UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

PhD Program in Signal Theory and Communications

Contribution to the Development of Wi-Fi Networks through Machine Learning based Prediction and Classification Techniques

Author:
Seyedeh Soheila Shaabanzadeh

Thesis Advisor:
Dr. Juan Sánchez-González

Signal Theory and Communications Department
Barcelona, June 2024

Abstract

The growing number of Wi-Fi users and the emergence of bandwidth-intensive services have necessitated an increase in Access Point (AP) density, resulting in more complex network configuration, optimization, and management tasks. Concurrently, advancements in data monitoring and analytics technologies in wireless networks offer opportunities to extract valuable insights into network and user behavior, facilitating more efficient network management. In this thesis, we propose a general architecture of how our work enhances Wi-Fi network management using machine learning techniques, focusing on three aspects: user connectivity prediction, Wi-Fi traffic prediction, and Wi-Fi traffic classification.

The first aspect of our work involves extracting knowledge related to the next AP of users. We present a methodology to predict the AP a user will connect to in a Wi-Fi network based on historical AP connections. Various approaches are proposed to capture different time-based (i.e. hourly, daily, weekly) user activity behaviors, comparing two prediction algorithms, Neural Networks and Random Forest. This methodology is evaluated using real data from a large Wi-Fi network deployed on a university campus. Predicting the next AP of users in Wi-Fi network enables a proactive approach to network reconfiguration, enhancing efficiency. For instance, it aids in optimizing Pairwise Master Key caching and Opportunistic Key Caching techniques, reducing re-authentication time when users switch between APs. Furthermore, the prediction of the next AP to which the User Equipment (UE) will connect, allow to identify in which geographical region the UE will be placed. Additionally, the extracted knowledge from next AP prediction of users can be leveraged for commercial purposes. Companies can utilize this information for targeted advertising, customizing messages based on the geographical location of the users, thereby enhancing the effectiveness of their marketing strategies.

Secondly, we introduce a solution for traffic prediction at the AP by proposing a methodology to predict the aggregated traffic of all the UEs connected at the AP. This methodology leverages spatial and temporal correlations of traffic handled by neighboring APs to improve prediction accuracy. Real measurements are used to evaluate the proposed methodology. We assess various Deep Learning methods, including Convolutional Neural Network (CNN), Simple Recurrent Neural Network (SRNN), Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM), and Transformer, for both prediction approaches. Furthermore, a hybrid prediction methodology that combines spatial processing using CNN and temporal prediction using RNN is presented. This hybrid approach enhances prediction accuracy with minimal increase in Training Computational Time and negligible impact on Prediction Computational Time. Forecasting future values of traffic in each AP enables a more accurate characterization of the load space/time distribution among the APs. On the one hand, this knowledge can be used as input for different resource management techniques such as admission control, congestion control, load balancing, etc. On the other hand, the prediction of a very low traffic at specific APs at certain periods of time can be a useful input for energy saving techniques (e.g. switching off APs with near-zero traffic at some specific time periods).

Finally, traffic classification stands as another crucial element in enhancing network performance. By being able to classify different kinds of services, it allows for better resource allocation and service prioritization, giving higher priority to specific services with stringent latency requirements. The increase in demand for eXtended Reality (XR)/Virtual Reality (VR) services in recent years presents a significant challenge for Wi-Fi networks in meeting strict latency requirements. Latency is particularly critical in VR over Wi-Fi, as users expect immediate responses to their interactions. Delays can disrupt the user experience, potentially leading to motion sickness and service abandonment. To address this, distinguishing interactive VR traffic from Non-VR traffic in Wi-Fi networks can help reduce latency for VR users, improving Wi-Fi Quality of Service (QoS). In this study, we propose a machine learning-based method for identifying interactive VR traffic in a Cloud-Edge VR environment. We analyze the correlation between downlink and uplink data and extract features from single-user traffic characteristics. We compare six common classification techniques (i.e., Logistic Regression, Support Vector Machines, k-Nearest Neighbors, Decision Trees, Random Forest, and Naive Bayes) to create an effective model. Using datasets generated by different VR applications, including both single and multi-user cases, and Wi-Fi network simulations, we assess the efficacy of our approach in enhancing VR traffic identification and prioritization. Our results demonstrate a notable reduction in VR traffic delays compared to scenarios without prioritization, with minimal impact on background (BG) traffic latency related to Non-VR services.

Resumen

El creciente número de usuarios Wi-Fi y el aumento en la demanda de servicios de alta capacidad ha propiciado un aumento de la densidad de puntos de acceso, dando lugar a procesos de configuración, optimización y gestión de red más complejos. Por otro lado, el reciente desarrollo en tecnologías de monitorización y análisis de datos en redes inalámbricas, ofrece la oportunidad de extraer información relevante acerca del comportamiento de la red y los usuarios, facilitando una gestión de red más eficiente. En esta tesis, se propone una descripción general de como este trabajo mejora la gestión de sistemas Wi-Fi mediante el uso de técnicas de Machine Learning, poniendo el foco en tres aspectos principales: predicción de conectividad de usuario, predicción de tráfico y clasificación de tráfico.

El primer aspecto de este trabajo consiste en la extracción de conocimiento relacionado con el próximo punto de acceso asociado a cada usuario. Este trabajo presenta una metodología para predecir el punto de acceso al cual un usuario se conectará basado en su histórico de conexiones. Se proponen varias alternativas para capturar comportamientos de actividad de los usuarios en diferentes escalas temporales (horaria, diaria, semanal). Por otro lado, se comparan dos algoritmos de predicción, uno basado en redes neuronales y el otro basado en técnicas de Random Forest. Estas metodologías se evalúan mediante el uso de medidas reales de una red Wi-Fi desplegada en un campus universitario. La predicción del próximo punto de acceso permite una reconfiguración de red de manera proactiva, mejorando la eficiencia. Por ejemplo, la predicción del próximo punto de acceso al cual se conectará el usuario, permite la optimización de técnicas de Pairwise Master Key caching y Opportunistic Key Caching, reduciendo el tiempo de autenticación cuando los usuarios cambian de punto de acceso. Por otro lado, este tipo de predicciones permiten identificar la región geográfica en la cual se ubicará el usuario. Esta información puede ser aprovechada con fines comerciales. Empresas pueden utilizar esta información para publicidad dirigida, mensajes personalizados, etc. permitiendo mejorar sus estrategias de marketing.

En segundo lugar, se presenta una solución de predicción de tráfico en un punto de acceso mediante la propuesta de una metodología de predicción del tráfico agregado por todos los usuarios conectados al punto de acceso. Esta metodología, hace uso de correlaciones espaciales y temporales de tráfico de puntos de acceso vecinos con el objetivo de mejorar la predicción. Para evaluar la metodología propuesta, se han utilizado medidas reales. Este trabajo evalúa diferentes metodologías de Deep Learning, incluyendo técnicas como Convolutional Neural Network (CNN), Simple Recurrent Neural Network (SRNN), Gate Recurrent Unit (GRU), Long Short Term Memory (LSTM) y Transformers. Además, se presenta una metodología de predicción híbrida que combina procesado espacial mediante CNN y procesado temporal mediante RNN. Esta metodología híbrida mejora la predicción obtenida a costa de un reducido incremento en el tiempo computacional de entrenamiento y un impacto despreciable en el tiempo computacional de predicción. La predicción de futuros valores de tráfico en un punto de acceso permite una caracterización más precisa de la distribución de la carga entre puntos de acceso en el espacio y en el tiempo. Por un lado, esta información puede ser utilizada por diferentes técnicas como el control de admisión, control de congestión, balance de tráfico, etc. Por otro lado, la predicción de valores reducidos de tráfico en algunos puntos de acceso en ciertos instantes de tiempo puede ser útil para aspectos de Energy Saving (por ejemplo, para la desconexión de puntos de acceso con tráfico cercano a cero en algunos periodos de tiempo).

Finalmente, las técnicas de clasificación de tráfico son otro aspecto crucial para la mejora del funcionamiento de la red. La capacidad de clasificar diferentes tipos de tráfico permite una mejor asignación de recursos y priorización de servicios, proporcionando mayor prioridad a servicios con requerimientos estrictos de retardo. El aumento de la demanda de servicios de realidad virtual/extendida en estos últimos años supone un reto para las redes Wi-Fi actuales para satisfacer estos requerimientos estrictos de retardo. El retardo es particularmente crítico en servicios de realidad virtual sobre redes Wi-Fi, dado que los usuarios necesitan respuestas inmediatas a sus interacciones. Un retardo elevado puede degradar la calidad percibida por el usuario puede provocar mareo y abandono del servicio. Para solucionar esto, la diferenciación de tráfico de realidad virtual del resto de tráfico puede reducir la latencia del tráfico de realidad virtual, mejorando la calidad del servicio. En este estudio, se propone una metodología basada en Machine Learning para identificar tráfico de realidad virtual en un entorno Cloud Edge VR. Esta metodología analiza la correlación entre los datos transmitidos en los dos sentidos de la comunicación para extraer ciertas características de tráfico. Se han comparado seis técnicas de clasificación (Logistic Regression, Support Vector Machines, k-Nearesst Neighbors, Decision Tree, Random Forest y Naive Bayes). Mediante el uso de datasets generados por diferentes aplicaciones de VR (tanto para un único usuario como para casos multi-usuario) y mediante simulaciones de la red Wi-Fi, se ha evaluado la eficacia de la metodología mejorando la clasificación y priorización de tráfico. Los resultados obtenidos demuestran una reducción notable de los retardos en el tráfico de realidad virtual, en comparación con escenarios sin priorización, con un impacto mínimo en el retardo del tráfico relacionado con el resto de servicios.

Acknowledgement

First and foremost, I extend my deepest gratitude to my thesis advisor, Juan Sánchez-González, whose invaluable guidance and insightful feedback have significantly enhanced the quality of this dissertation. I am privileged to have been mentored by him during my doctoral studies, and I am sincerely thankful for his unwavering support and dedication to my academic development.

Special appreciation goes to Prof. Boris Bellalta at Universitat Pompeu Fabra, whose guidance and support have been instrumental in shaping my doctoral journey and fostering my growth as a researcher. The nine months of collaboration with him have been exceptionally rewarding.

I am also indebted to Costas Michaelides for his patience, motivation, and profound expertise, which have greatly contributed to the progress of this thesis. Additionally, my gratitude extends to Marc Carrascosa-Zamacois for his willingness to address my technical inquiries and share his knowledge, enriching this work with valuable insights.

Finally, I express my heartfelt gratitude to my late father, whose unwavering encouragement ignited my passion for learning and inspired me to pursue higher education. I am deeply grateful to my mother, sister, brother and friends for their enduring love and steadfast support throughout this transformative journey. Their encouragement has been a constant source of strength and inspiration, sustaining me through challenges and triumphs alike.

Contents

List of Figures

List of Tables

Abbreviations

ABP	Adaptation-Based Programming
Ack	Acknowledgement
ADDFS	Adaptive Distribution Distance-based Feature Selection
AF	Activation Function
AI	Artificial Intelligence
AIFS	Arbitration Inter-Frame Space
AISVM	Authenticator Incremental Support Vector Machine
ALVR	Air Light Virtual Reality
A-MPDU	aggregate-MAC Protocol Data Unit
A-MSDU	aggregate-MAC Service Data Unit
ANN	Artificial Neural Network
AP	Access Point
AR	Augmented Reality
ARF	Auto Rate Fallback
ARIMA	Autoregressive Integrated Moving Average
ARMA	Autoregressive Moving Average
ARQ	Automate Repeat Request
BG	Background
BS	Base Station
BSS	Basic Service Set
CAGR	Compound Annual Growth Rate
CARA	Collision-Aware Rate Adaptation

CDF	Cumulative Distribution Function
CG	Cloud Gaming
CMSVM	Cost Sensitive Support Vector Machine
CNN	Convolutional Neural Network
COTS	Commercial off-the-shelf
CSI	Channel State Information
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
CT	Computational Time
CTS	Clear to Send
CW	Contention Window
DBCA	Dynamic Bandwidth Channel Access
DCB	Demand Channel Bonding
DCF	Distributed Coordination Function
DDPG	Deep Deterministic Policy Gradient
DIFS	Distributed Inter-Frame Space
DL	Deep Learning
DNN	Deep Neural Network
DPI	Deep Packet Inspection
DPP	Determinantal Point Process
DQL	Deep Q-Learning
DQN	Deep Q-Network
DRL	Deep Reinforcement Learning
DSL	Deep Supervised Learning
DT	Decision Tree
EDCA	Enhanced Distributed Channel Access
EHT	Extremely High Throughput
EIED	Exponential-Increase Exponential-decrease

EM	Expectation Method
EMA	Expectation Modification Algorithm
EMLSR	Enhanced Multi-Link Single-Radio
ETS	Exponential Smoothing
FCS	Frame Check Sequence
FER	Frame Error Rate
FIFO	First In First Out
FL	Federated Learning
fps	frames per second
GBRT	Gradient Boosted Regression Tree
GNN	Graph Neural Network
GP	Gaussian Process
GRU	Gated Recurrent Unit
HA-DBCA	Hybrid Adaptive-Dynamic Bandwidth Channel Access
HARQ	Hybrid Automatic Repeat Request
HMM	Hidden Markov Model
HMD	Head Mounted Display
IAT	Inter-Arrival Time
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
iQRA	intelligent Q-learning based Resource Allocation
ISVM	Incremental Support Vector Machine
ITE	Iterative Trial and Error
JBP	Joint Based Prediction
kNN	k-Nearest Neighbours
LiBRA	Learning-based Beam and Rate Adaptation
LMT	Logistic Model Tree
LoS	Line of Sight

LR	Logistic Regression
LSTM	Long Short-Term Memory
MAB	Multi-Armed Bandit
MAC	Medium Access Control
MADDPG	Multi-Agent Deep Deterministic Policy Gradient
MAE	Mean Absolute Error
MDP	Markov Decision Process
MFNN	Multi-layer Feed-forward Neural Network
MH-GAN	Metropolis Hastings Generative Adversarial Network
ML	Machine Learning
MLO	Multi-Link Operation
MLP	Multi Layer Perceptron
mm-Wave	millimeter-Wave
MPDU	MAC Protocol Data Unit
MR	Mixed Reality
MSDU	MAC Service Data Unit
MSE	Mean Square Error
Nack	Negative Acknowledgement
NB	Naive Bayes
NBKDE	Naive Bayes Algorithm with Kernel Density Estimation
NCA	Normalized Channel Access
NDP	Neighbour Discovery Protocol
NFV	Network Function Virtualization
NLoS	Non-Line of Sight
NN	Neural Network
OBSS	Overlapping Basic Service Set
OFDMA	Orthogonal Frequency Division Multiple Access
OKC	Opportunistic Key Caching
PBDP	Prediction Based on Daily Patterns

PBTP	Prediction Based on Time-period Patterns
PBWP	Prediction Based on Weekly Patterns
PCT	Prediction Computational Time
PDS	Post-Decision State-based
PHY	Physical
PMK	Pairwise Master Key
PNN	Probabilistic Neural Network
QAM	Quadrature Amplitude Modulation
QNN	Q Neural Network
QoE	Quality of Experience
QoS	Quality of Service
REPT	Reduced Error Pruning Tree
RF	Random Forest
RFR	Random Forest Regressor
RL	Reinforcement Learning
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
RSSI	Received Signal Strength Indicator
RTS	Request to Send
SBCA	Static Bandwidth Channel Access
SC	Small Cell
SD	Software Defined
SDN	Software Defined Network
SDR	Software Defined Radio
SGI	Short Guard Interval
SIFS	Short Inter-Frame Spacing
SINR	Signal-to-Interference-plus-Noise-Ratio
SL	Supervised Learning
SLA	Stochastic Learning Automata

SLO	Single-Link Operation
SNR	Signal-to-Noise Ratio
SR	Spatial Reuse
SRNN	Simple Recurrent Neural Network
SS	Spatial Stream
SSID	Service Set Identifier
STA	Station
SVM	Support Vector Machine
SVR	Support Vector Regression
TCT	Training Computational Time
TS	Thomson Sampling
TWT	Target Wake Time
TXOP	Transmission Opportunity
UCB	Upper Confidence Bound
UDN	Ultra Dense Network
UE	User Equipment
UHR	Ultra High Reliability
USL	Unsupervised Learning
VANET	Vehicular Ad-hoc Network
VNF	Virtual Network Function
VR	Virtual Reality
WLAN	Wireless Local Area Network
WLC	Wireless LAN Controller
XGBoost	eXtreme Gradient Boosting Tree
XR	eXtended Reality

Chapter 1

Introduction

1.1 Motivation and Objectives

As Wi-Fi networks continue to advance, driven by updates within the 802.11 protocol family and the introduction of new features, they become more intricate and multifaceted. This complexity is particularly pronounced in densely deployed scenarios and shared frequency bands, where multiple Access Points (APs) operate in close proximity. Traditional optimization methods struggle to effectively manage the myriad challenges associated with resource allocation, network configuration, and Quality of Service (QoS) provisioning in such environments.

One of the primary challenges is the efficient management of resources amidst the growing demands for high-speed, reliable wireless connectivity. With the proliferation of wireless devices and the increasing reliance on Wi-Fi for various applications, network administrators face mounting pressure to ensure optimal performance and user satisfaction. However, the dynamic and unpredictable nature of network traffic poses a significant hurdle, making it challenging to forecast and respond to changing demands effectively.

Moreover, the need for proactive network management strategies becomes increasingly imperative as network complexity grows. Predictive insights into future network conditions, such as user mobility patterns and traffic load fluctuations, are essential for optimizing network performance and resource utilization. Predicting how users will move within the network and how traffic will vary over time allows for better allocation of network resources and more effective management of network traffic. For instance, predicting user mobility patterns helps in predicting areas of high user concentration, allowing network administrators to allocate resources accordingly and maintain QoS. Similarly, forecasting traffic load fluctuations enables administrators to predict periods of high demand and allocate additional resources as needed to ensure smooth network operation. The recent development of network monitoring tools allows for the collection of extensive measurements to evaluate the network status and performance accurately. Additionally, advancements in Big Data technologies enable the management of large amounts of historical information and measurements efficiently. Leveraging these developments, sophisticated Machine Learning (ML) techniques are employed to process this vast amount of information and extract actionable insights. These Artificial Intelligence and Machine Learning (AI/ML) techniques play a crucial role in obtaining knowledge that aids in making better proactive decisions in the network, thereby enhancing overall network efficiency and performance.

Another key challenge lies in ensuring seamless connectivity and QoS for latency-sensitive applications, such as interactive Virtual Reality (VR) traffic. The integration of VR applications into daily activities necessitates prioritizing and efficiently handling VR traffic to deliver a seamless user experience. However, accurately differentiating between various types of network traffic, including VR and Non-VR traffic, presents a formidable challenge. Traditional methods may struggle to discern between different traffic types accurately, leading to suboptimal resource allocation and degraded performance for latency-sensitive applications.

In this context, the motivations of the thesis revolve around leveraging ML-driven solutions to address the challenges of modern Wi-Fi networks, enhance network performance, optimize network management, and meet the evolving demands of emerging technologies. By proposing innovative approaches to network optimization and management, the thesis aims to pave the way for more efficient and reliable wireless communication systems.

Therefore, the main objective is to propose different ML-driven solutions with the aim to improve the network performance and network management. These proposed solutions cover aspects of prediction of User Equipment (UE) connectivity, AP traffic prediction, and Wi-Fi traffic classification. The three specific objectives are defined to achieve this main objective:

1. Prediction of the future AP that the UE will connect to: Wi-Fi communication systems often exhibit specific patterns in user mobility, showcasing a variety of periodic movements influenced by factors such as daily, weekly, and seasonal variations. These periodicities and seasonal trends play a crucial role in shaping user mobility behaviors within Wi-Fi networks. Understanding these patterns, provides valuable insights into user movement dynamics. Predicting the future AP that the UE will connect to is indeed useful for various aspects of enhancing network performance. For instance, by capturing weekly, daily, and hourly user activity-based behaviors, the predictive approach allows for effective monitoring of user mobility within the network. This enables network administrators to identify periods with high user density, where significant traffic management may be required. Additionally, predicting the next AP for each User Equipment (UE) (prediction at user level) facilitates a proactive network reconfiguration approach. In Wi-Fi networks, this prediction can enhance techniques such as Pairwise Master Key (PMK) caching and Opportunistic Key Caching (OKC), reducing re-authentication time when roaming to a new AP. PMK caching involves storing security credentials between a client device and an AP, while OKC pre-authenticates a client with nearby APs to expedite roaming. Additionally, the extracted knowledge related to user location, mobility, habits, and interests holds commercial value. For example, predicting future user location and connection duration enables targeted Location Based Advertising, allowing advertisers to personalize messages based on user location and interests. Moreover, predictive analytics adjust configurations such as the transmitted power and the selected channel according to the prediction of the next AP to which the UE will connect.

   In this regard, the contributions in the context of the user connectivity prediction are in the following:

   - The proposal of a prediction methodology that is able to extract user periodical patterns at different time levels in order to capture hourly, daily and weekly user activity-based behaviours. In this prediction methodology, three approaches are defined to create each pattern (i.e. Prediction Based on Time-period Patterns, Prediction Based on Daily Patterns, Prediction Based on Weekly Patterns). Then, an approach with a combination of the three previous approaches is formulated to be able to provide better prediction accuracy than the rest of the approaches, separately, with a relatively low computational time (CT) per user.

- Two prediction algorithms based on a supervised learning process are compared, one using a Neural Network (NN) and the other one based on Random Forest (RF). This comparison is based on prediction accuracy and CT among different approaches. Moreover, an extensive evaluation for the best approach is done to illustrate the impact of the amount of data used for training.
- The proposed methodology is evaluated for a large Wi-Fi network with 429 APs in a University Campus with 33 buildings with four floors per building.

2. Prediction of future values of traffic load for a given AP in a Wi-Fi network: The proposed methodology can indeed be highly beneficial for various aspects of network management and performance optimization. For example, (i) predicting future network traffic can help in deciding whether to add or relocate APs in areas where there is expected to be high traffic load. By strategically placing APs based on predicted traffic, performance can be optimized. While immediate physical relocation or addition of APs is impractical with a next medium-term predictions (next few minutes) traffic prediction window, these predictions enable network administrators to make short-term adjustments to optimize performance. Furthermore, patterns identified through these short-term predictions contribute valuable data for long-term traffic characterization, ultimately guiding strategic decisions on permanent AP placement and infrastructure development. (ii) Predicting peaks of traffic at certain APs allows for proactive channel and band selection to mitigate potential congestion and performance degradation. For example, an AP may work at 2.4 and 5GHz bands. In case that the traffic/load (or number of UEs) in 2.4GHz band is too high, then some users that are capable of working in 5GHz band can be moved to this band to avoid congestion in 2.4GHz. (iii) Predictive insights into future network conditions may help to improve the performance of some techniques such as admission control, congestion control or load balancing. APs can distribute traffic more evenly across available channels and resources, ensuring optimal utilization and minimizing the risk of overloads or bottlenecks. (iv) the prediction of large periods with very low traffic at some APs can be useful for the energy-saving aspect. In this situation, these APs are switched off to reduce energy consumption. Users in these areas should connect to neighbouring APs.

   The contributions of the network prediction methodology are the following:

   - Proposal of an adaptive and context-aware prediction framework that is able to predict future values of a given parameter in a target AP (e.g. future traffic values) based on the historical values of this metric for this target AP and its neighbouring APs. First, the methodology runs spatio-temporal correlations among the target AP and its neighbours making use of historical collected measurements. In case that any of the neighbouring APs does not exhibit a high correlation with the target AP for the specific metric, then, the prediction is based only on the historical previous measurements of the target AP. In case that a high correlation is observed between the target

AP and some of its neighbouring APs, the historical measurements collected at these highly correlated neighbouring APs are included in the prediction process with the aim of increasing the prediction accuracy. By intelligently selecting whether to include information from neighbouring APs based on identified spatial correlations, our methodology overcomes the challenge of incorporating spatial dynamics into Wi-Fi metric prediction.

- In order to achieve a more comprehensive and data-driven solution to Wi-Fi metric prediction, different prediction algorithms are compared in terms of prediction accuracy and CT. In particular, four Deep Learning (DL) based algorithms are considered, namely, Long short-term memory (LSTM), Gated Recurrent Unit (GRU) and Convolutional Neural Network (CNN) and Transformer. On the other hand, three hybrid DL algorithms that combine CNN and Recurrent Neural Network (RNN) are also evaluated: CNN-SimpleRNN, CNN-LSTM, and CNN-GRU. To the best authors' knowledge, this is the first time that different spatio-temporal prediction-based methodologies are evaluated and compared for traffic prediction in a Wi-Fi network.
- The proposed methodology is evaluated in a real Wi-Fi network with measurements collected in 100 APs during three months. This addresses the challenge of scalability in designing prediction systems for Wi-Fi networks, ensuring that the methodology can effectively handle larger network environments without compromising performance.
- The proposed methodology is particularized for the prediction of future values of traffic at the AP and the prediction of transmission failures, but the methodology could be easily extended for other metrics.

3. Traffic classification through ML Techniques: Traffic classification is essential for enhancing Wi-Fi network performance and QoS by accurately distinguishing between different types of network traffic, such as interactive VR traffic and Non-VR traffic. The increasing use of VR applications necessitates prioritizing VR traffic to ensure a seamless user experience, as VR demands low-latency communication to avoid issues like motion sickness and user discomfort. Prioritizing traffic with strict latency requirements, like VR, is crucial because delays can significantly impact the user experience. ML-driven traffic classification techniques are pivotal in enabling Wi-Fi networks to prioritize critical traffic types and optimize resource allocation. By ensuring low-latency guarantees for latency-sensitive applications, networks can deliver consistent performance for demanding applications like VR. The classification results from ML models provide valuable insights for network management, enabling proactive measures to mitigate potential congestion and performance degradation. Prioritizing latency-sensitive traffic ensures that critical applications receive the necessary resources, maintaining high QoS standards. Accurate traffic classification allows for dynamic traffic shaping and policy enforcement, where Wi-Fi networks can adjust QoS policies in real-time based on the classification results. This dynamic adjustment ensures that mission-critical traffic types, such as interactive VR traffic, are prioritized over less critical traffic. By focusing on real-time prioritization of latency-sensitive traffic, Wi-Fi networks can optimize resource utilization, ensuring that users experience consistent and reliable QoS across various network conditions and usage scenarios.

   In particular, the key contributions of the proposed approach are as follows:

- Extraction of statistical features computed from single-user traffic characteristics, over a time interval of a certain duration. Among these statistical features, correlation between downlink and uplink traffic, computed during certain period of time, emerges as a valuable feature for distinguishing VR from Non-VR traffic.
- A binary classification ML model is proposed to identify interactive VR traffic. In this model, a heterogeneous traffic dataset is used with statistical features computed from single-user traffic characteristics over certain duration of time interval.
- Comparison and evaluation of six ML classification models (i.e., Logistic Regression (LR), Support Vector Machines (SVM), k-Nearest Neighbors (kNN), Decision Tree (DT), RF, and Naive Bayes (NB)). For each classifier, a process involving hyperparameter tuning and feature selection is done.
- The most accurate model is obtained among different considered settings. This model is evaluated on packet traces from a multi-user experimental setup involving three users, and using single-user traces from a VR framework not included in the training traces. In addition, Wi-Fi QoS is enhanced by reducing VR traffic delay through prioritizing VR traffic. The proposed prioritization methodology has been evaluated by means of a Wi-Fi simulator.

1.2 Thesis Structure

This thesis is organized in 6 chapters, whose structure is graphically summarized in Fig. 1.1.

Chapter 2 presents some necessary background information on (i) Wi-Fi system overview, evolution towards Wi-Fi networks, Wi-Fi 7 and Wi-Fi 8 features, Wi-Fi architecture, the 802.11 MAC protocol, and mobility management, (ii) a state of the art description related to the applicability of ML techniques in different areas of Wi-Fi networks that includes essential Wi-Fi features such as QoS improvement by traffic prioritization, newer Wi-Fi features, Wi-Fi connectivity management such as next user connectivity prediction, Wi-Fi traffic prediction, and other types of Wi-Fi scenarios, and (iii) a description of the ML techniques used in the thesis. Moreover, we provide a comprehensive literature review on the application of ML in addressing these network problems in this Chapter. Finally, according to this background and our work, a general AI/ML control loop utilized in our study is presented in a functional framework.

Chapter 3 treats the next AP prediction by proposing a methodology based on different approaches. The problem of next AP prediction is first introduced and, accordingly, the general architecture of Chapter 2 Section 2.4 is particularized for the ML-based next AP prediction. In this methodology, different approaches are formulated to capture hourly, daily and weekly user activity-based behaviours, while a joint solution combining the three aforementioned approaches is presented. Furthermore, the performance of the proposed methodology with the different approaches is evaluated using a scenario of large Wi-Fi network deployed in a University Campus. Some valuable results are obtained including an example of the next AP prediction methodology, comparison of the different approaches based on prediction accuracy, and CT per user for each ML technique, the impact of the amount of data used for training, and the impact of the size of the sliding window.

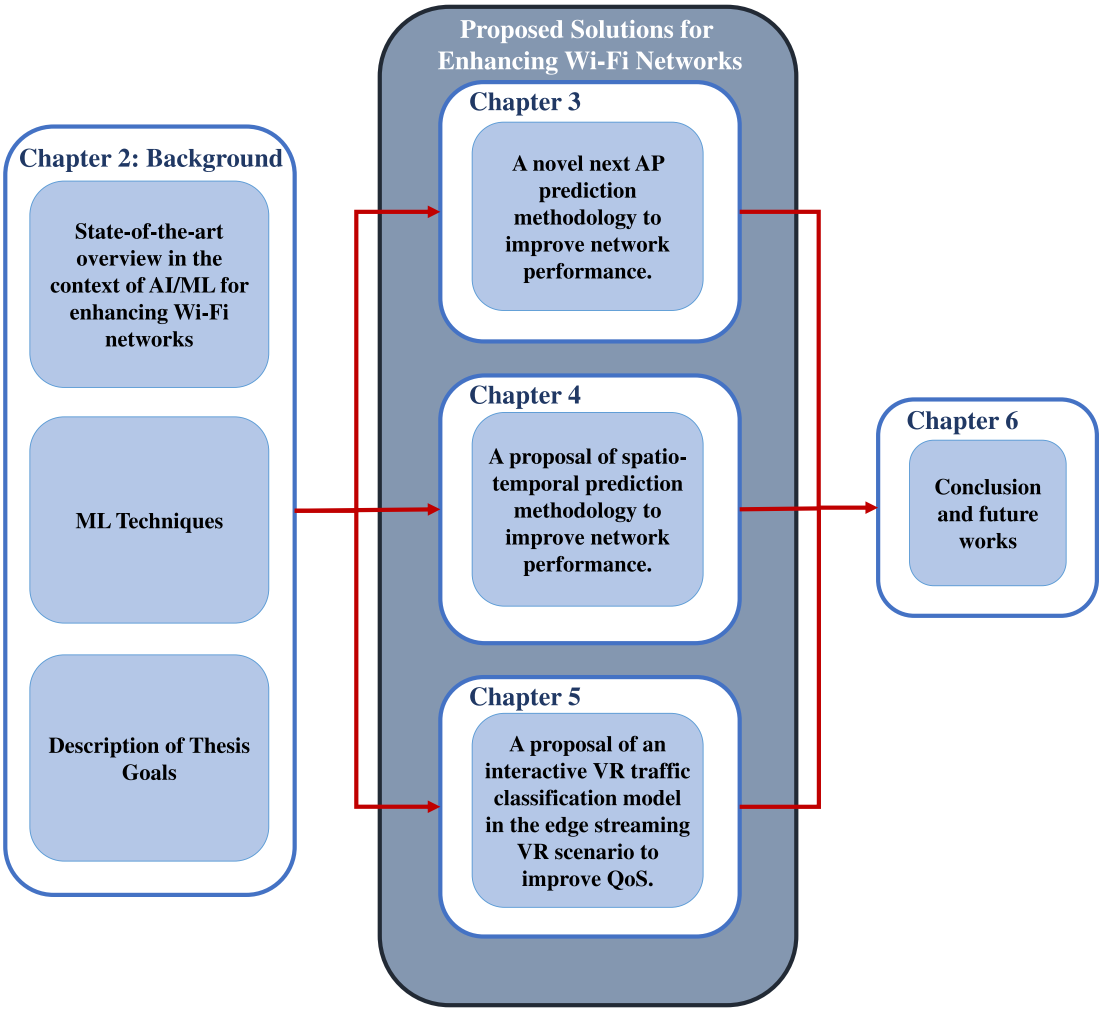

Proposed Solutions for Enhancing Wi-Fi Networks
Chapter 2: Background
State-of-the-art overview in the context of AI/ML for enhancing Wi-Fi networks
ML Techniques
Description of Thesis Goals
Chapter 3
A novel next AP prediction methodology to improve network performance.
Chapter 4
A proposal of spatio-temporal prediction methodology to improve network performance.
Chapter 5
A proposal of an interactive VR traffic classification model in the edge streaming VR scenario to improve QoS.
Chapter 6
Conclusion and future works

Fig. 1.1. Structure of the thesis.

In Chapter 4, based on general architecture outlined in Chapter 2, Section 2.4, we delve into traffic prediction using a proposed methodology. After introducing the challenge, the chapter presents our prediction methodology, which is based on a specific architecture designed to predict future values of network traffic load within a target AP using historical data from the target AP and its neighbouring APs. Initially, we estimate the correlation between each target AP and its neighbours, determining our prediction approach as either only temporal or spatio-temporal. To evaluate the methodology, we consider a scenario involving multiple APs within a real university network. We analyze various results obtained from temporal and spatio-temporal predictions using both single and hybrid time series DL methods. Lastly, we discuss the implementation aspects of our methodology in a real network setting, including the definition and consideration of Key Performance Indicators (KPIs) relevant to the specific network environment and operational requirements. These KPIs, such as Training Computational Time (TCT), Prediction Computational Time (PCT), and prediction accuracy, are crucial for assessing the performance of prediction methods in real-world scenarios.

