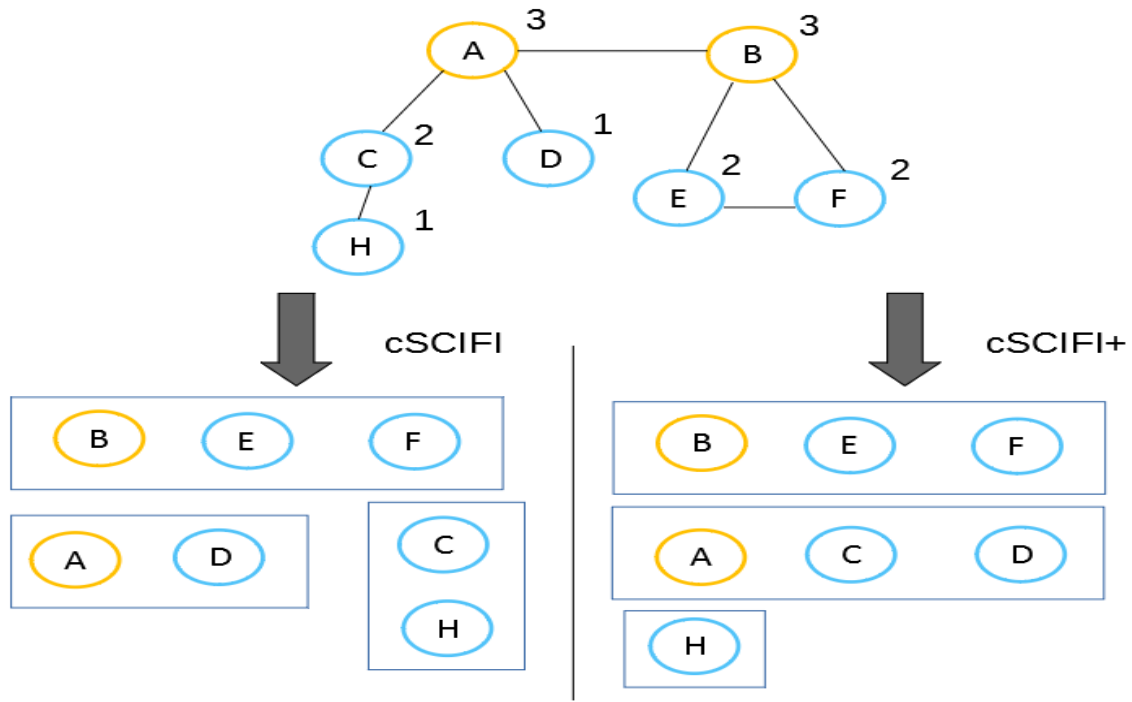


Figure 5.6: cSCIFI and cSCIFI+ cluster formation comparison

cSCIFI also has two types of clusters: clusters that have a special AP and clusters that do not have one. The cluster head in clusters with a special AP will be the special AP, once its special condition already guarantees that they are the AP with the highest traffic demand from all the other APs and therefore are more likely to be switched on due to its users' demand. In clusters without a special AP, the cluster head will change throughout the day. In those clusters, the average association of each AP is calculated for the night (0AM - 7AM), morning (7AM - 1PM) and afternoon/evening (1PM - 11PM 59') periods. The AP with the highest night average will be selected to be the cluster head for the night period and so on.

5.1.3.2 Energy State Decision Algorithm

The energy state decision algorithm is responsible for providing the energy scheduling of all APs for a date. The energy state decision algorithm runs once a day and it uses the traffic demand estimated by the hybrid machine learning model (the user association number in our SCIFI network scenario) to calculate the cluster demand for specific moments of a date and then decide which APs in a cluster can be switched off. The energy state decision algorithm is the last step on the eSCIFI energy saving mechanism and it is responsible for actively deciding which APs will be switched on or off and to provide the energy scheduling to the SCIFI controller. The SCIFI controller, based on this energy scheduling, will then

control the AP wireless interface switching on and off for the specified periods.

Our energy state decision algorithm uses the RoD policy proposed in the work of Dalmaso et al. [14]. However our energy state decision algorithm works using machine learning occupancy estimations instead of real traffic data and therefore presents some modification in the RoD policy design. This RoD policy has two main components: the time window and the double threshold criteria. The time window defines how long it will take before the algorithm reconfigure the AP's energy state. The time window size tw informs on which frequency the network will be reconfigured and also the demand estimation resolution. A small time window will allow the energy state decision algorithm to perceive short bursts in the traffic demand variations. A large time window on the other hand will only perceive the average traffic where instant or momentarily bursts in the traffic demand will fade. At first a smaller time window size seems always the best choice, however a smaller time window size means more rounds of energy state decisions will have to be made by the algorithm and that the controller will have to reconfigure the network more frequently. Related work [18, 46, 14, 16, 37] state that small time window sizes are not necessary. In fact, depending on the network traffic profile, those changes in the traffic demand can take hours to happen. Therefore the selection of the time window size is a parameter that needs to be decided based on the network scenario. Later on Section 6.2, we will deeply discuss the selection of the time window size.

The main concept behind our energy saving strategy is moving the traffic demand from switched off APs to the cluster head AP or other switched on APs in the cluster that can handle them. On the work of Damalso et al. [14], the APs in a cluster can be switched off based on the actual traffic demand (real time traffic data) at the beginning of each time window. However the eSCIFI mechanism uses machine learning models to estimate demand. Therefore in our energy decision algorithm, the decisions made for each time window will take into consideration the demand estimated for its whole duration and not just the demand at the beginning of the time window.

All APs in the network have the same maximum user threshold T_{max} for a time window. This maximum user threshold T_{max} defines how much traffic (or how much associations in our case) the APs can handle for the duration of the time window. The cluster head of every cluster will always be switched on guaranteeing a traffic capacity of T_{max} for the cluster. In our energy state decision algorithm, the double threshold criteria defines which APs in a cluster can be switched off based on the traffic demand estimated by the machine learning hybrid model for the assessed time window. However this energy

state decision algorithm varies depending whether the cSCIFI or the cSCIFI+ algorithm is used.

In the cSCIFI algorithm, all APs in a cluster are neighbors between themselves. Therefore in the cSCIFI case, the double threshold criteria defines that APs with estimated traffic demand below a minimum threshold T_{min} for the whole time window are switched off as long as the available traffic capacity provided by all APs that are switched on can handle their estimated traffic. Considering DM_i as the traffic demand of AP i for a time window, d as the number of switched on APs, o as the number of switched off APs, $\sum_{a=1}^d DM_a$ as the traffic demand of all d switched on APs and $\sum_{a=1}^o DM_a$ as the traffic demand of all o switched off APs, we can define how our energy state decision algorithm decides if an AP i will be switched off based on the double threshold criteria if the cSCIFI algorithm is used. Equation 5.3 shows the double threshold criteria, where the first criterion defines if the traffic demand is too low for the AP i to be switched on and the second criterion defines if the cluster switched on APs can handle the AP i traffic. If there are more d switched on APs in the cluster, the cluster maximum traffic capacity CCA increases to $T_{max}(d + 1)$ because cSCIFI guarantees that all APs inside a cluster can provide connection to any mobile station trying to connect to any AP in the cluster.

$$\begin{cases} DM_i < T_{min} \\ CCA - (\sum_{a=1}^d DM_a + \sum_{b=1}^o DM_b) \geq DM_i, \quad \text{where } CCA = T_{max}(d + 1) \end{cases} \quad (5.3)$$

On the other hand, in the cSCIFI+ algorithm, all APs in a cluster are neighbors only to the cluster head. Therefore in the cSCIFI+ case, APs with estimated traffic demand below a minimum threshold T_{min} for the whole time window are switched off as long as the available traffic capacity provided by the cluster head can handle their estimated traffic. Equation 5.4 shows the double threshold criteria if the cSCIFI+ algorithm is used, where the first criterion defines if the traffic demand is too low for the AP i to be switched on and the second criterion defines if the cluster head can handle the AP i traffic. In the cSCIFI+ case, the cluster maximum traffic capacity CCA is fixed to T_{max} because the cSCIFI+ guarantees that only the cluster head can provide connection to any mobile station trying to connect to any AP in the cluster except the AP itself.

$$\begin{cases} DM_i < T_{min} \\ CCA - (\sum_{a=1}^d DM_a + \sum_{b=1}^o DM_b) \geq DM_i, \quad \text{where } CCA = T_{max} \end{cases} \quad (5.4)$$

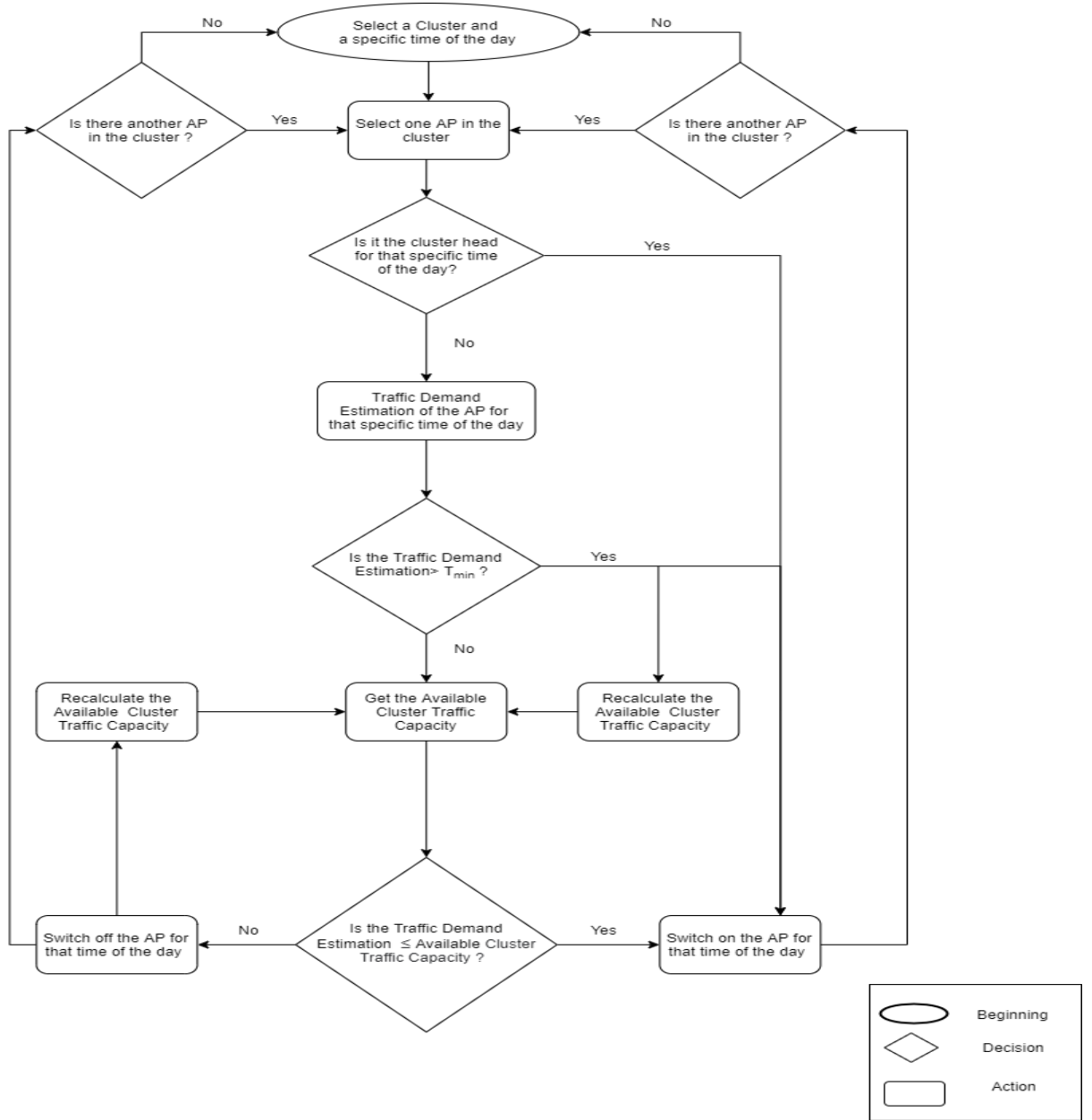


Figure 5.7: eSCIFI energy state decision algorithm flowchart

Figure 5.7 shows our energy state decision algorithm flowchart. At the beginning of a time window, our hybrid model provides the traffic demands for all APs in the clusters for the whole time window. Each cluster is individually evaluated and the decisions of switching off or switching on APs within the cluster are taken. Then the algorithm iterates through all APs in a cluster. The cluster head and APs that have a traffic demand higher than T_{min} are or remain switched on. For those APs that obey the first threshold criterion, our energy state decision algorithm keeps evaluating if their summed demand with the demand of the APs that were already schedule to be switched off is lower than the cluster maximum capacity CAA . If it is lower than the CAA , the AP is scheduled to be switched off and, if it is not, the AP is switched on and the maximum capacity is recalculated.