Chapter 5 introduces the proposal of an interactive VR ML traffic classification model within the edge streaming VR scenario. Initially, the chapter delves into the challenge of VR traffic identification within Wi-Fi networks. Subsequently, it analyzes dataset creation, encompassing the network setup for data generation, data collection procedures including traffic analysis for both VR and Non-VR, and feature engineering methods for extracting and selecting features to enhance accuracy. Following this, the chapter specifies the framework detailed in Chapter 2, Section 2.4, for the ML-driven traffic classification model. It also provides a comprehensive description of common hyperparameters for each classifier utilized in the study. After that, the results obtained from both the feature extraction and traffic classification phases are analyzed. Subsequently, the performance of six common classifiers is compared to assess the effectiveness of the model in traffic classification. Finally, the classification model undergoes testing using packet traces from three users engaged in VR gaming within a multi-user experimental setup, and using single-user traces from a VR framework not included in the training traces. Additionally, a network simulator is utilized to evaluate the impact of the VR traffic identification model on the existing Wi-Fi network.

Finally, Chapter 6 discusses the conclusions of the presented work and outlines the potential directions for future investigations.

Chapter 2

Background

2.1 Wi-Fi System Overview

The IEEE 802.11 standard [1], commonly known as WiFi, defines the architecture and specifications for wireless LANs (WLANs), enabling devices to connect without cables using high-frequency radio waves. Wi-Fi plays a crucial role in providing users with quick and convenient wireless internet access. The widespread adoption of Wi-Fi-enabled devices, projected to be three times the global population, reflects the technology's success [2]. In 2022, 3.8 billion Wi-Fi products were shipped [3], and it was estimated that by 2023, there will be 628 million public Wi-Fi hotspots, a significant increase from 169 million in 2018 [2]. One recent study highlights Wi-Fi's substantial contribution to the global economy, projected to reach $3.3 trillion in 2021 and potentially $4.9 trillion in 2025, when considering a wide range of factors including business and consumer connectivity needs, technology research and development, spectrum access, and wider macroeconomic impacts [4]. Over the years, Wi-Fi data rates have remarkably increased, from 1 Mbps in the first generation to a theoretical peak of nearly 30 Gbps in the latest products, offering affordable high-speed wireless services in unlicensed spectrum bands [5]. The development of Wi-Fi technologies is guided by two primary organizations, (i) the Institute of Electrical and Electronics Engineers' (IEEE) 802 Committee, setting standards for essential Wi-Fi technologies focusing on Medium Access Control (MAC) and Physical Layer (PHY) protocols for Wireless Local Area Networks (WLAN) [6, 7], and (ii) the Wi-Fi Alliance, responsible for ensuring interoperability, security, and reliability through Wi-Fi product certification and promoting Wi-Fi evolution[8, 9].

2.1.1 Evolution towards Wi-Fi networks

Wi-Fi always strive to get better and provide a great user Quality of Experience (QoE). Wi-Fi, a highly successful wireless technology, keeps evolving to improve performance, use the available spectrum more efficiently, reduce costs, and most importantly, make the user QoE even better. Alongside 5G, Wi-Fi ensures that more people stay connected, including those who are not yet connected. That is why, even when Wi-Fi 6 became available in 2019, experts were already working on the next version, Wi-Fi 7, as part of the IEEE 802.11be Extremely High Throughput (EHT) working group. The QoS and QoE were improved with the advent of a new Wi-Fi generation. Moreover, the the advanced network features aim to decrease latency, particularly in congested environments, and support real-time applications like gaming, eXtended Reality (XR) including Virtual Reality (VR), and Industrial Internet of Things (IoT).

Wi-Fi 5 (802.11ac) [10], introduced in 2013, marked a significant advancement in wireless technology, succeeding the previous Wi-Fi 4 generation. It responded to the growing demand for robust wireless connectivity, offering improved data rates, reduced latency, increased reliability, and enhanced security in both consumer and enterprise environments. Wi-Fi 5 accommodated the needs of smart homes, IoT, and Industrial IoT applications. Key features included data speeds of up to 3.5 Gbps, 160 MHz channel bandwidth, operation in the 2.4 GHz and 5 GHz bands, enhanced network efficiency with technologies like Orthogonal Frequency Division Multiple Access (OFDMA) and Multi-user Multiple-Input Multiple-Output (MU-MIMO), improved spectrum efficiency, and support for high-density deployments. Wi-Fi 6 (802.11ax) [11], launched in 2019, represented a substantial leap forward from Wi-Fi 5. Driven by increased reliance on Wi-Fi and demands for higher data rates, lower latency, reliability, and security, Wi-Fi 6 excelled in supporting applications in smart homes, IoT, and industrial settings. Key features included data speeds up to 9.6 Gbps, 160 MHz channel bandwidth, introduction of the 6 GHz band (Wi-Fi 6E), enhanced network efficiency with OFDMA, MU-MIMO, and transmit beamforming, improved spectrum efficiency, and the introduction of target wake time (TWT) for enhanced network efficiency and prolonged device battery life.

2.1.2 Wi-Fi 7 and Wi-Fi 8 features

Wi-Fi 7 (802.11be) [12] is currently in the standardization process, expected to be finalized by 2024, with deployment planned shortly after in unlicensed spectrum bands [13]. The commercially recognized Wi-Fi 7 introduces significant technical improvements to achieve Extremely High Throughput (EHT), resulting in higher data rates and lower latency. These enhancements encompass the utilization of 320 MHz of channel bandwidth, higher modulation and coding schemes, efficient use of noncontiguous spectrum through multiple resource unit allocation, and multi-band/multi-channel aggregation and operation. Additionally, there are improvements in QoS management, including the implementation of a restricted target wake time [14]. With a theoretical peak data rate exceeding 40 Gbps, Wi-Fi 7 offers a substantial advancement over its predecessor, Wi-Fi 6, which has a peak speed just under 10 Gbps. For an overview of Wi-Fi's evolution, Table 2.1 provides a detailed comparison of current and future radio technical specifications across recent Wi-Fi standards [9].

In comparison to Wi-Fi 6, which initially introduced the spectrum utilization of the 6 GHz band, Wi-Fi 7 enhances performance through the introduction of advanced features such as larger channel bandwidths (up to 320 MHz) and 4K Quadrature Amplitude Modulation (QAM) [15], significantly improving throughput. A key addition in Wi-Fi 7 is the Multi-Link Operation (MLO) framework, allowing Wi-Fi devices to concurrently operate on multiple channels through a single connection [16, 17]. This framework includes variations such as Enhanced Multi-link Single-radio (EMLSR) and Enhanced Multi-link Multi-radio (EMLMR). Studies indicate that in ultra-dense and crowded scenarios, where both available links are often busy, MLO achieves the highest throughput gains by capitalizing on multiple intermittent transmission opportunities, unlike traditional Single-Link Operation (SLO). Furthermore, MLO outperforms SLO in terms of latency by one order of magnitude, making it crucial for applications with latency-sensitive use case requirements [18]. Research also suggests that upgrading to MLO from legacy SLO effectively contains delay while simultaneously increasing traffic [19]. The current timeline for the final amendment of Wi-Fi 7 is expected by 2024, as shown in Fig. 2.1[9].

Table 2.1. Technical capabilities across legacy and current wireless standards [9].

Features	Wi-Fi 4 (802.11n)	Wi-Fi 5 (802.11ac)	Wi-Fi 6/6E (802.11ax)	Wi-Fi 7 (802.11be)	Wi-Fi 8 (speculative)
Peak data rate	600 Mbps	7 Gbps	9.6 Gbps	$\leq$ 46.4 Gbps	$>$ 46.4 Gbps
Carrier Frequency (GHz)	2.4, 5	5	2.4, 5, 6	2.4, 5, 6	2.4, 5, 6 at a minimum
Channel Bandwidth (MHz)	20, 40	20, 40, 80, 160	20, 40, 80,160	Up to 320	$>$ 320
Frequency multiplexing	OFDM	OFDM	OFDM and OFDMA	OFDM and OFDMA	OFDM and OFDMA
OFDM symbol time (μs)	3.2	3.2	12.8	12.8	12.8 at a minimum
Guard interval (μs)	0.4, 0.8	0.4, 0.8	0.8, 1.6, or 3.2	0.8, 1.6, or 3.2	0.8, 1.6, or 3.2
Total symbol time (μs)	3.6, 4.0	3.6, 4.0	13.6, 14.4, 16.0	13.6, 14.4, 16.0	13.6, 14.4, 16.0
Modulation	$\leq$64-QAM	$\leq$256-QAM	$\leq$1024-QAM	$\leq$4096-QAM	$>$4096-QAM
MU-MIMO	N/A	DL	DL and UL	DL and UL	DL and UL
OFDMA	N/A	N/A	DL and UL	DL and UL	DL and UL
MIMO	4×4	8×8	8×8	8×8	16×16

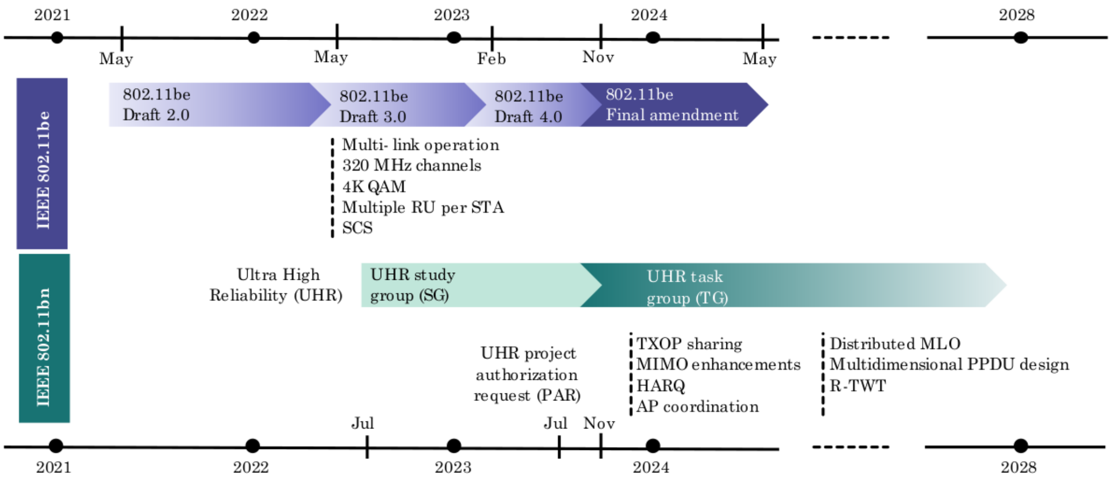


Fig. 2.1. Timeline for Wi-Fi 7 (802.11be) and Wi-Fi 8 (802.11bn) [9].

When 6G becomes available, the wireless industry will have progressed to the eighth generation of Wi-Fi technology, known as Wi-Fi 8 or Ultra High Reliability (UHR). In the context of this forward-looking assessment, the potential features under consideration for the final standard of Wi-Fi 8 include advanced technologies such as higher-order MIMO, Hybrid Automatic Repeat Request (HARQ), AP coordination, and the exploration of higher spectrum bands at 45 GHz and/or 60 GHz [9].

Wi-Fi 8 is designed with UHR as its primary characteristic, in contrast to previous standards that emphasized increasing peak throughput [5]. The key challenge for next-generation Wi-Fi technologies, including Wi-Fi 8, is achieving ultra-low deterministic latency [20]. As illustrated in Fig. 2.1, the standardization cycle for Wi-Fi 8 (see 802.11bn in the figure) is expected to conclude in 2028, with the UHR Study Group established in July 2022 to define protocol functionalities for future products [21]. The four main areas of focus for Wi-Fi 8 include improving throughput at lower Signal-to-Interference-plus-Noise (SINR) ratios, reducing tail latency and jitter, enhancing spectral reuse, and achieving greater power savings and peer-to-peer operations [5, 22].

2.1.3 Wi-Fi Architecture

Fig. 2.2 illustrates the main components of the 802.11 wireless LAN architecture. The core element of this architecture is the basic service set (BSS), which consists of one or more wireless stations (STAs) or hosts, and an AP serving as a central base station. The AP plays a crucial role in the wireless network infrastructure, managing data transmission to and from wireless STAs associated with it. It coordinates the transmission among multiple connected STAs [23]. STAs, which can be devices like smartphones, tablets, laptops, or IoT devices such as sensors and appliances, establish a wireless communication link with a base station or with other wireless STAs. Different wireless link technologies offer varying transmission rates and coverage distances. In the depicted architecture (Fig. 2.2), APs within BSSs connect to a Wireless LAN Controller (WLC), a crucial backend component that manages and coordinates APs within the network. The WLC is typically connected to the LAN, often through an Ethernet connection to the network router or switch. Functioning as a centralized management entity, the WLC oversees tasks such as AP configuration, firmware updates, security policies, and client authentication. It serves as the gateway for all wireless traffic from the APs, handling functions like radio resource management, load balancing, and seamless roaming between APs. Typically, a home network comprises one AP and one router, commonly integrated into a single unit, facilitating the connection of the BSS to the Internet.

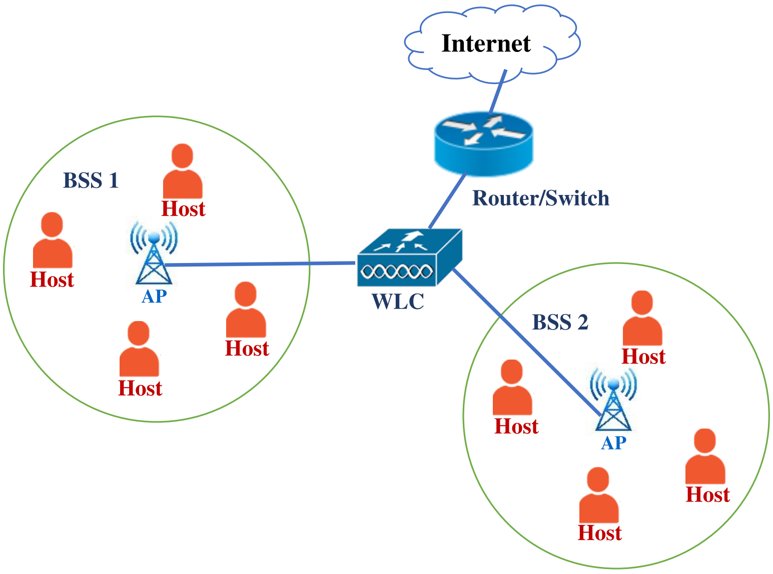


Fig. 2.2. Wi-Fi Architecture.

Similar to Ethernet devices, each 802.11 wireless station possesses a 6-byte MAC address stored in its adapter's (i.e., 802.11 network interface card) firmware. APs also have MAC addresses for their wireless interfaces, administered by the IEEE and intended to be globally unique.

Wireless LANs employing APs are known as infrastructure wireless LANs, where APs, along with the wired Ethernet infrastructure linking them and a router, form the infrastructure. Alternatively, IEEE 802.11 stations can form ad hoc networks, lacking central control and external connections, established dynamically by mobile devices in proximity for communication purposes. Ad hoc networking has garnered significant interest due to the proliferation of portable devices, facilitating communication in scenarios lacking centralized APs.

As mobile stations move between the coverage areas of different base stations, they switch their association to maintain network connectivity, a process known as handoff or handover.

2.1.3.1 Channel and Association

In 802.11, every wireless station must associate with an AP in order to transmit or receive network-layer data. When setting up an AP, the network administrator assigns a one or two-word Service Set Identifier (SSID) to it. Additionally, the administrator assigns a channel number to the AP. In the frequency range of 2.4 GHz to 2.4835 GHz, 802.11 defines 11 partially overlapping channels. Notably, channels 1, 6, and 11 form the only set of three non-overlapping channels. This allows an administrator to create a wireless LAN with a maximum aggregate transmission rate of three times the peak data rate shown in Table 2.1. This can be achieved by installing three 802.11 APs at the same physical location, assigning channels 1, 6, and 11 to them, and connecting each AP to a switch. For instance, in many New York City cafés, where multiple AP signals overlap, known as a Wi-Fi jungle, wireless stations can detect signals from several nearby APs. These APs may belong to the café or nearby residential apartments, each likely assigned to different IP subnets and channels. When a new device enters this Wi-Fi jungle seeking Internet access, it must associate with one AP. Association involves establishing a virtual connection between the device and the chosen AP. Only the associated AP sends data frames to the device, and the device sends data frames to the Internet through this AP.

According to the 802.11 standard, APs periodically broadcast beacon frames containing their SSID and MAC address. Wireless devices scan the 11 channels, looking for these beacon frames. After detecting available APs, the device selects one for association. The standard does not specify the selection algorithm, leaving it to the device's firmware and software designers. Typically, the device chooses the AP with the strongest signal. However, signal strength alone does not guarantee optimal performance, as an AP with many connected devices may offer a strong signal but limited bandwidth. Various alternative AP selection methods have been proposed to address this issue. For example, some methods consider the load on each AP, selecting the one with fewer connected devices to balance traffic and reduce congestion. Another approach involves evaluating the QoS metrics, such as latency and throughput, to choose an AP that provides a better overall user experience.

The process of scanning channels and listening for beacon frames is called passive scanning. Alternatively, a wireless device can conduct active scanning by sending out a probe frame, which is received by all APs within its range. APs respond to this probe frame with a probe response frame. The wireless device can then choose an AP to associate with based on these responses.

Once an AP is selected, the wireless device sends an association request frame to the AP, which responds with an association response frame. This second handshake is necessary for active scanning because an AP responding to the initial probe request frame does not know which AP the device will choose to associate with. Similarly, a DHCP client selects from multiple DHCP servers. After associating with an AP, the device aims to join the subnet to which the AP belongs. To obtain an IP address on the subnet, the device sends a DHCP discovery message via the AP. Once it obtains an address, the device is perceived by the rest of the network as another host or station with an IP address in that subnet.

2.1.4 The 802.11 MAC Protocol

The 802.11 MAC Protocol governs how devices communicate over Wi-Fi networks. Once a Wireless device is associated with an AP, it can send and receive data frames. However, since multiple devices may want to transmit simultaneously, a multiple access protocol is needed to coordinate transmissions. The 802.11 protocol uses Carrier Sense Multiple Access with collision avoidance (CSMA/CA), similar to Ethernet's Carrier Sense Multiple Access/ Collision Detection (CSMA/CD) but with some key differences. Instead of collision detection, 802.11 employs collision-avoidance techniques and a link-layer acknowledgment/retransmission (ARQ) scheme due to the higher error rates in wireless channels. The 802.11 MAC protocol does not have collision detection like Ethernet. This is because detecting collisions requires simultaneous transmission and reception, which is difficult and costly to implement in wireless devices due to signal strength differences and technical challenges like the hidden terminal problem and signal fading. Without collision detection, once a station starts transmitting a frame in 802.11, it completes the transmission even if collisions occur, which can degrade performance. To address this, 802.11 uses collision-avoidance techniques.

Additionally, 802.11 employs a link-layer acknowledgment scheme to ensure reliable transmission. When a station sends a frame, the destination station sends back an acknowledgment frame after a brief wait time called the Short Inter-frame Spacing (SIFS). If the transmitting station does not receive acknowledgment within a specified time, it assumes an error and retransmits the frame using the CSMA/CA protocol. If acknowledgment is not received after several attempts, the transmitting station discards the frame. The 802.11 CSMA/CA protocol works like this:

1. If the station senses that the channel is idle, it sends its frame after a brief wait time called the Distributed Inter-frame Space (DIFS).
2. If the channel is busy, the station picks a random backoff value using binary exponential backoff and waits for the channel to become idle. It freezes the countdown while the channel is busy.
3. When the countdown reaches zero and the channel is idle, the station sends the entire frame and waits for an acknowledgment.
4. If acknowledgment is received, the station knows its frame reached the destination. If there is more data to send, it repeats the process from step 2. If acknowledgment is not received, it goes back to step 2 with a longer backoff interval.

2.1.4.1 Hidden Terminals

The 802.11 MAC protocol also uses the Request to Send (RTS) and Clear to Send (CTS) mechanism, which helps prevent collisions caused by hidden terminals. Imagine two wireless stations and one AP in range. Although both stations can communicate with the AP, they cannot hear each other due to signal limitations. This scenario can lead to collisions if one station starts transmitting while the other wants to send data.

To avoid this, a station can send an RTS frame to the AP before transmitting its data frame, requesting permission and indicating how long it needs to transmit. The AP responds with a CTS frame, granting permission and informing other stations not to transmit during that time. This reservation mechanism prevents wasted channel time and reduces collisions. Using the RTS and CTS frames can enhance performance in two key ways:

1. It helps alleviate the hidden station problem by ensuring that a long DATA frame is sent only after reserving the channel.
2. Since RTS and CTS frames are short, any collision involving them lasts only for the duration of these frames. After successful transmission of RTS and CTS frames, subsequent DATA and ACK frames should be transmitted without collisions.

However, the RTS/CTS exchange introduces delay and utilizes channel resources. Thus, it is typically employed only when transmitting long DATA frames. Each wireless station can set an RTS threshold, deciding when to use the RTS/CTS sequence based on frame length. Often, the default RTS threshold value exceeds the maximum frame length, leading to the omission of the RTS/CTS sequence for all DATA frames sent.

2.1.5 Mobility Management

To extend the coverage of a wireless LAN, organizations often deploy multiple BSSs within the same IP subnet. This raises the issue of how wireless stations move seamlessly between BSSs while maintaining ongoing TCP sessions. Mobility is straightforward within the same subnet, but more complex protocols are needed when stations move between subnets. For instance, consider a scenario where a host (H1) moves from one BSS to another within the same subnet. If the interconnected BSSs share the same IP subnet, H1 can retain its IP address and ongoing TCP connections as it moves between BSSs. However, if the interconnection device is a router, H1 would need to acquire a new IP address upon moving, disrupting existing TCP connections. To prevent this disruption, network-layer mobility protocols like mobile IP can be utilized.

When H1 moves from one BSS to another, it detects a weakening signal from the first AP1 and starts searching for a stronger signal. Upon receiving beacon frames from the second AP2, H1 disassociates from AP1 and associates with AP2, all while maintaining its IP address and ongoing TCP sessions. While this solves the handover issue from the perspective of the host and AP, the switch must also be informed of the host's movement. Switches typically learn and update their forwarding tables automatically. However, this mechanism is not designed for highly mobile users switching between BSSs. To address this, AP2 can send a broadcast Ethernet frame with H1's address to the switch after the new association. Upon receiving this frame, the switch updates its forwarding table, allowing data destined for H1 to be directed via AP2. The 802.11f standards group is working on an inter-AP protocol to address these issues.

2.2 ML in Wi-Fi Networks

In recent years, there has been a significant increase in user traffic within wireless networks, largely driven by the emergence of new multimedia services such as high-definition video and augmented/virtual reality experiences. These services come with stringent demands for reliability, low latency, and minimal bit error rates [2]. To address these demands and accommodate the growing traffic, a viable solution involves deploying small cells that utilize cellular technologies (e.g., 4G, 5G) alongside the implementation of Wi-Fi technologies in specific hotspots using unlicensed frequency bands [24, 25].

The advent of (Big) Data monitoring and analytics technologies represents a cornerstone in the evolution of cellular and Wi-Fi networks, aimed at addressing the aforementioned challenges. The monitoring system provides the ability to collect information about the users and the network, while the analytics system allows to extract knowledge of the collected data by means of AI mechanisms [26]. There are different ways to extract knowledge (e.g. by using classification, clustering and prediction mechanisms) [27]. By leveraging monitoring tools to collect extensive user and network data, coupled with the application of AI/ML techniques, networks can become smarter, more adaptive, self-aware, and cost-effective, thereby facilitating self-optimization [25].

ML applications are also becoming increasingly vital in addressing challenges and optimizing performance as Wi-Fi networks undergo continuous evolution. ML techniques are utilized across various aspects of Wi-Fi networks, improving efficiency, security, and overall functionality. The rising complexity of Wi-Fi networks, coupled with decentralized management and network densification, poses potential challenges for future 802.11 networks. To tackle these challenges, ML, a branch of AI capable of learning from data patterns, emerges as a promising solution [28]. Recognizing the significance of ML, a new IEEE 802.11 topic interest group has been formed, focusing on areas such as feedback compression of Channel State Information (CSI) using ML, enhanced sharing of ML models, and ML-driven distributed channel access [15, 29]. ML techniques are also crucial for implementing multi-AP coordination mechanisms envisioned for Wi-Fi 8. Researchers have explored ML-based approaches for a range of networking tasks, from configuring physical layer parameters to traffic prediction [30–33]. Key areas where ML is commonly applied in Wi-Fi networks are classified into four categories, including essential Wi-Fi features, newer Wi-Fi features, Wi-Fi connectivity and traffic management and other types of Wi-Fi scenarios [30, 33, 34].

2.2.1 Essential Wi-Fi Features

The latest IEEE 802.11 amendments bring enhanced functionalities to ensure robust network operation and an improved user QoE. Notably, amendments such as IEEE 802.11n/ac/ax boost data rates up to 9 Gbit/s by utilizing multiple spatial streams (SSs) and employing techniques like channel bonding, multi-user transmissions, Short Guard Interval (SGI), and high modulations (up to 1024QAM for 802.11ax) [35–37]. Characterizing the impact of these parameters on network performance is challenging due to the variability in Wi-Fi environments and user dynamics. However, the presence of performance metrics, both at the user and AP levels, along with historical data, creates an ideal setting for ML methods to model and optimize network performance. ML-based approaches excel in gaining knowledge, generalizing information, and learning from experience, making them suitable for creating intelligent systems using the advanced functionalities of the IEEE 802.11 standard. In this context, ML techniques are applied to select Physical (PHY) features, optimize channel access, configure frame aggregation and link parameters, choose data rates, and address admission control, QoS by traffic prioritization and traffic classification. Table 2.2 illustrates a summary of related work in the context of essential Wi-Fi features.

2.2.1.1 Channel Access

Channel access mechanisms are a frequently addressed topic concerning the enhancement of Wi-Fi performance using ML. Most proposed optimizations are related to the fundamental 802.11 MAC protocol, namely the Distributed Coordination Function

Table 2.2. Summary of related works for essential Wi-Fi features.

Wi-Fi Aspects	Ref.	ML Application	Novelty
Channel Access	[38]	Selecting the value of CW	Applying Q-learning in dense network scenario
	[39]	Selecting the minimum value of CW	Applying DQN to configure DCF
	[40]	Selecting CW update rule	Applying ABP framework to configure DCF
	[41]	Selecting the minimum value of CW	Improving fairness and robust to selfish stations
	[42]	Selecting the value of CW	Applying a fixed-share algorithm to configure DCF
	[43]	Selecting time slot for transmission	Applying FL to configure slotted transmissions
	[44]	Selecting the value of CW	Applying two DRL algorithms to configure DCF
	[45]	Selecting the value of backoff	Applying PDS to configure DCF
	[46]	Setting the value of AIFS and CW	Considering QoS requirements by traffic prioritization
	[47]	Selecting the value of CW	Considering QoS requirements by traffic prioritization
	[48]	Selecting time slot for transmission	Stations self-organize into channel access based on slot
	[49]	Selecting time slot for transmission	Taking into account interference from non-ML based devices
Link Configuration	[50]	Selecting transmission rate	Applying the RF method for rate selection
	[51]	Selecting transmission rate	Providing extensible framework of rate selection
	[52]	Selecting transmission rate	Applying iterative learning for rate selection
	[53]	Selecting transmission rate	Utilizing the input of channel conditions, number of stations, and traffic intensity
	[54]	Selecting transmission rate	Utilizing packet timeouts to train RL model
	[55]	Classifying channel type	Applying SL to classify channel
	[56]	Selecting guard interval	Applying TS for guard interval selection
	[57]	Prediction the throughput of link-layer	Applying SL for link adaptation
Frame Aggregation	[58]	Selecting CW and frame size	Applying ML for frame-size optimization
	[59]	Selecting frame size	Applying SL for frame-size optimization
	[60]	Selecting frame size	Applying ML for optimization in SDN architecture
	[61]	Selecting frame size	Taking into account energy-consumption constraint
	[62]	Selecting frame size and transmission rate	Combined frame-size and transmission-rate optimization
PHY Features	[63]	Estimating probability of deferral	Analyzing interference relations between stations
	[64, 65]	Decoding frames	Alternative 802.11 protocol based on channelization in OFDM
	[66]	Estimating interference level	Applying DPP learning to determine interference distribution
	[67]	De-noising signals	Applying DL to enhance the quality of radio signal
	[68]	Classifying signal source	Applying ML to signal identification based on CSI
	[69]	Predicting signal strength	Applying DL to predict the quality of radio signal
	[70]	Predicting link-layer throughput	Utilizing SL to model the impact of PHY/MAC interactions on throughput

(DCF), serving as the baseline mechanism to prevent collisions among devices accessing a shared radio channel [71]. The key parameter influencing DCF performance is the Contention Window (CW), determining the range from which stations randomly select waiting periods (backoff counters) to avoid collisions during channel access. Larger CW values decrease collisions but increase idle times, thereby reducing throughput. On the other hand, smaller CW values enhance the chance of station transmission but also raise collision probabilities, consequently reducing throughput. Several studies explore the optimization of contention window (CW) values to enhance throughput by minimizing collisions and idle periods, often employing both supervised learning (SL) and reinforcement learning (RL) models. The evaluation criteria include reduced collisions [38, 39], increased differentiation between successful and collided frames [40], improved channel utilization [41], higher successful channel access attempts [42, 43], enhanced throughput [44], elevated network utility [72], and a combination of improved throughput, reduced energy consumption, and decreased collision instances [45]. We classify related works here into five categories, presented as follows:

1. **Collision Reduction:** In high-density 802.11ax WLANs, RL with the intelligent Q-learning based resource allocation (iQRA) is considered in [38]. The work utilizes RL with Q-learning for resource allocation, dynamically adjusting CW based on channel observation. By minimizing cumulative reward (considering collision probabilities), it optimizes CW size. Results from simulations show improved throughput compared to baseline 802.11ax protocol, with similar delay. The paper [40] implements a programming paradigm called adaptation-based programming (ABP), where implements RL with a reward system based on successful transmissions vs. collisions. It optimizes RL for two actions: halv-

ing or maintaining CW size after a successful transmission. Simulations reveal a reduction in dropped packets. The RF algorithm is applied in a supervised manner to balance the minimum CW size among users for fair channel access [41]. Using channel variables, it builds a DT to optimize settings. Simulations in 802.11ac scenarios show significant improvements in throughput, latency, and fairness compared to the standard. The size of the CW can also be adjusted by directly increasing the access to the channel through weighted CW range possibilities [42]. After successful transmissions, weights of users with larger CWs are reduced, while weights of users with smaller CWs are increased. Simulations in heavily loaded scenarios demonstrate improved throughput and reduced end-to-end delay compared to DCF.

2. **Scalability:** The paper [44] Utilizes DRL models, including Deep Q-Network (DQN) and Deep Deterministic Policy Gradient (DDPG), to maintain stable throughput as the number of stations increases. A three-phase algorithm evaluates collision probabilities, trains DRL models to maximize throughput, and deploys them in the network. Simulations demonstrate that these algorithms maintain stable throughput compared to the 802.11ax standard, which experiences decreased throughput with an increasing number of stations. Moreover, a post-decision state-based (PDS) learning algorithm is applied in [45] to enhance scalability by leveraging previous system knowledge, such as CW and transmission buffer occupancy. Unlike Q-learning (QL), PDS achieves faster convergence by predefining CW values in specific states, eliminating the need to learn transition probabilities. This approach results in improved throughput, especially under moderated network load, compared to Q-learning, the standard 802.11 approach, and other deterministic mechanisms like exponential-increase exponential-decrease (EIED).

3. **User Fairness:** The CW can also be adjusted considering user fairness metrics [43]. To that end, Federated Learning (FL) and Q neural network (QNN) models are implemented in APs and stations, respectively, as a part of a distributed method. Initially, stations randomly initialize their QNN parameters, leading to varied strategies for channel access. To ensure fairness, the AP collects a global model of QNNs through FL and broadcasts updated CW values to stations. Simulations with up to 50 stations demonstrate a 20% improvement in throughput compared to the DCF. Additionally, an enhanced DQN is trained to select minimum CW sizes and deployed at stations to achieve per-user fairness [39]. Rainbow agents incorporate six improvements, including double DQN, prioritized reply, dueling networks, multi-step learning, distributional RL, and noisy nets. Simulation results in ns-3, with 32 stations transmitting at a constant rate, show that this solution achieves near-optimal results and outperforms an RF-based method.

4. **QoS improvement by traffic prioritization:** QoS improvement in wireless networks is crucial for ensuring a satisfactory user experience, especially in scenarios where multiple types of traffic coexist and compete for network resources. One approach to enhance QoS is through the use of traffic prioritization mechanisms, which allocate network resources based on the priority of different types of traffic.

   Traditionally, the DCF in the IEEE 802.11 standard has been used for MAC in wireless LANs. However, with the increasing demand for differentiated services, the Enhanced Distributed Channel Access (EDCA) was introduced as part of the 802.11e amendment [73, 74]. EDCA extends DCF by introducing new or modified MAC parameters for different traffic classes. For example, while the CW parameter existed in DCF, EDCA introduced a more dynamic adjustment of CW values based on traffic priorities. Additionally, EDCA introduced new parameters such as Arbitration Inter-Frame Space (AIFS) and Transmission Opportunity (TXOP) limit, which further support traffic differentiation and prioritization.

The choice of appropriate values for parameters like AIFS and CW is critical in balancing the trade-off between delay and throughput. AIFS determines the time interval a station must wait before it can access the medium, with different AIFS values assigned to different traffic classes to prioritize certain types of traffic. CW defines the range of time slots a station must wait before attempting to access the medium. TXOP limit specifies the maximum duration a station can hold the medium once it gains access, allowing for efficient transmission of larger bursts of data for high-priority traffic. To address this challenge, researchers have explored ML techniques to determine the optimal combination of AIFS and CW values. For example, a three-phase scheme proposed by [46] utilizes DT algorithms like J48 for classification and M5 for prediction to select the best AIFS and CW values. Simulation results have shown high throughput prediction accuracy across various scenarios, indicating the effectiveness of ML-based approaches in optimizing network performance.

In addition to ML-based approaches, QL has been employed to achieve priority-driven channel access under the EDCA scheme. By estimating network density and dynamically adjusting the CW value based on traffic priorities, the QL model ensures efficient utilization of network resources [47]. For instance, in EDCA, smaller CW values are configured for high-priority traffic categories such as voice and video, thereby reducing contention and improving QoS for critical applications. Simulations conducted using tools like the ns-3 simulator have demonstrated the effectiveness of QL-based approaches in enhancing throughput for different traffic types compared to standard EDCA mechanisms.

5. **Time-Slotted Access:** In these mechanisms, collisions are prevented by scheduling users to transmit during specific time slots [48]. Each station maintains a table containing available time slots for transmitting frames. These time slots are determined using RL techniques to select optimal actions for accessing the channel. Furthermore, the research described in [49] explores a scenario involving channel access between two APs: the primary AP equipped with an agent and a secondary AP referred to as the ‘outsider’. Time is divided into slots, allowing both APs to independently decide when to transmit. The objective of the agent in the primary AP is to maximize throughput by learning the transmission probabilities based on the behavior of the outsider AP. A robust adversarial RL framework, employing game theory, models the interactions between the two APs and facilitates learning of optimal transmission policies through Q-learning.

2.2.1.2 Link Configuration

The IEEE 802.11n/ac/ax standards have been developed to meet the increasing demands of users by providing high-throughput wireless connections [56]. These standards introduce various enhancements at both PHY and MAC layers. At the PHY layer, features such as channel bonding, multi-SS transmissions, SGI, and high modulations like 1024-QAM (in 802.11ax) contribute to achieving high data rates. These functionalities are aimed at improving wireless link performance [35–37]. Meanwhile, at the MAC layer, frame aggregation and block acknowledgment are implemented as key features to enhance the maximum link throughput.

Link configuration, involving the selection of appropriate PHY and MAC parameters, is crucial for achieving optimal throughput under varying network and channel conditions. Rate adaptation, a key component of link configuration, determines the suitable MCS for each transmission, balancing between high data rates and potential transmission errors. In dynamic Wi-Fi environments, such as those affected by user mobility or interference, rate adaptation faces challenges like the trade-off between high data rates leading to increased error rates and lower data rates resulting in underutilization of the channel. ML models are increasingly utilized to assess this trade-off and optimize rate adaptation strategies to cope with changing channel conditions effectively. Below, we outline the contributions related to selecting the most suitable MCS and SGI values, as well as several trade-offs encountered at the PHY layer.

1. **Rate Adaptation:** Rate adaptation methods assess the likelihood of successful transmissions for each potential MCS option, selecting the data rate associated with the most favorable outcome. These predictions rely on signal-to-noise ratio (SNR) [50, 51] or adopt a cross-layer strategy based on acknowledgment (ACK) or negative acknowledgment (NACK) feedback [52–54]. SNR is favored for promptly updating channel conditions, particularly in mobile environments like vehicular ad hoc networks (VANETs) [50]. However, more precise assessments are achieved when channel status updates are guided by ACK and NACK feedback [50]. SNR-based predictions benefit from a two-tiered data rate search method employing an artificial neural network (ANN) model [51] or an RF algorithm [50]. The ANN serves as a coarse estimator to identify a preliminary set of optimal data rate candidates, followed by a refined selection process to determine the best candidate. This approach yields a notable enhancement of at least 25% in mobile scenarios compared to baseline rate adaptation algorithms like Minstrel [51]. Additionally, the RF algorithm implemented in [50] enhances uplink data rate adaptation in VANETs by leveraging car position and velocity data to estimate SNR. By predicting transmission success probabilities for each potential data rate option and selecting the most suitable candidate, this approach achieves a minimum goodput improvement of 27% relative to existing solutions such as collision-aware rate adaptation (CARA).

   To address the variability of SNR and packet loss caused by fast fading, [52] introduces a method inspired by stochastic learning automata (SLA), which avoids assuming a predefined relationship between SNR and packet loss. This algorithm updates a selection probability vector mapped to available data rates, adjusting it based on throughput as the reward function and using ACK frames for channel feedback. Consequently, the probability corresponding to the most rewarding rate is updated, yielding a 15% throughput enhancement compared to other solutions. Additionally, [53] employs ML models to establish thresholds for successfully and unsuccessfully received packets, enhancing aggregate throughput by counting ACKs. Building on the auto rate fallback (ARF) algorithm, the data rate adjusts based on whether the total ACK count surpasses predefined thresholds, with adjustments informed by an ANN's estimation of correlation with achievable throughput. The results demonstrate a 10% increase in aggregate output in networks with ten stations.

Rate selection methods include SL, as seen in [55], which classifies channel conditions (e.g., residential or office environments) to determine the appropriate MCS level based on selected characteristics of an 802.11 frame's preamble. In contrast, Q-learning, as demonstrated in [54], dynamically adjusts the MCS level based on the total number of received ACKs, with network state observation inferred from timeout events (total number of missing ACKs). Simulations in ns3-gym consider dynamic scenarios, such as receiver station movement at 80 m/s, yielding throughput comparable to Minstrel. Additionally, MCS selection may account for available bandwidth and selected spatial streams. In [75], the DDQN model, incorporating goodput as a reward and employing prioritized training, history-based initialization, and adaptive training interval, significantly outperforms default mechanisms when implemented in hardware.

2. **SGI Adaptation:** The choice of SGI values, another aspect of link configuration, can be optimized using ML models. SGI offers two (802.11ac) or three (802.11ax) different values, and their selection is managed through Thompson sampling (TS) as detailed in [56]. TS, an online learning method, adapts to channel quality fluctuations like signal interference, fading, and attenuation. Evaluations conducted via ns-3 simulations on an 802.11ac network with up to 40 stations demonstrate slight throughput enhancements when SNR varies randomly between 20–60 dB, compared to static SGI settings.

3. **PHY Layer Trade-Offs**: In the PHY layer, several trade-offs exist, such as choosing wider channels leading to more interference, selecting MCS based on required SNR, and considering frame aggregation versus packet loss. These trade-offs can be collectively managed to enhance overall performance using ML techniques like multi-armed bandit (MAB) [35], [76–78], and DL [57].

   An online learning approach based on the Multi-Armed Bandit (MAB) framework is developed for configuring links in 802.11ac networks [35], [76–78]. This method, utilizing a MAB-based adaptive learning (AL) (i.e., the ϵ-greedy algorithm) with fuzzy logic, considers network load and channel conditions to enhance performance by exploring various configurations. Through this approach, throughput is notably improved, with up to a 358% increase compared to existing methods. Additionally, a study by [57] introduces a two-step algorithm that leverages Deep Neural Networks (DNNs) to estimate throughput based on PHY and MAC layer parameters such as channel bonding, MCS, and frame aggregation settings. This algorithm then utilizes a predictive control-based search to identify optimal parameter values, leading to superior performance in terms of delay and throughput compared to baseline algorithms, as demonstrated through experiments with IEEE 802.11ac client boards on laptops.

2.2.1.3 Frame Aggregation

Frame aggregation is a technique used in 802.11 networks to enhance throughput by transmitting multiple data frames in a single transmission. This method reduces protocol overhead by consolidating multiple packets into a single transmission, thereby reducing the number of headers required. There are two types of frame aggregation in 802.11: MAC Service Data Unit (MSDU) aggregation, which bundles multiple packets from the upper layer into an aggregate-MSDU (A-MSDU) with a single PHY header, and MAC Protocol Data Unit (MPDU) aggregation, which consolidates multiple frames from the MAC layer into an aggregate MPDU (A-MPDU) transmitted with a single PHY header.

Frame aggregation significantly affects communication efficiency by balancing useful transmitted data with overhead [79]. Efficiency is evaluated based on packet decoding errors, where larger frames reduce overhead impact but are more susceptible to transmission errors. To optimize efficiency, frame aggregation techniques determine the ideal frame size. The 802.11 standard introduces two main aggregation methods: aggregated MAC service data unit (A-MSDU) and aggregated MAC protocol data unit (A-MPDU) [80]. These aggregations can also be used together [81]. While A-MSDU is more efficient but prone to errors due to a single frame check sequence (FCS), A-MPDU is more robust with multiple FCSs but introduces higher overhead. However, these methods lack dynamic adjustment for varying CSI in wireless links.

ML techniques, including SL and RL, are employed to optimally select frame sizes in dynamic conditions across various 802.11 network standards. SL and RL methods are utilized to maximize throughput in generic 802.11 networks [58], improve goodput in 802.11n networks [59, 60], and balance energy and throughput in 802.11ac networks [61]. These techniques are also applied to estimate aggregation levels in 802.11ac networks [82]. For instance, paper [59] introduces a low-complexity approach using a random forest regressor (RFR) model to configure aggregation and MCS settings in the downlink direction. This method, tested in small to medium-sized networks with up to 20 stations, reduces retransmission rates and improves goodput by 18.36% compared to traditional 802.11 aggregation methods.

ML-supported aggregation methods are applied in software-defined WLANs (SD-WLANs) within an AI-based operating system [60]. Using low-complexity models like M5P and RFR, these methods aim to optimize frame length for each user to maximize goodput. Training with real Wi-Fi measurements in scenarios with up to 10 stations, the RFR model achieves the highest goodput improvement (55%) compared to A-MSDU. Additionally, predicting MCS levels through an ANN [62] involves training the model in client devices using received packets from an AP within a 1-second window, improving throughput by at least 13% over baseline algorithms. ML techniques are also utilized in estimating aggregation levels to manage queue backlogging [82]. By employing LR estimator models on hardware-level timestamps, accurate aggregation level estimation with low computational complexity is achieved, particularly in non-rooted hardware acting as client devices, with accuracy reaching close to 100%.

Frame aggregation settings can be optimized considering energy costs by selecting the aggregation level with the smallest frame error rate (FER) based on channel conditions, as indicated by the SNR value [61]. Using an online learning algorithm combined with fuzzy logic, this approach defines suitable aggregation levels and selects the most appropriate one to minimize FER. Resulting in a 14% improvement in energy efficiency compared to standard A-MSDU and A-MPDU mechanisms with 10 stations. Additionally, addressing channel conditions and collision impacts, [58] adjusts frame size and CW together using an ANN model trained with frame size-throughput patterns. Simulation results for 10 mobile users demonstrate throughput improvement compared to optimizing only the frame size without considering the optimal CW.

2.2.1.4 PHY Features

ML techniques are applied at the PHY layer to enhance Wi-Fi network performance by addressing various issues such as collision detection characterization [63] and its mitigation [64, 65], interference power-level characterization [66] and its mitigation [83], signal de-noising [67], source detection to improve spectral efficiency [68], prediction of signal strength variability [69], and the enhanced modeling of the PHY and MAC layer interactions to improve throughput [70].

1. **Collision Reduction:** To estimate collision occurrences in the channel, stations' network activity is modeled using a hidden Markov model (HMM) [63]. RL techniques are employed to learn model parameters, allowing the calculation of collision probabilities. The transition probabilities in the model are evaluated using the expectation modification algorithm (EMA). By estimating the total number of simultaneous transmitters, collision probabilities are directly computed. Results from a study deploying seven APs with an equal number of clients over two building floors demonstrate the accuracy of estimated deferring probabilities, reflecting real-world conditions. Additionally, to enhance the decoding of request to send (RTS) frames during collisions, [64, 65] implement an ML model using techniques like NB, naive Bayesian tree, J48 DT, and SVM. This approach, integrated with a kϵ-greedy algorithm for channel allocation, improves system performance significantly compared to legacy 802.11 operations.

2. **Interference Estimation:** The interference level in the network is estimated using a determinantal point process (DPP) model [66]. A SL process is employed to determine the number and positions of active transmitters that could potentially interfere with each other. The interference is assessed by computing the cumulative density function for the total number of active users. This information is utilized to model the power of interference signals using a path-loss model for individual links. Experimental results demonstrate a close correspondence with theoretical models regarding the cumulative density function of interference levels.

3. **Signal Quality Estimation and Management:** The paper [69] employs DL techniques to predict received signal strength. Using a RNN model, it utilizes encoder and decoder components to capture and predict the CSI variability. The model is trained under three different schemes to balance convergence speed and performance: (i)guided training, which employs current measured signal strength for faster convergence; (ii)unguided training, which utilizes predicted signal strength for improved prediction performance; and (iii)curriculum training, which combines both methods to balance speed and prediction accuracy. The curriculum training scheme notably enhances prediction accuracy of the signal strength compared to linear regression and auto-regression methods.

   DL techniques, as discussed in [67], contribute to enhancing the quality of received signals at the PHY layer. Using an ANN, the preamble of 802.11 protocols undergoes de-noising in the spectrogram domain, where noise is separated from the useful signal. By processing the spectrogram as an image, the ANN functions as a convolutional de-noising auto-encoder to estimate the original signal patterns, achieving a reconstruction accuracy of approximately 85%. Additionally, to improve Wi-Fi spectral efficiency by addressing the exposed terminal problem, [68] employs sender identification based on CSI. Models trained using kNN and ANN techniques successfully predict potential interference among senders, achieving an accuracy of 90% in indoor scenarios with 20 wireless stations, particularly excelling with the kNN model when total samples are limited.

4. **Interaction With the MAC Layer:** The PHY layer collaborates with the MAC layer to assess the impact of various factors on observed throughput [70]. Input features like received power, channel width, spectral separation, traffic load, and physical rates are selected to construct a mathematical function that correlates these features with throughput, using SL. This function serves as a predictive model, enabling the optimization of throughput. Regression techniques such as regression tree, gradient boosted regression tree (GBRT), and support vector regressor (SVR) are utilized to derive this function. Simulation outcomes reveal that GBRT and SVR offer superior accuracy compared to a benchmark model.

2.2.2 Newer Wi-Fi Features

ML techniques play a crucial role in supporting recent Wi-Fi amendments, including 802.11ac, 802.11ax, and 802.11be. These amendments introduce advanced and intricate techniques such as multi-user communications (OFDMA, MU-MIMO) [84], spectrum aggregation and opportunistic spectrum access (channel bonding [85], multi-link operation [86–88]), spatial reuse [89], and multi-AP coordination [90, 91]. While these techniques hold the promise of significant performance improvements in terms of both throughput and latency, they also present new challenges. Table 2.3 presents a summary of related work that has utilized ML in this area of Wi-Fi networking.

2.2.2.1 Beamforming

Transmissions within the millimeter-wave (mmWave) 60 GHz band in Wi-Fi networks, particularly in scenarios like short-range (indoor) and long-range (outdoor) communication, known as fixed wireless access (FWA) [92], require beamforming to combat increased signal attenuation. Beamforming, introduced in 802.11ad and extended in 802.11ay, aims to optimize beam sector pairs between transmitters and receivers. Traditional methods involve exhaustive beam searches, which can be time-consuming. To address this, [93] employs NN algorithms to predict optimal beam sectors, including historical data. [94] further enhances this approach by reducing training duration through SL-based feature extraction and reinforcement learning-based beam selection. Additionally, [95] presents DeepBeam, a framework utilizing DL to infer beam sector selection without the need for time-consuming beam sweeping procedures, by passively listening to other transmissions.

ML techniques are increasingly employed in beamforming for mmWave Wi-Fi networks. In [96], camera images are utilized to expedite beam alignment predictions, significantly reducing setup time. Similarly, [97] leverages ML on camera images to swiftly and accurately predict received power for beam selection. Additionally, camera-based ML predictions aid in preemptively detecting link outage, improving handovers in mmWave networks [98]. In scenarios necessitating dense AP deployment, such as in 802.11ad/ay networks, beam coordination and interference management are crucial. [99] employs statistical learning to construct a radio map, reducing cross-beam interference. Meanwhile, [100] presents a DNN-based solution for optimizing beams in centrally-managed deployments, achieving performance comparable to optimization algorithms with reduced computational time.

Table 2.3. Summary of related works for newer Wi-Fi features.

Wi-Fi Aspects	Ref.	ML Application	Novelty
Beamforming	[93]	Selecting beam pair	Reducing number of beam sectors
	[94]	Selecting beam pair	Combining feature extraction with training beam selection
	[95]	Selecting beam pair	Listening to ongoing transmissions passively
	[96]	Selecting beam pair	Using camera imagery
	[97]	Predicting received power	Using camera imagery
	[98]	Predicting link outage	Using camera imagery
	[99]	Selecting AP and beam	Reducing cross-beam interference
	[100]	Selecting beam directions, transmitted power and beamwidths	Performing combined beam management and interference coordination
	[101]	Selecting AP and band	Considering multiple APs and bands
	[102]	Classifying channel type	Classifying channel using received preamble
	[103]	Predicting channel characteristics	Applying prediction method to massive MIMO channels
	[104]	Selecting transmission rate	Utilizing PHY layer features for rate selection
	[105]	Selecting transmission rate and beam pair	Combined transmission rate and beam pair selection
	[106]	Estimating channel frequency response	Utilizing transceiver location information
Multi-User Communication	[107]	Selecting users and configuring links	Combined parameter optimization
	[108]	Selecting users, spatial mode and transmission rate	Addressing MU-MIMO challenges
	[109]	Compressing CSI	Minimizing CSI feedback airtime
	[110]	Predicting benefits of MU-MIMO	Addressing MU-MIMO station mobility
	[111]	Selecting MU-MIMO group	Utilizing ML to improve MU-MIMO performance
	[112]	Scheduling RUs	Applying DRL to resource allocation in DLNK OFDMA
	[113]	Scheduling RUs	Applying DRL to resource allocation in ULNK OFDMA
	[114]	Scheduling RUs	Improvement of Q-value estimation
	[115]	Scheduling RUs	Resource allocation based on queue
	[116]	Compressing CSI, scheduling RUs and allocating power	Combined MU-MIMO and OFDMA optimization
Spatial Reuse	[83]	Selecting channel and transmitted power	Applying QL with training based on event
	[117]	Selecting carrier sense threshold, transmission rate and power	Combined optimization of these parameters
	[118]	Deciding whether to transmit simultaneasly	Adaptive CCA per frame
	[119]	Selecting antenna orientation	Directional transmission in multi-AP indoor setting
	[120]	Selecting channel and transmitted power	Dynamic channel assignment in a decentralized setting
	[121]	Selecting channel, transmitted power and sensitivity threshold	SR Improvement in dense and uncoordinated WLANs
	[122]	Selecting channel and transmitted power	Evaluating new strategies of action selection
	[123]	Selecting channel and transmitted power	Deploying SR methods in a SDN architecture
	[124]	Selecting carrier sensitivity threshold	Combining local and global operations
	[125]	Selecting transmitted power and sensitivity threshold	Subsampling state space to solve dimensionality problem
Channel Bonding	[126]	Selecting channel and bandwidth	Applying stateless RL to channel allocation
	[127]	Selecting channel and bandwidth	Packet scheduler supporting dynamic bandwidth channel access
	[128]	Selecting channel and bandwidth	Considering non-continuous channels
	[129]	Selecting channel and bandwidth	Utilizing laser oscillations to action space
	[130]	Selecting channel and bandwidth	Applying DL to channel bonding issue
	[131]	Selecting channel and bandwidth	Spatio-temporal changes in traffic demands
	[132]	Identifying stations under starvation	Fairness of dynamic bandwidth channel access
	[133]	Selecting channel and bandwidth	Proposing general spectrum management framework
	[134]	Selecting channel and bandwidth	Considering traffic load per channel
	[135]	Predicting throughput	Taking into account impact of channel bonding
	[136]	Predicting throughput	Applying GNN to performance prediction
	[137]	Selecting channel and bandwidth	Taking into account hidden stations
MLO, MIMO, and Full Duplex Communication	[138]	Predicting channel idle state	Applying PNN to multi-band Wi-Fi network
	[139]	Selecting channel and clustering APs	Combined parameter optimization for distributed MU-MIMO networks
	[140]	Finding pairs for transmission simultaneously	Applying ML to orchestrating full-duplex transmissions
	[141]	Selecting band for retransmission	Solution in retransmissions over multi-band Wi-Fi network

In dense deployment scenarios, associating user stations with APs becomes challenging, particularly with the advent of multi-homing capable stations. This issue is addressed using ML methods, as demonstrated in [101], where users autonomously learn which APs to connect to and which bands to use. After identifying suitable beam sectors in 802.11ad/ay, rate adaptation becomes necessary. For mmWave transmissions, MCS selection relies on accurate channel classification, distinguishing between Line of Sight (LoS) and Non-Line of Sight (NLoS) channels. ML augmentation is seen in [102], employing RF techniques for channel classification. Furthermore, predicting the statistical channel characteristics is crucial, with studies like [103] utilizing CNNs to forecast channel statistics, particularly in indoor locations across various wireless technologies using massive MIMO.

Alternatively, rate adaptation can rely on conventional metrics available in commercial off-the-shelf (COTS) devices. In [104], optimal MCS settings are predicted using three ML models: DT, RF, and SVM, with RF showing superior performance over SNR-based rate selection strategies. This approach is extended in the learning-based beam and rate adaptation (LiBRA) framework [105], where ML-based classification methods determine whether rate selection or beam selection offers better performance for a given link. Furthermore, Improving the data rate of mmWave links can be achieved through enhanced channel estimation techniques. In [106], transceiver location information is combined with a DNN to assess the channel frequency response. This method reduces the number of transmitted pilot signals, thereby allocating more bandwidth for user data.

2.2.2.2 Multi-User Communication

The IEEE 802.11ac amendment introduced support for downlink MU-MIMO transmissions, enabling simultaneous transmission to different stations within the same TXOP using spatial multiplexing. This feature was extended in IEEE 802.11ax, which supports both downlink and uplink MU-MIMO, along with OFDMA. OFDMA divides the available bandwidth into sub-channels, known as RUs, allocated to different users. These features will continue to be significant in future IEEE 802.11be networks, which will support wider channels and more spatial streams, as well as improvements like multiple RU allocation to the same user and implicit channel sounding [142].

Several papers tackle challenges in Multi-User MU-MIMO enabled WLANs through various ML strategies. In [107], an ϵ-greedy strategy is employed to optimize configurations using experience. The study by [108] employs an SVM classifier for robust MCS selection. [109] introduces a method using DNNs to compress and decompress CSI to reduce overhead. Conversely, [110, 111] utilize a policy gradient technique to determine client participation in MU-MIMO transmissions, with a NN representing the policy function. These approaches consistently demonstrate significant improvements in network throughput. In OFDMA, whether initiated by the AP or stations, selecting the group of stations and RUs allocation is crucial. In AP-initiated transmissions, the AP determines the station group and optimal RU allocation, while in uplink transmissions, stations may select their RUs [143]. This challenge is tackled using DRL techniques in [112–114]. The papers [113, 114] focus on decentralized RU selection for uplink, employing CNN-based DQN methods, showing significant gains over random RU selection. Conversely, [112] focuses on AP-initiated downlink transmissions, employing DRL-based scheduling using per-station channel quality and traffic information. The study in [115] employs a deep deterministic policy gradient (DDPG) algorithm to address OFDMA resource allocation, demonstrating improved latency meeting and fairness compared to baseline solutions. Additionally, In the study [116], the focus is on optimizing MU-MIMO and OFDMA together. The paper employs Deep Supervised Learning (DSL) to develop DeepMux, executed at APs. DeepMux utilizes DNNs to mitigate channel sounding impact and determine a nearly optimal resource allocation policy. Experimental findings reveal throughput improvements of up to 50%.

2.2.2.3 Spatial Reuse

The IEEE 802.11ax amendment introduced spatial reuse (SR) to Wi-Fi networks [89], aimed at enabling concurrent transmissions among devices from different BSSs. Despite its effective gains, the conservative rule-based design of IEEE 802.11ax SR could be further enhanced by adaptive mechanisms. ML techniques can play a crucial role in making SR adaptive to various scenarios, determining when and how devices can benefit from spatial reuse opportunities, potentially leading to higher throughput and latency improvements. Future amendments, such as IEEE 802.11be, are expected to extend Wi-Fi SR capabilities, possibly allowing neighboring APs to coordinate transmissions. ML techniques are poised to enhance coordinated SR by addressing the challenge of identifying devices that can transmit simultaneously, leveraging collected CSI data from multiple devices.

ML solutions have increased some attention for addressing the SR problem, particularly using RL techniques. Methods like Q-learning and MABs have been widely explored in recent studies [117–123]. These approaches involve multiple agents, each empowered with ML capabilities, learning online to determine the best configuration for APs. However, some studies highlight challenges when agents compete without collaboration, making convergence difficult. Other approaches use SL techniques, such as NNs, to select appropriate SR parameters based on known scenario characteristics [124, 144]. These methods contribute to optimizing SR in wireless networks.

The study [117] employs a Q-learning algorithm to optimize power, transmission rate, and clear channel assessment (CCA) in Wi-Fi networks. Agents, located at each device, act selfishly, and states are defined based on transmission power, interference, and the used MCS. Actions involve adjusting transmission power and MCS to enhance performance. The study [118] utilizes Q-learning to enhance IEEE 802.11ax's SR mechanism, where agents decide whether to transmit concurrently or wait, considering current interferers. In this paper, the learning rate is adapted to ensure quick adjustments in non-stationary scenarios. Meanwhile, Q-learning and MABs are explored in various scenarios [120–122]. Stateless Q-learning optimizes channel selection and transmission power allocation, while MABs consider different action-selection strategies. Results demonstrate that even when networks operate selfishly, optimal proportional fairness is achieved, with sequential action taking reducing throughput variability among BSSs. Additionally, the paper [125] proposes a centralized MAB-based solution to dynamically adjust spatial reuse parameters, mitigating starvation by selecting the best configuration using Thompson Sampling (TS). Simulation results confirm the effectiveness of this approach in enhancing performance in dense WLANs.

SL methods like Multi-layer Perceptron (MLP) and DTs are explored in [124] to optimize spatial reuse parameters for both APs and stations. These models are trained offline using diverse datasets encompassing various scenarios. The paper [144] proposes a centralized NN approach to configure all BSSs for maximizing SR, considering the correlation between device throughput and associated link parameters. In another approach, [83] tackles interference mitigation by jointly optimizing APs' transmit power and channel allocation policies using a Q-learning model. This model undergoes a learning process with reduced iterations triggered by network status changes, resulting in a 16% throughput enhancement compared to state-of-the-art mechanisms. Meanwhile, [119] adopts a distinct strategy for achieving spatial reuse with directional transmissions, treating antenna orientation selection as a non-stationary MAB problem. Results from a software-defined radio (SDR) implementation demonstrate its operational correctness and resilience to co-channel interference.

2.2.2.4 Channel Bonding

The capability to enable channels wider than 20 MHz was initially introduced in IEEE 802.11n, supporting up to 40 MHz channels. Subsequent amendments like IEEE 802.11ac and IEEE 802.11ax expanded the maximum channel width to 80 and 160 MHz, respectively. In IEEE 802.11be, channel width is further increased, accommodating up to 320 MHz channels. Wider channels facilitate higher transmission rates and thus improved performance. However, in dense environments, wider channels may intensify contention among neighboring BSSs, potentially leading to decreased performance. Therefore, determining when to utilize wider channels, their appropriate size, and the specific channels to use are crucial for enhancing WLAN performance. However, there is no universal solution to this question, as it depends on various factors such as the number and positioning of contending devices, BSS loads, and available channels in each specific scenario. ML techniques offer a solution to optimize channel allocation and bonding configurations in various scenarios. Online learning, particularly when combining RL techniques with prediction models, emerges as a suitable approach for rapid convergence in addressing this challenge [126].

MABs typically determine the optimal channel widths for maximizing WLAN performance without considering additional network or user-related information [127–129]. However, when factors like traffic loads, delay, and throughput are taken into account, DRL techniques are proven to be effective in channel bonding decision-making [130, 131].

The paper [127] proposes enhancing dynamic channel bonding by considering individual station needs and access category requirements, using a MAB algorithm, Upper Confidence Bound (UCB). Testbed results indicate performance gains reaching 100% in certain scenarios. Similarly, the study [128] introduce an Iterative Trial and Error (ITE) mechanism, employing trial and error to determine optimal channel bonding strategies, including both contiguous and non-contiguous 20 MHz channels. ITE, implemented using an ϵ-greedy strategy, outperforms default mechanisms and enhances both Static Bandwidth Channel Access (SBCA) and Dynamic Bandwidth Channel Access (DBCA) performance. Additionally, the study [132] proposes Hybrid Adaptive DBCA (HA-DBCA) to address starvation issues in DBCA devices. HA-DBCA employs a polling-based adaptive mechanism for contention-free access and utilizes the UCB algorithm to identify starving stations, enabling them to transmit during contention-free access periods. The paper [129] also models the channel bonding problem as a MAB, using chaotically oscillating waveforms generated by semiconductor lasers for exploration. The thresholds dynamically adjust based on waveform amplitudes, demonstrating superior throughput compared to traditional MAB algorithms. Furthermore, the study [133] advocates for model-free RL techniques for channel bonding, designing a comprehensive RL framework and showcasing the effectiveness of lightweight MABs through simulations. In this study, lightweight MABs offer efficient adaptation without the complexity of Q-learning or deep reinforcement learning, presenting a viable alternative for rapid learning in realistic scenarios.

DRL techniques are explored for optimizing channel bonding configurations in WLANs. The paper [130] focuses on minimizing latency by dynamically allocating channels to different BSSs based on their expected load and performance using an on-demand channel bonding (DCB) algorithm. By employing multi-agent deep deterministic policy gradient (MADDPG) for training, suitable channel allocations are determined. The paper [131] addresses the channel assignment problem in WLANs with channel bonding, considering spatio-temporal changes in traffic demands. The DRL solution adapts to varying service requirements by learning from historical traffic loads, minimizing interactions with other BSSs when unnecessary. Additionally, the study [134] proposes an opportunistic contiguous and non-contiguous channel aggregation scheme for 802.11ax WLANs, using DRL to adjust aggregation probabilities of secondary channels based on traffic load. Results demonstrate the superiority of this strategy over predefined rules such as aggregating all channels or randomly selecting one or two channels. The study [135] focuses on throughput prediction in dense WLANs with channel bonding. The SL techniques including ANNs, graph neural networks (GNNs), RF regression, and gradient boosting are employed to build predictors. Using data from the IEEE 802.11ax-oriented Komondor network simulator, these predictors are trained and validated. The accuracy achieved demonstrates the suitability of ML for predicting complex WLAN throughput. Similarly, the paper [136] predicts Wi-Fi performance using a GNN model that integrates topology information of the deployment. Additionally, the study [137] addresses the issue of collisions with hidden stations in channel bonding scenarios. In this paper, Smart Bond is proposed, employing a recursive neural network called Metropolis-Hastings generative adversarial network (MH-GAN), to predict neighboring BSS activity. Results indicate that Smart Bond reduces the probability of transmission errors due to hidden stations.

2.2.2.5 Multi-Link Operation, Network MIMO, and Full-Duplex

ML techniques play a vital role in enhancing various advanced mechanisms such as multi-band WLAN operation [138], multi-AP coordination for network MIMO [139], and in-band full-duplex [140]. Both RL and SL techniques are employed. For instance, DRL optimizes distributed MIMO transmissions by considering channel allocation and AP clustering [139]. Likewise, NNs predict channel states to enhance the performance of multi-band WLANs [138] and identify groups of stations facilitating full-duplex communication at APs [140]. Below, we delve into these solutions in more detail.

1. **Multi-link Operation:** Multi-link operation allows a WLAN device to transmit over multiple interfaces concurrently, either on the same or different bands. This capability, currently being developed in the IEEE 802.11be task group, enhances transmission efficiency. In the synchronous version, idle interfaces bond together for transmission when any active backoff instance reaches zero. However, if some interfaces are busy but expected to become idle soon, it may be more efficient to wait and aggregate these links instead of transmitting immediately using a single interface. To address this uncertainty, [138] proposes a solution using a probabilistic neural network (PNN) to predict when an interface will become idle. Additionally, [141] explores multi-band operation combined with Hybrid Automatic Repeat Request (HARQ) to enhance packet retransmission efficiency. SL determines whether retransmissions should use the same band, a different one, or all available bands simultaneously, leading to improved network utilization and higher throughput.

2. **Network MIMO:** The paper by [139] explores the challenge of optimizing channel allocation and AP clustering in distributed MU-MIMO for Wi-Fi networks. This problem is addressed using DRL, aiming to maximize per-user throughput. Given the NP-hard nature of both problems, heuristic solutions dominate the literature. Their DRL framework, utilizing a DNN agent and a distributed MIMO Wi-Fi simulator, achieves a 20% enhancement in user throughput. Furthermore, it can concurrently optimize multiple objectives, including throughput and fairness.

3. **Full-Duplex Communication:** Full-duplex communication, particularly in WLANs, enables simultaneous transmission and reception, effectively doubling channel capacity. A significant challenge is the user pairing problem, wherein groups of stations must be identified to enable the AP to transmit to one while receiving from another. Addressing this combinatorial problem, [140] employs a

DSL architecture. This approach offers the advantage of not requiring re-training of the NN when the input length changes within an expected range. Results demonstrate that the DSL-based solution surpasses low-complexity methods like greedy and random assignment.

2.2.3 Wi-Fi Connectivity and Traffic Management

Connectivity management stands as a crucial task in Wi-Fi networks, encompassing tasks such as channel allocation, band selection, and AP selection. The complexity of this task arises from the fact that altering the configuration of a single link not only impacts its performance but can also affect neighboring networks, particularly in densely deployed environments. ML-based approaches are employed to address the challenges of connectivity management, tackling subtasks related to optimizing channel allocation and other parameters. Additionally, in this context, ML-based methods prove valuable for predicting next user connectivity based on historical data derived from human activity detection. Furthermore, ML approaches play a role in forecasting future traffic loads and assessing the health of Wi-Fi link connections. These techniques enable proactive network configuration updates, minimizing outage probability, and enhancing user QoE, especially in scenarios involving rapid changes in communication conditions [33, 145]. Table 2.4 illustrates a summary of related work that has used ML in this area of Wi-Fi networking.

Table 2.4. Summary of related works for Wi-Fi connectivity and traffic Management.