The eSCIFI mechanism has several parameters that must be configured and that may depend on the network usage profile such as the selection of the time window size and T_{min} value. On the next chapter we evaluate how the different components in the eSCIFI architecture affects the mechanism energy saving capacity and the network coverage to its users. We also compare the eSCIFI to other energy saving mechanisms present in our related work that are applicable in our evaluation scenario.

Chapter 6

eSCIFI Evaluation

To evaluate the eSCIFI impacts on the network performance, we performed trace-driven simulations using the real association trace data collected from the UFF SCIFI network. Using real association traces, it is possible to evaluate if eSCIFI can cope with users demand while saving energy. To perform our simulations, we are going to use the association data collected for one week on September 2018 from the H building at UFF. The week used in our collected data is formed by a weekend (September 1 and 2, 2018) and 5 weekdays (September 24,25,26,27,28, 2018). The weekend used are apart from the weekdays dates because there was not complete association history for the weekend before or after those weekdays. This might have happened for several reasons such as energy outages or network failures for example. However the weekend (September 1 and 2, 2018) contains the association data for all APs in the H building and therefore we will use them to represent Saturday and Sunday in our trace-driven simulations. We are also going to use the Brazil's Independence day public holiday (September 7) to compare and evaluate how eSCIFI impacts the network on holidays.

The work of [25] presents a mathematical formula, indicated in Equation 6.1, that allows us to determine the energy saving factor ESF achieved with the AP wireless network interface shut down during periods of time. The formula gives the percentage of energy that could be saved by shutting down the AP wireless network interface compared to the total energy that would be consumed if the AP wireless network interface stayed working the whole time.

Terms P_{ext_on} and P_{ext_off} of Equation 6.1 represent the measured power values in Watts, in the AP external power source, for the cases where the wireless network interface is switched on and off respectively. Terms t_{on} and t_{total} represent the period of time that the AP stayed with its wireless interface switched on and total period of time that is taken

into analysis respectively. The result given by Equation 6.1 gives the percentage of energy that could have been saved from the total of the energy used, by switching off the wireless interfaces of the AP during the idle time slots. This formula can be easily extended to also provide the network's energy saving factor. To do so, the terms t_{on} and t_{total} must change in order to represent the sum of time that all APs on the network stayed with its wireless interface switched on and total time that is taken into analysis multiplied by the number of APs in the network respectively.

$$ESF = \frac{P_{ext_on} - P_{ext_off}}{P_{ext_on}} \left(1 - \frac{t_{on}}{t_{total}}\right) \quad (6.1)$$

From Equation 6.1 it is possible to notice that the energy saving factor (ESF) reaches its maximum power saving factor value, ESF_{max} , when $t_{on} = t_{total}$. This condition represents the scenario where the wireless interface of all APs in the network are switched off during the whole time. However it is also possible to notice that depending on the scenario and switching off scheme the ESF_{max} can assume several values. Therefore the normalized energy saving factor, \overline{ESF} given by Equation 6.2 can better indicate the performance of the mechanism in different scenarios. The normalized energy saving factor \overline{ESF} is limited between 0% and 100% and represents the percentage of the maximum energy saving factor that could be saved.

$$\overline{ESF}(\%) = \frac{ESF(\%)}{ESF_{max}(\%)} \quad (6.2)$$

The work of [18] defines the coverage ratio loss CR formula, indicated in Equation 6.3. The coverage ratio loss is the number of uncovered clients U_l by the energy saving mechanism over the total clients in the network U within a certain period of time. The coverage ratio loss gives the percentage of clients that could not successfully access the network under the evaluated period.

$$CR(\%) = \left(\frac{U_l}{U} * 100\right) \quad (6.3)$$

The analysis in this section will evaluate the normalized energy saving factor (Equation 6.2) and the coverage ratio (Equation 6.3) to compare how the eSCIFI mechanism impacts on the network performance. To calculate the coverage ratio, we must know the parameter T_{max} that indicates the maximum number of association an AP might support in a time slot. Based on Table D.1, we defined $T_{max} = 300$ which is roughly the maximum

number of APs associated in a time slot registered plus 10%. In our experimental scenario only the wireless network interface will be switched off. Table 6.1 shows the consumed power measured for the AP model present in the UFF SCIFI network when the wireless interface is switched on and off (P_{ext_on} and P_{ext_off}). Table 6.1 also shows what would be the maximum power saving factor, ESF_{max} , which represents the power saving factor percentage if the wireless interface of all APs were switched off the whole time. Therefore in our evaluation scenario the maximum energy saving factor percentage that could be reached by switching off the wireless interface of the entire network during the whole evaluation period is 23,93%. Those information are required by the normalized energy saving factor calculations.

Table 6.1: AP's consumed power and maximum power saving factor percentage

P_{ext_on} (W)	P_{ext_off} (W)	ESF_{max} (%)
1,111	0,845	23,93

We are going to evaluate how several components from the eSCIFI architecture impact on the network performance. eSCIFI using the cSCIFI and the cSCIFI+ clustering algorithms will also be compared with other mechanisms proposed in the literature. The eSCIFI energy saving mechanisms will be compared with SEAR, ACE and ECMA mechanisms proposed by Jardosh et al. [29], Fang et al. [18] and Silva et al. [46] respectively. The SEAR mechanism uses the green clustering algorithms and a single threshold where only the T_{min} parameter is used as the RoD strategy. In the SEAR mechanism, the network APs are grouped into clusters, the cluster head is always switched on and the other APs in the clusters remain switched off as long as their traffic demand is lower than the T_{min} . The ACE mechanism uses an inactivity time window based on machine learning occupancy detection results as its RoD strategy and does not have a coverage guarantee. APs that remain unused by a whole time window size are switched off the whole time window duration period. The ECMA mechanism uses the SEAR mechanism for night hours (between 0AM - 6:59AM) and keeps the whole network switched on the rest of the day. The Baseline mechanism where all the APs in the network remain switched on between 7AM - 11:59PM and switched off between 0AM - 6:59AM is also used for comparison.

We will evaluate how the size of the special AP set, the time window size and the minimum threshold value affect the network performance. We will also compare the eSCIFI mechanism performance on regular weekday with its performance on a public holiday. We are going to compare eSCIFI using the demand estimation hybrid model and using the

real demand to evaluate how the hybrid model inaccuracies may affect the mechanism performance. After that, we will compare the SEAR green clustering, eSCIFI cSCIFI and cSCIFI+ clustering algorithms performances using different neighborhood lists. Our last analysis will compare how day periods may affect the mechanism performance. Our trace-driven simulations script is developed in Python.

6.1 Number of Special APs Analysis

In Section 5.1.3.1, we have defined how the special AP set is formed and how it is used in our cluster formation algorithm. Table D.1 shows the statistics taken into consideration for our experimental scenario as defined in Section 5.1.3.1. From these statistics, it is possible to calculate the overall rank as defined in Equation 5.2 and extract the special AP set. Table E.1 in Appendix E shows the overall rank for our UFF SCIFI evaluation scenario. A special AP set of size s will select the first s APs from the overall rank. We will select different special AP set sizes ranging from 3 ($\sim 10\%$) to 9 ($\sim 30\%$). We will compare the performance of these special AP sets with an empty special AP set (size 0). The idea is to evaluate how the special AP set size will affect the network cluster formation and to verify the validity of using a special AP set in our scenario.

Table F.1 in Appendix F shows how the special AP set size using the cSCIFI affects the number of clusters present in the network's cluster set. However a bigger special AP set does not always create a bigger number of clusters and sometimes it does not even change the cluster set formed. There are two main explanations for that. One explanation is that imposing them a special AP characteristic does not change the cluster set formed because those APs have already been selected by the algorithm to form clusters. Other explanation is that selecting more APs to be in the special AP set creates clusters of different sizes but in such way that the number of cluster does not change.

Table F.2 in Appendix F shows how the special AP set size using the cSCIFI+ affects the number of clusters present in the network's cluster set. We can see that the more aggressive clustering cSCIFI+ nature generates the same behavior already discussed for cSCIFI, but with smaller cluster set sizes. However in the cSCIFI+ algorithm, we can see that the absence of special APs has actually lead to an increase in the number of clusters when compared to the cluster set formed using a special AP set of size 3. Therefore different special AP sizes might generate different clusters and/or different number of clusters. At first we see that smaller special AP set sizes generate smaller clusters which

can be an advantage. However a smaller cluster set may not be enough to guarantee a better performance in our energy saving mechanism proposal.

Next we evaluate the normalized energy saving factor of those different clusters set formed using different special AP set sizes. To do so, we selected a fixed combination of T_{min} and tw values that guarantees the highest normalized energy saving factor \overline{ESF} and no coverage ratio CR loss for all special AP set sizes. Therefore the results showed in Figure 6.1 show the normalized energy saving factor of the eSCIFI mechanism where $T_{min} = 72$ and $tw = 120$ (120 minutes or 12 time slots) using the cSCIFI and the cSCIFI+ cluster set formed with different special AP set sizes. As we can see from Figure 6.1, the special AP set sizes impacts on the normalized energy saving factor. Smaller special AP set sizes generate bigger energy savings. This energy saving differences are related to the cluster set size. A smaller number of clusters may generate bigger savings since less APs will be required to be switched on during idle periods. However Figure 6.1 shows that the number of clusters is not enough to define which cluster set configuration will lead to bigger energy saving. For the cSCIFI algorithm for example, special AP sets of size 7 and 8 form cluster sets of equal size, but they present different energy saving percentages. The cSCIFI+ cluster set formed without any AP in the special AP set is bigger than the cluster set formed with 3 APs in the special set, but it still had bigger energy savings.

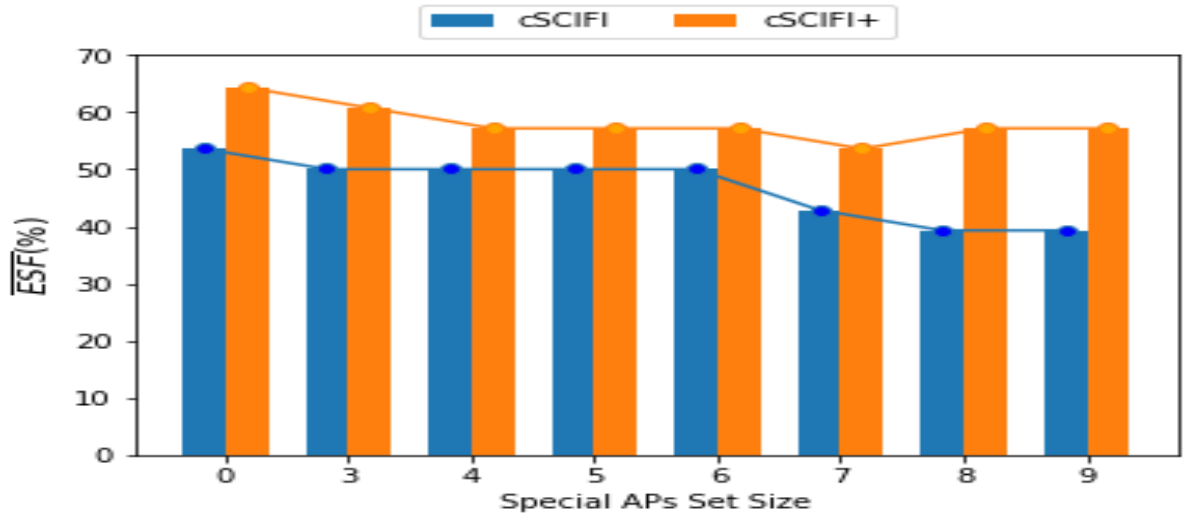


Figure 6.1: Normalized Energy saving factor for different special APs set sizes

The number of clusters and the APs that are gathered inside them are equally important to determine which configuration will generate the biggest energy saving. Another important thing is that the cluster formed by the cSCIFI and cSCIFI+ algorithms without using the special APs criteria (special AP of size 0) got better results. The special AP set had not improved the cluster formation process in our scenario. The results shows

that using the special APs criteria in our cluster formation algorithm may not generate better clusters which may lead to energy saving losses. The clusters formation is intimately related with the neighborhood of the APs. Special APs are on the other hand selected only based in their statistics. This may lead to the selection of APs with smaller neighborhoods which generate smaller clusters and consequentially changes the clusters set size or even the distribution of APs inside clusters. That happened on our scenario where APs 21, 519 and 419 were selected as special APs but have fewer neighbors than other APs in the network such as APs 224 and 288. Therefore the selection of special APs without taking the neighboring condition into account may create a network cluster formation that is not the best one.