Wi-Fi Aspects	Ref.	ML Application	Novelty
Channel and Band Selection	[146]	Selecting AP for association	Applying MAB for AP selection
	[147]	Selecting AP for association	New Approach for AP selection to reduce network dynamics
	[148]	Selecting Channel and AP	Joint channel allocation and AP selection
	[149]	Selecting AP for association	Apply SL for AP selection
	[150]	Selecting channel	Applying traffic prediction methods for channel allocation
	[151]	Selecting APs for association	Applying DQN to multi-AP association
	[152]	Selecting AP for association	Centralized SDN-based solution
	[153]	Selecting AP for association	Taking into account connection establishment
	[154]	Selecting AP for association	Using frame aggregation characteristics to derive expected throughput
	[155]	Handover decision	Monitoring RSSI patterns for upcoming handover
	[156]	Handover decision	Predicting user location, AP load, and signal strength to preserve QoS during handover
	[157]	handover decision	Considering wireless signal spatial and temporal characteristics
Next User Connectivity	[158, 159]	Selecting next connectivity for association	Applying SVM to uncover regularity for next cell prediction
	[160]	Selecting next connectivity for association	Applying Bayesian Model to uncover regularity for next cell prediction
	[161]	Selecting next connectivity for association	Applying ANN to uncover regularity for next cell prediction
Management Framework	[162]	Optimizing slice configuration	Applying DRL to network slicing
	[163]	AP load prediction	Enabling APs dynamically based on load predictions
Health of Wi-Fi Connection	[164]	Classifying Wi-Fi problems	Providing real-time diagnostics for Wi-Fi problems
	[165]	Classifying Wi-Fi problems	Providing real-time diagnostics for Wi-Fi problems
	[166]	Classifying and predicting link status	Creating analysis framework to decide for user.
	[167]	Classifying Wi-Fi problems	Providing real-time diagnostics for Wi-Fi problems
	[168]	Evaluating QoE of video streaming	Applying ML to estimate QoE by monitoring Wi-Fi parameters
	[169]	Evaluating QoE of video conferencing	Applying ML to estimate QoE by monitoring Wi-Fi parameters
Wi-Fi Traffic	[170]	Predicting transmission throughput	Applying ML for Wi-Fi throughput
	[171]	Predicting traffic intensity	Applying SVM to Wi-Fi traffic prediction
	[172]	Predicting network congestion	Applying both regression and clustering to predict congestion

2.2.3.1 Channel and band selection

In the context of Wi-Fi connectivity management, channel allocation stands out as a critical challenge in dense Wi-Fi networks. These networks face the task of sharing a limited set of available channels among numerous co-located Wi-Fi BSSs. Inadequate channel allocation leads to significant contention among APs and stations, resulting in reduced throughput for each station. The primary research goal in proposed solutions is to allocate channels in a manner that prevents interference among APs using the same channel and avoids assigning the same channel to highly loaded APs (a form of load balancing). It is important to note that in scenarios with variable traffic load, channel allocation must be performed periodically. ML-based algorithms offer solutions to the challenges of channel and band selection. These algorithms provide models capable of considering changing interference relations (e.g., due to station mobility) and variable traffic loads (e.g., as a result of stations transitioning between active and passive states) [33].

1. **AP Selection and Association:** the widespread deployment and concentration of Wi-Fi networks often result in the presence of multiple overlapping Wi-Fi cells in the same spatial area. Consequently, a station must make a decision on which of the identified APs to connect to. The 802.11 association method relies on stations selecting the AP with the strongest signal. However, this simplistic approach can lead to the inefficient use of some APs and overcrowding of others. As a result, various approaches for AP selection and load balancing have been extensively investigated to enhance network throughput. For instance, [146, 147] propose a decentralized AP selection procedure in which stations utilize a Multi-Armed Bandit (MAB)-based approach to dynamically learn the optimal mapping between APs and stations. This procedure aims to distribute stations evenly among available APs. Each station independently explores different APs within its coverage range and selects the one that best meets its requirements. An innovative opportunistic ε-greedy approach with stickiness halts exploration once a suitable AP is found. The station then remains associated with that AP while satisfied, only resuming exploration after several unsatisfactory association periods. Results indicate that this approach increases the number of satisfied stations and the aggregated network throughput by up to 80% in the case of dense AP deployments. In a related work, [148] investigates Multi-Armed Bandit (MAB)-based solutions for decentralized channel allocation and AP selection challenges in enterprise WLAN settings. The proposed approach involves the use of agents by both APs and stations. Employing a Thompson sampling algorithm, these agents explore and learn: (i) at the AP side, the optimal channel to use, and (ii) at the station side, the best AP to associate with. In the paper, results from a custom-built simulator reveal that the learning-based approach consistently outperforms the static one across varying network densities and traffic requirements.

   In the study [149] a cognitive AP selection scheme is introduced, where stations choose an AP expected to deliver optimal throughput based on past performance. This scheme, part of the SL family, employs a multi-layer feed-forward neural network (MFNN) to learn the correlation between environmental conditions (e.g., SNR, probability of failure, beacon delay) and achieved performance (throughput). Results from an 802.11 testbed demonstrate the effectiveness of the approach, surpassing legacy AP selection strategies in diverse scenarios. Similarly, two papers of [150, 173] employ a predictive performance approach within the constraints of AP selection. Moreover, paper [151] introduces an interesting RL-based approach for user-to-multiple AP association. Two distributed association methods, leveraging deep Q-learning (DQL), empower stations to autonomously learn the optimal set of APs to connect to. This learning process relies solely on local knowledge of the wireless environment or limited feedback from the APs. Each device is equipped with multiple wireless interfaces. The goal is to maximize the long-term sum-rate while adhering to various constraints, such as AP load or application QoS constraints. Numerical evaluations demonstrate that the algorithms enhance targeted objectives and improve fairness among applications. The paper [152] proposes a centralized approach, introducing an RL-based client-AP association algorithm aimed at improving aggregated throughput in dense Wi-Fi networks. This Q-learning-based algorithm is centrally deployed within a Software-Defined Network (SDN) controller, managing the associations of new users and executing re-associations of connected stations. Simulation results demonstrate that the proposed approach surpasses the standard 802.11 association procedure, particularly in scenarios with non-uniform user distributions, and performs comparably well in cases of uniform distributions.

The study [153] conducts a comprehensive study using extensive measurements to identify factors influencing the Wi-Fi connection set-up process. Analyzing data from 0.4 billion Wi-Fi sessions collected via the Wi-Fi Manager mobile app from 5 million mobile devices, a 45% failure rate is observed in Wi-Fi connection attempts, with approximately 5% taking more than 10 seconds. In response, an SL-based AP selection algorithm is proposed that significantly enhances Wi-Fi connection setup performance. The algorithm utilizes RF to categorize candidate APs into slow or fast sets, considering features such as the hour of the day, Received Signal Strength Indicator (RSSI), mobile device model, AP model, and encryption status. Using this classification, a station avoids connecting to APs in the slow set. Evaluation results demonstrate that this approach reduces connection failures to 3.6% and improves connection setup time by over 10 times. The study [154] highlights the effectiveness of frame aggregation in providing a concise representation of anticipated throughput, enhancing AP selection. The distinctive characteristics of subframes within frame aggregation encapsulate the utilization, interference, and backlog traffic pressure for an AP. Employing an SL-based approach, straightforward regression models (utilizing linear regression and DT regression) predict the expected throughput of APs to improve AP selection. The outcomes reveal a prediction accuracy exceeding 80%.

2. **Station Handovers:** in mobile scenarios, it is common for a station to move from the coverage area of one AP to that of another. In such cases, the station needs to execute a handover from the old AP to the new one. It is crucial to make the handover decision early to prevent low data rates or connectivity outage. ML methods can predict network conditions, facilitating accurate handover decisions. For instance, [155] utilizes ML to predict upcoming handovers. In this context, an AP monitor the RSSI of connected stations is made and a NN is used for pattern recognition in the RSSI evolution. This approach demonstrates good prediction accuracy and resilience to noise, speed variations, and fading phenomena. Moreover, [156] proposes ABRAHAM (mAchine learning Backed multi-metRic Handover AlgorithM), a proactive handover algorithm based on ML that leverages multiple metrics to forecast the future positions of stations and the upcoming AP load. By employing LSTM, it predicts future RSSI values. These predictions are then utilized to optimize AP load by strategically handing over stations, ensuring the preservation of QoS and QoE metrics. ABRAHAM demonstrates a 139% increase in overall throughput compared to the traditional 802.11 handover algorithm. In the paper [157] a handover management approach is designed for dense WLAN networks, utilizing Deep Reinforcement Learning (DRL), specifically a deep Q-network. The approach empowers the NN to learn

from user behavior and network conditions, adjusting its learning in dynamic and densely populated WLANs. The handover decision is formulated as a Markov Decision Process (MDP), taking advantage of the temporal correlation property, and relies on real-time network statistics to make informed decisions. Simulation results indicate that this solution effectively enhances the data rate during the handover process, surpassing the performance of the 802.11 handover scheme.

2.2.3.2 Next User Connectivity Prediction

In this section, we mention connectivity management from a distinct perspective, focusing on human activity detection rather than the network access side. Next cell/AP and mobility prediction emerge as potent tools for network operators seeking to optimize overall network performance [145]. In conventional approaches aimed at preventing connection disruptions during handovers, each base station (BS) typically allocates a fixed set of resources for prospective incoming users. However, inadequate resource reservation may lead to inefficiencies or suboptimal user experiences. To mitigate the challenges posed by user mobility, an effective strategy involves the implementation of mobility prediction [174]. Successful forecasting of user mobility enables network operators to employ a proactive (or preemptive) reservation policy [175], ensuring seamless service continuity while avoiding excessive resource allocation. The successful implementation of passive reservation relies on the effectiveness of user mobility prediction, a pivotal factor influencing resource allocation performance in wireless networks. Recognizing that humans often exhibit recurring patterns of behavior, it becomes feasible to predict individual movements based on historical location information. While studies have demonstrated a remarkable 93% predictability of user mobility [176], achieving such high precision poses challenges, particularly in contexts where sufficient contextual data and specific movement patterns are lacking. In the following, we describe the associated issues and characteristics concerning the predictability of next-cell/AP transitions [177].

In our daily lives, our movements are not entirely random but often have direction or specific destinations. People tend to follow particular routes regularly, such as students commuting to school or office workers following their weekday commute paths. In these trajectories, the cells to which mobile users connect have a high probability of being fixed. By observing users' mobility patterns over a certain period, it becomes possible to identify the regularities in their movements and predict the subsequent cells they will connect to while in motion. The work conducted by [176] explores the predictability of human behavior by measuring the entropy of numerous trajectories of mobile phone users. In this study, it is discovered that the potential predictability of user mobility could reach up to 93%, and this predictability lacks variability. Essentially, there is strong regularity in human mobility, theoretically allowing for the development of accurate prediction models. Vehicular mobility predictability has also been examined in [178], revealing that about 78%-99% of the location and over 70% of the staying time are predictable. The study highlights the strong regularity in everyday vehicular mobility, providing a basis for the development of practical prediction algorithms.

User movement can be seen as a combination of regular and random movements [158]. Regular movements exhibit patterns that can be discovered through users' long-term (weeks or months) historical trajectories. The literature suggests various prediction approaches to uncover regularity for next cell/AP prediction, including Markov chain [179], HMM [180], Bayesian network [160], SVM [159], and ANN [161].

However, when a user arrives at a new location, there is insufficient historical data to analyze its mobility pattern, resulting in irregular movement. While predicting random movements is more challenging due to their irregular nature, determining the next cell/AP is possible to some extent using various strategies, such as real-time monitoring [181].

In terms of spatial granularity, next-cell/AP prediction, unlike position and trajectory predictions, involves a broader location forecast. Its goal is to identify the cells/APs to which users will be connected, rather than pinpointing accurate locations. Consequently, predicting a cell/AP or sequence of cells/APs is generally considered easier than predicting fine-grained positions or trajectories.

2.2.3.3 Management Framework

The management of Wi-Fi networks involves adjusting numerous parameters across various devices, which can be challenging. Several management frameworks utilize AI-based control planes to simplify this task.

In the paper [60], an AI-based operating system called aiOS is introduced for 802.11-based Software-Defined WLANs (SD-WLANs). This system employs ML toolboxes to create a global intelligence platform, enhancing network performance through adaptive frame length selection. Tests in a real-world environment demonstrate up to a 55% improvement in network throughput.

The study in [162] utilizes DRL to dynamically optimize the configuration of network slices in Wi-Fi networks. Each slice configuration comprises various parameters such as CCA sensitivity level, MCS, and transmit power level. Unlike traditional approaches, the chosen action does not entail absolute configuration values but rather adjusts the current parameter values. A basic DQN agent with DDQN, experience replay, and fitted Q-learning techniques are enhanced to improve convergence speed and stability. Simulations in the ns-3 simulator demonstrate that this approach achieves optimal performance comparable to exhaustive search methods. Additionally, DDQN enables real-time optimization without requiring detailed AP deployment information or knowledge about coexisting networks.

The paper discussed in [163] utilizes data from a university Wi-Fi system comprising 8,000 APs and 40,000 active users. Through a detailed analysis, it examines the AP load and traffic throughput. It notes an idle phenomenon where many APs remain unused and identifies a skewed distribution of AP load, with most APs serving few users while a few serve hundreds. To address this, a management system called LAM (large-scale AP management) intelligently switches off unused APs based on user association patterns. LAM employs M algorithms to predict AP load, achieving up to 90% accuracy. This strategy saves over 70% of power energy while ensuring 92% Wi-Fi coverage, resulting in significant cost savings of $59,000 annually for the university Wi-Fi system.

The paper [182] proposes an SDN-based Wi-Fi control system to manage APs. A central controller handles configuration tasks like channel and transmission power settings for the APs. Decisions are made based on learned data, using ML techniques such as reduced error pruning trees (REPTs) to predict both Wi-Fi and non-Wi-Fi activities. By predicting these activities, the system can deploy more effective configurations. The framework significantly reduces channel congestion by up to 47%.

2.2.3.4 Predicting the Health of Wi-Fi Connections

The Wi-Fi networks operating in unlicensed bands are experiencing overcrowding due to numerous deployments managed by different users. This situation worsens issues like hidden terminals, flow starvation, and performance anomalies. Detecting these problems in real-world scenarios is challenging because they manifest as performance degradation, which can stem from various causes needing distinct solutions. ML emerges as a suitable approach for detecting specific impairments because it can process vast amounts of raw measurement data and learn to identify the current operational state, employing classification methods. The study presented in [164] introduces Wi-Dia, an automatic diagnostic tool designed to identify the causes of performance issues in Wi-Fi networks by understanding the wireless operating environment. Wi-Dia adopts a data-driven approach and employs M techniques to classify various Wi-Fi problems like hidden stations and flow starvation. It utilizes network topology features and measures channel utilization without disrupting regular network functions. The classifier of Wi-Dia is trained using both simulated and experimental data, leveraging the flexibility of network simulators and the realism of wireless testbeds. Results demonstrate that Wi-Dia achieves high accuracy in detecting Wi-Fi issues in real-world scenarios.

In a study by [165], the detection of Wi-Fi performance issues, such as contention with other devices, low SNR, hidden terminals, or capture effect, is addressed. A centralized Wi-Fi controller gathers performance metrics from connected APs, including normalized channel access (NCA) and frame delivery ratio (FDR). NCA represents the ratio of channel access attempts to the maximum possible attempts calculated using 802.11 models, while FDR indicates the ratio of successful transmissions to channel access attempts. Through data modeling and feature extraction, four different algorithms—DT, RF, SVM, and kNN—are employed for classification. After parameter optimization, the algorithms achieve a high detection accuracy of 99.2% using the kNN algorithm.

In the study by [166], WiNetSense is introduced, which is a centralized sensing framework. It gathers Wi-Fi link quality statistics like RSSI from network devices to establish a global network topology and real-time network health information. ML algorithms such as kNN and NB are then applied to analyze the collected data and predict wireless link health. This information aids in making decisions related to load balancing, seamless handovers, and dynamic power control. Additionally, [167] proposes an anomaly-detection approach utilizing a self-organizing hidden Markov model map. This model, based on a self-organizing map trained via USL, demonstrates enhanced accuracy and sensitivity in detecting anomalies compared to other hidden Markov model (HMM)-based methods, as verified through simulations.

Furthermore, papers [168] and [169] present a new ML-based method for assessing the perceived QoS in video streaming and video conferencing, respectively, using only network performance data from Wi-Fi APs. The research utilizes datasets containing specific network performance metrics from 802.11n/ac/ax standards, paired with mean opinion scores. Various ML algorithms such as logistic model tree (LMT), reduced error pruning tree (REPT), Naive Bayes Tree (NBT), and MLP are trained on these datasets, achieving high accuracy ranging from 93% to 99% in estimating QoS classes. Among these algorithms, LMT and REPT are identified as the most suitable for video streaming and conferencing, respectively, considering accuracy, interpretability, and computational efficiency. Furthermore, the developed ML model can be implemented as a lightweight script on APs for continuous QoS monitoring.

2.2.3.5 Wi-Fi Traffic Prediction

Traffic prediction techniques play a vital role in enhancing network management operations for effective short- and long-term planning. Strategic planning, incorporating methods like traffic forecasting, congestion control, power conservation, bandwidth allocation, and buffer management, contributes to an enhanced user QoE. For example, leveraging predicted traffic patterns enables APs to enhance load balancing and implement effective admission control. Real-time traffic prediction poses a considerable challenge in Wi-Fi networks, given the dynamic nature of channel conditions, evolving network topologies, and unpredictable user traffic patterns. The accuracy of traffic estimation is influenced by various factors, including the total number of users in the network, SNR on the link, and the communication capabilities of both users and APs [170]. In such complex scenarios, ML models provide a solution to handle the diverse conditions of Wi-Fi networks, offering insights that are difficult to obtain through traditional analytical methods.

In the proposed solution of the paper [171], a SVM is utilized to predict traffic evolution, focusing on forecasting one step ahead. Moreover, by recursively applying this one-step-ahead approach, the study extends the prediction to multiple steps ahead. The SVM model, employing a Gaussian radial basis function, is trained with 100 samples to predict the subsequent 100 samples. The application of the SVM model results in a reduction of at least 33% in the error for predicting upcoming traffic compared to the performance of ANN. The study of [170] assesses various ML models—MLP, SVR, DT, and RF—to predict traffic. Training these models involves extracting multiple features from both simulation and real data (e.g., Wireshark network trace). Extracted features include the number of connected users, signal strength, modulation scheme, data rate, inter-arrival time, packet arrival rate, number of re-transmissions, and various other channel parameters. The evaluation is conducted in a Wi-Fi network with 10 users and a single 802.11 AP. The reported prediction accuracy reaches a maximum of 96.2%, 94.5%, 93.3%, and 91% for MLP, DT, RF, and SVR, respectively. Additionally, the study explores the real-time complexity of these models by reporting the time elapsed for each. MLP is identified as the most time-consuming, followed by RF, SVR, and DT. Moreover, the study [172] employs both SL and USL models to predict network congestion levels. Using attributes captured from data, such as the number of clients, throughput, frame retry rate, and frame error rate, SVR and polynomial regressor models predict corresponding values for a specific location, day, and time. These predicted values are then input into an expectation maximization algorithm to predict congestion levels, forming three distinct clusters. Each cluster represents high, medium, and low congestion levels based on the numeric values of the clustered samples. The achieved accuracy is 24%, 50%, and 26% for low, medium, and high levels of network area congestion, respectively.

2.2.4 Other Types of Wi-Fi Scenarios

In addition to Wi-Fi categories (explained in detail previously), there are also types of (i) co-existance scenarios of Wi-Fi with other technologies, and (ii) multi-hop Wi-Fi scenarios, that ML has been used to improve network performance.

- ML-optimized coexistence of Wi-Fi with other technologies: Research into the coexistence of Wi-Fi and cellular technologies has become increasingly popular and appealing. While these technologies are highly advanced, with their latest generations boasting peak data rates in the order of Gbit/s, coexistence scenarios in unlicensed bands (e.g., with LTE-LAA) still rely on relatively basic coexistence schemes based on energy sensing. This reliance results in frequent collisions and significant throughput degradation, sometimes reaching up to 90% [183, 184]. Coexistence schemes face challenges due to the heterogeneity of the underlying technologies, including different MAC and PHY implementations, separate management by operators, and a lack of native support for inter-technology communication in spectrum sharing. Consequently, achieving fair sharing of unlicensed radio resources remains an open challenge [185]. ML is applied to optimize these coexistence scenarios, particularly in areas such as channel sharing, network monitoring, and cross-technology signal classification.

- Multi-hop Wi-Fi deployment: The primary design objective of IEEE 802.11 networks is to function as a single-hop access network (as discussed in detail in previous subsections). However, it can also be employed in various multi-hop scenarios, such as ad hoc or vehicular networks, either using the mainline standards (802.11a/b/g/n/ac/ax) or dedicated amendments (such as 802.11ah for IoT and 802.11p for vehicular networks). ML techniques is also used in multi-hop Wi-Fi deployments, including ad hoc networks, mesh networks, sensor networks, vehicular networks, and relay networks.

2.3 ML Techniques

ML techniques encompass three primary categories: Supervised Learning (SL), Unsupervised Learning (USL), and Reinforcement Learning (RL). SL involves training models on labeled datasets, where each input is matched with a corresponding output label, facilitating the model's ability to learn patterns. Within this realm, classification tasks entail assigning input data points to predefined categories, while regression tasks focus on predicting continuous numerical values. USL, on the other hand, operates without labeled data, seeking to uncover inherent structures and patterns within the dataset independently. Clustering, a prevalent task in USL, groups similar data points based on shared characteristics. Reinforcement learning diverges from the others by enabling the model to learn through iterative interactions with an environment, receiving feedback in the form of rewards or penalties. The objective in reinforcement learning is to develop strategies that maximize cumulative rewards over time.

Databases contain valuable hidden information that can be leveraged for informed decision-making. Within data analysis, classification and prediction emerge as two approaches aimed at extracting models to describe significant data classes or predict future data trends. Through such analysis, we gain a deeper understanding of the data. While classification predicts categorical (discrete, unordered) labels, prediction models continuous-valued functions. Various methods for classification and prediction, proposed by researchers in ML, pattern recognition, and statistics, have been designed. Many existing algorithms assume a small data size and reside in memory. However, recent advancements in data mining research focus on scalable techniques for classification and prediction capable of handling large datasets residing on disk [186]. Table 2.5 provides an summary of various ML techniques employed in this thesis, along with their respective advantages, disadvantages, and the tasks they are best suited for.

A more comprehensive description of the ML algorithms used for classification and prediction in this thesis is presented in the following:

- **Logistic Regression (LR):** is a versatile statistical method primarily used for binary- and multi-class classification problems. Despite its name, it is more com-

Table 2.5. Summary of comparison among different ML techniques used in the thesis.

Algorithm	Advantages	Disadvantages	Suitable Tasks
LR	Simple, interpretable	Assumes linear decision boundary	Classification
SVM	Effective in high-dimensional spaces	Computationally intensive for large datasets	Classification, Regression
kNN	Intuitive, no explicit training phase	Sensitive to choice of k	Classification, Regression
DT	Interpretable, captures non-linear relationships	Prone to overfitting	Classification, Regression
RF	Robust, handles high-dimensional data well	Requires more computational resources	Classification, Regression
NB	Simple, efficient	Assumes independence among features	Classification
NN	Captures complex patterns, adaptive	Requires more data for training, black-box nature	Classification, Regression
ARIMA	Captures temporal dependencies	Requires stationary data	Time Series Prediction
SRNN	Captures temporal dependencies	Prone to vanishing gradient problem	Time Series Prediction
LSTM	Captures long-term dependencies	More computational resources	Time Series Prediction
GRU	Efficient architecture	shorter memory compared to LSTM	Time Series Prediction
CNN	Captures complex features	Not originally designed for time-series data	Time Series Prediction
CNN-RNN	Integrates spatial and temporal features	Complexity	Time Series Prediction
Transformer	Captures intricate dynamics	Complexity	Time Series Prediction

monly employed for classification rather than regression tasks. The fundamental idea behind LR is to model the relationship between a binary dependent variable (target) and one or more independent variables (features) by estimating probabilities using a logistic or Sigmoid function. In LR, a linear combination of input features, weighted by coefficients, is computed. This linear combination is transformed using the Sigmoid function, producing probabilities that an instance belongs to the positive class. A decision threshold is then applied to classify instances based on these probabilities. Training involves adjusting the coefficients to minimize the difference between predicted probabilities and actual class labels, typically using Maximum Likelihood Estimation. LR is simple, interpretable, and works well when the decision boundary is linear [186–188].

- **Support Vector Machine (SVM):** stands as a versatile and potent M algorithm adept at handling classification and regression tasks. Its fundamental principle involves determining an optimal hyperplane that maximally separates data points belonging to different classes in a feature space. The emphasis lies in creating a margin, the space between the hyperplane and the nearest data points of each class, with these critical data points termed as Support Vectors. For classification, SVM initially assumes a linear separation, striving to find the hyperplane that maximizes the margin while minimizing classification errors. The optimization process involves formulating a cost function, penalizing misclassified points. SVM extends its utility to non-linearly separable data through the kernel trick, wherein kernels transform input features into higher-dimensional spaces, enabling the discovery of hyperplanes in these transformed dimensions. In the realm of prediction problems, SVM adapts seamlessly for regression tasks by modifying its formulation. Instead of classifying into discrete categories, SVM regression predicts continuous outputs. The concept of epsilon-support vectors is introduced to allow for a certain degree of error or deviation from predicted values. It is effective in high-dimensional spaces and can handle non-linear data using kernel functions [186–189].

- **k-Nearest Neighbors (kNN):** stands as a versatile M algorithm used for both classification and regression tasks. Its methodology revolves around the concept of proximity in the feature space. For classification, when tasked with categorizing a data point, k-NN calculates the distances to all other data points in the dataset. Subsequently, it selects the k-nearest neighbors based on the smallest distances and employs a majority voting mechanism to assign a class label to the target data point. In regression tasks, k-NN follows a similar procedure, computing distances and selecting k-nearest neighbors. However, instead of majority voting, it predicts the target value for the data point as the average or weighted average of the target values of its k-nearest neighbors. Notably, k-NN does not involve an explicit training phase; rather, it memorizes the entire dataset, contributing to its simplicity and ease of implementation. The choice of the crucial parameter, k (number of neighbors), plays a pivotal role in the performance of k-NN. A smaller k results in a more flexible model, potentially sensitive to noise, while a larger k provides a smoother decision boundary, albeit with a risk of overlooking local patterns. Additionally, the distance metric used, typically Euclidean distance, can be adapted based on the characteristics of the data.It is intuitive and works well when data is locally clustered [186–189].

- **Decision Tree (DT):** is a ML algorithm utilized for tasks involving both classification and regression. This algorithm adopts a tree-like structure, wherein internal nodes represent features or attributes, branches represent decision rules, and leaf nodes signify the final outcomes. Understanding how DTs function in both classification and prediction problems involves several key steps. For classification tasks, the process starts with the entire dataset, and the algorithm selects the most informative feature to split the data based on criteria such as Gini impurity or information gain. This splitting process continues recursively, creating subsets until reaching stopping criteria, such as a maximum tree depth or a minimum number of samples in a leaf node. Each leaf node corresponds to a class or label, and the path from the root to a leaf node represents the decision rules for classifying a data point. In regression tasks, DTs operate similarly but focus on predicting numerical values. The algorithm selects features to split the data based on criteria like mean squared error. The recursive splitting process continues until the stopping criteria are met, and the leaf nodes now contain predicted numerical values. To make predictions for a new data point, one traverses the tree from the root to a leaf node following the decision rules, and the predicted value is associated with that leaf. DT is interpretable and can capture non-linear relationships in the data. However, it is prone to overfitting [186–189].

- **Random Forest (RF):** is based on building multiple DTs, generated during the training phase, and merge them together in order to obtain a more accurate and stable prediction or classification [27, 186]. Different from the single DT algorithm, where each node of the tree is split by searching for the most important feature, in RF, additional random components are included (i.e. for each node of each tree the algorithm searches for the best feature among a random subset of features). Once all the trees are built, the result of the prediction or classification corresponds to the most occurred prediction from all the trees of the forest. RF is robust, handles high-dimensional data well, and reduces overfitting [186–189].

- **Naive Bayes (NB):** is a probabilistic ML algorithm renowned for its efficacy in classification tasks, particularly in scenarios involving text classification and spam filtering. Its foundation lies in Bayes' theorem, a mathematical formula describing the probability of an event based on prior knowledge of related conditions. The “naive” aspect of Naive Bayes stems from the assumption of conditional independence among features given the class label. This simplifying assumption, though not always strictly met in real-world scenarios, facilitates computational efficiency and ease of implementation. In the classification process, Naive Bayes undergoes training where it calculates the prior probabilities of each class and the conditional probabilities of features given the class. During prediction, the algorithm computes the posterior probability of each class for a new instance, using Bayes' theorem. The class with the highest posterior probability is then assigned as the predicted class for the given instance. NB is simple and efficient [186–189].

- **Neural Networks (NN):** in this case, the prediction is done by means of a feed-forward NN that consists on an input layer, one or more hidden layers and an output layer [186]. Each layer is made up of processing units called neurons. The inputs are fed simultaneously into the units of the input layer. Then, these inputs are weighted and are fed simultaneously to the first hidden layer. The outputs of the hidden layer units are input to the next hidden layer, and so on. The training of this network involves a SL technique known as back-propagation, where errors in predictions are iteratively minimized by adjusting the weights during the training process, ensuring the NN refines its ability to classify and make accurate predictions over time [190].

- **ARIMA Algorithm:** AutoRegressive Integrated Moving Average (ARIMA) algorithm is a popular time series prediction method [191]. It works by capturing and modeling the patterns and trends within a time series dataset. The "AR" stands for AutoRegressive, indicating that the model considers the relationship between a data point and its previous values, helping to account for temporal dependencies. The "I" stands for Integrated, which implies differencing the data to make it stationary, enabling more accurate modeling. The "MA" stands for Moving Average, suggesting the consideration of past forecast errors to improve predictions. The order of an ARIMA model is denoted as (p, d, q), where "p", "d", "q", are the autoregressive, differencing and moving average order, respectively. In our study, in order to apply ARIMA to measurements of the i-th target AP, a two-step process is followed. Initially, the appropriate model order is determined using the auto-arima function. Once the optimal order is identified, we utilize ARIMA with that specific order in conjunction with a "Recursive Sampling" technique [192] (to increase prediction performance) to yield our prediction results. Recursive Sampling in the context of ARIMA refers to a technique where the prediction model is applied sequentially, with each new prediction used to update the model for next prediction. It involves iterating through the time series data, making one-step-ahead forecasts, and then incorporating each forecasted value into the dataset for the subsequent prediction. This iterative process continues until the desired forecast horizon is reached.

- **RNN Algorithms:** Recurrent Neural Networks (RNN) are a family of Neural Networks that have been successfully used for time-series prediction since they recurrently make use of data that was previously fed into the neural network resulting in a higher prediction accuracy. One of the first considered approach was SimpleRNN (SRNN), that not only considers the current input, but it also takes into account the information from previous time steps [193]. However, SRNN suffers from the vanishing gradient problem, because it can only remember the most recent information. Therefore, in order to make use of longer historical measurements, Long Short-Term Memory (LSTM) was proposed. LSTMs process sequential input data element by element. The network maintains memory cells to store and manage information over time, and three gates—input gate, forget gate, and output gate—regulate the flow of information into and out of these cells. This gating mechanism allows LSTMs to selectively remember or forget information, making them adept at learning temporal patterns and dependencies. In fact, LSTM cell is able to control the data flow by forgetting unnecessary information and keeping the important one. The output layer of the LSTM produces a continuous value representing the prediction for the next time step in the sequence [194, 195]. However, more computational power and time is required to process the data due to the algorithm complexity. Gated Recurrent Unit (GRU) is another variant of RNNs that, like LSTM, addresses the challenge of modeling sequential dependencies. GRU simplifies the architecture compared to LSTM by combining the memory cell and hidden state into a single state vector. It introduces two gates, an update gate and a reset gate, to control the flow of information. The update gate governs how much of the previous state information should be carried forward, while the reset gate determines how much of the past information should be forgotten. GRU's design results in a more streamlined architecture, reducing computational complexity and making it more computationally efficient compared to LSTM. While both LSTM and GRU share the fundamental goal of capturing long-term dependencies, GRU's architecture offers a trade-off between efficiency and expressiveness, making it a suitable alternative for tasks involving sequential data and prediction. However, GRU may not have such a long memory as LSTM [196].

- **CNN Algorithm:** Convolutional Neural Network (CNN) is also a type of NN which is not originally designed for modeling time-series data. However, several approaches [194][197] have adapted CNN algorithms for time series data processing leading to good performance results. A possible way to adapt the input data for CNN algorithm is to divide the sequence into multiple input/output patterns, where a certain number of time steps are used as input and a time step is used as output to predict the next step. CNN will make use of three kinds of layers (1. filter layer, 2. pooling layer, 3. fully connected layer), combined together with desired sequence to generate the output. It captures complex features by convolving input with some filter, pooling layer that compacts the learned patterns for less computational resource, and fully connected layers which are designed to extract non-linear features [194, 197].

- **CNN-RNN hybrid algorithm:** this architecture integrates CNN and RNN to take advantage of the good capabilities of CNN to learn spatial domain features and RNN to capture temporal dependencies [177]. Previous successful applications of this architecture include activity recognition and video description like in [198].

- **Transformer algorithm:** A model architecture forsaking recurrence, relies solely on an attention mechanism to establish global dependencies between input and output [199]. The Transformer architecture consists on an encoder and a decoder stack [200]. The encoder is constructed using a stack of identical layers, each containing two sub-layers: a multi-head self-attention mechanism, and a position-wise fully connected feed-forward network. Residual connections are integrated around each of these sub-layers, followed by layer normalization. All sub-layers in the model, and the embedding layers, generate encoder outputs to support these residual connections. Furthermore, the decoder is composed of a stack of identical layers, similar to the encoder. Each encoder layer includes the two sub-layers present in the encoder, with the addition of a third sub-layer for multi-head attention over the output of the encoder stack. Residual connections and layer normalization are applied around each sub-layer. Moreover, positions are prevented from attending to subsequent positions through appropriate masking in the decoder self-attention sub-layer. The concept of self-attention, entails an attention mechanism that establishes relationships among different positions within a single sequence to generate a comprehensive representation of the sequence. The attention function involves mapping a query and a set of key-value pairs to produce an output, wherein all entries involved query, keys, values and output, are represented as vectors. The output is then computed as a weighted sum of the values, with the weight assigned to each value determined by a compatibility function that relates the query to its corresponding key [200]. In the context of time series data, Transformer diverges from traditional sequence-aligned models by eschewing ordered sequential processing. Instead, it operates on an entire sequence of data and leverages self-attention mechanisms to discern dependencies within the sequence. As a result, Transformer-based models exhibit the capacity to capture intricate dynamics inherent in time series data, thereby surpassing the capabilities of conventional sequence models in handling complex temporal Patterns [199].

2.4 General Framework of Thesis Goals

In this section, we present a framework for the AI/ML control loop utilized in our study. This framework is depicted in Figure 2.3, utilizing the infrastructure-based Wi-Fi network shown in Figure 2.2.

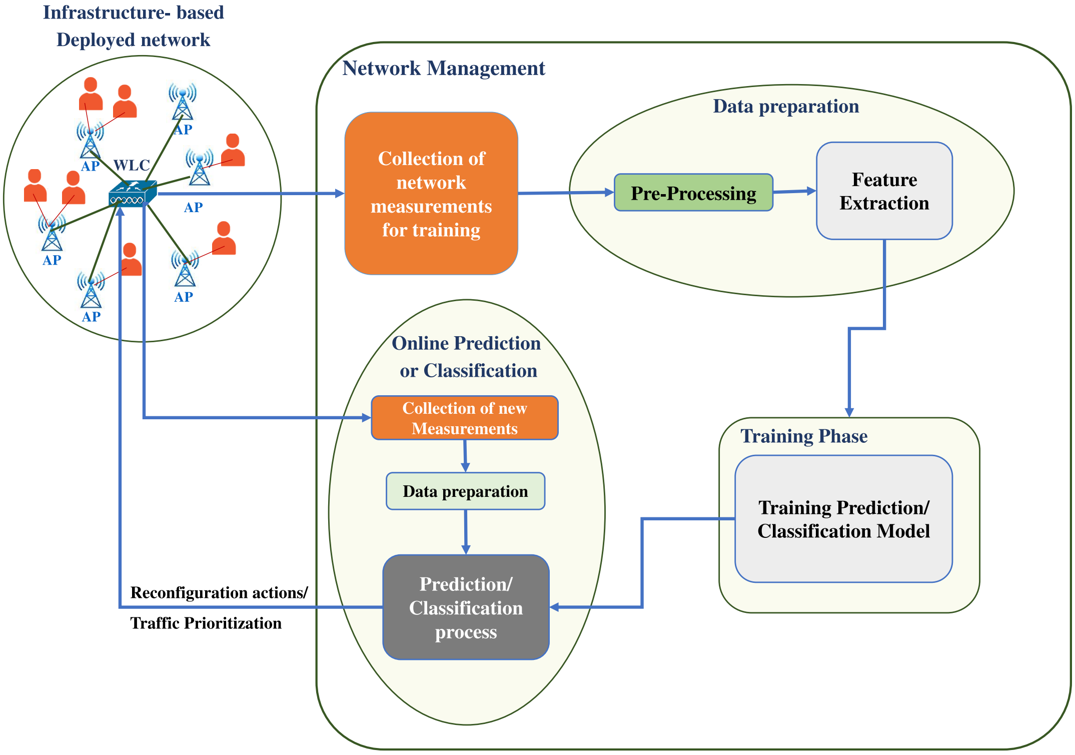


Fig. 2.3. AI/ML control loop of three objectives.

As depicted in Fig. 2.3, our framework consists of various entities responsible for managing the performance of this Wi-Fi network. These entities encompass management tools that control Wi-Fi operations. Additionally, we implement processes or optimization loops wherein network measurements such as association time, session duration, AP names, connected users, signal strength, traffic load and packet traces are collected and preprocessed. Subsequently, features are extracted from some computational operations. Then, the training model is made for predicting the next AP to which users will connect (Chapter 3), prediction of the amount of traffic data transmission in the user connections (Chapter 4), or classifying the types of traffic for the AP traffic prioritization (Chapter 5). The trained model is used for prediction (Chapter 3 and 4) and classification (Chapter 5) of future value of new measurements collected from the deployed network after data preparation. The particularization of this Figure is also presented in each chapter. These predictions and traffic classification serve as valuable inputs for making informed decisions or taking actions within the network.

In particular, if we can detect or predict the next connectivity and predict the associated traffic load for each connection, we can obtain valuable insights for implementing network reconfiguration actions. These actions encompass tasks such as channel allocation, AP selection, and handovers for predicting the next user connectivity. Additionally, they involve congestion control, load balancing, and resource allocation for traffic prediction. For instance, adjusting the transmitted power of the APs can aid in network reconfiguration efforts. In the context of traffic classification, once implemented in a real system, the objective is to prioritize packets to enhance QoS.

Chapter 3

UE Connectivity Prediction

3.1 Introduction

Prediction of UE connectivity within Wi-Fi communications networks is crucial for optimizing resource allocation, enhancing QoS, and improving network management strategies. Forecasting the next locations of users within Wi-Fi networks empowers operators to allocate resources effectively and enhance user experiences. This prediction capability is particularly valuable in scenarios where users move between different APs within a network, influencing network load distribution and QoS provision. Wi-Fi communication systems often exhibit distinct patterns in user mobility, characterized by periodic movements and transitions between APs. However, conventional ML algorithms face challenges in effectively capturing these mobility patterns due to the complex and dynamic nature of Wi-Fi network environments. Sparse and incomplete data resulting from irregular user reporting further complicates prediction tasks, limiting the ability of ML models to accurately predict user movements between APs.

Drawing from the insights gained from Wi-Fi communication networks, our work is guided by the understanding that predicting the next user location (AP) relies not only on current sequential activities but also on periodic patterns evident in historical data. Recognizing this, we propose various approaches to assess the impact of different periodic factors on user mobility prediction. Additionally, incorporating time representations into the embedding process empowers ML models to capture the time-dependent nature inherent in Wi-Fi network dynamics. By integrating both sequential and periodic information, our approach aims to enhance the accuracy of next user location prediction within Wi-Fi networks. Within this context, the novelty of this work is the proposal of a methodology that predicts the future user connections to the different APs of a Wi-Fi network according to historical user records. This prediction leverages information from the user's previous AP connections to capture periodical patterns in user activity across different time intervals (e.g., hourly, daily, and weekly behaviors). By comprehensively considering these temporal factors, our model achieves better prediction performance, capturing the semantic motivation behind user mobility patterns. An extensive evaluation of this methodology has been done with a large data set of measurements of a real Wi-Fi network.

The remaining of the chapter is organized as follows. Section 3.2 provides an overview of the existing literature. Section 3.3 presents the proposed AP prediction methodology. The results are presented in Section 3.5, while Section 3.6 summarizes the conclusions.

3.2 Related Work

Multiple applicability examples of data monitoring and analytics in the context of Wi-Fi or cellular networks can be found in the literature. As an example related to mobile cellular networks and concerning the characterization of user habits, the collection of the Base Station and the mobile terminal communication activity (messages, calls, etc.) has been used for urban and transportation planning purposes to identify daily motifs, given that human daily mobility can be highly structural and organized by a few activities essential to life [201]. Similarly, [202] proposed a methodology to partition a population of users tracked by their mobile phones into four predefined user profiles: residents, commuters, in transit and visitors. Applications envisaged are traffic management, to better understand how traffic is affected by the residents mobility compared to the commuters, or studying how the city is receiving people from outside and how their movements affect the city. The paper [203] proposed an agglomerative clustering to identify user's daily motifs according to the cells in which the user is camping during the day. Real measurements obtained from a 3G/4G network were used. The obtained knowledge related to the network status, the performance of the services, user habits, user requirements, etc. can also be useful for supporting different decision-making processes over the network (e.g. adjusting the usage of the network resources) which will lead to more efficient network management tasks related to network reconfiguration and optimization. As an example, the use of prediction methodologies for identifying the future Small Cell/Access Point (SC/AP) which the user will be connected to, together with an estimation of the future user traffic volume and perceived user performance may provide a more accurate future user characterization. This can be useful for carrying out a more proactive network reconfiguration approach. In the context of Wi-Fi networks, the prediction of APs to which a client will connect in the future can be useful for a more efficient Pairwise Master Key (PMK) caching or Opportunistic Key Caching (OKC) techniques which can reduce the time for re-authentication when roaming to a new AP [204]. The extracted knowledge related to user location and mobility, user habits and interests, etc. can also be useful for companies for commercial purposes. As an example, a reliable prediction of future user location and length of stay connected to the different SC/AP enables the use of Location Based Advertising mechanisms. This presents the possibility for advertisers to personalize their messages to people based on their location and interests [205].

Furthermore, in the context of wireless communication, two common problems related to device mobility are handover prediction and AP selection [34, 206]. To address these two issues, literature employs mathematical expressions (e.g., Markov Model [207, 208] and Hidden Markov Model [209–211]) and ML techniques [34, 159, 212–214].

In the context of theoretical frameworks for AP selection, the paper [207] introduces an innovative mobility prediction algorithm that leverages both long-term and short-term user trajectories. The algorithm captures the regularity in user movements by training a Markov Renewal Process using long-term trajectory data. Short-term trajectory data within the current cell is then incorporated to account for potential randomness in user behavior. Each neighboring cell is assigned two probabilities for being the next crossing cell—one from the Markov Renewal Process and another based on the direction of movements across the current cell. The assigned probabilities, derived from the two trajectory datasets, are combined using Dempster-Shafer theory to make the most informed decision about the future crossing cell. The study [208] utilized a real dataset containing information about APs, users, and the mobility paths derived from it. The network was modeled with 4 APs in the research as a second-order Markov chain, employing information about the current and preceding locations of users to predict their potential next location. This prediction process involved computing transition probabilities derived from the users' historical movement patterns.

Concerning the AP selection in Wi-Fi networks, [209] proposed a mobility prediction methodology based on a Hidden Markov Model that is used to forecast the next AP that users will connect to, based on current and historical user location information. In addition, in order to solve handover problem by mathematical solution, paper [210] introduces an enhanced handoff mobility prediction scheme utilizing Hidden Markov Models (HMM). The HMM acts as a predictor by analyzing the geographic locations of mobile users to identify the most probable Femtocell AP for connection. The paper regarded femtocell AP as hidden states and users' geographical location as observable states. Specifically, it selects the Femtocell AP in the user's direction with the best signal quality. The model's performance was evaluated against Handoff-to-Nearest neighbor Femtocell and Handoff-to-Randomly selected Neighbor Femtocell approaches across varying network densities. Results indicated that the proposed scheme reduces the number of handoffs and ping-pong handoffs compared to the other algorithms. Furthermore, users utilizing this approach experience extended connections to the same Femtocell AP, leading to increased dwell time and improved communication quality. The paper [211] employs a Dual Hidden Markov Model (HMM) for effective prediction of Wi-Fi APs. This model incorporates two hidden states and two observable states. The initial HMM utilizes the user's present location as input to predict future locations. Subsequently, these predicted locations become the input for the second HMM, which then determines the optimal Wi-Fi AP for the next connection. Specifically, the first HMM uses the mobile user's future geographical coordinates as hidden states and their current geographical coordinates as observable states to predict the user's location. In contrast, the second HMM utilizes optimal Wi-Fi AP and geographic location as hidden and observable states, respectively, to predict the next Wi-Fi AP for the user.

In recent years, the authors made use of ML models to solve these issues (i.e. handover and AP selection) within wireless communication. ML models can adapt to complex and dynamic patterns in data. They can learn and generalize from diverse datasets, making them suitable for various scenarios. Additionally, ML models can effectively handle high-dimensional data and consider a wide range of input features simultaneously, enhancing prediction accuracy [34, 145]. In this context, from the point of view of traffic offloading from cellular to Wi-Fi systems, length of stay prediction at an AP can be useful for user bandwidth allocation e.g. giving higher priority to soon-to-depart Wi-Fi users so that the larger amount of traffic is sent through the Wi-Fi before performing the handover to the cellular system [214]. The paper [34] introduced data-driven ML approaches to effectively tackle these two issues (i.e. handover prediction and AP selection) in wireless LAN networks. The proposed solution involved a centralized network architecture utilizing a SDN controller that incorporated ML algorithms for handover prediction and AP selection. RF was employed for handover prediction, while MLP and SVR were utilized for AP selection. Firstly, the scheme anticipated potential handover events and determines their necessity, reducing the likelihood of unnecessary handovers, especially in ultra-dense deployments with Overlapping Basic Service Set (OBSS). Secondly, it addressed AP selection by predicting post-selection network throughput to identify the optimal AP.

To tackle AP selection issue by ML techniques, the paper [159] investigated a user location prediction method for 5G Ultra Dense Networks (UDN) utilizing SVM. Addressing the specific characteristics of density and mobility in UDN, the approach employs SVM to analyze the vector index of the mobile terminal and utilizes a regression algorithm for predicting the terminal's location in the next 5 seconds. The proposed scheme achieved real-time and highly accurate predictions of the mobile user's position in UDN. Additionally, the utilization of SVM enhanced the prediction scheme's performance while concurrently decreasing hardware complexity. The study [212] focused on a standard rectangular system network where users can establish connections with one of the three closest Base Stations (BSs). The distance between a user and a BS was determined through RSSI and a path loss model. For either process of training and prediction, Recurrent Neural Networks (RNNs) were employed, with the sequences of RSSI values as the input for the NN. This paper [213] evaluated AI-assisted mobility predictors. Specifically, mobility prediction was modeled as a multi-class classification challenge in order to predict the future association of the mobile users with base stations. This is performed using Extreme Gradient Boosting Trees (XGBoost) and DNN.

Our work primarily offers a comprehensive approach to user mobility prediction within Wi-Fi networks by considering both sequential and periodic information, extending beyond the scope of existing literature primarily focused on cellular network-related issues. Existing works predominantly focus on issues within cellular networks, such as characterizing user habits based on BS and mobile terminal communication activity [201], partitioning user populations into predefined profiles [202], and identifying user daily motifs based on cell camping patterns [203]. Furthermore, our work extends beyond traditional handover prediction and AP selection methodologies by integrating time representations into the embedding process, allowing ML models to capture the time-dependent nature of Wi-Fi network dynamics. This differs from existing literature, which primarily utilizes mathematical expressions (i.e. MM, or HMM) or ML techniques for handover prediction and AP selection in wireless communication networks.

3.3 Methodology Description

We particularize the framework of AI/ML control loop depicted in Fig. 2.3 to focus only on the objective of predicting the next user connectivity. The proposed prediction methodology designed for this specific objective is illustrated in Fig. 3.1. The methodology assumes a Wi-Fi Network with monitoring capabilities for the collection of measurements reported by the users when connected to the different APs. As we described in Section 2.4, network measurements collected at the APs are sent to the WLC and finally stored in a centralized database that contains the historical measurements of the different APs. In the *Collection of Network Measurements for training* process of the Fig. 3.1, a list of metrics for each u-th user ($u = 1,\ldots,U$) is collected when connected to each AP (e.g. the instants of time when the user begins and ends a connection to each AP, the average SNR-Signal to Noise Ratio-, the average RSSI-Received Signal Strength Indicator-, the amount of bytes transmitted/received during the connection of the user to each AP, etc.). All this information is stored in a database. From this information, the instants when the user starts and ends a connection to each AP, along with the AP names, are used to generate the final dataset. Then, for each user, a pre-processing of the collected data is done so that the measurements collected during each d-th day (with $d = 1,\ldots,D$) are split in M

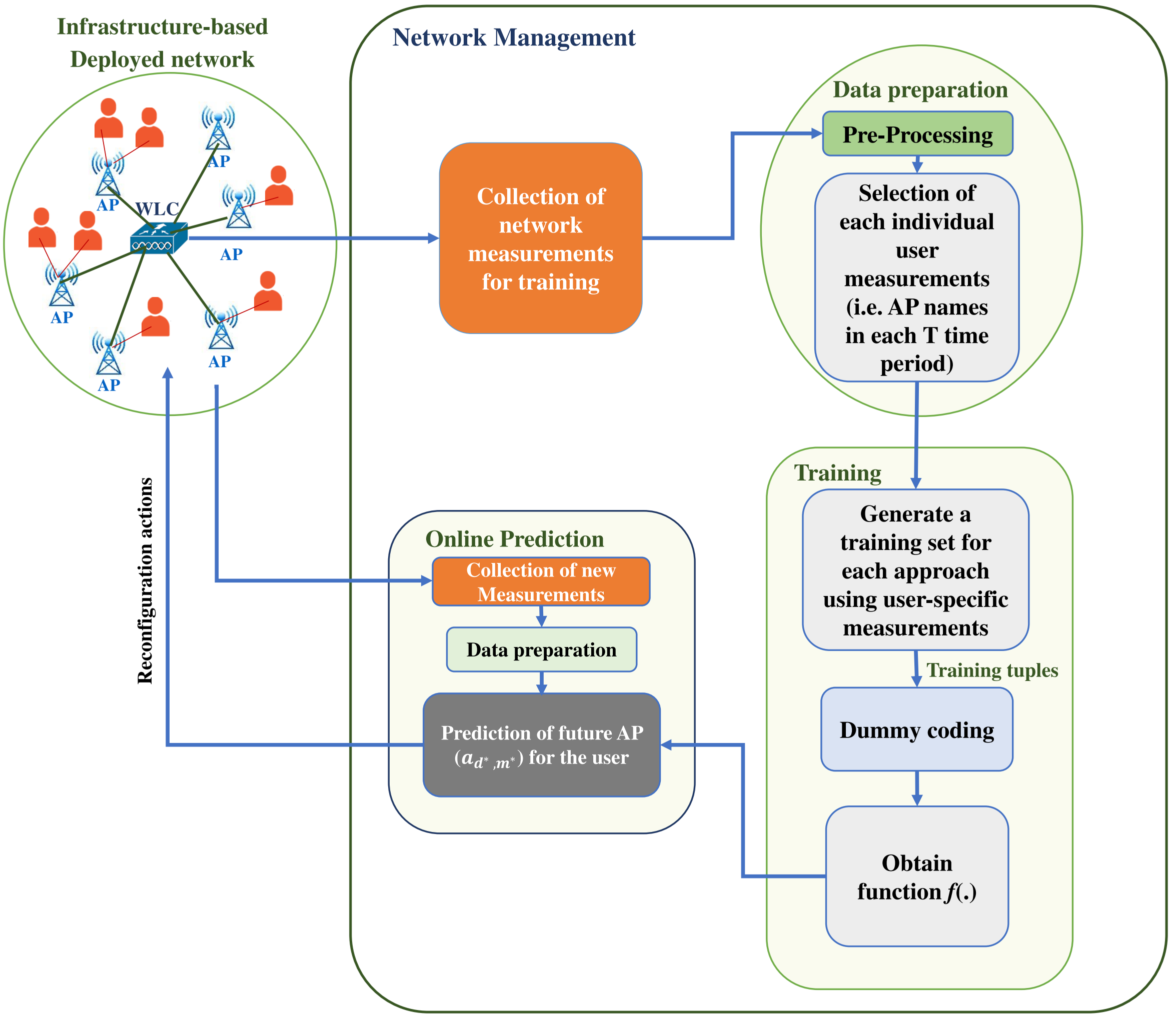

Infrastructure-based
Deployed network
AP
WLC
Reconfiguration actions
Network Management
Collection of network measurements for training
Data preparation
Pre-Processing
Selection of each individual user measurements (i.e. AP names in each T time period)
Training
Generate a training set for each approach using user-specific measurements
Training tuples
Dummy coding
Obtain function f(.)
Online Prediction
Collection of new Measurements
Data preparation
Prediction of future AP (a_{d^*,m^*}) for the user

Fig. 3.1. General framework of proposed prediction methodology.

time periods with equal duration T. In particular, the pre-processing step generates a matrix A for each user, so that each term $a_{d,m}$ (with $m = 1,\ldots,M$ and $d = 1,\ldots,D$) represents the AP identifier to which this user was connected during the m-th time period of the d-th day. In case that the user connects to more than one AP at the same m-th time period, it is assumed that the term $a_{d,m}$ will correspond to the AP with the highest connection duration.

For the prediction of the AP to which the user will be connected in a specific m^*-th time period of a specific d^*-th day in the future (a_{d^*,m^*}), the proposed methodology makes use of some historical information of the AP to which the user was connected in the past and a prediction function $f(\cdot)$ that is obtained by means of a supervised learning. For that purpose, the *Selection of historical data* process selects some specific terms in matrix A. Different approaches are presented below:

- **Prediction Based on Time-period Patterns (PBTP):** in this case, the prediction of a_{d^*,m^*} is based on the APs to which the user was connected in the last N previous time periods (i.e. a_{d^*,m^*-N},...,a_{d^*,m^*-n},...,a_{d^*,m^*-1}), as illustrated in Fig. 3.2a. In order to obtain the prediction function $f(\cdot)$, a vector B =(b_1,...,b_f,..., b_F) is built from all the $F = D{\cdot}M$ previous time periods in the last D days, i.e. B =($a_{1,1}$,...,$a_{1,m}$,...,$a_{1,M}$,...,$a_{d,1}$,...,$a_{d,m}$,...,$a_{d,M}$,...,$a_{D,1}$,...,$a_{D,m}$,...,$a_{D,M}$). Then, B is split into I different training tuples B^i, each one composed of the i-th element and its N previous elements. This split is done by applying a sliding window of length N over the set of F measurements, resulting in a training set of $F-N$ training tuples of the form B^i =(b_{i-N},...,b_{i-n},...,b_i) with i = 1,...,$F-N$. The $f(\cdot)$ function is learnt by a supervised learning that consists on observing the relationship between b_i and its N previous elements (b_{i-N},...,b_{i-1}) for all the $I = F-N$ tuples. The rationale of this approach is to identify user frequent AP connectivity patterns in N consecutive time periods.

- **Prediction Based on Daily Patterns (PBDP):** in this case, the prediction of a_{d^*,m^*} is done according to the APs to which the user was connected in the previous N days at the same time period of the day (i.e. a_{d^*-N,m^*}, ..., a_{d^*-n,m^*}, ..., a_{d^*-1,m^*}), as depicted in Fig. 3.2b. In order to obtain the prediction function $f(\cdot)$, a set of M vectors B_m =($b_{1,m}$, ..., $b_{f,m}$, ..., $b_{F,m}$) is built. Each B_m consists of the AP to which the user connected in the last $F = D$ days at the m-th time period of the day, i.e. B_m =($a_{1,m}$, ..., $a_{D-d,m}$,..., $a_{D-1,m}$, $a_{D,m}$). Then, each B_m is split into I different training tuples B^i_m, each one composed of the i-th element and its N previous elements, i.e. B^i_m =($b_{i-N,m}$,...,$b_{i-n,m}$,...,$b_{i,m}$) with $i = 1,\ldots,D-N$. Then, a total of $(D-N)\cdot M$ tuples with size $N+1$ are generated. The rationale of this is to identify user periodical AP connectivity patterns in N consecutive days at the same time of the day.

- **Prediction Based on Weekly Patterns (PBWP):** In this case, the prediction of a_{d^*,m^*} is done according to the APs to which the user was connected in the N previous weeks at the same day of the week and time period of the day (i.e. $a_{d^*-7N,m^*},\ldots,a_{d^*-7n,m^*},\ldots,a_{d^*-7,m^*}$), as demonstrated in Fig. 3.2c. Again, a set of M vectors B_m =($b_{1,m},\ldots,b_{f,m},\ldots,b_{F,m}$) is built, each one consisting on the AP to which the user connected in the last $F = W$ weeks at each m-th time period of a specific day of the week, i.e. B_m =($a_{d-7W,m},\ldots$, $a_{d-7w,m}$,...,$a_{d-7,m}$,$a_{d,m}$). Then, each B_m is split into different I training tuples B^i_m, each one composed of the i-th element and its N previous elements, i.e. B^i_m =($b_{i-N,m},\ldots,b_{i-n,m},\ldots,b_{i,m}$) with $i = 1,\ldots,W-N$. In this case, the total number of tuples is $(W-N)\cdot M$. The rationale of this is to identify weekly AP connectivity patterns at the same day of the week and time of the day.

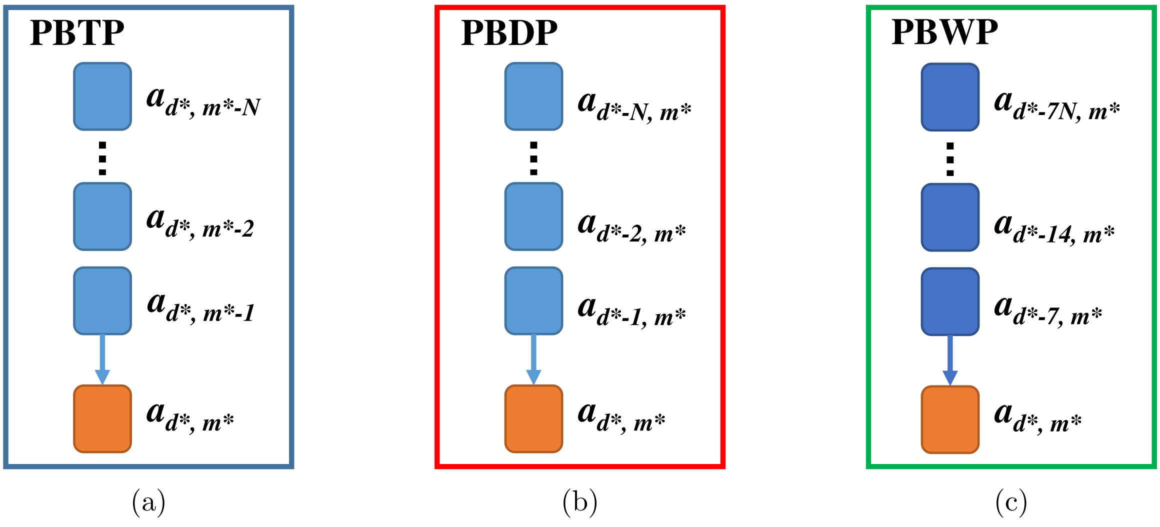


Fig. 3.2. The process of Prediction of a_{d^*,m^*} based on the last N previous time periods for Time-period Patterns (PBTP) (a), Daily Patterns (PBDP) (b), and Weekly Patterns (PBWP) (c).

- **Joint Based Prediction (JBP):** It consists on the same methodology but doing a combination of the three approaches described previously. In this case, the prediction of a_{d^*,m^*} is done according to the APs to which the user connected in the last N time periods (i.e. a_{d^*,m^*-N},. . . ,a_{d^*,m^*-n},. . . ,a_{d^*,m^*-1}), the APs at the same m-th time period for the last N days (i.e. a_{d^*-N,m^*},. . . ,a_{d^*-n,m^*},. . . ,a_{d^*-1,m^*}) and the APs at the same day of the week and time period of the day for the last N weeks (i.e. a_{d^*-7N,m^*},. . . ,a_{d^*-7n,m^*},. . . ,a_{d^*-7,m^*}). The procedure is illustrated in Fig. 3.3 to enhance comprehension of JBP. The prediction function $f(\cdot)$ is learnt in a similar way as before by observing the relationship of specific $a_{d,m}$ and the observations in the last N time periods, the last N days at the same time period and the last N weeks at the same day of the week and time period of the day. A total number of $(W - N)\cdot M$ tuples with size $3N + 1$ are obtained.

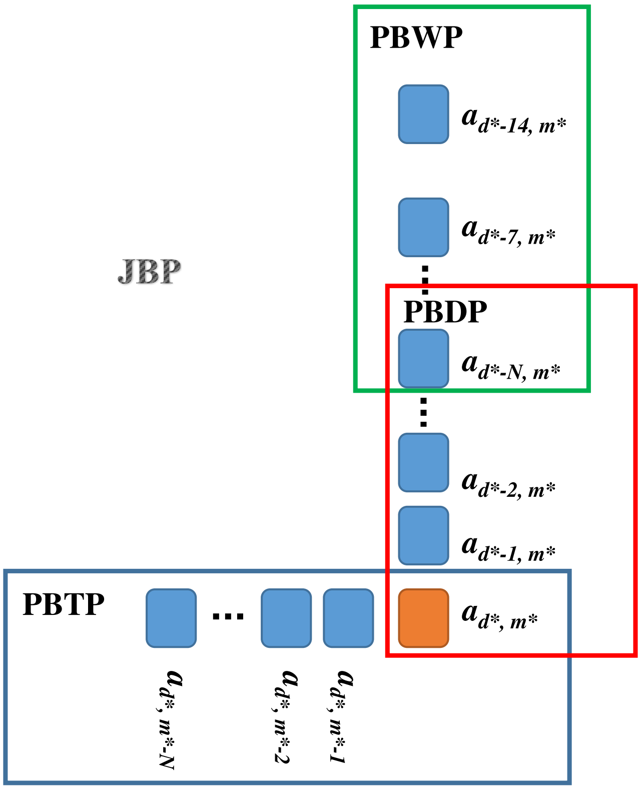


Fig. 3.3. The process of Joint Based Prediction (JBP) approach.

It is worth noting that all these terms $a_{d,m}$ in matrix A correspond to categorical values (e.g. an AP identifier). For both the training and prediction, these terms are converted into numerical attributes by means of the so-called dummy coding process [215]. A dummy variable is a binary variable coded as 0 or 1 to represent the absence or presence of some categorical attribute. Therefore, each of the N elements used for prediction a_λ ($\lambda = 1$,. . . ,N) are converted into a set of G dummy variables c_λ =($c_{\lambda,1}$,. . . ,$c_{\lambda,g}$,. . . ,$c_{\lambda,G}$), where G is the number of different APs in the set of N measurements, so that the term $c_{\lambda,g} = 1$ if a_λ corresponds to the g-th AP and $c_{\lambda,g} = 0$ otherwise. Then, the resulting number of dummy variables $D = N\cdot G$ are used for the prediction of a_{d^*,m^*} according to (3.1). Before the training, the same dummy coding is also done for all the training tuples of the training set.

$$a_{d^*,m^*} = f(c_{1,1}, \ldots, c_{1,G}, \ldots, c_{\lambda,1}, \ldots, c_{\lambda,G} \ldots, c_{N,1}, \ldots, c_{N,G}) \tag{3.1}$$

To illustrate the process of dummy coding, let's consider a scenario where we have three APs: AP1, AP2, and AP3. If a_λ corresponds to AP1, then $c_{\lambda,1} = 1$, the dummy variables would be (1, 0, 0), denoting the presence of AP1 and absence of AP2 and AP3. Similarly, the dummy variables would be (0, 1, 0) and (0, 0, 1) for AP2 and AP3, respectively.

In the online prediction for the prediction of a_{d^*,m^*}, (see Fig. 3.1), the N measurements of the user are used as input. Then, this input data is prepared, including preparation of the input data and also dummy coding process, in a manner similar to that in the training step. Finally, the prediction step makes use of the input preprocessed data and the learnt $f(.)$ function in order to make the prediction for this specific user.

3.4 Scenario

The considered scenario involves a large Wi-Fi network with 429 APs deployed on a university campus of the Universitat Politècnica de Catalunya, located in Barcelona, comprising 33 buildings, each with four floors. The reported user measurements are collected by the Cisco Prime Infrastructure tool [216]. The users' measurements were collected during $D = 84$ consecutive days (i.e. $W = 12$ weeks). The prediction methodology was run for $U = 967$ users. According to the methodology described in Section 3.3, the matrix A is built for each user by determining the AP to which the user is connected in each of the $M = 96$ periods of $T = 15$ minutes for each of the $D = 84$ days. According to this data, the proposed prediction methodology is run in order to predict the AP which each user will be connected to in all the $M = 96$ time periods of $T = 15$ minutes in the subsequent week. The obtained predictions are compared to the real APs which the user has been connected to. The prediction accuracy is calculated as the percentage of time periods that have been predicted correctly in the range between 6:00h and 22:00h for all the weekdays (from Monday to Friday) for all the users that connected to the Wi-Fi network at least one time every day. Two ML techniques, namely NN and RF, are chosen for this study. NN is selected for its ability in recognizing complex patterns and managing non-linear relationships within the data [190]. On the other hand, RF is chosen for its ensemble learning approach, which combines multiple decision trees to improve accuracy, reduce the risk of overfitting, and effectively address non-linear relationships [27, 186]. A brief description of these two ML prediction techniques has been provided in Chapter 2 Section 2.3. The prediction methodology is implemented by means of Rapidminer Studio [217]. The parameters of each supervised learning algorithm have been tuned to obtain the maximum prediction accuracy. In particular, the NN is configured with the following hyperparameters: a learning rate of 0.05, which controls the step size during gradient descent; a momentum of 0.9, which accelerates gradient descent in the relevant direction and dampens oscillations; 100 training cycles, which denote the number of iterations over the entire dataset during training; and 1 hidden layer of size 20, which specifies the number of neurons in the hidden layer. Additionally, the RF is set with 100 trees, which are individual decision trees used for ensemble learning; gain_ratio criterion, which measures the quality of a split in the decision tree based on information gain ratio; and a maximal depth of 10, which limits the maximum depth of each decision tree in the forest.

3.5 Results

To demonstrate the effectiveness of our prediction methodology, we initially concentrate on predicting the AP for a specific user across all time periods on Wednesday. Subsequently, we broaden our analysis to assess the methodology's performance across all users and time periods throughout an entire week. Furthermore, we evaluate the effect of the sliding window size on the Joint Based Prediction (JBP) method for a dataset spanning $D = 84$ days. Lastly, we investigate the influence of varying amounts of historical data utilized in constructing the training set.

3.5.1 Example of the AP prediction methodology

To illustrate the performance of the proposed prediction methodology, we first focus on the AP prediction for a specific user for all the time periods on Wednesday. Assuming here the PBWP approach, the AP prediction at the m-th time period is based on the APs to which the user connected to in the previous $N = 6$ Wednesdays at the same m-th time period of the day. The training set is built by using the last $F = 12$ weeks. For validation purposes, the predictions are compared to the real AP where the user connected to during this Wednesday. According to this, Fig. 3.4 presents this comparison when using the NN algorithm. As shown, for this particular user, the methodology is able to correctly predict the AP in 60 out of 64 periods of 15 minutes (i.e. a 93.75% of prediction accuracy). In general, most of the transitions between AP are correctly predicted. In fact, only an error of one period of 15 minutes is observed in predicting the user time of arrival while a slightly higher error is also observed for the prediction of departure. As shown, the user arrive at university at 11:00 AM. and leave at 6:30 PM, while the NN algorithm predict to 10:45 AM and 7:15 PM, respectively. During lunch time, the connection to AP XSFD4P202 is not correctly predicted, but the prediction was AP XSFD4P102 that is located just in the lower floor within the same building. This indicates that, although the AP is not well predicted in this case, the methodology predicted correctly the region where the user was located.

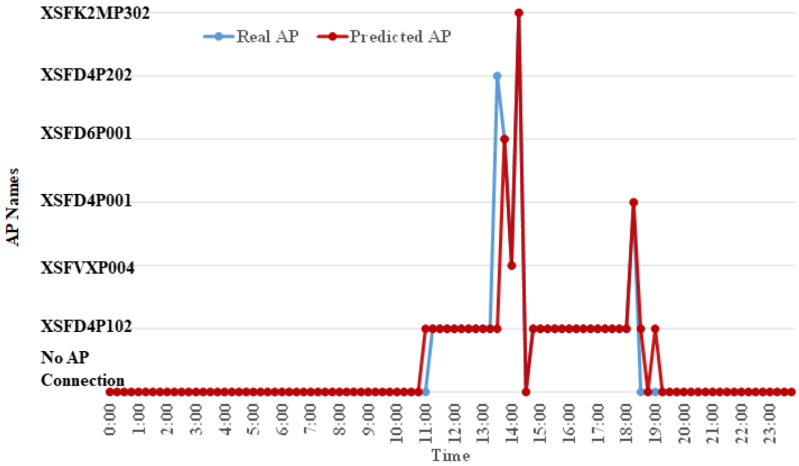


Fig. 3.4. Comparison of real and predicted AP for a specific user on Wednesday with NN.