The special AP set has clear connection with the cluster formation process since heavily used APs are not good to be put together with each other since they may always be switched on. However the special APs as defined in this work was not enough to guarantee better cluster formation and energy savings. Therefore further analysis would be required to improve the special APs selection criteria to improve its results when compared to the clustering algorithm that does not differentiate APs.

6.2 Time Window Size Analysis

In Section 5.1.3.2 we have seen that the time window tw defines how long it takes before reconfiguring the network APs energy state. A bigger time window is desired since it will minimize the number of times the controller will need to change the APs working status, which will minimize the controller tasks over a day. On the other side, a bigger time window may not notice small traffic demand bursts, which may lead to network coverage losses during these bursts due to unnoticed behaviors. Therefore we need to evaluate how the eSCIFI time window size may affect the network energy saving and coverage loss. We tested the eSCIFI mechanisms using 5 different time window values (10 minutes, 30 minutes, 1 hour, 1:30 hours and 2 hours). Those time window values were selected based on our time slots size and correspond to 1, 3, 6, 9 and 12 time slots respectively. Those time window were selected based on the lecture duration time at UFF that usually takes 2 hours. The real and predicted association values for time windows bigger than one time slot (10 minutes) is the sum of the devices connected during the corresponding amount of time slots. We evaluated the eSCIFI mechanisms using the cSCIFI and cSCIFI+ algorithms without a special AP set (size 0) and $T_{min} = 72$ with different time windows to evaluate the normalized energy saving factor and coverage loss. Those fixed parameters

were used because they delivered the best normalized energy saving factor percentage and coverage ratio loss to all possible time windows. We will also evaluate the time window size effect in the SEAR, ACE, ECMA and Baseline mechanisms.

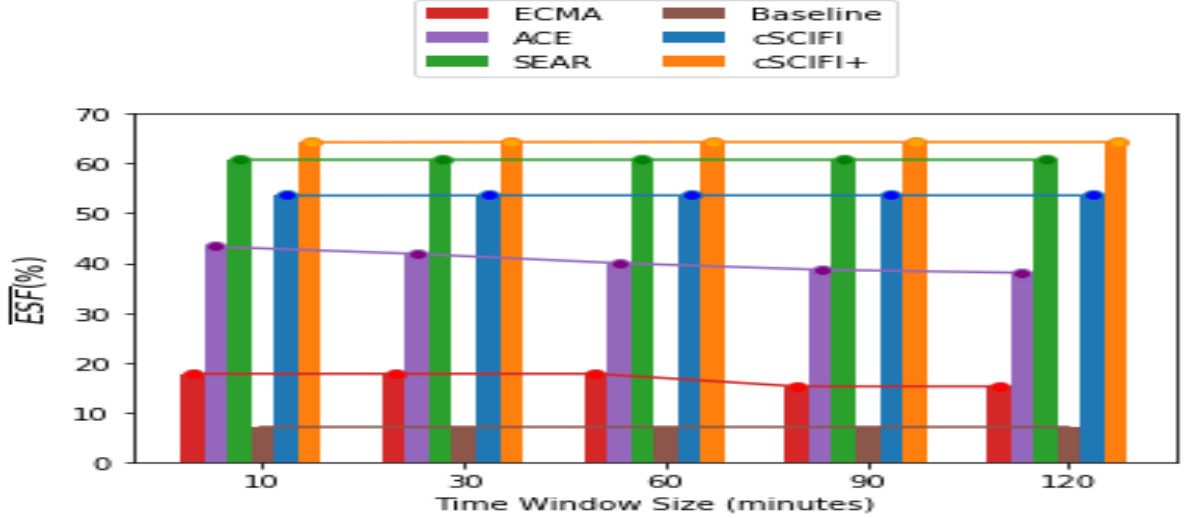


Figure 6.2: Normalized Energy saving factor for different time window sizes

As we can see in Figure 6.2, the selected time window sizes have not affected the normalized energy saving factor \overline{ESF} for the SEAR mechanism and the eSCIFI mechanisms using the cSCIFI and the CSCIFI+ clustering algorithms. Only ECMA and ACE mechanisms had their normalized energy saving factor negatively affected by the time window size. The baseline estimator does not depend on the time window (its scheduling presents fixed switching on/off periods) and therefore we can see that its normalized energy saving factor does not change. This result means that for our evaluation scenario it is possible to use a 2-hour time window resolution without affecting the normalized energy saving factor for our eSCIFI mechanism. This would allow the eSCIFI mechanism to compute less energy state changes in the APs and consequently less tasks to be executed by the network controller.

Figure 6.3 shows how the different time window sizes affects the coverage ratio. As we can see only the Baseline and ACE mechanism presented coverage ratio losses in this evaluation scenario. The baseline estimator has a fixed coverage loss that does not depend on the time window size. The Baseline loss occurs due to unattended users in the night hours where all APs are switched off. However the ACE mechanism shows a small decrease in the coverage loss as the time window grows. That result was expected because the ACE mechanism uses the time window size as an inactivity criteria to switch off APs and therefore a large time window would require a longer period of inactivity which would be harder to achieve and consequently would lower the chances of mistakenly switching

off APs that should not.

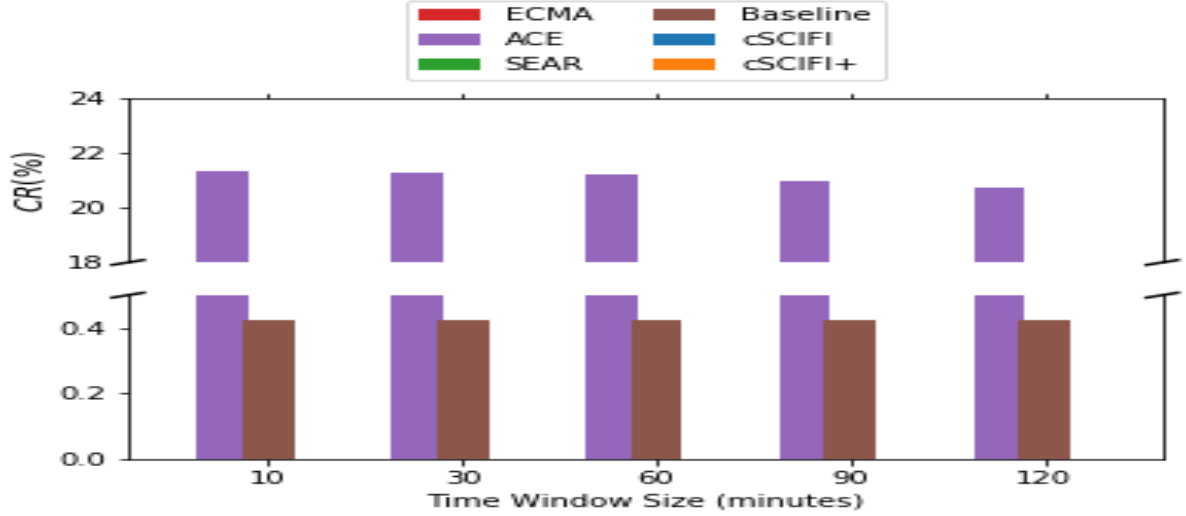


Figure 6.3: Coverage loss for different time window sizes

6.3 Minimum Threshold Analysis

The last parameter on our eSCIFI mechanism that needs to be evaluated is the T_{min} value selection. The T_{min} defines the minimum number of associations that an AP must have during the time window duration to be switched on. If the number of associations is below T_{min} , the AP will be evaluated to be switched off by the energy state decision algorithm. In this section we evaluate how the T_{min} value affects the normalized energy saving factor and the coverage ratio. To do so we varied the value assumed by T_{min} during one time slot including all multiples of 9 ranging from 9 to 90. Therefore the T_{min} value will be proportional to the time window size used. So if the time window has a size w of time slots, the T_{min} values assumed will be $w \times T_{min}$. We fixed the special AP set size to 0 (no special APs selected) and the time window size to 12 time slots (2 hours or 120 minutes).

Figure 6.4 shows the normalized energy saving factor achieved by the eSCIFI mechanism using the cSCIFI and cSCIFI+ clustering algorithms, SEAR, ACE and ECMA mechanisms. As it can be seen in Figure 6.4, SEAR and eSCIFI mechanism using the cSCIFI+ clustering algorithm got the best energy saving percentages on our evaluation scenario. The eSCIFI mechanism using the cSCIFI cluster formation had a smaller energy saving percentage because it has a different cluster set that is bigger than the ones formed by the SEAR and eSCIFI mechanism using the cSCIFI+ clustering algorithm. From Figure 6.4 we can also see that the normalized energy factor \overline{ESF} grows as T_{min} value grows until it reaches $T_{min} = 54$, after that the energy factor stays the same for all

mechanisms. This result was expected and it is the same result achieved by Dalmasso et al. [14] in his work. This asymptotic characteristic in the normalized energy saving factor curve happens because, for higher values of T_{min} than 54, the cluster maximum capacity CAA threshold is reached requiring those same APs to be turned on anyway.

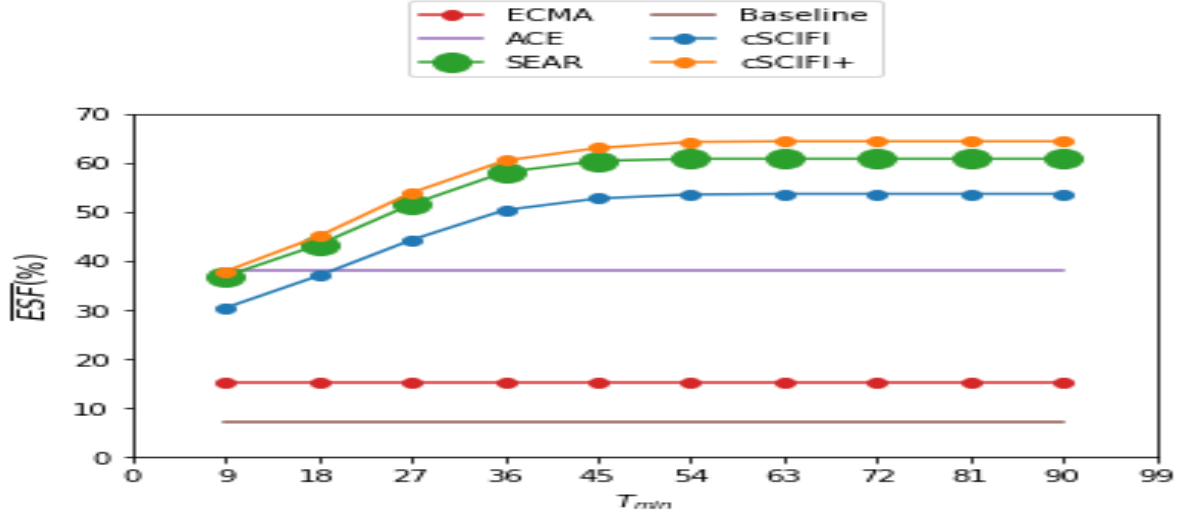


Figure 6.4: Normalized Energy saving factor for different T_{min} values

Higher T_{min} values mean that the APs will require a higher number of associations in a time window to be switched on according to the first criteria which means it will be harder to them to be switched on. However those APs will have their demand transferred to the cluster head (or other switched on AP in the case where the cSCIFI algorithm has been used). That will mean that the cluster maximum capacity CAA threshold will be reached sooner and the APs will have to be turned on anyway. Therefore the normalized energy saving factor is limited and there is a T_{min} value that reaches it. Increasing T_{min} after its optimum value will not change the normalized energy saving factor. The SEAR mechanism does not have this second threshold criteria and still it has its normalized energy saving factor capped. The possible explanations behind that may be that after $T_{min} = 54$, the SEAR mechanism already reaches the minimum required APs to guarantee coverage (only the cluster heads may be switched on) or the switched on APs after that value present traffic demands much higher than the maximum value of $T_{min} = 90$. Figure 6.4 shows that the ECMA mechanism has a steady normalized energy saving factor that does not depend on the T_{min} value. This might happen because the ECMA mechanism apply the SEAR mechanism in night hours (between 0AM - 6:59AM) and keeps the whole network switched on the rest of the day. The network has very little traffic demands in night hours, therefore few APs are required to be turned on or will have enough traffic to trigger the cluster maximum capacity threshold. That way, the ECMA

already reaches its highest energy saving factor with a $T_{min} = 9$. The ACE and Baseline mechanism do not present a T_{min} parameter for energy state decision and therefore their normalized energy saving factors do not change according to it.

Now we evaluate how different T_{min} values affect the coverage ratio. As we can see in Figure 6.5, the eSCIFI mechanism using both clustering algorithms (cSCIFI and cSCIFI+), SEAR and ECMA strategy had no coverage ratio loss at all for any value of T_{min} . This results showed that none of the mechanisms had overpassed the maximum cluster capacity at any moment. Only the Baseline and ACE mechanism present coverage losses. However as we have already mentioned previously, those mechanisms do not change their energy state decisions based on a minimum threshold T_{min} parameter. Therefore their coverage ratio results are the same showed in Figure 6.3 where the time window size is $tw = 120$ minutes.

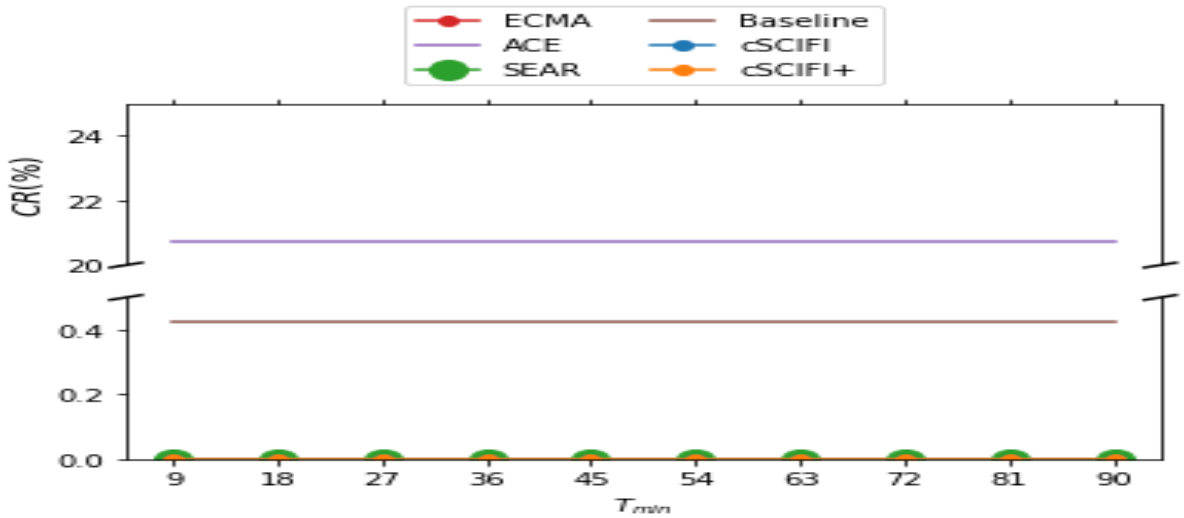


Figure 6.5: Coverage ratio for different T_{min} values

As we have seen there is an optimum T_{min} value that reaches the maximum energy factor. However this T_{min} value is directly related to the time window size tw . Figure 6.6 shows how the optimum T_{min} value might change depending on the time window size tw for the SEAR mechanism and the eSCIFI using both clustering algorithms. As it can be seen in Figure 6.6, bigger time windows require smaller T_{min} values to reach the highest normalized energy saving factor. This might happened because on bigger time window resolutions the traffic demand is averaged by a longer period of time and smaller traffic bursts might disappear into the average demand. Therefore bigger time windows require smaller T_{min} to reach its highest energy saving value.

The results in this section showed that there is an optimum T_{min} value that maximizes

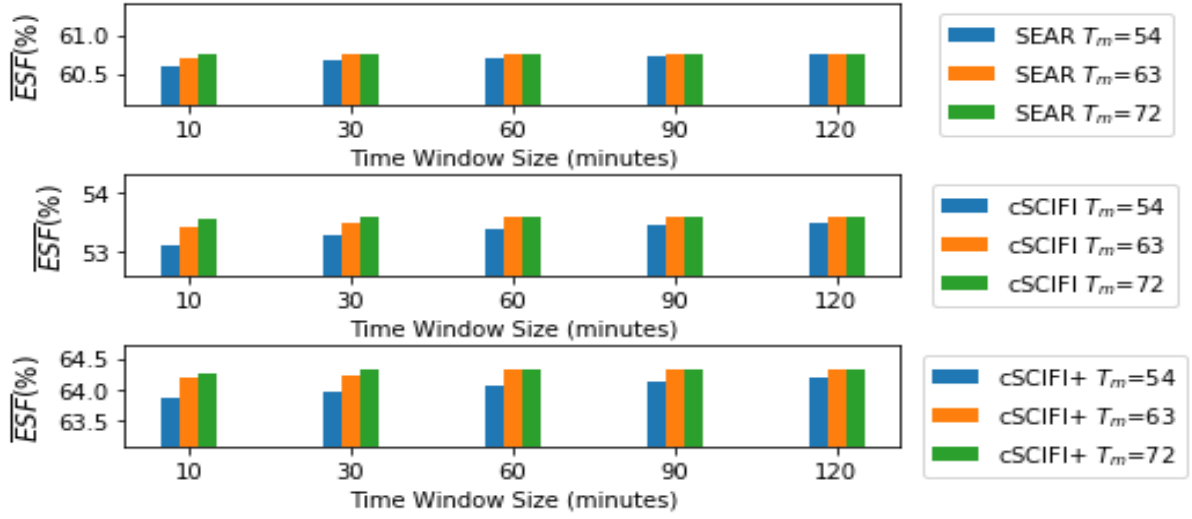


Figure 6.6: Normalized Energy saving factor for different T_{min} and tw values

the normalized energy saving factor. It also showed that this optimum T_{min} value changes according to the time window size. For our experimental scenario the eSCIFI mechanism using both clustering algorithms, a special APs set of size 0 and $tw = 120$ minutes reached this optimum value when $T_{min} = 54$. The results showed that a bigger time window can lower the optimum value. This result is positive because smaller T_{min} will trigger the mechanism to switch on more APs when necessary and that can counterbalance the large time window size disadvantage of not being sensible to small traffic bursts. That way we guarantee that the eSCIFI mechanism in our scenario can switch on more APs to a smaller traffic demand (be more sensible to smaller traffic demands) without affecting the energy saving.

6.4 Weekday Versus Holiday Analysis

The eSCIFI uses machine learning prediction models to estimate traffic demands. In our scenario the hybrid model uses holiday input feature that distinguishes normal weekdays from public holidays and university student holidays. The hybrid model uses this feature to differentiate the network demand variation that happens between regular day and holidays as we have seen in Section 3.2. Here we will evaluate if our eSCIFI mechanism using the hybrid model can better cope with the holiday demand than the SEAR, ACE and ECMA mechanisms. We compare Brazil's Independence Day public holiday (Friday, September 7, 2018) and the Friday used in our regular week. We compared the mechanism using the parameters that gave the best normalized energy saving factor and smallest coverage ratio loss (without a special AP set (size 0), $T_{min} = 54$ and $tw = 120$ minutes).

Figure 6.7 shows the normalized energy saving factor achieved by the different mechanisms. As we can see, the eSCIFI mechanism using both clustering algorithms and the SEAR mechanism kept the normalized energy saving factor stable. The eSCIFI mechanism using the cSCIFI+ clustering algorithm has the biggest normalized energy saving factor for holiday and weekdays. The Baseline and ECMA mechanism also remain with their normalized energy saving factor unchanged. In the Baseline case, this happens because the Baseline decision is only based on time schedules and not in traffic demand estimations and therefore is unaffected by it. In the ECMA case, this result happened because the traffic demand for our holiday or weekday remains unchanged, which did not change the SEAR APs switching on/off schedule during night hours. Only the ACE mechanism had a reduction on the normalized energy saving factor in our holiday evaluation.

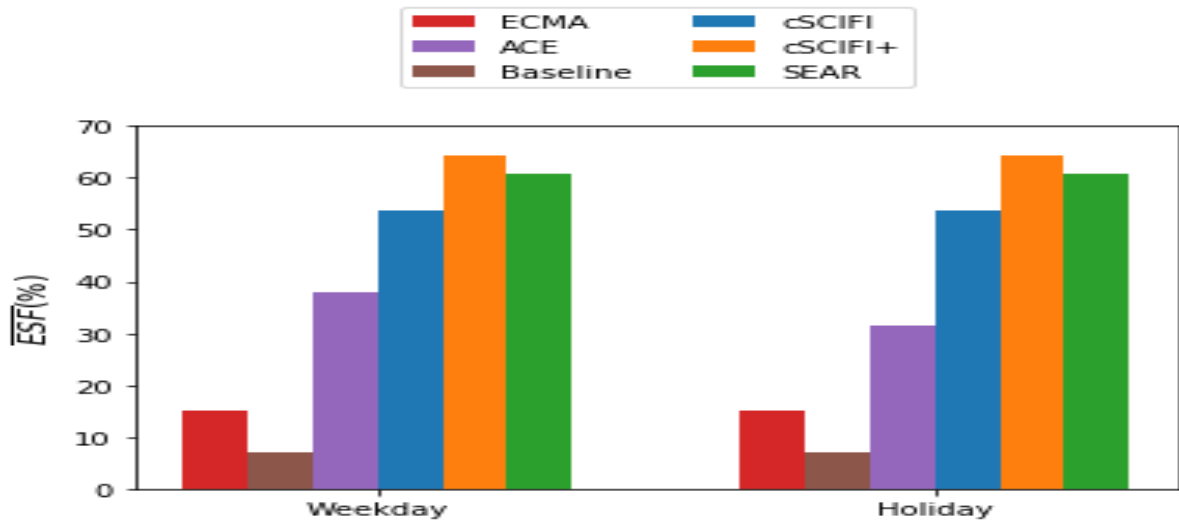


Figure 6.7: Normalized Energy saving factor comparison between a holiday and a weekday

As we have seen in Figure 5.3, the demand on that holiday (Friday, September 7, 2018) was much smaller than the demand presented for the regular weekday (Friday, September 28, 2018). Figure 5.3 also shows that the holiday demand predicted by the hybrid model is much bigger than the real one, differently from the regular Friday where the hybrid model prediction was very close to the real traffic. Those results would first suggest that the normalized energy saving factor achieved by the eSCIFI and SEAR mechanisms for the public holiday should have been smaller as it happened with the ACE mechanism case. However Figure 5.3 shows that the hybrid model wrong estimations have not even reached 200 associated devices for the whole network in any moment of the day on September 7. Figure 5.3 also shows that the regular Friday has not even reached 500 associated devices for the whole network on September 28. Those association values are

very low considering that $T_{max} = 300$. Therefore we can presume that, for the evaluated regular and holiday Friday, the network is working with the minimum set of APs switched on (only the cluster heads) and that is the reason why the SEAR and eSCIFI mechanism using both clustering algorithms have their normalized energy saving factor unchanged by it. In fact, we analyzed how the real data would affect the normalized energy saving factor results in that analysis for the mechanisms and it showed that it would not have changed much (less than 3.3% for all mechanisms) in the results.

Figure 6.8 shows the mechanisms' coverage ratio loss for the regular weekday and for the holiday. Only the ACE and Baseline mechanism present some energy loss since they are the only mechanisms that do not have a coverage guarantee. The Baseline coverage loss remains the same, which shows that the traffic demand for night hours (0AM - 6:59AM) on both our holiday or weekday remains unchanged. The smaller coverage ratio loss for our holiday when compared to our weekday on the ACE mechanism case can be explained by the smaller traffic demand estimated for the whole day. Another explanation for the ACE mechanism reduced coverage ratio can be on the fact that the ACE mechanism has a smaller normalized energy saving factor on our holiday, which means it has a smaller number of APs switched off or that they are switched off for a short period of time.