3.5.2 Comparison of the different proposed approaches

To gain insights into the performance of the proposed methodology, the prediction process has been run for all the set of users in all the time periods of 15 minutes during a whole week. Then, the predictions were compared to the AP to which each user connected at each time period. Table 3.1 presents the percentage of users in which each approach provides the best prediction accuracy for PBWP, PBDP and PBTP. As shown for both NN and RF predictors, PBTP approach provides better prediction accuracy than PBWP or PBDP for most of the users. This indicates that the AP to which a user was connected in the most recent time periods is the most useful information for prediction. However, it is worth noting that, for a relatively high percentage of users, the best approach is obtained with PBWP or PBDP (e.g. around 24% and 18% for NN and RF, respectively). This result indicates that the daily or weekly periodical behavior of some users can be captured better in NN by PBDP or PBWP approaches, respectively.

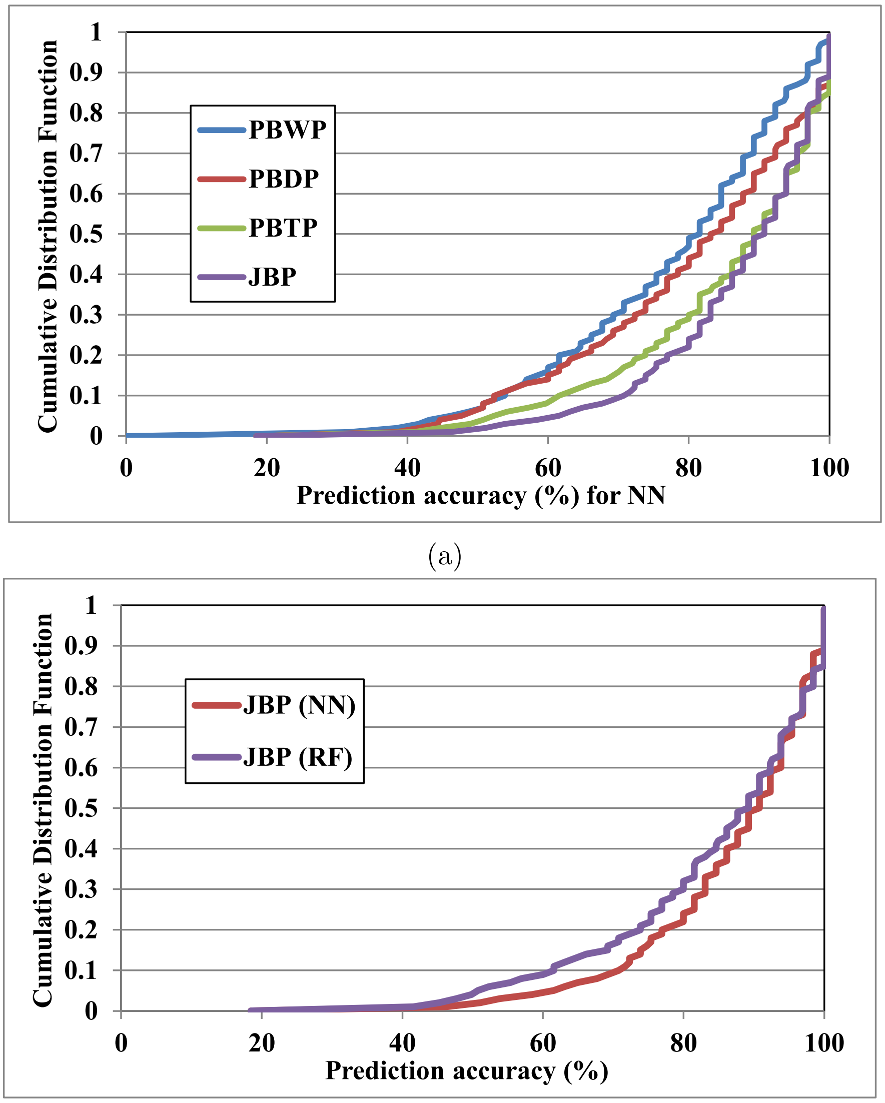
Table 3.1. Percentage of users in which the different approaches provide the best accuracy.

	NN	RF
PBWP	24.07	18.62
PBDP	6.01	0
PBTP	69.90	81.37

Fig. 3.5a presents the Cumulative Distribution Function of the prediction accuracy for the different prediction approaches with NN. For comparison purposes, in all the approaches, the prediction is based on the 6 previous observations. Therefore, in PBWP, PBDP and PBTP, the sliding window is set to $N = 6$. In JBP approach, the sliding window is set to $N = 2$, i.e. the prediction is based according to the APs to which the user connected to in the last $N = 2$ time periods (i.e. $a_{d^*,m^*-2}, a_{d^*,m^*-1}$), the APs at the same m-th time period for the last $N = 2$ days (i.e. $a_{d^*-2,m^*}, a_{d^*-1,m^*}$) and the APs at the same day of the week and time period of the day for the last $N = 2$ weeks ($a_{d^*-14,m^*}, a_{d^*-7,m^*}$). As shown in Fig. 3.5a, the JBP approach is able to provide better prediction accuracy than the rest of the approaches separately. The reason is that the JBP is able to jointly capture the hourly, daily and weekly user behavior. Moreover, Fig. 3.5b presents a comparison between NN and RF in JBP. As illustrated in the figure, the JBP approach using NN achieves better prediction accuracy than using RF.

In Table 3.2, the average prediction accuracy and the average CT, required for running the methodology for each user, are compared for the different approaches for both NN and RF. The methodology was executed in a computer with a Core i5-3330 processor at 3.00 GHz and RAM memory of 8GB running Microsoft Windows 10. It has been observed that the CT is mainly due to the process of training while the time for the prediction step is negligible. As shown in Table 3.2, the PBTP and PBDP approaches exhibit higher CT per user since they make use of larger number of training tuples, leading to longer training times. As shown in Table 3.2, the JBP approach provides the best prediction accuracy with a relatively low CT.

It is worth noting that the CT required for the training may impose some restrictions in the maximum number of users that can included in the AP prediction or the frequency in which the training is updated. The values of the average CT per user obtained in Table 3.2 may be excessively high in a Wi-Fi network that may have several



(a)

(b)

Fig. 3.5. Cumulative Distribution Function (CDF) of the prediction accuracy for the different approaches using NN in (a) and a comparison of the prediction accuracy between NN and RF in (b).

thousands of simultaneous user connections. As a consequence, running the proposed methodology for such a high amount of users may require the parallelisation of the proposed methodology using multiple processors, each of them to run the prediction of a group of users.

Table 3.2. Comparison of the different approaches.

	Average Prediction accuracy(%)		Computational Time (CT) per user (s)	
	NN	RF	NN	RF
PBWP	77.16	79.29	0.87	0.22
PBDP	79.88	79.64	33.33	3.74
PBTP	84.84	84.53	38.95	3.94
JBP	86.73	84.26	3.21	0.64

3.5.3 Impact of the size of the sliding window

This section investigates the influence of the sliding window size on the JBP approach, considering a dataset spanning $D = 84$ days for training set generation. Table 3.3 provides an overview of the average prediction accuracy and CT per user for varying sliding window sizes (N). As depicted in Table 3.3, employing a sliding window that is too small diminishes the method's ability to capture weekly, daily, and hourly user behavioral patterns, consequently leading to decreased prediction accuracy. Conversely, utilizing an excessively large sliding window results in a reduction of tuples available for training set generation, thereby compromising the quality of the training process and subsequently reducing prediction accuracy. Therefore, selecting an optimal sliding window size is crucial to ensure an effective balance between capturing periodic user behavior and maintaining an adequate training dataset size. However, the value of N does not significantly impact the results; the differences in accuracy and CT are not so high.

Table 3.3. Impact of the size of the sliding window on the JBP approach.

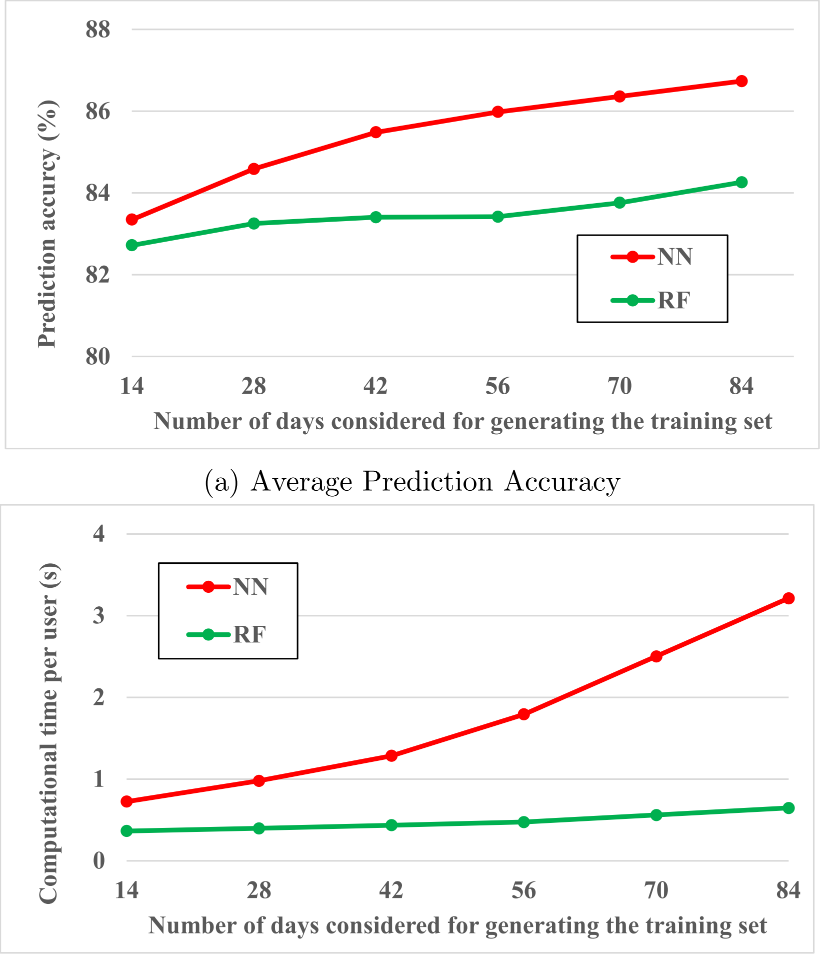
	Average Prediction accuracy(%)		Computational Time (CT) per user (s)	
	NN	RF	NN	RF
N=2	86.73	84.26	3.21	0.64
N=6	90.71	84.75	4.75	0.79
N=10	88.79	82.96	2.54	0.71

3.5.4 Impact of the amount of data used for training

This section delves into the impact of historical data quantity on training set construction. Specifically, Fig. 3.6a illustrates a comparison of prediction accuracy for the JBP method with $N = 2$, utilizing NN and RF predictors, relative to the number of days considered for training set generation. Conversely, Fig. 3.6b displays the average CT per user. As depicted in Fig. 3.6a, the NN predictor consistently outperforms RF in terms of average prediction accuracy. Moreover, Fig. 3.6a highlights that leveraging a larger volume of data for training set generation leads to enhanced prediction accuracy. However, processing an increased volume of data necessitates a longer CT, particularly evident for NN, as illustrated in Fig. 3.6b. These findings underscore the trade-off between prediction accuracy and CT efficiency, particularly in NN.

3.6 Conclusions

This chapter has proposed a methodology for the prediction of future APs to which users will be connected in a Wi-Fi network. The proposed methodology is based on a supervised learning that makes use of historical user connectivity to build a prediction model. Different approaches have been defined depending the historical data that is used. In general, the PBTP approach, in which the prediction is based according to the most recent APs to which the user connected, provides the best prediction accuracies. However, PBDP or PBWP perform better for users that follow some daily or weekly periodical behavior. As shown, a joint approach (JBP) is able to provide better prediction accuracy than the rest of the approaches separately with a relatively low CT per user. The impact of the training set size has been illustrated for the



(a) Average Prediction Accuracy

(b) Average Computational Time (CT) per user

Fig. 3.6. The impact of the amount of data.

JBP approach in terms of prediction accuracy and CT. As shown, higher amount of days with measurements considered for generating the training set provides higher prediction accuracy at expenses of higher CT, especially for NNs. The impact of the size of the sliding window has been also evaluated. A too low value of the sliding window results in a worse capability to detect weekly, daily and hourly user periodical behavior while a too large value leads to a too low number of tuples for training. The results indicate that the prediction based on the NN provides a higher prediction accuracy than the prediction based on RF at expenses of an increase in the CT.

Therefore, the following key performance indicators of prediction accuracy and CT efficiency have been addressed in this chapter of our study. The study evaluates the prediction accuracy of different approaches, such as PBTP, PBDP, PBWP, and JBP, for predicting future AP connections of users in a Wi-Fi network. The comparison between these approaches demonstrates their effectiveness in achieving prediction accuracy. The study also assesses the CT required by the prediction models, particularly emphasizing the relatively low CT per user achieved by the joint approach (JBP). This indicates consideration for CT efficiency in model development. The study was conducted in a large Wi-Fi network environment comprising 429 APs deployed across a university campus with extensive coverage, including 33 buildings, each with four floors. Additionally, the study involved collecting user measurements over a span of 84 consecutive days, which amounts to a significant volume of data. Furthermore, the prediction methodology was applied to a substantial number of users, totaling 967 users to evaluate its capability to handle a large-scale Wi-Fi network environment and a considerable volume of user data.

Chapter 4

AP Traffic Prediction

4.1 Introduction

Insights gained from predictive analysis of network characteristics play a pivotal role in facilitating various decision-making processes within network management, particularly concerning network reconfiguration actions to improve network performance [1, 218]. For example, the prediction of a future peak of traffic in the area covered by an AP may be useful for e.g. the reconfiguration of the admission control threshold in order to control the amount of traffic at the AP or the activation of load balancing processes to encourage users to connect to less overloaded APs/radios. Numerous methodologies have been proposed for cellular [219, 220] and Wi-Fi [170, 172, 221–223] networks in this regard.

In real-world Wi-Fi networks, the dynamics of network traffic, performance, and environmental conditions are constantly changing, necessitating predictive capabilities that can adapt to these evolving circumstances. Temporal prediction models often rely solely on historical data from the target AP itself, overlooking valuable contextual information provided by neighbouring APs. However, in complex network environments, the behavior of an individual AP is influenced not only by its own historical data but also by the surrounding network topology, user mobility patterns, and interference from neighbouring APs. Therefore, an adaptive and context-aware prediction framework that considers the spatio-temporal relationships between the target AP and its neighbours is essential for accurate forecasting.

By incorporating information from neighbouring APs, the framework can capture spatial dependencies and temporal correlations that impact the performance of the target AP. For example, fluctuations in network traffic at neighbouring APs may indicate potential changes in user behavior or network conditions that will affect the target AP in the near future. A very large number of UEs in a specific AP means that there is a concentration of many people. Then, when this large amount of people move in this area, a large number of UEs may be expected in neighbour APs. Similarly, a high number of failures in neighbour APs can help us to identify a high number of failures in the target AP. It means that a large number of users in a region may lead to an increase in interference that causes a larger number of failures. Moreover, the presence of an external interference affecting a specific area, may affect not only to the target AP but also to its neighbours. Likewise, variations in environmental factors such as signal strength or interference levels can provide valuable insights into the expected trajectory of network metrics at the target AP.

In this context, the key novelty compared to prior studies lies in introducing a comprehensive framework for forecasting future values of a particular network metric within a Wi-Fi network's AP. The proposed prediction methodology explores various DL techniques, leveraging historical network measurements archived in a database. It autonomously determines whether to incorporate historical data from neighbouring APs based on identified spatial correlations among APs. A thorough assessment of this methodology has been conducted using an extensive dataset comprising real-world Wi-Fi network measurements.

The rest of the chapter is organized as follows. Section 4.2 provides an overview of the existing literature. Section 4.3 explains the details of the proposed prediction methodology. Section 4.4 describes the techniques used for prediction. The scenario description presented in Section 4.5, and the obtained results are discussed in Section 4.6. Section 4.7 explains the implementation aspects necessary for deploying the methodology in a real network. Finally, conclusion is drawn in Section 4.8.

4.2 Related Work

In the context of traffic prediction, multiple approaches can be found in the literature. The works such as, [224–227] make use of a Gaussian Model for traffic prediction. For example, [224] proposed a wireless network traffic prediction model based on Bayesian Gaussian tensor decomposition and Recurrent Neural Network with rectified linear unit (BGCP-RNN-ReLU model), which effectively predict short-term changes in the upstream and downstream network traffic. In turn, [225] proposed a wireless traffic prediction model by applying the Gaussian Process (GP) method based on real 4G traffic data. Finally, [228] proposed a data traffic prediction model based on autoregressive moving average (ARMA) utilizing the time series wireless data.

Other studies consider both temporal and spatial traffic analysis to enhance the prediction accuracy. Spatio-temporal prediction combines spatial dependencies and interactions, capturing how traffic at one location impacts neighbouring areas. It considers traffic propagation effects, handles spatial variability, and adapts to dynamic network conditions. By modeling spatial and temporal features together, it provides contextually aware, accurate, and adaptive predictions for network metrics like traffic load and failures, optimizing network management and resource allocation. Several works proposed a joint space/time traffic prediction, assuming that the traffic is temporally and spatially correlated [177, 220, 229–231]. The study [231] includes a correlation-based sub-predictor and a causality-based sub-predictor. The first one predicts the regular traffic component (i.e. traffic volume, time stamps, cell location) based on traffic spatio-temporal correlation, and the second focuses on forecasting the traffic causality related features (e.g. existence of social events and user mobility information). The predicted traffic is generated by combining the results of these two sub-predictors in order to improve the prediction accuracy at the expenses of increasing the complexity of the prediction methodology. The methodology was evaluated at an International Airport in China. In particular, in the context of cellular networks, [177, 230] presented a spatio-temporal prediction with CNN-RNN. Authors in [220] proposed a strategy by combining auto-encoder and LSTM for spatio-temporal prediction of cellular network traffic. Authors of [229] stated that auto-encoder may fail to learn the fully characterized features for spatial dependencies between neighbouring cells so that they proposed a CNN based framework. On the other hand, [177] claimed that a joint CNN-LSTM method outperformed the CNN and LSTM for spatio-temporal mobile traffic prediction. In recent years, other novel approaches have been proposed for prediction. As an example, transformer-based deep learning models have been proposed to tackle the task of traffic time series prediction [232–235]. In this context, [232] introduced a Transformer-based spatio-temporal model for traffic prediction by making use of two New York City Taxi and Bike datasets. The paper presents a predictive model capable of forecasting the inflow and outflow of different regions at a specific future time by utilizing historical aggregated data. It makes use of a data-driven approach to learn spatio-temporal dependencies between regions, without making assumptions about spatial locality. The model considers dynamic spatio-temporal dependencies for individual time slots and also incorporates the so-called attention mechanism to capture potential periodic characteristics of spatio-temporal dependencies across multiple time slots. Moreover, in [233], the authors extracted intricate temporal, local, and global spatial features utilizing the spatio-temporal transformer. Additionally, a downsampling transformer is used to effectively capture global spatial features encompassing the entire city. Their proposed model comprises four integral components: a spatio-temporal transformer to model the spatio-temporal correlations within mobile traffic data, spatio-temporal fusion to effectively integrate information obtained from both spatial and temporal transformers, a downsampling transformer designed to capture a wider range of spatial features, and a prediction layer, responsible for aggregating the spatio-temporal features.

To the best of the authors' knowledge, the majority of the papers in the literature focus on traffic prediction in cellular networks. However, very few papers deal with traffic prediction in Wi-Fi networks. In this context, in the paper of [172] different congestion levels (low, medium or high) in different locations are predicted in a University Campus. In order to do this, first, several network features (such as number of clients, obtained throughput, frame transmission retries, frame errors) are predicted by means of a support vector regression (SVR). Then, an expectation method (EM) is used to obtain different clusters that represent different levels of congestion. The obtained clustering model is used for the prediction of future levels of congestion. In [170], data-driven ML techniques are employed to predict the transmission throughput in Wi-Fi networks. Additionally, [221] extracted three categories of features, i.e., individual features, spatial features, and temporal features from network data to label each AP. Then, these features are used to construct a DL architecture comprising two separate deep RNN models. Leveraging the training model and historical client traffic load input, future client traffic loads are predicted in real-time. In [223], a temporal prediction of channel and traffic utilization was performed to proactively allocate resources more efficiently. Regarding this topic, the paper [222] is the only one that is directly close to our work (i.e., spatio-temporal prediction in Wi-Fi systems). It underscores the significance of integrating measurements from highly correlated AP neighbours to enhance prediction accuracy. It analyzes data from 8 APs, while our investigation extends this analysis to encompass a broader array of APs. Additionally, our work incorporates contemporary prediction methods, including Transformers, alongside other DL techniques posited in [222]. Unlike the approach of [222], which provides a single time complexity for both training and prediction, our study separately calculates TCT and PCT that can effectively aim to tackle implementation aspects, challenges, and real-world deployment considerations.

4.3 Methodology Description

We particularize the framework of AI/ML control loop depicted in Fig. 2.3 to concentrate only on the objective of traffic prediction. The proposed prediction methodology designed for this specific objective is illustrated in Fig. 4.1. As shown in Fig. 4.1, network measurements collected at the APs are sent to the WLC and finally stored in a centralized database that contains the historical measurements of the different APs. Then, with this data, an offline training is run in order to extract knowledge from the collected data. This offline training is done with certain periodicity in order to keep updated the obtained model according to recent collected measurements. Then, the prediction phase is done online and consists of using new measurements collected at the APs together with the prediction function $f(.)$ obtained in the training phase.

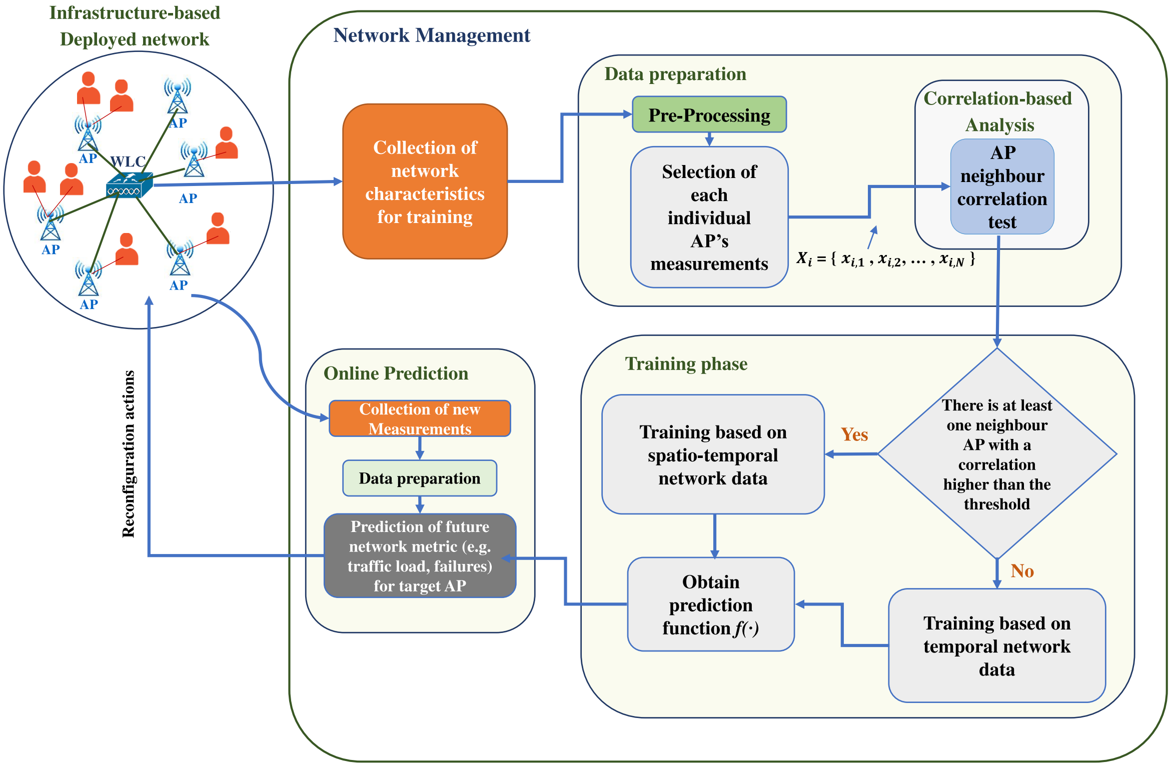

Infrastructure-based Deployed network
WLC
AP
AP
AP
AP
AP
AP
AP
Reconfiguration actions
Network Management
Collection of network characteristics for training
Data preparation
Pre-Processing
Selection of each individual AP's measurements
$X_i = \{ x_{i,1}, x_{i,2}, \dots, x_{i,N} \}$
Correlation-based Analysis
AP neighbour correlation test
Online Prediction
Collection of new Measurements
Data preparation
Prediction of future network metric (e.g. traffic load, failures) for target AP
Training phase
Training based on spatio-temporal network data
Yes
There is at least one neighbour AP with a correlation higher than the threshold
No
Obtain prediction function $f(\cdot)$
Training based on temporal network data

Fig. 4.1. General framework of proposed prediction methodology.

The proposed methodology, assumes a Wi-Fi Network consisting of I APs. Measurements are collected from the different I APs with a periodicity of T over a total of D days. The time sequence of a specific collected metric of the i-th AP is represented as $X_i = \{x_{i,1}, x_{i,2}, ..., x_{i,N}\}$ where $i = 1, 2, \dots, I$, and N is the total number of samples for each AP collected during the D days. With this data, the proposed methodology aims to predict future values of this specific metric in a given AP by using the last L collected measurements. The methodology can be used for the prediction of different possible metrics (e.g. related to network traffic, network failures, network performance, etc.).

After data collection, in the data preparation phase, the methodology runs a data pre-processing according to the following steps: (i) The first step consists on automatically determining missing data in some time periods. This may happen due to e.g. possible errors in the collection of measurements, etc. Missing data can be completed according to different possible techniques. In this work, backfilling [194] is used, in which missing values at certain time period are filled with the data of the previous time period. If there are more than two consecutive time periods with missing data, the measurements of this day are removed and are not considered for training. (ii) The methodology determines the optimum value of the L. This process is done by the Augmented Dickey-Fuller Test that uses an autoregressive model for determining the most adequate value of L [194]. The Augmented Dickey-Fuller test iteratively assesses the autocorrelation structure of time series data for various lag values. It selects the lag value that minimizes the test statistic, indicative of optimal lag for the time series, to evaluate its stationarity. For each AP, the optimum value of L is obtained. Then, the most frequent value is selected. (iii) The input data X_i is normalized so that all values are within the range between 0 and 1.

After the pre-processing step, the proposed methodology calculates the correlation between each AP and all its neighbours. AP neighbours are determined by means of the Neighbour Discovery Protocol (NDP) [236]. The criterion for the finding neighbour APs is received power. According to this, each AP measures the received power from nearby APs and determines a list of neighbours automatically with the APs detected with a received power higher than a specific threshold. In particular, the correlation between the i-th and the j-th APs (with $i \neq j$) is calculated according to (4.1):

$$C_{i,j} = \frac{N(\sum_{n=1}^{N} x_{i,n}x_{j,n}) - (\sum_{n=1}^{N} x_{i,n})(\sum_{n=1}^{N} x_{j,n})}{\sqrt{[N\sum_{n=1}^{N} x_{i,n}^2 - (\sum_{n=1}^{N} x_{i,n})^2][N\sum_{n=1}^{N} x_{j,n}^2 - (\sum_{n=1}^{N} x_{j,n})^2]}} \tag{4.1}$$

where $x_{i,n}$ and $x_{j,n}$ are the n-th measurement of the selected i-th AP and j-th AP, respectively. These correlations $C_{i,j}$ are used to determine the list of highly correlated neighbours of each AP. Then, if the correlation $C_{i,j}$ is higher than a specific threshold (i.e. $C_{i,j} > Ths$), the j-th AP is included in the list of highly correlated neighbours of the i-th AP. Then, for each i-th AP, a list of M_i highly correlated neighbours is determined. As shown in Fig. 4.1, in case that no highly correlated neighbour has been found for a given i-th AP (i.e. $M_i = 0$), then, the proposed methodology makes a temporal based prediction using the L previous measurements of this AP in order to predict the future value of this specific metric. Otherwise, (i.e. $M_i > 0$), a spatio-temporal prediction is done based on the L previous measurements of the i-th AP and also the L previous measurements of its M_i highly correlated neighbours.

With this data, an offline training is done in order to obtain a $f(.)$ function that will be used in the prediction step, as detailed hereinafter. For the case of the only temporal-based prediction, this $f(.)$ function is determined by observing the relationship between a specific measurement and its previous consecutive L measurements for a give AP. Specifically, the prediction function $f(.)$ is determined according to the following. The set of measurements of the i-th AP $X_i = \{x_{i,1}, ..., x_{i,N}\}$ is split into P different training tuples $X_i^P = \{x_{i,p}, ..., x_{i,p+1}, ..., x_{i,p+L}, x_{i,p+L+1}\}$ with $p = 1, ..., N-L-1$. Therefore, the $f(.)$ function is learnt by a DL process that consists on observing the relationship between $x_{i,p+L+1}$ and its L previous elements $= \{x_{i,p}, ..., x_{i,p+1}, ..., x_{i,p+L}\}$ for all the $P = N - L - 1$ tuples, as depicted in Fig. 4.2 for L measurements. The rationale of this approach is to identify patterns for target AP's time series values of a network metric in L consecutive time periods. In fact, this concept is quite similar to the PBTP approach explained in Section 3.3 for predicting the next AP.

In turn, for the case of space-time-based prediction, not only the L previous measurements of the corresponding i-th AP but also the L previous measurements of each of the M_i highly correlated neighbours are used for the prediction of the next sample of i-th AP (i.e. the measurements $\{X_i^P\}$ and the measurements of the M_i highly cor-

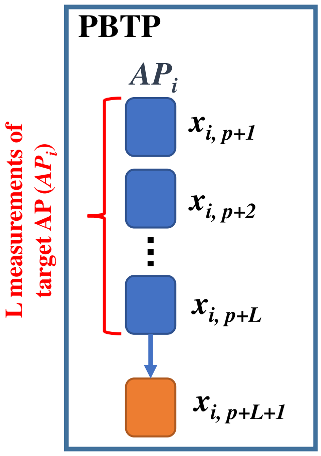


Fig. 4.2. Input and output data in the temporal prediction process (based on PBTP approach).

related neighbouring APs $\{X_m^P\}$ with $m = 1, \ldots, M_i$), for the prediction of $x_{i,p+L+1}$, as illustrated in Fig. 4.3.

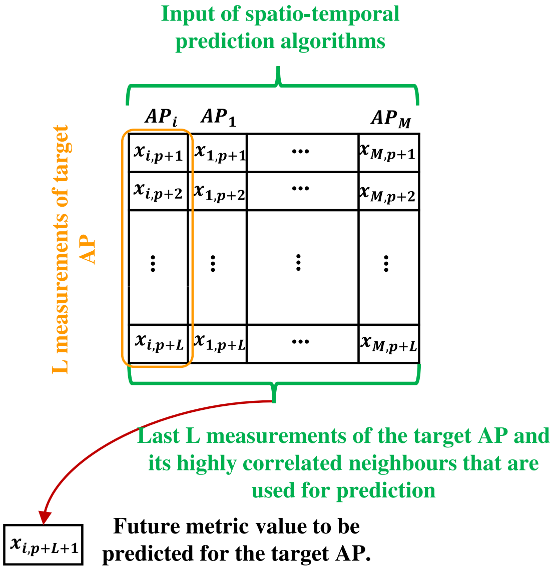

Input of spatio-temporal prediction algorithms
L measurements of target AP
AP_i
AP_1
AP_M
$x_{i,p+1}$
$x_{1,p+1}$
...
$x_{M,p+1}$
$x_{i,p+2}$
$x_{1,p+2}$
...
$x_{M,p+2}$
$x_{i,p+L}$
$x_{1,p+L}$
...
$x_{M,p+L}$
Last L measurements of the target AP and its highly correlated neighbours that are used for prediction
$x_{i,p+L+1}$
Future metric value to be predicted for the target AP.

Fig. 4.3. Input and output data in the spatio-temporal prediction process.

In the online prediction for a specific i-th AP, (see Fig. 4.1), the N measurements of this i-th AP and its M_i highly correlated neighbours are used as input. Then, this input data is prepared in a manner similar to that in the training step. Finally, the prediction step makes use of the input pre-processed data and the learnt $f(.)$ function in order to make the prediction in this specific AP. All the methodology processes are presented in Algorithm 1. Based on this approach, correlations are initially calculated for a given AP, identifying highly correlated neighbours, and subsequently, a model is trained and saved. During predictions at this AP, the highly correlated neighbours are utilized, employing the trained model represented by the function $f(.)$. Periodically, correlations are recalculated for this AP to monitor changes in the list of highly correlated neighbours, which may occur due to AP additions/removals or environmental changes. In such cases, retraining the model for this AP and updating the $f(.)$ function for prediction becomes necessary.

Algorithm 1 Temporal and Spatio-Temporal Prediction by RNN and CNN-RNN

```
Step 1: Data
Load historical data of target i-th AP: $X_i = \{x_{i,1}, x_{i,2}, ..., x_{i,N}\}$.
Determine neighbours of each AP based on received power using NDP.
Calculate correlations between each AP and its neighbours using Equation (4.1).
Step 2: Division to Training and Test
for each AP do
    Split the measurements into training and test sets:
    Training set: $X_{\text{train}} = \{x_{i,1}, x_{i,2}, ..., x_{i,N-s}\}$     s = the number of test samples
    Test set: $X_{\text{test}} = \{x_{i,N-s+1}, x_{i,N-s+2}, ..., x_{i,N}\}$
end for
Step 3: Define Offline Training Model
for each AP do
    if highly correlated neighbours exist then
        Extract measurements of the AP and its highly correlated neighbours.
        Split the measurements into training tuples $X_m^P$.
        Define the hybrid CNN-RNN model:
            Define: Model = Sequential()
            Set: Layer.CNN(nodes, activation)
            Layer.MaxPooling(stride_size)
            Layer.CNN(nodes, activation)
            Layer.MaxPooling(stride_size)
            Layer.RNN(nodes, activation, dropout)
            Layer.RNN(nodes, activation, dropout)
            Layer.Dense(1)
            Model.compile(optimizer='adam', loss='mse')
        Train the hybrid CNN-RNN model:
        for each training tuple $X_m^P$ do
                Model($X_m^P$) to predict $x_{m,p+L+1}$ given $L$ previous measurements.
        end for
    else
        Split the measurements into training tuples $X_i^P$.
        Define the RNN model:
            Define: Model = Sequential()
            Set: Layer.RNN(nodes, activation, dropout)
            Layer.RNN(nodes, activation, dropout)
            Layer.Dense(1)
            Model.compile(optimizer='adam', loss='mse')
        Train the RNN model:
            for each training tuple $X_i^P$:
                Model($X_i^P$) to predict $x_{i,p+L+1}$ given $L$ previous measurements.
    end if
end for
Step 4: Evaluation
Evaluate the predictions using MSE, RMSE, MAE, and $R^2$ score.
Step 5: Online Prediction
for each AP do
    if highly correlated neighbours exist then
        Use the spatio-temporal CNN-RNN model to predict the next sample.
    else
        Use the temporal RNN model to predict the next sample.
    end if
end for
```

4.4 ML Prediction Techniques

In this chapter, we perform both temporal prediction and spatio-temporal prediction. Before the advent of DL techniques, traditional or classical methods (e.g. Autoregressive Moving Average (ARMA)/Autoregressive Integrated Moving Average (ARIMA), Exponential Smoothing (ETS)), were commonly performed for time series analysis and prediction [191]. ARIMA, a widely used statistical method for analyzing and forecasting time series data, is particularly useful when handling recurring patterns or fluctuations. ARIMA makes predictions by taking into account factors such as previous historical data and data trends.

However, the use of DL algorithms may provide better predictions thanks to their inherent capacity to automatically learn complex and hierarchical representations from the input data within a reasonable computational time.

DL algorithms are suitable for identifying intricate patterns that cannot be captured using conventional statistical ML. Owing to their great performance, they have been used for different purposes, including time series data prediction [191, 237]. As highlighted by [177, 222], DL algorithms such as LSTM, GRU, CNN, and a combination of them are adequate to model and predict wireless traffic usage time series data. In recent years, new predictors (e.g. based on Transformers [200]) have been proposed and developed in order to enhance the prediction accuracy.

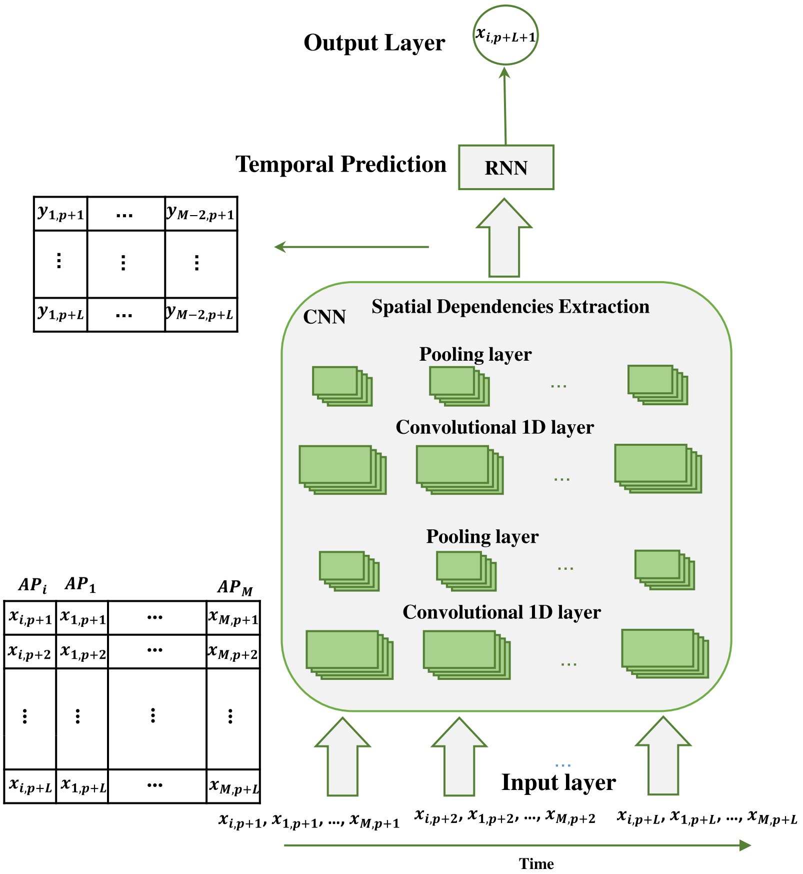
In the case of Wi-Fi traffic prediction, DL algorithms can effectively capture the spatial and temporal patterns present in the data, which are essential for accurate predictions. On the other hand, CNNs can analyze spatial patterns across different APs and identify relevant features that contribute to traffic fluctuations. Meanwhile, LSTM and GRU architectures, equipped with memory cells and gating mechanisms, excel at capturing long-term dependencies within the data. This ability is particularly advantageous for Wi-Fi traffic prediction, where past traffic patterns may have a significant impact on future ones. In addition, when adapted to time series data, the Transformer architecture can effectively model long-range dependencies in the context of Wi-Fi traffic prediction. Moreover, the aptitude of DL models to deliver improved performance with larger datasets further strengthens their utility. Given the substantial volume of Wi-Fi traffic data available, DL models can leverage their high capacity more effectively to generate accurate predictions without necessitating extensive handcrafted feature engineering. The amalgamation of these advantages renders DL algorithms highly suitable for Wi-Fi traffic prediction tasks, yielding superior results in comparison to traditional time series algorithms.

NNs, and in particular DL, capture the underlying pattern behind the measurements by giving certain weights to the connection between neurons of two sequential layers that derive the model. The DL methodology iteratively processes the dataset and adjust the weights so that the sum of differences between predicted and original values (which is known as the cost) is minimized, by running a process called Back-propagation [190].

On the other hand, Transformer, originally designed for natural language processing [200], has recently become an important interesting tool for time series prediction. It includes a so-called attention mechanism that can be useful for capturing dependencies between different time steps in a sequence.

A brief description of the classical and single time series methods considered in this work has been presented in Chapter 2 Section 2.3. A detailed description of how the two more novel ML methods are applied to available real data measurements in spatio-temporal prediction of our proposed methodology is presented below.

- **CNN-RNN hybrid algorithm:** We provide a detailed description of how the available real data measurements are processed in a way that can be used as input for the joint CNN-RNN algorithm-based prediction for spatio-temporal context. Fig. 4.4 illustrates the proposed architecture that combines CNN and RNN, where CNN is used for the spatial dependency extraction and RNN for the temporal analysis. As depicted in Fig. 4.4, the CNN architecture integrates the temporal measurements of the i-th AP and its M highly correlated APs to analyze spatial dependencies. Following two convolutional layers with 1D filters and two pooling layers, the dimensionality of the feature set is decreased. For instance, in the initial input vector $\{x_{i,p+1}, x_{1,p+1}, ..., x_{M,p+1}\}$, the resulting CNN output becomes $\{y_{1,p+1}, ..., y_{M-2,p+1}\}$. Here, $x_{i,p+1}$ denotes the target AP, while $\{x_{1,p+1}, ..., x_{M,p+1}\}$ represents the highly correlated APs, totaling M+1 features for input and M-2 features for output. The CNN output serves as input for the subsequent RNN phase (see Fig. 4.4) to predict the metric's value in the subsequent time period ($x_{i,p+L+1}$).


Output Layer
$x_{i,p+L+1}$
Temporal Prediction
RNN
CNN
Spatial Dependencies Extraction
Pooling layer
Convolutional 1D layer
Pooling layer
Convolutional 1D layer
Input layer
$x_{i,p+1}, x_{1,p+1}, ..., x_{M,p+1}$
$x_{i,p+2}, x_{1,p+2}, ..., x_{M,p+2}$
$x_{i,p+L}, x_{1,p+L}, ..., x_{M,p+L}$
Time

$y_{1,p+1}$	...	$y_{M-2,p+1}$
⋮	⋮	⋮
$y_{1,p+L}$	...	$y_{M-2,p+L}$

AP_i	AP_1		AP_M
$x_{i,p+1}$	$x_{1,p+1}$	...	$x_{M,p+1}$
$x_{i,p+2}$	$x_{1,p+2}$	...	$x_{M,p+2}$
⋮	⋮	⋮	⋮
$x_{i,p+L}$	$x_{1,p+L}$	...	$x_{M,p+L}$

Fig. 4.4. Hybrid CNN-RNN Architecture.

- **Transformer algorithm:** A comprehensive description is provided to illustrate how the available real data measurements are processed to generate an input data suitable for predictive modeling in the spatio-temporal context using Transformer. Fig. 4.5 depicts the general Transformer architecture for spatio-temporal prediction. The available data is used as input to the encoder while the decoder operates on the sequence input of $x_{i,p+L}$ to produce the decoder output $x_{i,p+L+1}$. Throughout the encoding phase, the traditional canonical self-attention mechanism is substituted with the ProbSparse self-attention as implemented in [238]. All the other sub-layers of the encoder remain unchanged and follow the

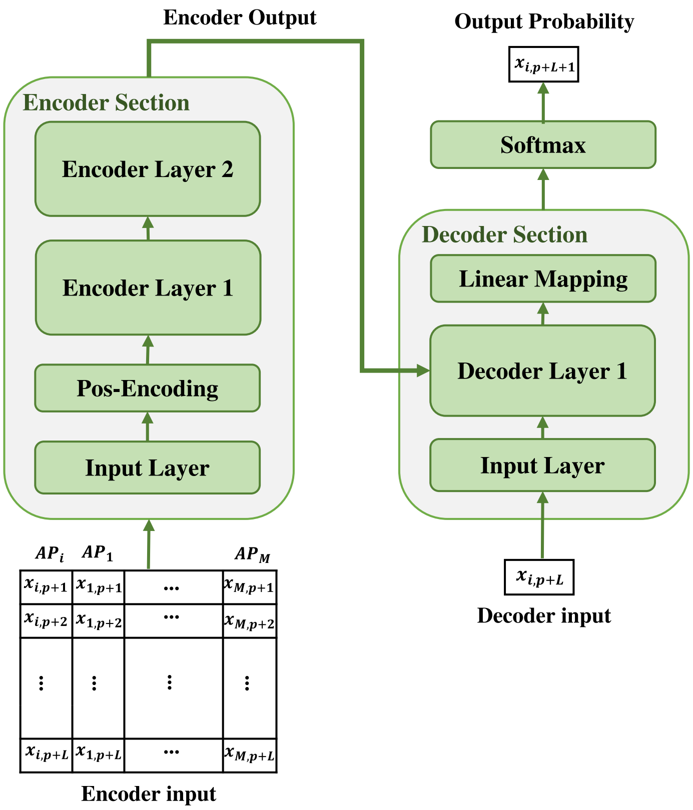


Fig. 4.5. Transformer Architecture.

standard configuration of a normal encoder. In ProbSparse attention, each key is permitted to attend exclusively to the dominant queries, a distinctive characteristic that enhances the efficiency of the model and performance in handling spatio-temporal predictions. Concerning the decoder, a standard decoder structure based on [200] is adopted. The decoder is composed of a stack of two identical multi-head attention layers, which facilitate the ability of the model to effectively capture complex dependencies and patterns in the data.

4.5 Scenario

To evaluate the proposed methodology, real measurements were collected from a total of 100 APs deployed in a University Campus of Universitat Politécnica de Catalunya, located in Barcelona over a three-month period. The collection and management of these measurements is done by means of the Cisco Prime Infrastructure [216]. This tool periodically collects a large amount of AP measurements to capture the network status and centralises all this information to generate useful statistics for prediction purposes and other analytical tasks. It's notable that the proposed prediction approach utilizes the data accessible through Cisco Prime Infrastructure, without necessitating additional information or causing extra communication overhead.

The proposed prediction methodology can be useful for the prediction of different user metrics. As an example, it's applied to predict future AP traffic values and transmission failures. The data is preprocessed in a form of time series data for each AP, separately. The prediction methodology utilizes the measurements collected on weekdays from September 9th to December 22th, 2019, spanning from 5:30 to 22:00 for every day with a periodicity T=30 minutes. This comprises 34 time periods per day, resulting in a total of D=75 days and N=2550 (34*75) measurements for each metric.

After pre-processing, the traffic load and transmission failures measurements for each single AP are derived. These two metrics are defined by equations (4.2) and (4.3). Equation (4.2) computes the normalized traffic by dividing the number of transmitted packets in each time period (TxC) by the maximum observed value in the measurement data ($MaxV$). On the other hand, the traffic transmission failures are calculated according to equation (4.3) where, FTx, STx and $STxR$ denote the number of failed transmitted packets, the number of successfully transmitted packets and the number of successfully transmitted packets after retransmissions, respectively.

$$\text{Traffic Load} = \frac{TxC}{MaxV} \tag{4.2}$$

$$\text{Failures} = \frac{FTx}{STx + STxR} \tag{4.3}$$

To demonstrate the methodology, we begin with a small scenario comprising 6 APs, depicted in Fig. 4.6. Within this scenario, the temporal prediction focuses on predicting future traffic and failure values for a specific AP (i.e. XSFA4PS205). Subsequently, we extend our analysis to encompass 100 APs distributed across seven buildings, each spanning four floors, to validate the obtained results.

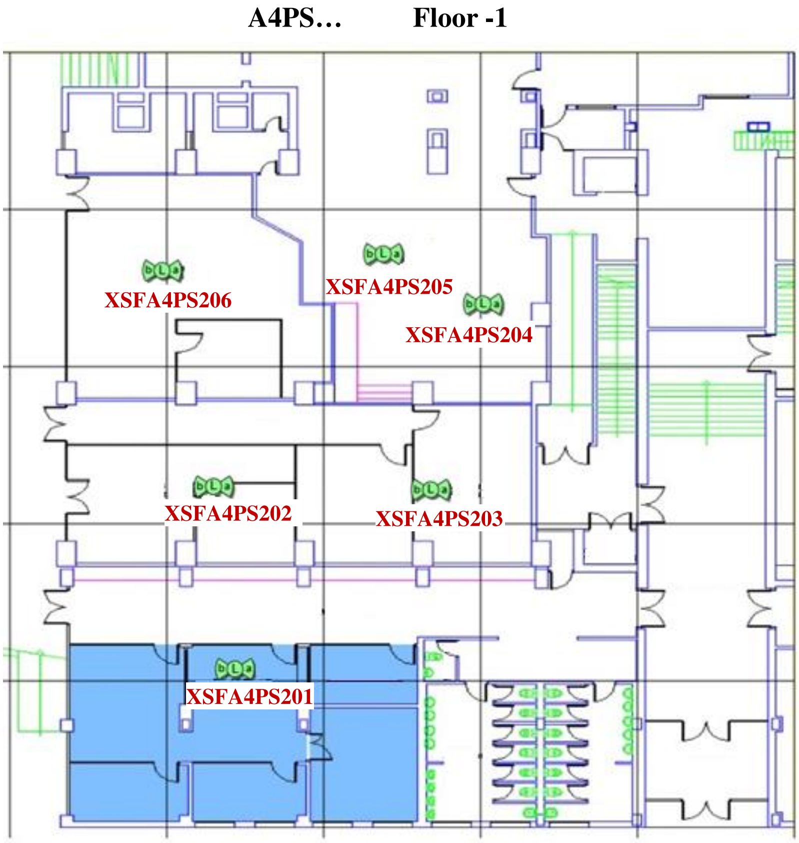


Fig. 4.6. Location of APs in study and class area.

The collected dataset comprises numerous features, but we specifically utilize the following:

- Time: this corresponds to the timestamp when the measurements have been taken.

- AP_name: this feature is related to the name of AP.
- Radio_type: it indicates the radio interface where the measurements are taken, in our case, predicting traffic in the 2.4GHz band.
- Number of transmitted packets (TxC): this metric is a counter that is increased every time a MSDU is transmitted.
- Number of successfully transmitted packets (STx): this metric is a counter that is increased every time a MSDU is transmitted successfully without the need of retransmission.
- Number of successfully transmitted packets after retransmissions (STxR): this is a counter that is increased every time a MSDU is successfully transmitted after one or more retries.
- Number of failed transmitted packets (FTx): this metric is a counter that is increased every time a MSDU frame is not transmitted successfully after a maximum number of retransmission attempts.

These counters are collected with a given periodicity (T=30 minutes) and set again to zero. Each time period in the dataset contains values for all features. The ”AP_name” and ”Radio_Type” values are strings, ”Time” is in date/time format, and the remaining features are integers. To prepare the dataset for prediction, we undergo several preprocessing steps. Initially, traffic load and failure measurements are derived for each AP by using equations 4.2 and 4.3, respectively (i.e. after this computation we have 2 features of traffic load and failures per AP). These measurements are then normalized by dividing them by the maximum collected measurement. Subsequently, they are converted to measurements with a specific periodicity of T=30 using the “asfreq” technique [194] in the Pandas Python library. The ”asfreq” technique in time series data facilitates the adjustment of data frequency to a specified frequency. This process fills in any missing values as needed, ensuring that the data aligns with the desired periodicity. During this conversion, any missing data is filled using the backfilling technique, ensuring that no more than one subsequent missing data point is encountered. Following these preprocessing steps, the total of 2550 observations are available for each AP, ready for further analysis and prediction. Before beginning the training phase for each AP, we selected M=5 as the maximum number of neighbours because considering the five most correlated neighbour APs captures sufficient spatial dependencies for accurate predictions without overwhelming the model with excessive information. In addition, we utilize the Augmented Dickey-Fuller test to determine the optimal lag. When conducting temporal prediction with only one feature and one column, the test straightforwardly identifies the lag. However, in the spatio-temporal prediction, where we consider multiple correlated APs (with column 1 denoting the main AP measurement and additional columns representing highly correlated APs), we select the maximum lag value among all columns. This approach ensures effective capture of the temporal dependencies inherent in the data. However, we have observed that the lag values for each column are frequently very close, resulting in minimal impact on prediction performance.

After preprocessing the dataset for each RNN, CNN, and Hybrid model implementation, we utilize a validation method known as walk-forward validation (also referred to as expanding window validation) [239], which is well-suited for time series data. This method involves splitting the dataset into training and validation sets, with a split ratio of 0.3, where 30% of the data is allocated for validation. The model is trained by using past data and subsequently validated on the next data point. This process is repeated by updating the training set each time with the most recent data. By doing so, the model can effectively adapt to changes in the time series pattern, providing insights into its predictive performance in real-time scenarios [239]. The hyperparameters that yield the optimal average result across all validation windows are then selected for the ML prediction models.

A comparative analysis of various DL prediction algorithms is performed. The comparison primarily focuses on evaluating prediction accuracy of the different algorithms. This assessment utilizes several metrics, including the R2 Score, Mean Square Error (MSE), Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) [191, 222]. The R2 Score, also referred to as the coefficient of determination, serves as a statistical measure quantifying the proportion of variability in the observed time series explained by a predictive model. It measures the model's goodness of fit by comparing the differences between the actual data and the model's predictions. It provides insight into the model's goodness of fit: a higher R2 score indicates that the model accounts for more of the variability in the data, while a lower score suggests that the model may not be a good fit for the data. Ranging from 0 to 1, the score represents no-fit to perfect fit, respectively. MSE calculates the squared average of the differences between predicted and actual values, whereas RMSE is the square root of MSE. On the other hand, MAE computes the absolute average of the residuals by summing the differences and dividing the result by the total number of samples. These metrics collectively provide comprehensive insights into the accuracy of the prediction models.

Furthermore, the various prediction methods are evaluated with respect to CT. This evaluation involves calculating the time spent during the training phase (TCT) in seconds for a specific AP or the average TCT for all the APs, as well as determining the runtime for online prediction (PCT) of a specific future value for a given AP or the average PCT for all APs, also in seconds. By considering both training and test runtime, our analysis provides valuable insights into the computational costs associated with the different prediction methods. The CT of the proposed methodologies is estimated utilizing a computer equipped with the following specifications: Intel(R) Core(TM) i5-11500 CPU @ 2.70GHz, 16.00 GB RAM, x64 based processor.

Hyperparameter tuning is a critical phase in enhancing the prediction accuracy of DL algorithms. However, when working with time series NN models, exploring numerous hyperparameter combinations can significantly prolong the training process [194]. Therefore, our focus was on tuning the hyperparameters with the most substantial impact on prediction accuracy. Our approach involved experimenting with various combinations of hyperparameters to identify the optimal ones for specific performance metrics. Initially, we examined with commonly used hyperparameters for time series forecasting as documented in [177, 194, 222]. In CNN model, we tested different combinations of the number of kernels $\{16, 32, 64, 128, 256\}$ and the number of layers $\{1, 2, 3\}$ to determine the best combination. Among these, the configuration with 16 kernels and 2 layers demonstrated the highest accuracy across the experiments. Similarly, in the RNN model, particularly using LSTM, we explored combinations of the number of cells $\{32, 50, 64, 100, 128, 256\}$ and the number of layers $\{1, 2, 3\}$. Notably, the configuration with 50 cells in each of the 2 layers demonstrated superior performance and was chosen as the optimal configuration. Finally, we optimized the hyperparameters for the CNN-RNN model by experimenting with various combinations of batch sizes $\{16, 32\}$ and numbers of epochs $\{50, 100\}$. The combination that yielded the highest performance consisted of a batch size of 16 and 100 epochs. Details of all the hyperparameters considered in the NN models are provided in Table 4.1.

Table 4.1. Hyperparameters of the CNN-RNN Algorithms (RNN refers to either SRNN, LSTM or GRU).

	Layers	Kernels	Kernel Size	Pool./Stride Size	AF
CNN	Conv1D.1	16	2	-	RelU
	Pooling1	-	-	Max/2	-
	Conv1D.2	16	2	-	RelU
	Pooling2	-	-	Max/1	-
RNN	Layers	Cells	Cell Type	Dropout Layer	AF
	Layer1	50	RNN	0.3	RelU
	Layer2	50	RNN	0.3	RelU

In our implementation, we considered hyperparameters similar to those outlined in the work by [238]. The Transformer architecture comprises 2 encoder layers and 1 decoder layer. We utilized the 'Adam' optimizer for the optimization process, initializing the learning rate at e^{-4}, and decayed it by 0.5 after each epoch. For the Transformer model, we configured the batch size to 16, the dimension of the model (d_model) to 512, the number of attention heads (n_head) to 8, the dropout rate to 0.3, and the dimension of the feed-forward layer (d_ff) to 2048. The training process is performed over approximately 10 epochs, incorporating suitable early stopping mechanisms (patience = 3) to prevent overfitting and ensure optimal model performance.

4.6 Results

4.6.1 Overview

This section outlines the results obtained from both the proposed temporal and spatio-temporal prediction methods. In order to demonstrate the performance of the proposed methodology, first, a straightforward scenario consisting of 6 APs located on the same floor (refer to Fig. 4.6) is examined. Subsequently, the analysis is extended to the 100 APs located in seven distinct buildings within the University Campus. The evaluation encompasses several metrics, including prediction accuracy, TCT, and PCT for all DL methods in the proposed methodology. The TCT involves two distinct phases: (i) preprocessing, which encompasses tasks such as computing traffic load and failure measurements from raw data, frequency conversion to 30-minute intervals, backfilling, and Augmented Dickey-Fuller test; and (ii) training, incorporating walk-forward validation. According to this, section 4.6.2 presents the results obtained only from temporal prediction, while section 4.6.3 describes the results of the spatio-temporal predictions.

4.6.2 Temporal Prediction

The prediction results for the deployed 100APs when running the temporal-based prediction are depicted in Table 4.2 and Table 4.3 for traffic load and transmission failures, respectively. In this case, the obtained prediction metrics are averaged across all APs. As shown in the tables, for all the methods, the results indicate highly accurate predictions, with LSTM demonstrating a slightly superior performance compared to other algorithms, although GRU closely follows LSTM and exhibits slightly better performance in MAE. Notably, LSTM achieves an R2 score of 89.73% and 89.87% for traffic load and failures prediction, respectively. This superior accuracy of LSTM can be attributed to its inherent capability to retain memory and capture sequential dependencies, enabling effective modeling of complex relationships. Furthermore, the CT required for training the model remains relatively low (in the order of tenths or hundreds of seconds) across all methods, facilitating rapid offline training and retraining processes to keep the model frequently updated, if needed.

Table 4.2. Comparison among different Prediction Techniques for all 100 APs Traffic time series data.

	R2 Score	MSE	RMSE	MAE	TCT(s)	PCT(s)
ARIMA	0.8646	0.00036	0.01889	0.01641	427.072	0.00024
SRNN	0.8826	0.00020	0.01428	0.00830	69.414	0.245
LSTM	**0.8973**	**0.00019**	**0.01364**	0.00813	110.862	0.300
GRU	0.8968	0.00020	0.01421	**0.00792**	110.204	0.296
CNN	0.7821	0.00084	0.02896	0.01671	58.258	0.204
Trans.	0.8704	0.00035	0.01881	0.01630	410.244	1.360

Table 4.3. Comparison among different Prediction Techniques for all 100 APs Failures time series data.

	R2 Score	MSE	RMSE	MAE	TCT(s)	PCT(s)
ARIMA	0.8745	0.00386	0.06216	0.04921	325.374	0.00025
SRNN	0.8830	0.00336	0.05795	0.04221	70.295	0.248
LSTM	**0.8987**	**0.00254**	**0.05042**	0.03854	112.134	0.303
GRU	0.8942	0.00257	0.05068	**0.03842**	110.358	0.300
CNN	0.7397	0.00686	0.08281	0.05958	59.318	0.207
Trans.	0.8872	0.00313	0.05592	0.03885	411.755	1.462

As shown in the tables, the Transformer method exhibits a marginally lower prediction accuracy (e.g. R2 score) compared to the other methods and entails longer TCT. Additionally, CNN demonstrates a relatively low TCT but provides the worse prediction accuracy because CNN is more suited for space-based predictions rather than temporal-based prediction. Furthermore, the performance of all DL prediction models is compared with the ARIMA model. As shown, ARIMA model yields slightly worse predictions than the other DL algorithms, except for CNN but requires more TCT. This is mainly because ARIMA uses the technique of recursive sampling to improve its performance. Moreover, there is a noticeable difference in the ARIMA's TCT for the case of traffic load and failures prediction. This difference is due to the fact that finding optimal order for each target AP measurements to achieve accurate predictions requires more TCT for traffic load than for failures. It is noteworthy that approximately 40 to 50 seconds are allocated to determining the best order using the auto-arima function. The auto-arima function is a feature in the ARIMA modeling technique that automatically selects the optimal order of the ARIMA model based on the provided data, helping to streamline the modeling process and improve prediction accuracy.

As per the PCT, it is observed from Table 4.2 and Table 4.3 that ARIMA demonstrates the lowest PCT, whereas Transformer exhibits the highest PCT due to its more complex architecture compared to other methods. For NNs, the PCT indicated in the tables, excluding preprocessing, is very low, typically falls between 0.20 to 0.30. However, the preprocessing time, which typically ranging between 0.6 to 0.8, needs to be added to the PCT. Consequently, the overall prediction time remains relatively low.

Fig. 4.7 illustrates a comparison between real values and predicted values of traffic load using LSTM and Transformer models during weekdays of the last week in autumn for a single AP (XSFA4PS205). As observed, the majority of the predicted measurements closely align with the real data.

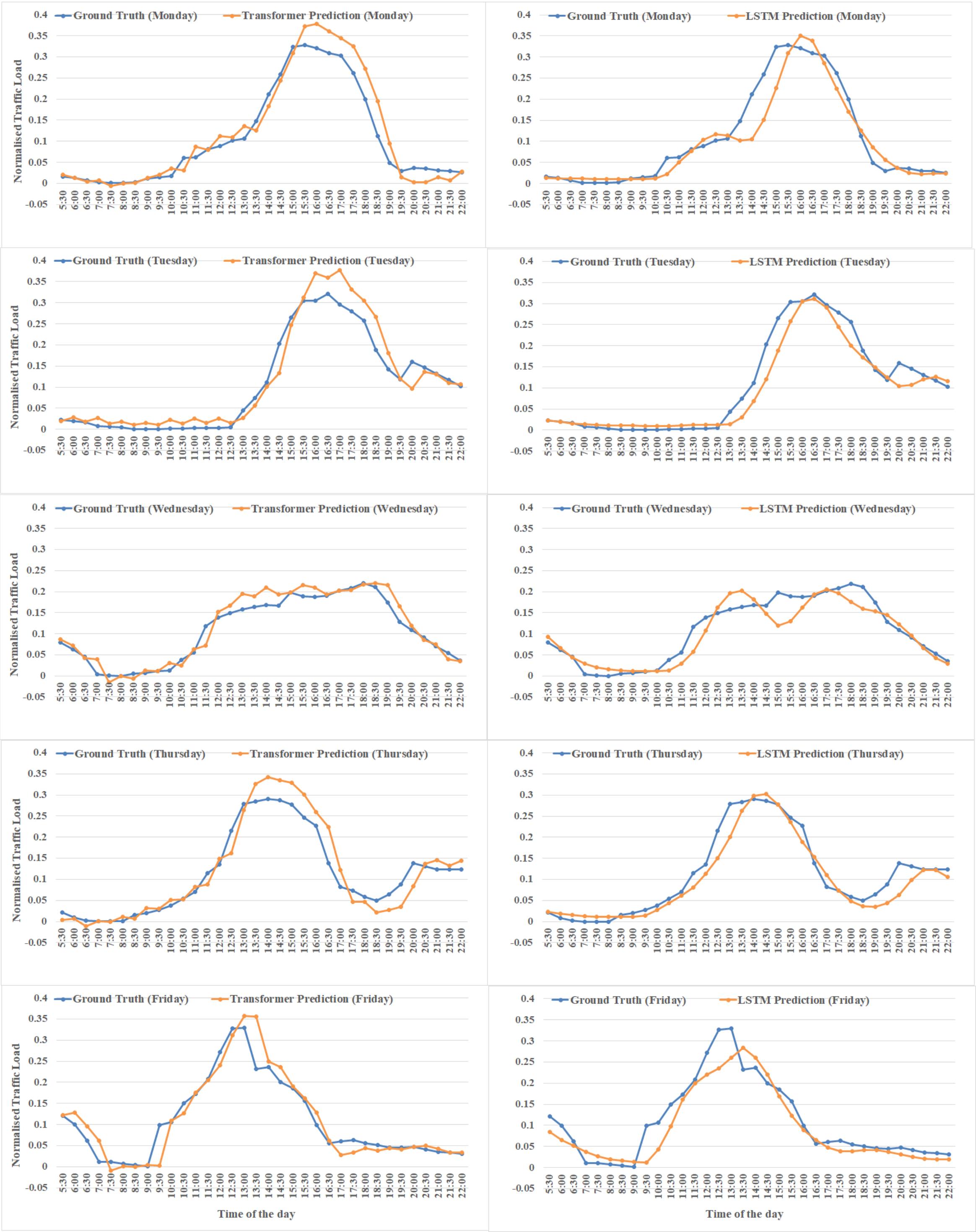


Fig. 4.7. Comparing the Real and Prediction traffic load of target AP(XSFA4PS205) using LSTM and Transformer on weekdays of the last week.

4.6.3 Spatio-temporal Prediction

In order to demonstrate the performance of the spatio-temporal prediction methodology in a relatively simple scenario, we consider the scenario of 6 APs depicted in Fig. 4.6. These APs communicate among themselves and neighbours are detected according to measured power. The APs are automatically designated as neighbours based on an established threshold of -80dBm. Subsequently, the Pearson correlation [220, 222] among these 6 APs is presented in Table 4.4 and Table 4.5. For the prediction of traffic and failures pertaining to XSFA4PS205, the methodology identifies highly correlated neighbouring APs with a correlation $C_{i,j} > 0.50$. Note that the term $C_{i,j}$ is calculated according to equation 4.1. The threshold of 0.50 is empirically determined to optimize the resulting R2 score. In the example delineated in Table 4.4, AP XSFA4PS205 and its four highly correlated neighbour APs (i.e., XSFA4PS201, XSFA4PS203, XSFA4PS204, and XSFA4PS206) are utilized for spatio-temporal prediction of traffic data. However, for the spatio-temporal prediction of failures in XSFA4PS205, all five neighbouring APs are considered, as illustrated in Table 4.5.

Table 4.4. Spatial Correlation for Traffic data (In order to simplification, the prefix "XSFA4PS" at the beginning of the AP name has been omitted).

Traffic	201	202	203	204	205	206
201	1	0.4950	0.5622	0.6824	0.5101	0.4907
202	0.4950	1	0.4185	0.5042	0.4172	0.4356
203	0.5622	0.4185	1	0.5985	0.5489	0.5049
204	0.6824	0.5042	0.5985	1	0.5384	0.4729
205	0.5101	0.4172	0.5489	0.5384	1	0.5130
206	0.4907	0.4356	0.5049	0.4729	0.5130	1

Table 4.5. Spatial Correlation for Failures data (In order to simplification, the prefix "XSFA4PS" at the beginning of the AP name has been omitted).

Failures	201	202	203	204	205	206
201	1	0.7122	0.8494	0.8612	0.7659	0.5229
202	0.7122	1	0.7123	0.7239	0.7038	0.4849
203	0.8494	0.7123	1	0.8221	0.7831	0.6046
204	0.8612	0.7239	0.8221	1	0.7985	0.5508
205	0.7659	0.7038	0.7831	0.7985	1	0.6041
206	0.5229	0.4849	0.6046	0.5508	0.6041	1

Based on this, the results derived from the spatio-temporal prediction methodology are presented in Table 4.6 and Table 4.7 and are also compared with the results of temporal prediction of this AP. Regarding temporal prediction, LSTM demonstrates superior performance compared to other methods, with the exception of MAE for traffic prediction, where GRU and LSTM exhibit similar performance, albeit with a slightly better performance in GRU. For the sake of benchmarking, within the spatio-temporal context, the initial three methods listed in the Tables (i.e., LSTM, GRU, and CNN) pertain to individual DL algorithms, whereas the last three methods (CNN-SRNN, CNN-LSTM, and CNN-GRU) correspond to hybrid CNN-RNN prediction algorithms. As illustrated in Tables 4.6 and 4.7, the results yielded by the spatio-temporal LSTM, CNN and GRU methods provide higher prediction accuracy, albeit accompanied by an increase in the TCT compared to the methods only based on temporal prediction.

Table 4.6. Comparison of different Traffic Prediction Methods for AP XSFA4PS205.