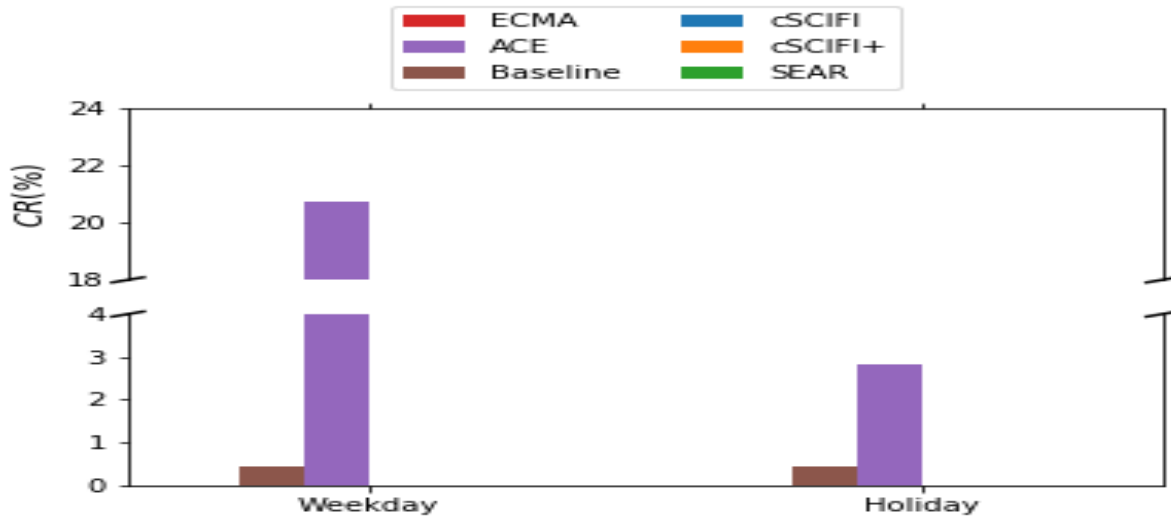


Figure 6.8: Coverage ratio loss comparison between a holiday and a weekday

The results showed in this section can not give us a precise conclusion on whether or not our algorithm could cope with the holiday demand without sacrificing the normalized energy saving factor. A large number of holidays in distinct weekdays and with distinct demand estimations would be necessary to understand it better. However results have indicated that the mechanism performance on holidays is not related to any change on its function, but it is in fact intimately related to the correct traffic estimations given by

the hybrid model when compared to the real traffic data.

6.5 Real Data vs. Model Predictions

Energy saving mechanisms that use scheduling driven approaches usually present smaller normalized energy saving factor when compared to mechanisms that use demand driven approaches. That happens because the real traffic demands are not exactly the same obtained by traffic estimations. However if the traffic estimations are very close to the real data traffic demands those approaches may give the same result. In this section we compare the normalized energy saving factor of the eSCIFI mechanism using real traffic data and model prediction demand estimations. We will compare the eSCIFI mechanism using both clustering algorithms with the parameters that gave the best normalized energy saving factor and no coverage ratio loss (without the use of a special AP set, $T_{min} = 54$ and $tw = 120$ minutes). Figure 6.9 shows the normalized energy saving factor for both clustering algorithms using real traffic data and the hybrid model's predictions as demand estimation. The results shows that there is no difference between the normalized energy saving factor achieved with them for both clustering algorithms. The explanation behind these results is that the cluster maximum capacity CAA value is huge when compared to the hybrid model's prediction errors. The errors introduced by the hybrid model could not reach the cluster maximum capacity CAA threshold and change the decisions taken by the energy state algorithm. Therefore in our evaluation scenario, none of the errors introduced by the hybrid model on the demand estimations were sufficient to change the algorithm energy state decisions for APs when compared to the decisions taken by the algorithm using real demand estimations. The cluster maximum capacity CAA value had a direct impact on these results however it is not simple to change the CAA value, once it is directly related to the T_{max} value that was based on the maximum value registered in our history data.

This result shows that for our experimental scenario the eSCIFI energy saving mechanisms can effectively use machine learning models to predict traffic demands without any impact over the normalized energy saving factor. This result also shows that the eSCIFI scheduling driven strategy can not only cope with the network real data acquisition constraint but also deliver a normalized energy saving factor comparable to scenarios where it would be possible to use real data acquisition.

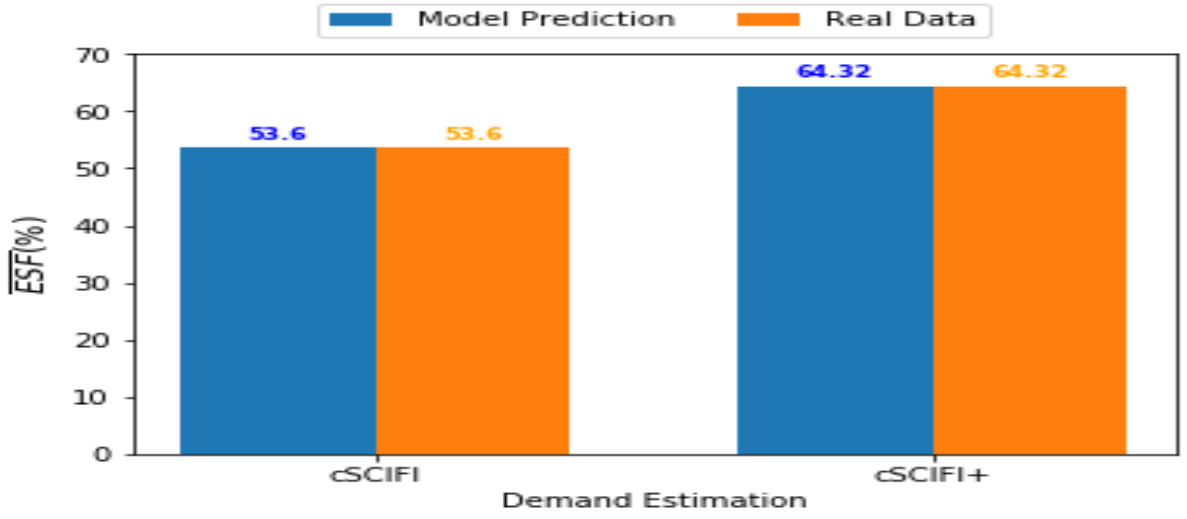


Figure 6.9: Normalized Energy saving factor comparison between the eSCIFI mechanism using real traffic data and using model prediction demand estimations

6.6 SEAR vs. eSCIFI Clustering Algorithms

As we have seen in Section 6.2, the SEAR mechanism had a better normalized energy saving factor result than the eSCIFI mechanism using the cSCIFI clustering algorithm. However the clustering algorithm developed by Jardosh et al. [29] does not have the same optimizations criterion we have implemented on our both algorithms. Therefore the work of Jardosh et al. [29] is susceptible to the order of appearance of APs in the neighborhood list of other APs. This order affects which will be the next APs selected by the Jardosh et al. [29] green clustering algorithm to fill the cluster in case of ties between the number of neighbors. The order of appearance of APs in the neighborhood list impacts the Jardosh et al. [29] green clustering algorithm result since it does not have any tie breaker rule in the selection of the next AP to be put in the cluster in case of a tie in the number of neighbors between APs. The cSCIFI and cSCIFI+ algorithms do not have this disadvantage and therefore we can guarantee that the clusters formed will not depend on the order of appearance. Here we will compare the normalized energy saving factor achieved by the SEAR and eSCIFI mechanism using both clustering algorithm using 3 different orders of appearance of APs in the neighborhood lists of the APs. The two first neighborhood lists present cases where the APs position inside the neighborhood lists are randomized and the third represent the neighborhood list we have used for all tests we have done before for the SEAR mechanism. We will compare the SEAR and eSCIFI mechanism using both clustering algorithms with the parameters that gave the best normalized energy saving factor and no coverage ratio loss (without the use of a special AP set, $T_{min} = 54$ and

$tw = 120$).

As we can see in Figure 6.10, the SEAR mechanism energy saving result is heavily affected by the order of appearance of APs in the neighborhood list, while the eSCIFI algorithm using cSCIFI and cSCIFI+ are not affected at all. This result shows that the changes we have implemented on the cSCIFI and cSCIFI+ have turned our algorithm unaffected by the order of appearance of AP in the neighborhood list. This is a clear advantage since it will not require an optimization on the neighborhood list formation process that in a huge network topology might be impractical to be done.

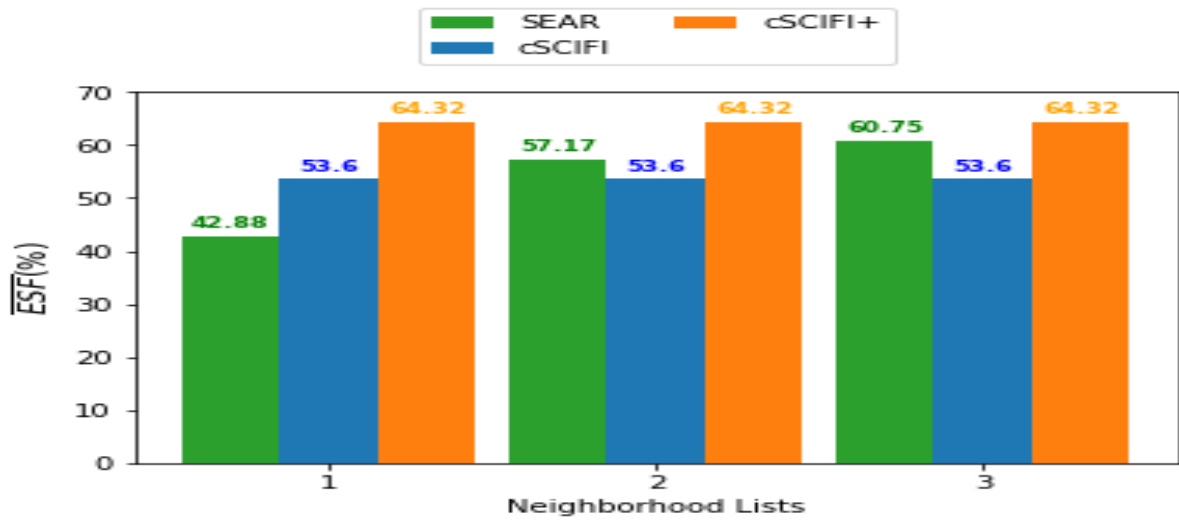


Figure 6.10: Normalized Energy saving factor comparison between the SEAR and eSCIFI mechanism using both clustering algorithm with different neighborhood lists

Chapter 7

Conclusion

We presented the eSCIFI energy saving mechanism and its main architecture. The eSCIFI energy saving mechanism used traffic demand estimations given by machine learning model to manage the energy state of APs and it was designed to cope with a broader variety of wireless network, specially those wireless network that can not collect traffic data in a real time manner and/or have a limited CPU power. We proposed and used a unified methodology to find the best machine learning model for our evaluation scenario. We conducted an experimental analysis using our proposed methodology and two datasets we created from the UFF's SCIFI network APs, belonging to a classroom building, over a period of 6 months, from April to September 2018. Our results showed that the collective single-target machine learning model using the the decision tree as a machine learning algorithm and using only the APs identification number, the holiday and weekday as input features (Col/DT/ST/APHDWD machine learning model) achieved achieved the best $\overline{RMSP\bar{E}}$ results for the regression problem (with an \overline{RMPSE} value of 0.29). The collective single-label machine learning model using the the decision tree as a machine learning algorithm and using only the APs identification number, the holiday and weekday as input features (Col/DT/SL/APHDWD machine learning model) achieved the best \bar{A} accuracy results for the classification problem (with an \bar{A} of 86.69%). Our experimental analysis showed that the proposed methodology could broadly and extensively evaluate the machine learning models.

eSCIFI uses a hybrid model that combines the classification and regression models to provide a better occupancy estimation. The hybrid model has proven to be able to provide very accurate traffic demand estimation and that it can cope with the network distinct demands. We also proposed two clustering algorithms: cSCIFI and the cSCIFI+. cSCIFI is a less aggressive clustering strategy that may increase the cluster available

traffic capacity while cSCIFI+ is a more aggressive one that presents a limited maximum traffic capacity. After describing the eSCIFI mechanism main components and features, we conducted a trace-driven analysis to evaluate how well the eSCIFI mechanism would be in our motivational scenario.

We evaluated how the special AP set size, the time window size, the T_{min} value, the traffic demand estimation and holiday feature would affect the normalized energy saving factor and the coverage ratio loss of our proposed mechanism. We also reproduced and compared the eSCIFI mechanism results to the results achieved by the ACE, ECMA and SEAR mechanism. Those results showed that, for the UFF SCIFI network scenario, the eSCIFI mechanism produced good results and that the machine learning model produced results comparable to the ones obtained using real time data. The SEAR mechanism got comparable results to the eSCIFI mechanism using the cSCIFI clustering algorithm and therefore, at a first analysis, would be a better solution than the eSCIFI mechanism using the cSCIFI clustering algorithm. However the SEAR green clustering algorithm normalized energy saving factor changes depending on the order of appearance of APs in the neighborhood list, while with the eSCIFI algorithm using cSCIFI and cSCIFI+, the energy saving factor does not change, which is a clear advantage of our clustering algorithms. All of those results showed that, for the UFF SCIFI motivation scenario, the best energy saving mechanism was the eSCIFI using the cSCIFI+ mechanism that can save up to 64.32% of the total energy consumed in a week without affecting the network coverage and user's association capacity.

7.1 Limitations

Our unified methodology evaluates several regression and classification machine learning models and selects the best ones to the desired usage scenario. However all the evaluations made and the best regression and classification model selections were manually made. This manual selection of models took several hours of analysis due to the great number of distinct models evaluate and metrics used. However this analysis can become even more challenging if more configuration parameters are used. Therefore the automatic model selection and comparison between models is required in more complex evaluation scenarios.

The vicinity criteria defined in Section 5.1.3.2 uses spatial proximity bewteen APs to determine if two APs are neighbors. However this assumption is not always true. There

are numerous cases where radio interference caused by obstacles such as thick walls or doors between two spatially close APs make them practically invisible to each other. A better vicinity criteria would be using RSSI values measured between APs to create an interference matrix where the APs' radio interference above a certain threshold on others would indicate the proximity between them. However it was not possible to use the UFF SCIFI network interference matrix to create the neighborhood list because those data were unavailable.

Another limitation of this work is related to the analysis of our machine learning model prediction's horizon. The trained machine learning models can fairly predict the network traffic demand for a whole month (September) afterwards. However it is expected that this accuracy would decrease over time. It would be required to have a broad dataset to evaluate this question and define a model update parameter. Therefore for our case we simply assumed that the machine learning model predictions would keep the same accuracy throughout the weeks.

In our tests we used $T_{max} = 300$. This value represents that an AP could handle at least 300 user devices throughout a 10-minute time slot. This value does not mean that an AP could handle those 300 devices at once or throughout the whole time slots, but rather the maximum associations process that could be done in a time slot. This means that during those 10-minute time slots, there are mobile stations that associate and dissociate to that specific AP. Some APs located in transit places such as the building entrance might notice a huge number of association that are just transitory. However assuming $T_{max} = 300$ might be a practical problem in a real network scenario, since it may affect the coverage ratio loss. In a real scenario the APs used in the network could handle a number of associations at the same time considerably smaller than 300. Therefore the real impacts of assuming $T_{max} = 300$ on the coverage ratio loss might change in real network scenarios where the instantaneous cluster capacity might be reached. In a real network scenario the effects of assuming $T_{max} = 300$ might also affect the normalized energy saving factor. However changing T_{max} value is not simple, since it changes the cluster maximum capacity CAA that affects the energy saving algorithms decisions and give a complete different energy state schedule to the APs. Therefore this assumption shows some practical problems in real network scenarios that would need to be further analyzed.

7.2 Future Work

We plan to insert the AP location as input features in our constructed dataset. The AP location inside the building can bring relevant information about its association history and can therefore enhance the overall performance of our machine learning model.

We have plans to conduct a more complex evaluation and selection of machine learning models using our proposed unified methodology. We have plans to use the cross validation method to test our models with different training set sizes. We also have plans to use additional machine learning algorithms for classification models such as the XGBoost.

We have plans to create an overall assessment rank to evaluate and select machine learning models using our proposed unified methodology. An automatic selection based in an overall rank metric can decrease the time and effort required to evaluate and select those models using our unified methodology. Therefore an automatic assessment can ease the process and make it a more feasible solution to more complex smart building scenarios.

We have seen that the special AP selection criteria and the special AP set have not improved the eSCIFI normalized energy saving factor. The cSCIFI and cSCIFI+ clustering algorithms cluster set formed is strongly dependent on the neighborhoods. Therefore some future changes on its design are required to improve its results such as the inclusion of a minimum neighborhood criteria.

We also plan to propose new features to the eSCIFI design such as the machine learning model update scheme and a fail protection scheme. The model update scheme on our mechanism would allow the eSCIFI machine learning model to be updated with new training traffic data from time to time. This model update feature would help the model estimations accuracy to not significantly decrease over time. The implementation of such feature is required for an energy saving mechanism that would have to work throughout years and there would be necessary to evaluate strategies to perform this model update the best way possible. Another important feature that we plan to add to our energy saving mechanism is a fail protection scheme. Network elements might not be working temporarily due to some fails such as energy outages. The eSCIFI mechanism would be extremely sensible if the cluster heads or the controller fails for some reason. Therefore some protection schemes for that cases would be necessary to increase the eSCIFI resilience.

The eSCIFI mechanism has not being tested and implemented in real network scenarios. We plan to implement the eSCIFI mechanism on the UFF SCIFI controller and do

some future experiments using the real network infrastructure. The real network would give us some real insights about how the eSCIFI would cope with a real scenario and to properly tune its parameters according to a real implementation. We also plan to use more sophisticated metrics as the average throughput and delay to evaluate the network performance and user's coverage on those real network tests. We hope that those tests and new features will allow us to fully understand the eSCIFI possibilities and overcome its limitations.

References

- [1] APOSTOLO, G. H.; BERNARDINI, F.; MAGALHÃES, L. C. S.; MUCHALUAT-SAADE, D. C. An experimental analysis for detecting wi-fi network associations using multi-label learning. In *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)* (2020), IEEE, pp. 423–428.
- [2] APOSTOLO, G. H.; BERNARDINI, F.; MAGALHÃES, L. C. S.; MUCHALUAT-SAADE, D. C. A unified methodology to predict wi-fi network usage in smart buildings. *IEEE Access* 9 (2021), 11455–11469.
- [3] APOSTOLO, G. H.; MUCHALUAT-SAADE, D. C.; MAGALHÃES, L. C. S.; BERNARDINI, F. C. Análise de associações em redes wi-fi utilizando técnicas de aprendizado de máquina multirrótulo para economia de energia da rede. In *Anais do XXV Workshop de Gerência e Operação de Redes e Serviços* (2020), SBC, pp. 125–138.
- [4] BALAJI, B.; XU, J.; NWOKAFOR, A.; GUPTA, R.; AGARWAL, Y. Sentinel: occupancy based hvac actuation using existing WiFi infrastructure within commercial buildings. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems* (2013), pp. 1–14.
- [5] BARTH, W. *Nagios: System and network monitoring*. No Starch Press, 2008.
- [6] BUDZISZ, Ł.; GANJI, F.; RIZZO, G.; MARSAN, M. A.; MEO, M.; ZHANG, Y.; KOUTITAS, G.; TASSIULAS, L.; LAMBERT, S.; LANNOO, B., ET AL. Dynamic resource provisioning for energy efficiency in wireless access networks: A survey and an outlook. *IEEE Communications Surveys & Tutorials* 16, 4 (2014), 2259–2285.
- [7] BUITINCK, L.; LOUPPE, G.; BLONDEL, M.; PEDREGOSA, F.; MUELLER, A.; GRISEL, O.; NICULAE, V.; PRETTENHOFER, P.; GRAMFORT, A.; GROBLER, J.; LAYTON, R.; VANDERPLAS, J.; JOLY, A.; HOLT, B.; VAROQUAUX, G. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning* (2013).
- [8] CAMACHO, D.; NOVAIS, P. *Innovations and practical applications of intelligent systems in ambient intelligence and humanized computing*, 2017.
- [9] CAPONE, A.; MALANDRA, F.; SANSÒ, B. Energy savings in wireless mesh networks in a time-variable context. *Mobile Networks and Applications* 17, 2 (2012), 298–311.
- [10] CHEN, Y.-J.; SHEN, Y.-H.; WANG, L.-C. Achieving energy saving with qos guarantee for wlan using sdn. In *2016 IEEE International Conference on Communications (ICC)* (2016), IEEE, pp. 1–7.

- [11] CHIN, K.-W. A green scheduler for enterprise wlans. In *2011 Australasian Telecommunication Networks and Applications Conference (ATNAC)* (2011), IEEE, pp. 1–3.
- [12] CHRISTENSEN, K.; MELFI, R.; NORDMAN, B.; ROSENBLUM, B.; VIERA, R. Using existing network infrastructure to estimate building occupancy and control plugged-in devices in user workspaces. *International Journal of Communication Networks and Distributed Systems* 12, 1 (2014), 4–29.
- [13] CUI, Y.; MA, X.; WANG, H.; STOJMENOVIC, I.; LIU, J. A survey of energy efficient wireless transmission and modeling in mobile cloud computing. *Mobile Networks and Applications* 18, 1 (2013), 148–155.
- [14] DALMASSO, M.; MEO, M.; RENG, D. Radio resource management for improving energy self-sufficiency of green mobile networks. *ACM SIGMETRICS Performance Evaluation Review* 44, 2 (2016), 82–87.
- [15] DEBELE, F. G.; LI, N.; MEO, M.; RICCA, M.; ZHANG, Y. Experimenting resource-demand strategies for green wlans. *ACM SIGMETRICS Performance Evaluation Review* 42, 3 (2014), 61–66.
- [16] DONEVSKI, I.; VALLERO, G.; MARSAN, M. A. Neural networks for cellular base station switching. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)* (2019), IEEE, pp. 738–743.
- [17] FAINELLI, F. The openwrt embedded development framework. In *Proceedings of the Free and Open Source Software Developers European Meeting* (2008), p. 106.
- [18] FANG, L.; XUE, G.; LYU, F.; SHENG, H.; ZOU, F.; LI, M. Intelligent large-scale ap control with remarkable energy saving in campus WiFi system. In *2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)* (2018), IEEE, pp. 69–76.
- [19] GANJI, F.; BUDZISZ, Ł.; WOLISZ, A. Assessment of the power saving potential in dense enterprise wlans. In *IEEE 24th Intl. Symposium on Personal, Indoor, and Mobile Radio Communications* (2013), IEEE.
- [20] GARDNER, M. W.; DORLING, S. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment* 32, 14-15 (1998), 2627–2636.
- [21] GHAI, S. K.; THANAYANKIZIL, L. V.; SEETHARAM, D. P.; CHAKRABORTY, D. Occupancy detection in commercial buildings using opportunistic context sources. In *2012 IEEE International Conference on Pervasive Computing and Communications Workshops* (2012), IEEE, pp. 463–466.
- [22] GOMEZ, K.; SENGUL, C.; BAYER, N.; RIGGIO, R.; RASHEED, T.; MIORANDI, D. Morfeo: Saving energy in wireless access infrastructures. In *2013 IEEE 14th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)* (2013), IEEE, pp. 1–6.

- [23] GONÇALVES, E. C.; PLASTINO, A.; FREITAS, A. A. Simpler is better: a novel genetic algorithm to induce compact multi-label chain classifiers. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation* (2015), pp. 559–566.
- [24] GONÇALVES, E. C.; FREITAS, A. A.; PLASTINO, A. A survey of genetic algorithms for multi-label classification. In *2018 IEEE Congress on Evolutionary Computation (CEC)* (July 2018), pp. 1–8.
- [25] HARATCHEREV, I.; FIORITO, M.; BALAGEAS, C. Low-power sleep mode and out-of-band wake-up for indoor access points. In *2009 IEEE Globecom Workshops* (2009), IEEE, pp. 1–6.
- [26] HERRERA, F.; CHARTE, F.; RIVERA, A. J.; DEL JESUS, M. J. Multilabel classification. In *Multilabel Classification*. Springer, 2016, pp. 17–31.
- [27] HOBSON, B. W.; LOWCAY, D.; GUNAY, H. B.; ASHOURI, A.; NEWSHAM, G. R. Opportunistic occupancy-count estimation using sensor fusion: A case study. *Building and environment* 159 (2019).
- [28] JARDOSH, A. P.; IANNACCONE, G.; PAPAGIANNAKI, K.; VINNAKOTA, B. Towards an energy-star wlan infrastructure. In *Eighth IEEE Workshop on Mobile Computing Systems and Applications* (2007), IEEE, pp. 85–90.
- [29] JARDOSH, A. P.; PAPAGIANNAKI, K.; BELDING, E. M.; ALMEROTH, K. C.; IANNACCONE, G.; VINNAKOTA, B. Green wlangs: on-demand wlan infrastructures. *Mobile Networks and Applications* 14, 6 (2009), 798–814.
- [30] KONDO, Y.; YOMO, H.; TANG, S.; IWAI, M.; TANAKA, T.; TSUTSUI, H.; OBANA, S. Energy-efficient wlan with on-demand ap wake-up using ieee 802.11 frame length modulation. *Computer Communications* 35, 14 (2012).
- [31] KUMAZOE, K.; NOBAYASHI, D.; FUKUDA, Y.; IKENAGA, T. Station aggregation scheme considering channel interference for radio on demand networks. In *2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)* (2013), IEEE, pp. 265–270.
- [32] KUMAZOE, K.; NOBAYASHI, D.; FUKUDA, Y.; IKENAGA, T.; ABE, K. Multiple station aggregation procedure for radio-on-demand wlangs. In *2012 Seventh International Conference on Broadband, Wireless Computing, Communication and Applications* (2012), IEEE, pp. 156–161.
- [33] LEE, K.; KIM, Y.; KIM, S.; SHIN, J.; SHIN, S.; CHONG, S. Just-in-time wlangs: On-demand interference-managed wlan infrastructures. In *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications* (2016), IEEE, pp. 1–9.
- [34] LIU, L.; XU, S.; HU, J.; CUI, L.; MIN, G. Balancing of the quality-of-service, energy and revenue of base stations in wireless networks via tullock contests. In *Proceedings of the International Symposium on Quality of Service* (2019), pp. 1–8.

- [35] LORINCZ, J.; BOGARELLI, M.; CAPONE, A.; BEGUŠIĆ, D. Heuristic approach for optimized energy savings in wireless access networks. In *SoftCOM 2010, 18th International Conference on Software, Telecommunications and Computer Networks* (2010), IEEE, pp. 60–65.
- [36] LORINCZ, J.; CAPONE, A.; BOGARELLI, M. Energy savings in wireless access networks through optimized network management. In *IEEE 5th Intl. Symposium on Wireless Pervasive Computing* (2010), IEEE, pp. 449–454.
- [37] LYU, F.; FANG, L.; XUE, G.; XUE, H.; LI, M. Large-scale full WiFi coverage: Deployment and management strategy based on user spatio-temporal association analytics. *IEEE Internet of Things Journal* 6, 6 (2019).
- [38] MAGALHÃES, L. C. S.; BALBI, H. D.; CORRÊA, C.; VALLE, R. D. T. D.; STANTON, M. Scifi—a software-based controller for efficient wireless networks. In *UbuntuNet-Connect 2013* (2013), UbuntuNet Alliance.
- [39] MARSAN, M. A.; CHIARAVIGLIO, L.; CIULLO, D.; MEO, M. A simple analytical model for the energy-efficient activation of access points in dense wlans. In *Proceedings of the 1st international conference on energy-efficient computing and networking* (2010), pp. 159–168.
- [40] MCKINNEY, W. pandas: a foundational Python library for data analysis and statistics. *Python for High Performance and Scientific Computing* (2011).
- [41] NAGAREDA, R.; HASEGAWA, A.; SHIBATA, T.; OBANA, S. A proposal of power saving scheme for wireless access networks with access point sharing. In *2012 International Conference on Computing, Networking and Communications (ICNC)* (2012), IEEE, pp. 1128–1132.
- [42] OETIKER, T.; RAND, D. Mrtg: The multi router traffic grapher. In *LISA* (1998), vol. 98, pp. 141–148.
- [43] PANDEY, S.; HINDOLIYA, D.; MOD, R. Artificial neural networks for predicting indoor temperature using roof passive cooling techniques in buildings in different climatic conditions. *Applied Soft Computing* 12, 3 (2012).
- [44] ROSSI, C.; CASETTI, C.; CHIASSERINI, C.-F.; BORGIATTINO, C. Cooperative energy-efficient management of federated wifi networks. *IEEE Transactions on Mobile Computing* 14, 11 (2015), 2201–2215.
- [45] SANGOGBOYE, F. C.; IMAMOVIC, K.; KJÆRGAARD, M. B. Improving occupancy presence prediction via multi-label classification. In *2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)* (2016), IEEE, pp. 1–6.
- [46] SILVA, P.; ALMEIDA, N. T.; CAMPOS, R. Energy consumption management for dense wi-fi networks. In *2019 Wireless Days (WD)* (2019), IEEE, pp. 1–8.
- [47] SPYROMITROS-XIOUFIS, E.; TSOUMAKAS, G.; GROVES, W.; VLAHAVAS, I. Multi-label classification methods for multi-target regression. *arXiv preprint arXiv:1211.6581* (2012), 1159–1168.

- [48] STOLFI, D. H.; ALBA, E. Green swarm: Greener routes with bio-inspired techniques. *Applied Soft Computing* 71 (2018), 952–963.
- [49] TANAKA, T.; ABE, K.; AUST, S.; ITO, T.; YOMO, H.; SAKATA, S. Automatic and cooperative sleep control strategies for power-saving in radio-on-demand wlans. In *2013 IEEE Green Technologies Conference (GreenTech)* (2013), IEEE, pp. 293–300.
- [50] TANG, S.; YOMO, H.; KONDO, Y.; OBANA, S. Wake-up receiver for radio-on-demand wireless lans. *EURASIP Journal on Wireless Communications and Networking* 2012, 1 (2012), 42.
- [51] TRIVEDI, A.; GUMMESON, J.; IRWIN, D.; GANESAN, D.; SHENOY, P. ischedule: Campus-scale hvac scheduling via mobile WiFi monitoring. In *Proceedings of the Eighth Intl. Conference on Future Energy Systems* (2017).
- [52] VALLERO, G.; RENG, D.; MEO, M.; MARSAN, M. A. Greener ran operation through machine learning. *IEEE Transactions on Network and Service Management* 16, 3 (2019), 896–908.
- [53] WANG, W.; CHEN, J.; HONG, T. Occupancy prediction through machine learning and data fusion of environmental sensing and WiFi sensing in buildings. *Automation in Construction* 94 (2018), 233–243.
- [54] WANG, W.; CHEN, J.; HONG, T.; ZHU, N. Occupancy prediction through markov based feedback recurrent neural network (m-frnn) algorithm with WiFi probe technology. *Building and Environment* 138 (2018).
- [55] WU, W.; LUO, J.; DONG, K.; YANG, M.; LING, Z. Energy-efficient user association with congestion avoidance and migration constraint in green wlans. *Wireless Communications and Mobile Computing* 2018 (2018).
- [56] XU, C.; HAN, Z.; ZHAO, G.; YU, S. A sleeping and offloading optimization scheme for energy-efficient wlans. *IEEE Communications Letters* 21, 4 (2016), 877–880.
- [57] XU, C.; WANG, J.; ZHU, Z.; NIYATO, D. Energy-efficient wlans with resource and re-association scheduling optimization. *IEEE Transactions on Network and Service Management* 16, 2 (2019), 563–577.
- [58] YOMO, H.; KONDO, Y.; NAMBA, K.; TANG, S.; KIMURA, T.; ITO, T. Wake-up id and protocol design for radio-on-demand wireless lan. In *2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications-(PIMRC)* (2012), IEEE, pp. 419–424.
- [59] ZHANG, Y.; JIANG, C.; WANG, J.; HAN, Z.; YUAN, J.; CAO, J. Green wi-fi implementation and management in dense autonomous environments for smart cities. *IEEE Transactions on Industrial Informatics* 14, 4 (2017), 1552–1563.
- [60] ZOU, H.; JIANG, H.; YANG, J.; XIE, L.; SPANOS, C. Non-intrusive occupancy sensing in commercial buildings. *Energy and Buildings* 154 (2017).
- [61] ZOU, H.; ZHOU, Y.; JIANG, H.; CHIEN, S.-C.; XIE, L.; SPANOS, C. J. Winlight: A WiFi-based occupancy-driven lighting control system for smart building. *Energy and Buildings* 158 (2018), 924–938.

-
- [62] ZOU, H.; ZHOU, Y.; YANG, J.; SPANOS, C. J. Towards occupant activity driven smart buildings via WiFi-enabled iot devices and deep learning. *Energy and Buildings* 177 (2018), 12–22.

APPENDIX A – UFF SCIFI AP Positions in the H Building

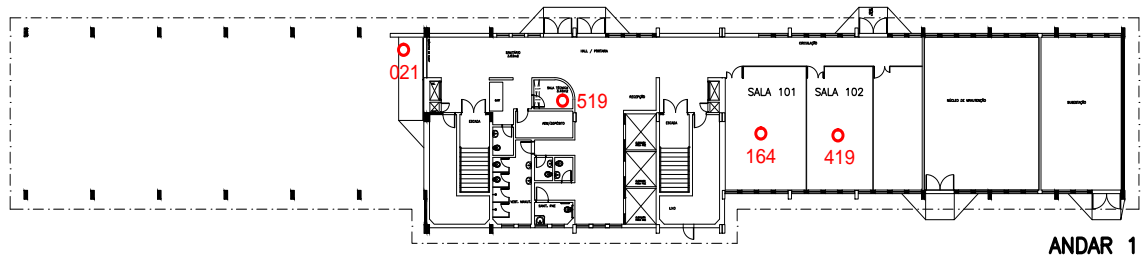


Figure A.1: H building ground floor blueprint showing the UFF SCIFI AP positions

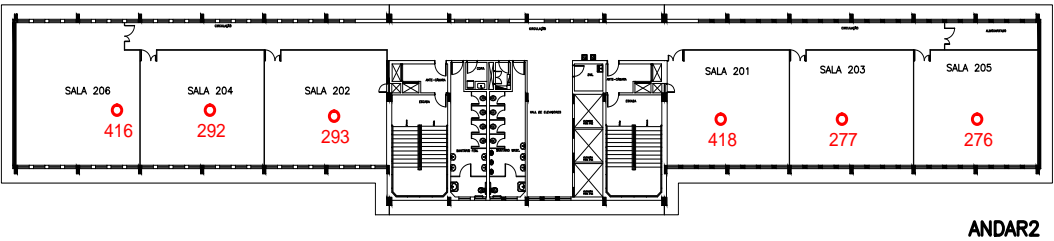


Figure A.2: H building second floor blueprint showing the UFF SCIFI AP positions

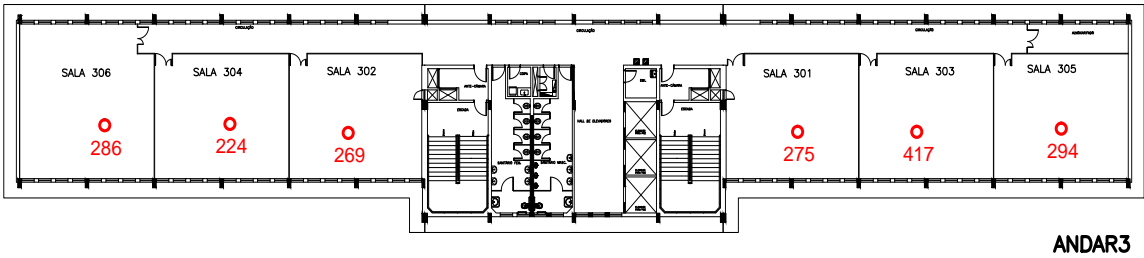


Figure A.3: H building third floor blueprint showing the UFF SCIFI AP positions

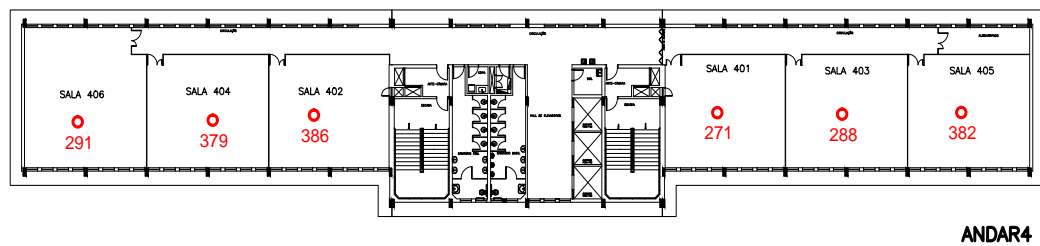


Figure A.4: H building fourth floor blueprint showing the UFF SCIFI AP positions

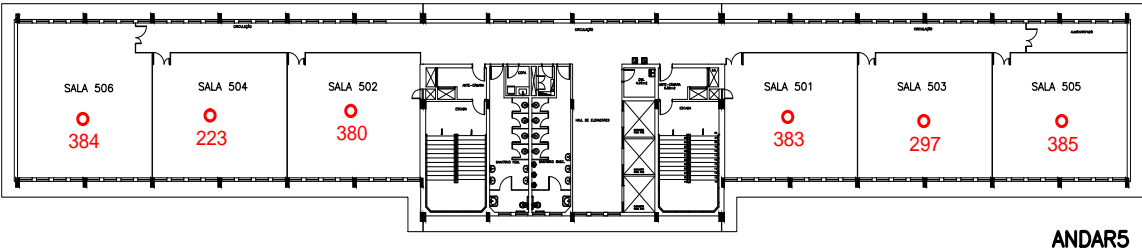


Figure A.5: H building fifth floor blueprint showing the UFF SCIFI AP positions

APPENDIX B – Datasets Example

Table B.1: A part of the constructed ML dataset, showing the input attributes and the occupancy detection history for the APs

APid	Holiday	Day of the Week	T_0	T_1	...	T_{143}
21	F	Sunday	0	0	...	0
164	F	Sunday	0	0	...	0
223	F	Sunday	0	0	...	0
.						
.
.						
269	T	Wednesday	0	0	...	1
276	T	Wednesday	0	0	...	0
277	T	Wednesday	0	0	...	1

Table B.2: A part of the constructed SL dataset, showing the input attributes and the occupancy count history for the APs

APid	Holiday	Day of the Week	Month	Day	Hour	Minute	Occupancy Counter
21	F	Sunday	4	8	0	0	0
164	F	Sunday	4	8	6	10	2
223	F	Sunday	4	8	15	20	23
.							
.
.							
269	F	Wednesday	9	26	0	30	1
276	F	Wednesday	9	26	7	40	7
277	F	Wednesday	9	26	16	50	50
277	T	Friday	9	7	13	0	2

APPENDIX C – UFF SCIFI Network Topology

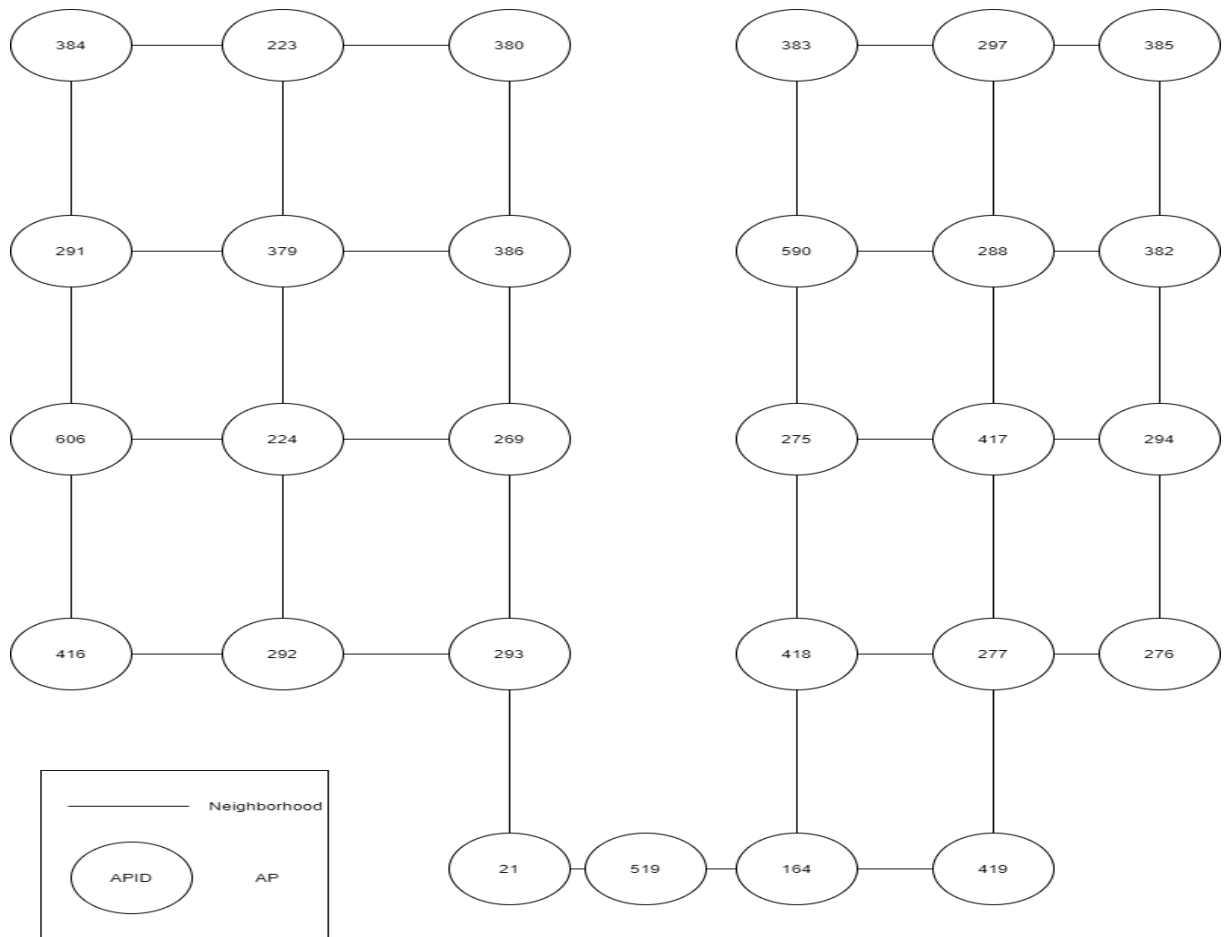


Figure C.1: H building UFF SCIFI network topology showing the neighbor APs

APPENDIX D - APs Statistics

Table D.1: Access Point statistics from April to August 2018.

Apid	Avg. Month	Avg. Day	Avg. Hour	Avg. Timeslot	Night	Morning	Noon	Total	Max
21	66151.40	2312.99	96.37	16.06	0.33	15.69	25.89	330757	272
164	28797.40	1199.89	50.00	8.33	0.08	7.42	13.88	143987	106
223	20144.60	839.36	34.97	5.83	0.12	5.15	9.69	100723	73
224	27962.00	1306.64	54.44	9.07	0.01	9.62	14.33	139810	87
269	30226.40	1280.78	53.37	8.89	0.01	9.99	13.74	151132	84
275	29384.00	1412.69	58.86	9.81	0.05	9.98	15.69	146920	83
276	39976.80	1417.62	59.07	9.84	0.12	8.33	16.61	199884	115
277	50013.80	1786.21	74.43	12.40	0.11	11.51	20.41	250069	109
288	25658.80	1105.98	46.08	7.68	0.01	7.94	12.23	128294	72
291	43522.00	1532.46	63.85	10.64	0.35	10.23	17.16	217610	98
292	26472.60	938.74	39.11	6.52	0.02	4.67	11.49	132363	79
293	27810.80	1000.39	41.68	6.95	0.04	5.72	11.83	139054	90
294	28812.20	1241.91	51.75	8.62	0.08	7.56	14.43	144061	77
297	24764.20	1049.33	43.72	7.29	0.24	6.93	11.79	123821	71
379	29640.40	1245.39	51.89	8.65	0.08	8.54	13.95	148202	77
380	26750.60	948.60	39.53	6.59	0.46	6.09	10.60	133753	82
382	34064.20	1207.95	50.33	8.39	0.11	7.92	13.70	170321	76
383	27055.20	1156.21	48.18	8.03	0.07	7.48	13.20	135276	86
384	34267.40	1215.16	50.63	8.44	0.54	6.73	14.19	171337	83
385	15401.80	641.74	26.74	4.46	0.03	3.84	7.50	77009	52
386	28956.20	1237.44	51.56	8.59	0.01	9.08	13.58	144781	81
416	37897.00	1353.46	56.39	9.40	0.08	8.18	15.76	189485	97
417	39262.40	1663.66	69.32	11.55	0.04	11.50	18.63	196312	107
418	30582.80	1124.37	46.85	7.81	0.05	7.18	12.89	152914	72
419	42073.60	1546.82	64.45	10.74	0.09	9.27	18.05	210368	128
519	44768.20	1805.17	75.22	12.54	0.13	11.59	20.64	223841	129
590	31248.40	1346.91	56.12	9.35	0.05	9.58	14.92	156242	88
606	24063.80	1037.23	43.22	7.20	0.01	8.08	11.14	120319	78

APPENDIX E – Overall APs Rank

Table E.1: Overall Rank of UFF SCIFI H building network

APid	WA value	APid	WA value
21	0.4137	382	0.1970
519	0.3940	386	0.1945
277	0.3842	164	0.1896
419	0.3596	294	0.1896
417	0.3497	418	0.1600
291	0.3448	383	0.1527
276	0.3374	293	0.1280
416	0.3004	380	0.0935
590	0.2733	288	0.0911
275	0.2586	606	0.0812
269	0.2413	292	0.0763
384	0.2266	297	0.0763
224	0.2241	223	0.0369
379	0.2093	385	0.0147

APPENDIX F – Clusters Formed With cSFICI and cSCFI+

Table F.1: Cluster formation of UFF SCIFI H building network using cSCIFI algorithm

Special APs Set Size	Special APs Set	Number of Clusters	Cluster Set Formed	Clusters Set Identical To (Cluster Set of Size)
0	-	10	[21, 519], [164, 419], [223, 384, 380],[224, 606, 269], [275, 417, 294], [277, 276, 418], [288, 590, 382], [291, 379, 386], [293, 292, 416], [297, 383, 385]	N/A
3	21,519,277	11	[277, 276, 418], [21, 293], [519, 164], [224, 269, 606], [275, 417, 294], [288, 590, 382], [291, 379, 386], [223, 384, 380], [292, 416], [297, 383, 385], [419]	4
4	21,519,277,419	11	[277, 276, 418], [21, 293], [519, 164], [419], [224, 269, 606], [275, 417, 294], [288, 590, 382], [291, 379, 386], [223, 384, 380], [292, 416], [297, 383, 385]	3
5	21,519,277,419,417	11	[277, 276, 418], [417, 275, 294], [21, 293], [519, 164], [419], [224, 269, 606], [288, 590, 382], [291, 379, 386], [223, 384, 380], [292, 416], [297, 383, 385]	6
6	21,519,277,419, 417,291	11	[277, 276, 418], [417, 275, 294], [291, 379, 386], [21, 293], [519, 164], [419], [224, 269, 606], [288, 590, 382], [223, 384, 380], [292, 416], [297, 383, 385]	5
7	21,519,277,419, 417,291,276	12	[277, 418], [417, 275, 294], [291, 379, 386], [276], [21, 293], [519, 164], [419], [224, 269, 606], [288, 590, 382], [223, 384, 380], [292, 416], [297, 383, 385]	N/A
8	21,519,277,419, 417,291,276,416	12	[277, 418], [417, 275, 294], [291, 379, 386], [276], [416, 292, 293], [21], [519, 164], [419], [224, 606, 269], [288, 590, 382], [223, 384, 380], [297, 383, 385]	N/A
9	21,519,277,419, 417,291,276,416,590	12	[277, 418], [417, 275, 294], [291, 379, 386], [590, 288, 382], [276], [416, 292, 293], [21], [519, 164], [419], [224, 606, 269], [223, 384, 380], [297, 383, 385]	N/A

Table F.2: Cluster formation of UFF SCIFI H building network using cSCIFI+ algorithm

Special APs Set Size	Special APs Set	Number of Clusters	Cluster Set Formed	Clusters Set Identical To (Cluster Set of Size)
0	-	10	[223, 380], [224, 606, 269, 379, 292], [288, 590, 382, 417, 297], [291, 384, 386], [293, 21, 416], [294, 276, 275], [383, 385], [418, 164, 277], [419], [519]	N/A
3	21,519,277	9	[277, 418, 417, 276, 419], [416], [519, 164], [224, 606, 379, 269, 292], [275, 590, 294], [288, 297, 382], [291, 384, 386], [223, 380], [383, 385]	N/A
4	21,519,277,419	10	[277, 418, 417, 276], [416], [519, 164], [419], [224, 606, 379, 269, 292], [275, 590, 294], [288, 297, 382], [291, 384, 386], [223, 380], [383, 385]	N/A
5	21,519,277,419,417	10	[277, 418, 276], [417, 288, 275, 294], [416], [519, 164], [419], [224, 606, 379, 269, 292], [291, 384, 386], [382, 385, 590], [223, 380], [297, 383]	N/A
6	21,519,277,419, 417,291	10	[277, 418, 276], [417, 288, 275, 294], [291, 379, 386, 606, 384], [416], [519, 164], [419], [224, 292, 269], [382, 385, 590], [223, 380], [297, 383]	N/A
7	21,519,277,419, 417,291,276	11	[277, 418], [417, 288, 275, 294], [291, 379, 386, 606, 384], [276], [416], [519, 164], [419], [224, 292, 269], [382, 385, 590], [223, 380], [297, 383]	N/A
8	21,519,277,419, 417,291,276,416	12	[277, 418], [417, 288, 275, 294], [291, 379, 386, 384, 606], [276], [416, 292, 293], [21], [519, 164], [419], [224, 269], [382, 385, 590], [223, 380], [297, 383]	N/A
9	21,519,277,419, 417,291,276,416,590	12	[277, 418], [417, 294, 288, 275], [291, 379, 386, 384, 606], [590, 383, 382], [276], [416, 292, 293], [21], [519, 164], [419], [224, 269], [223, 380], [297, 385]	N/A