		DL Methods	R2 Score	MSE	RMSE	MAE	TCT(s)	PCT(s)
Temporal Prediction	Single	SRNN	0.8737	0.00121	0.03479	0.02479	71.3415	0.267
		LSTM	**0.8957**	**0.00099**	**0.03162**	0.02207	106.557	0.305
		GRU	0.8908	0.00105	0.03235	**0.02184**	107.354	0.305
		CNN	0.8754	0.00119	0.03456	0.02608	57.164	0.207
		Transformer	0.8670	0.00128	0.03574	0.02495	406.238	1.521
Spatio-temporal Prediction	Single	LSTM	0.9241	0.00073	0.02698	0.01966	248.245	0.302
		GRU	0.9197	0.00077	0.02775	0.01971	238.142	0.286
		CNN	0.8854	0.00100	0.03164	0.02645	98.455	0.187
		Transformer	0.9174	0.00082	0.02868	0.02073	523.15	1.891
	Hybrid	CNN-SRNN	0.9350	0.00062	0.02496	0.01860	152.778	0.234
		CNN-LSTM	**0.9521**	**0.00046**	**0.02143**	0.01612	233.547	0.271
		CNN-GRU	0.9465	0.00051	0.02262	**0.01599**	223.0490	0.264

Table 4.7. Comparison of different Failures Prediction Methods for AP XSFA4PS205.

		DL Methods	R2 Score	MSE	RMSE	MAE	TCT(s)	PCT(s)
Temporal Prediction	Single	SRNN	0.8257	0.00939	0.09692	0.07217	72.159	0.256
		LSTM	**0.9073**	**0.00498**	**0.07060**	**0.05607**	108.337	0.307
		GRU	0.8896	0.00594	0.07708	0.05750	105.474	0.298
		CNN	0.8700	0.00699	0.08361	0.06699	57.170	0.254
		Transformer	0.9044	0.00506	0.07116	0.05614	406.684	1.421
Spatio-temporal Prediction	Single	LSTM	0.9286	0.00382	0.06177	0.04748	254.754	0.261
		GRU	0.9118	0.00471	0.06865	0.05178	247.456	0.248
		CNN	0.9041	0.00513	0.07157	0.05555	116.354	0.203
		Transformer	0.9265	0.00404	0.06354	0.05004	534.799	1.721
	Hybrid	CNN-SRNN	0.9551	0.00239	0.04895	0.03864	157.246	0.252
		CNN-LSTM	**0.9599**	**0.00214**	**0.04631**	**0.03698**	238.472	0.302
		CNN-GRU	0.9523	0.00255	0.05048	0.03919	237.576	0.294

In any case, this increase in TCT is not relevant since the training is performed offline. The online computational time (PCT) for both temporal and spatio-temporal predictions is similar. In Table 4.6 and Table 4.7, the Transformer's accuracy closely aligns with that of GRU and LSTM, respectively. Focusing on the hybrid CNN-RNN models based on the architecture depicted in Fig. 4.4, it is worth noting that they provide a superior prediction performance with a reduced TCT when compared to the single DL methods (see Tables 4.6 and 4.7). The reason is that the inclusion of the first CNN process in the hybrid methodology, in order to facilitate the identification of spatial relationships among APs, reduces the number of features after 1D-Conv and pooling layer, thus reducing the TCT of the latter RNN process. Tables 4.6 and 4.7 highlight that the hybrid CNN-LSTM exhibits the highest prediction accuracy for both traffic usage and failures of a target AP, except for MAE in traffic load, where CNN-GRU slightly outperforms CNN-LSTM (see Table 4.6). Note also that CNN-SRNN provides a relatively high prediction accuracy (e.g. in terms of R2 score) with a significantly lower TCT than CNN-LSTM. The PCT for both single and hybrid NN models exhibits minimal difference or is closely matched. The PCT of the Transformer model in spatio-temporal prediction is higher compared to other models. However, when compared to temporal prediction, it remains relatively similar or close.

It is noteworthy that having a higher number of correlated neighbours may not necessarily lead to higher accuracy in predictions. The quality of these correlations, meaning how strong they are, is also important. To demonstrate this point, we consider a quantitative example involving a target AP of XSFA4PS205 in failure measurements.

As depicted in Table 4.5 (related to spatial correlation for failures data), adjusting the correlation threshold from 0.50 to 0.77 yields two highly correlated APs for the target AP. Subsequently, we assess the performance of CNN-LSTM model on failures metric across this revised threshold. On the other hand, if we set a threshold of 0.8, there would not be any highly correlated neighbour. Therefore, the methodology runs only temporal prediction based on LSTM. Table 4.8 illustrates a slight enhancement in results as the threshold is increased from 0.5 to 0.77, leading to a reduction in the number of correlated APs. However, an excessively high correlation threshold can lead to situations wherein an AP lacks neighbours with such high correlation (e.g. threshold of 0.8 in the Table 4.8). This may cause that relevant information of neighbouring APs is not considered for prediction leading to worse prediction results. Furthermore, we have included Table 4.9 to analyze the impact of the threshold value on traffic load data. For threshold 0.5, the methodology utilizes spatio-temporal prediction with CNN-LSTM, while for a threshold of 0.77, the methodology employs only temporal prediction with LSTM (i.e. there are no highly correlated neighbours). In order to deal with this trade-off, we have selected a correlation threshold of 0.5.

Table 4.8. Comparison between three thresholds, (C_i, j)s for failures data of AP XSFA4PS205.

C_i, j	R2 Score	MSE	RMSE	MAE
0.5	0.9599	0.00214	0.04631	0.03698
0.77	0.9684	0.00167	0.04083	0.03252
0.8	0.9073	0.00498	0.07060	0.05607

Table 4.9. Comparison between two thresholds, (C_i, j)s for traffic load data of AP XSFA4PS205.

C_i, j	R2 Score	MSE	RMSE	MAE
0.5	0.9521	0.00046	0.02143	0.01612
0.77	0.8957	0.00099	0.03162	0.02207

The spatio-temporal prediction method has also been undergone evaluation across all the 100 APs situated in seven different buildings. After the analyzing the traffic correlation among the various APs, it was observed that 42APs have one or more neighbouring APs exhibiting a correlation $C_{i,j}$ higher than the specified threshold (i.e. $C_{i,j} = 0.50$). Regarding the failures' correlations among APs, 55APs with one or more highly correlated neighbours are identified. In order to compare the different spatio-temporal prediction DL methods, Table 4.10 and Table 4.11 present the obtained results when exclusively running the spatio-temporal prediction method for the APs with one or more highly correlated neighbours (i.e. 42 APs for traffic predictions and 55APs for failure predictions). The obtained prediction metrics correspond to the averaged values of each metric across all the APs. According to Table 4.10 and Table 4.11, the hybrid DL methods outperform the prediction accuracy metrics compared to single DL models. The average TCT for spatio-temporal prediction across 42 APs for traffic load and 55 APs for failures is relatively similar to that of AP XSFA4PS205. The PCT for NN methods ranges between 0.20 and 0.32 seconds, whereas Transformers exhibit a higher PCT compared to others.

Finally, the performance of the different prediction methods is summarized in Table 4.12 and Table 4.13 when considering the LSTM algorithm. Specifically, Table 4.12 compares the different proposed temporal and spatio-temporal predictions for

Table 4.10. Comparison of different Traffic Prediction Methods for the 42 APs with one or more highly correlated neighbours.

DL Methods		R2 Score	MSE	RMSE	MAE	TCT(s)	PCT(s)
Single	LSTM	0.9250	0.00061	0.02464	0.01518	247.488	0.301
	GRU	0.9202	0.00050	0.02240	0.01482	241.516	0.313
	CNN	0.8471	0.00114	0.03378	0.02186	100.225	0.206
	Transformer	0.8930	0.00080	0.02834	0.01962	435.09	1.457
Hybrid	CNN-SRNN	0.9313	0.00057	0.02392	0.01483	153.232	0.233
	CNN-LSTM	**0.9371**	**0.00047**	**0.02161**	**0.01364**	235.086	0.273
	CNN-GRU	0.9345	0.00050	0.02243	0.01422	225.047	0.271

Table 4.11. Comparison of different Failures Prediction Methods for the 55 APs with one or more highly correlated neighbours.

DL Methods		R2 Score	MSE	RMSE	MAE	TCT(s)	PCT(s)
Single	LSTM	0.9065	0.00278	0.05270	0.04141	248.958	0.259
	GRU	0.9058	0.00268	0.05178	0.04151	243.783	0.258
	CNN	0.8046	0.00574	0.07579	0.05967	114.481	0.207
	Transformer	0.9050	0.00270	0.05200	0.04096	486.110	1.768
Hybrid	CNN-SRNN	0.9081	0.00268	0.05178	0.04073	153.893	0.240
	CNN-LSTM	**0.9241**	**0.00223**	**0.04720**	**0.03727**	234.192	0.271
	CNN-GRU	0.9229	0.00225	0.04745	0.03733	232.667	0.251

the 42 APs with one or more highly traffic correlated neighbours and the 55 APs with one or more highly failure-correlated neighbours. As illustrated, the proposed hybrid spatio-temporal prediction methods, particularly CNN-LSTM, demonstrate slightly higher prediction accuracy (i.e. R2 Score) at the expense of an increase in the average TCT compared to the only temporal prediction for both traffic load and transmission failures. The PCT values for all methods range from 0.259 to 0.304 seconds, indicating relatively low computational overhead.

Table 4.12. Comparison of different Proposed Temporal and Spatio-temporal Prediction Methodologies for Traffic Load and Failures based on LSTM.

		Temporal Prediction			Spatio-temporal Prediction					
		LSTM			Single LSTM			Hybrid CNN-LSTM		
Considered APs	Metric	R2 Score	TCT(s)	PCT(s)	R2 Score	TCT(s)	PCT(s)	R2 Score	TCT(s)	PCT(s)
42 APs	Traffic Load	0.9175	111.700	0.298	0.9250	247.488	0.301	0.9371	235.086	0.273
55 APs	Tx. Failures	0.9043	113.972	0.304	0.9065	248.958	0.259	0.9241	234.192	0.271

Table 4.13. Performance of the Combined Temporal and Spatio-temporal Prediction Methodology for Traffic Load and Failures based on LSTM.

		Temporal Prediction			Combined Temoporal and Spatio-temporal Prediction					
		LSTM			Single LSTM			Hybrid CNN-LSTM		
Considered APs	Metric	R2 Score	TCT(s)	PCT(s)	R2 Score	TCT(s)	PCT(s)	R2 Score	TCT(s)	PCT(s)
100 APs	Traffic Load	0.8973	110.862	0.300	0.8989	171.265	0.299	0.9041	165.782	0.284
100 APs	Tx. Failures	0.8987	112.134	0.303	0.9000	191.341	0.271	0.9101	182.795	0.287

Table 4.13 demonstrates the performance of a combined methodology. For comparison purposes, the only temporal-based predictions with the LSTM algorithm (i.e. first column of prediction part in the Table) are presented as a benchmark in Table 4.13. This comparison is performed for all the 100 APs. The combined methodology

has two possibilities: (i) for APs where no highly correlated neighbour is identified, temporal prediction is performed based on LSTM and in cases where some highly correlated neighbours are identified, spatio-temporal prediction is conducted based on Single LSTM (the second column in Table 4.13), and (ii) for APs where no highly correlated neighbours are identified, only temporal prediction based on LSTM is executed, while spatio-temporal prediction based on CNN-LSTM is employed for APs with highly correlated neighbours (the third column in Table 4.13). In our dataset of 100 APs, the methodology applies spatio-temporal prediction to 42 APs for traffic load and temporal prediction to the remaining 58 APs. Similarly, for failures, spatio-temporal prediction is applied to 55 APs and temporal prediction to the other 45 APs. We present the performance of our data under these combined settings in Table 4.13. As shown, the combined temporal LSTM and spatio-temporal CNN-LSTM (the third column) exhibit slightly superior prediction accuracy compared to temporal LSTM alone, albeit with a higher TCT. However, the PCT values for both temporal prediction and the combined models remain very similar.

4.7 Implementation Aspects in Real Network

The escalating demand for enhanced network management strategies within Wi-Fi environments has been underscored by the rapid development and increased usage of Wi-Fi technologies. In response to this imperative need, a novel methodology has been devised to predict future network metrics. This proactive approach not only enhances network reliability and performance but also elevates the overall user experience by mitigating downtime and congestion. Additionally, the potential for cost optimization is evident through reduced reactive maintenance efforts and informed resource investments, thereby highlighting the significance of this research in tackling real-world challenges encountered in network management.

Implementing our methodology in a real network entails defining and addressing Key Performance Indicators (KPIs) relevant to the specific network environment and operational requirements. KPIs such as TCT, PCT, prediction accuracy serve as crucial metrics for evaluating the performance of prediction methods in a real network setting. In our evaluation, various methods were assessed for temporal and spatio-temporal prediction in a Wi-Fi network. Across all methods, the TCT for training the models remained relatively low, facilitating fast offline training and retraining processes. This ensures the adaptability of models, allowing for frequent updates if necessary, without incurring significant computational overheads. Moreover, the PCT associated with both temporal and spatio-temporal prediction methods was found to be minimal, enabling nearly real-time predictions. This fast prediction capability is crucial in supporting timely decision-making processes and enhancing network responsiveness to dynamic operational conditions.

To scale up to a larger number of APs (i.e., adding new APs), the process involves collecting data for each new AP and its neighbours, preprocessing the data (such as filling missing data), calculating correlations, and executing the training process to obtain the trained model. However, it may be necessary to retrain everything at certain instant of time in the future, perhaps every few months. Furthermore, in our scenario, adapting to new circumstances might involve either fine-tuning the model's trainable parameters or full retraining it. For instance, in the context of a University Campus, when there are changes in classroom setups or the introduction of new academic programs, adjusting the predictive model becomes necessary. This adaptation entails fine-tuning the model's parameters or retraining from scratch to better suit the updated circumstances. In our dataset, after splitting the training dataset to training and validation sets (70-30) using walk-forward validation, the model considers approximately one month for validation. The trained model remains valid throughout the entire period covered by our available dataset, but retraining from scratch might be necessary in the future. Generally, evaluating the frequency of retraining requires more available data, which is outlined as part of future work.

4.8 Conclusion

Prediction of future values of network metrics (such as future traffic values at the APs, future obtained performance metrics, etc.) can be useful to make proactive decisions over the network by means of e.g. load balancing, proactive resource allocation, congestion control, etc., that can enhance the network performance. This study proposes a new approach for predicting future specific parameter values based on historical data and leveraging spatial correlations among neighbouring APs. The methodology commences with an analysis of spatial correlations among a cluster of APs within a given area. Depending on the outcomes of this analysis, it either conducts a temporal prediction grounded in past measurements of the target AP or a spatio-temporal forecast that integrates historical data from both the target AP and its highly correlated neighbouring APs. The proposed prediction methodology harnesses NNs and unfolds in two phases. Initially, historical data is utilized to train the NN in an offline process. Then, newly collected data is fed into the trained NN during the prediction phase.

Various DL methods have been analysed, including SRNN, CNN, GRU, LSTM and Transformer for both temporal and spatio-temporal prediction tasks. Moreover, hybrid DL algorithms have been proposed, starting with a CNN phase to capture spatial correlations, followed by an RNN phase to leverage temporal correlations. The proposed methodology has been evaluated using real data obtained from a Wi-Fi network deployed on a University Campus to predict future AP traffic and transmission failures. Employing hybrid DL techniques, the methodology achieved impressive prediction accuracy, with average R2 scores reaching 93.7% and 92.4% for traffic load and failures, respectively. The computational overhead, indicated by TCT and PCT, remained modest, with TCT in the order of hundreds of seconds and PCT below 2 seconds. In particular, integrating spatial correlations among neighbouring APs significantly improved prediction accuracy in spatio-temporal predictions compared to temporal predictions alone, albeit with a marginal increase in TCT. However, PCT values for both temporal and spatio-temporal prediction NN models remained largely consistent. Furthermore, performance evaluation of the combined temporal and spatio-temporal prediction methodology demonstrated slight enhancements in prediction accuracy when exploiting spatial relationships among APs with one or more highly correlated neighbours.

The chapter has also discussed the practical implementation of our methodology within an real network context. This involved two key aspects: Firstly, a comparative analysis of various ML techniques was conducted, focusing on KPIs pertinent to real-world network operation and management. Metrics such as accuracy, TCT, and PCT were analyzed to assess the strengths and limitations of each technique. Secondly, considerations regarding scalability were briefly addressed, particularly with regards to accommodating a larger number of APs and the necessity of periodic model retraining to maintain prediction efficacy in dynamic network environments.

Chapter 5

Wi-Fi Traffic Classification

5.1 Introduction

The escalating need for high-speed data transmission has spurred considerable research into developing novel methods for identifying and categorizing network traffic. Ineffectively managed mechanisms in this regard can detrimentally affect critical network operational tasks such as Fault Tolerance, Traffic Engineering, QoS provisioning, and Dynamic Access Control. For example, in Fault Tolerance, quickly identifying traffic patterns can help in the rapid detection and isolation of network failures, ensuring minimal disruption to services. In Traffic Engineering, understanding the nature of the traffic allows for better routing decisions, optimizing the flow of data across the network and preventing congestion. QoS provisioning relies on precise traffic categorization to allocate the appropriate bandwidth and resources to different types of traffic, ensuring that high-priority services receive the necessary support to function effectively. Lastly, Dynamic Access Control can benefit from accurate traffic identification by adjusting access policies in real-time based on the type and behavior of the traffic, enhancing both security and efficiency. Essentially, an accurate traffic identification process enables the effective management of available network resources, thereby facilitating more precise and resilient resource allocation strategies [240]. Traffic classification methods are typically categorized into three main groups, namely port-based, payload-based known as Deep Packet Inspection (DPI), and ML-based techniques [240]. Port-based identification is a straightforward approach that associates network applications with well-defined port numbers, a method that still finds widespread use. While this approach has been dependable for many applications, it faces challenges with the growing trend of applications employing dynamic port numbers, which can impact its reliability. Payload-based methods involve the scrutiny of packet payloads to discern the nature of the traffic, a valuable technique in many contexts. However, the efficacy of this approach may be limited when dealing with privacy regulations and encryption practices, which can restrict access to packet payloads. Conversely, DPI classification examines both packet header and payload, entails substantial computational overheads and necessitates ongoing manual signature maintenance, making it a less practical option for traffic classification in some contexts. ML-based methodologies present a viable solution to circumvent certain constraints inherent in port-based and payload-based approaches. Specifically, ML techniques demonstrate the capacity to classify network traffic by leveraging traffic statistics that are independent of application protocols. These statistics encompass metrics such as flow duration, packet length variance, maximum and minimum segment sizes, window size, round trip time, and packet inter-arrival time. Moreover, ML approaches offer the advantage of reduced computational complexity and the ability to effectively discern encrypted traffic [189].

Building upon the foundation of traffic classification, the applicability of traffic classification extends to various network management tasks, including the prioritization of services with strict latency requirements such as XR traffic in Wi-Fi networks. The XR merges physical and virtual realms, fostering immersive environments through technologies such as Augmented Reality (AR), VR, and Mixed Reality (MR). XR's deep interactivity and immersion provide users with captivating virtual experiences globally [241]. In the future years, the widespread integration of XR into our daily lives, impacting areas like education, work, health, and entertainment, will bring about significant social changes. The XR market is expected for significant growth, with projection indicating an increase from USD 105.58 billion in 2023 to USD 472.39 billion by 2028, at a robust Compound Annual Growth Rate (CAGR) of 34.94% during the forecast period [242].

Wi-Fi technology serves as a vital component in enabling VR experiences [241]. A typical setup involves running a video game on a robust desktop system and utilizing a Wi-Fi network to transmit the game's visuals to end-users, often in the form of Head Mounted Displays (HMDs) or VR headsets. In the downlink process, the desktop sends the video frames of the VR game to the VR headset, allowing users to immerse themselves in the visual content. Concurrently, in the uplink phase, the VR headset transmits motion data back to the desktop. This bidirectional data exchange facilitates a feedback loop where the desktop generates subsequent VR video frames based on the user's current viewpoint [243, 244]. Throughout this discussion, we will refer to this setup as VR edge streaming over Wi-Fi.

The successful integration of XR technologies into society hinges on the ability of wireless networks to meet rigorous performance demands. Real-time and interactive XR applications involve the exchange of high-quality video and metadata, necessitating high throughput (hundreds of Mbps per user) and low latency (ms or even sub-ms), which often present conflicting objectives. Present wireless networks are not fully equipped to efficiently support the widespread utilization of XR applications, prompting both academic and industrial sectors to pursue innovative solutions, aiming for readiness by 2030. Despite Wi-Fi's expected continued dominance as the preferred wireless access solution in unlicensed bands, its current limitations in providing efficient low-latency guarantees highlight the need for further advancements. Latency emerges as a significant concern in VR, impacting both wired and wireless systems [243]. Excessive latency in VR services can create a noticeable delay between the actions in the VR video game and the corresponding visual feedback, potentially leading to motion sickness and diminishing the overall user experience [243]. This inconsistency arises not only from transmission delays but also from the jitter in feedback during uplink transmission [243]. Effective Wi-Fi QoS management holds the potential to significantly enhance user experiences by minimizing latency and jitter in real-time applications like gaming, thereby ensuring uninterrupted immersive experiences and reducing lag during access to cloud and edge services [245].

In light of the importance of network traffic classification and the need to identify VR traffic, the novelty of this chapter is the proposal of an interactive VR ML traffic classification model in the edge streaming VR scenario. The proposed model evaluates the use of some common classifiers, fed with the features previously extracted. The model is evaluated by using VR traffic traces coming from a multi-user VR scenario, and using single-user traces from a VR framework not included in the training traces. Finally, a Wi-Fi network environment is simulated in order to illustrate how the model can be used to detect VR traffic, and give it higher priority to improve its QoS.

The rest of the chapter is organized as follows. Section 5.2 provides an overview of the existing literature. Section 5.3 describes the experimental design, including dataset creation, and traffic classification methods. The obtained results in both feature extraction and traffic classification are presented in Section 5.4, while the testing process of the model in a multi-user experimental setup and the evaluation of VR traffic identification in a Wi-Fi environment simulation are discussed in Section 5.5. Finally, conclusion is drawn in Section 5.6.

5.2 Related Work

In the context of network traffic identification, the port-based and DPI traditional methods require manual tuning, whereas an ML-based approach works seamlessly after training. The early network traffic classification solutions used port numbers to classify network traffic to their correspondence protocols. The Internet Assigned Numbers Authority (IANA) assigned standard port numbers to different services or protocols [246]. These port numbers are divided into three ranges: system ports (0-1023) known as standard ports, user ports (1024-49,151), and dynamic ports (49,152-65,535). Typically, the classification process involves inspecting the port numbers in Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) packets. For instance, port number 80 is associated with the HTTP protocol. The paper [247] assessed the port-based identification approach for classifying network traffic, such as HTTP and FTP. Their evaluation showed an accuracy of no more than 70%, regardless of the number of packets observed. The paper [248] examined the effectiveness of port-based methods for classifying P2P traffic. Their data indicated that standard ports accounted for only 30% of the dataset, resulting in low performance for the classification. The DPI approach overcomes the shortcomings of port-based classification since it does not rely on port numbers. Consequently, it avoids issues related to masquerading and port randomization [249]. DPI inspects the payload of network traffic to identify the originating applications. By analyzing the content of packets—including characters, strings, bit patterns, and symbols—a signature is created to classify applications. A signature library contains records, each associated with a specific application. This method allows the classifier to examine the content of individual packets or groups of packets and compare them against the signature library. If a match is found, the traffic is identified and associated with an application. This technique enhances accuracy compared to port-based identification, even when applications use non-standard ports. The paper [250] proposed a framework named Lightweight DPI (LW-DPI) aimed at reducing the detection process's overhead while maintaining acceptable accuracy. This solution classifies network traffic by examining the content of a limited number of packets or a fraction of the payload in a given packet. The evaluation used network traffic collected from commercial ISPs and local university laboratories, covering multiple network protocols. The accuracy results were up to 99% for P2P and mail network traffic. The paper [251] introduced the Bitcoding framework, which generates bit-level DPI signatures for traffic classification. It analyzes the first n bits of the network flow, extracts signatures, and converts them into a state transition machine for further comparison. Hamming distance was integrated into their system to reduce collisions and increase the number of targeted applications.

The dynamic nature of ports and encryption techniques in both port-based and DPI methods has led to the increased prominence of ML-based approaches as widely adopted technologies for traffic identification. [252, 253]. In recent years, scholars have made research into the domain of traffic identification. Researchers in [254], identify cloud game video traffic by proposing a novel approach for extracting distinctive features from video scenes to distinguish traffic patterns and then an adaptive distribution distance-based feature selection technique for feature selection. In addition to conventional statistical features derived from standard network attributes (e.g. packet payload size, packet inter-arrival time), the authors defined the 'peak point' as the maximum data transmission in a given period for the feature extraction process. This innovation aimed to enhance the video identification process, reducing unnecessary costs, time overhead, and potential negative impacts on the identification model.

In the paper [255], researchers focused on classifying Cloud Gaming (CG) traffic to optimize the use of a low-latency queue within the L4S (Low Latency, Low Loss, Scalable Throughput) architecture. The objective is to minimize latency fluctuations that could negatively impact QoS at the network edge. Three classification models (Thresholds, DT, RF) were developed and evaluated to distinguish CG traffic from other high-bitrate UDP-based applications. These models were constructed using 12 features derived from network flow data, including packet sizes and Inter-Arrival Times (IAT). Additionally, the study introduced a fully functional implementation of the classifier in the form of a micro-service architecture, suitable for deployment as Virtual Network Functions (VNFs) following the Network Function Virtualization (NFV) paradigm. In [256, 257], researchers conducted model tuning in SVM and evaluated various SVM kernel functions, including linear, polynomial, sigmoid, and radial kernels, to classify Internet traffic categories such as database, mail, www, multimedia, game, and service. The results indicated that the Radial Kernel exhibited superior performance compared to the other kernels. The researchers employed a sequential forward feature selection algorithm to improve accuracy by reducing irrelevant and redundant features. Notably, in [256], the authors compared the results of SVM with unsupervised K-means clustering. The paper [258] compared the accuracy of C5.0 with C4.5, SVM, and NB learning algorithms. Extracted features comprised packet rate, data rate, and inter-arrival time statistics (minimum, mean, maximum, and standard deviation). Using a private dataset of 17 applications (e.g. FTP, HTTP, Skype, Game, BitTorrent) collected in their labs, the experimental evaluation demonstrated C5.0's superior performance.

In the paper [240], the authors introduced a ML framework based on NB designed for real-time application classification, particularly focusing on video network traffic. The framework was trained using 13 statistical features, encompassing packet arrival time, the average of decimal values, and the average value of IP datagram length. Feature extraction involved LAN frame length and IPv4 header fields, utilizing a feature extractor script in a three-packet window. This calculated values like Time between Packets, Mean and Variance of Time between Packets, IP Total Length, Time to Live, and Protocol field. They collected the dataset in their labs, comprising YouTube and Netflix videos, along with files downloaded. The classification module effectively distinguished between Netflix streaming video, YouTube streaming video, and background traffic, particularly involving two file downloads.

In the paper [259], the authors tackled the challenge of reducing the computational resources required by the traditional SVM learning classifier in the context of internet traffic classification. They introduced Incremental SVM (ISVM), a concept aimed at decreasing the high training costs associated with memory and CPU usage. Additionally, the authors put forth the Authenticator ISVM (AISVM) framework, leveraging valuable information from prior training datasets. Comparative experiments with NB and NBKDE (Naive Bayes algorithm with kernel density estimation) learning algorithms revealed lower accuracy than SVM. The study demonstrated that the proposed frameworks (ISVM and AISVM) yielded higher accuracy results while utilizing fewer computational resources than SVM. Note that features included simple statistics about packet length, inter-packet time, and information derived from traffic flows. The study [260], aimed to boost the accuracy of a traffic classification model based on SVM. It introduced two modules: one for feature selection and another for optimizing the SVM algorithm. The feature module used an innovative algorithm to extract the most representative features, while the classifier module employed an Improved Grid Search algorithm to enhance parameter selection. Evaluation on the Moore dataset showed high accuracy compared to SVM, NB, and kNN algorithms in a smaller feature space.

The paper [261], focused on the identification of Social Media, Audio, and Video applications, particularly when they are encrypted. A ML-based approach was assessed using a comprehensive dataset gathered from four distinct networks and generated via four different off-the-shelf traffic flow exporters. This dataset encompassed a diverse array of services, including Web Browsing, Email, Chat, Streaming, File transfer, VoIP, and P2P. The evaluation of the proposed system primarily centers on its accuracy in classifying the aforementioned applications. Within this investigation, four distinct feature sets derived from various off-the-shelf network traffic flow exporters (Tstat, Tranalyzer, SiLK, and Argus) were employed for the classification task using DT. While some features were common across the four exporters, each computed them differently (e.g., average Round Trip Time). Furthermore, certain features were exclusive to specific exporters (e.g., IP Time To Live change count exists solely within the Tranalyzer feature set).

The paper [262] introduced an improved SVM approach called cost-sensitive SVM (CMSVM) to address accuracy, computational cost, and data imbalance issues. CMSVM incorporates an active learning technique to dynamically assign weights to specific applications. Evaluation using MOORE_SET and NOC_SET datasets for classifying network flows into service groups demonstrated that the proposed solution is more effective than the traditional SVM classifier in terms of accuracy and addressing data imbalance. The paper [263] examined the effectiveness of classifying VPN and non-VPN network traffic using ensemble classifiers, focusing on precision, recall, and F1-score. Their experiments, using the ISCX dataset, revealed that Gradient Boosting (GB) and RF ensemble classifiers outperformed single classifiers like DT, Multi-Layer Perceptron (MLP), and kNN in terms of accuracy. In another work, The paper [264] introduced a ML scheduling framework for prioritizing network traffic in an IoT environment based on QoS requirements. The study compared the performance of seven supervised learning algorithms, including RF, kNN, MLP, NB, LR, and SVM. Using the UNSW dataset with 21 IoT/non-IoT devices, RF achieved the highest accuracy results.

This chapter's primary focus is the application of ML classifiers for identifying interactive VR traffic, a topic not thoroughly explored in prior research. VR traffic comprises both uplink (user tracking) and downlink (video batches) data, and it exhibits a distinct relationship between these two components. The interactive nature of VR traffic —in contrast to video streaming alone— makes it essential to differentiate from non-VR traffic, necessitating high-priority QoS profiles in Wi-Fi networks to ensure optimal performance and user experience.

5.3 Experimental Setup

In this section, we describe the process followed for the generation of the datasets. We also present the definition and selection of the features used for classification, the design of the proposed classifiers and the aspects related to the tuning the hyperparameters of the considered classifiers[1].

5.3.1 Dataset

The dataset that is used in this work consists on two different traffic datasets, one for VR traffic and one for Non-VR traffic. In our case, the first VR traffic dataset is related to a range of VR applications, including SteamVR Home, Half Life: Alyx, Budget Cuts, and a Custom Game developed in Unity called Alteration Hunting, in different configurations involving three distinct frame rates of 60, 90 and 120 frames per second (fps), as well as three different bit rates of 40, 50 and 100 Mbps. The second dataset representing Non-VR traffic focuses on commonly used application types such as non-game videos (i.e., multimedia streaming services like Youtube, Netflix), and online meetings (i.e., Zoom, Google Meet). This section provides an overview of the network environment in which the dataset was generated, outlines the data collection process, and presents the analysis of traffic data traces for both VR and Non-VR traffic.

1. **Network Setup:** the network setup consists of a Desktop computer, AP that supports Wi-Fi 6 [11, 80, 266], a Head Mounted Display Meta Quest 2 and a laptop, as shown in Fig. 5.1. The desktop computer is connected to the AP with an ethernet cable and the clients are connected to the AP via Wi-Fi. The equipment used is also illustrated in Table 5.1.

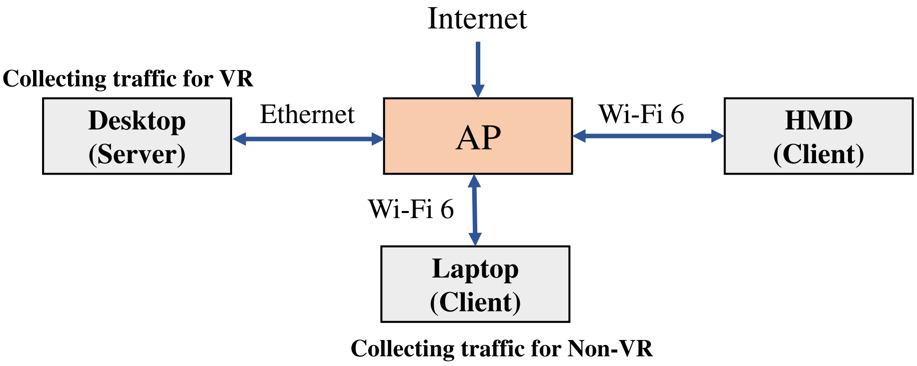


Fig. 5.1. The network setup for data collection.

For VR, we have used two platforms to play games: Steam and Unity. In the Steam platform, the Desktop computer and HMD in Fig. 5.1 are used as a server and a client, respectively for streaming. We install the SteamVR[2] platform, and the Air light VR (ALVR)[3] server on the Desktop in order to allow users to play PC-based VR games on their HMD via a Wi-Fi connection. We also connect HMD as a client using the ALVR application. So, Playing games here is done

[1] The raw packet traces, the input files and Python scripts for both the feature extraction and traffic classification are available on Zenodo [265]. We also reserve some special traces for the final test

[2] https://www.steamvr.com

[3] https://github.com/alvr-org/ALVR

Table 5.1. Equipment.

Desktop	OS	Windows 10 (Unity), Ubuntu 22.04 LTS (Steam)
	CPU	12th Gen Intel Core i5
	GPU	NVIDIA GeForce RTX 3080
	RAM	2 x Kingston 16GB DDR5
Laptop	OS	Ubuntu 22.04 LTS
	CPU	11th Gen Intel Core i7
	RAM	16GB
HMD	Model	Meta Quest 2
AP	Model	RT-AX58U

on the HMD while the Desktop handles rendering and processing, streaming downlink data (i.e. game content like video, audio and haptics) to the HMD and uplink data (i.e. pose tracking, controller input, and performance metrics) from the HMD back to the Desktop for processing. In the Unity platform, the Desktop computer and laptop in Fig.1 are used as a server and a client, respectively for streaming. In order to develop a custom VR game, first the VR game is created and integrated in Unity, enabling VR support for the laptop. Downlink and uplink traffic in Unity is similar to ALVR.

For Non-VR, we just made use of the laptop connected to the AP linked to the internet in the Fig.5.1. For the streaming process of multimedia services, we played the content on the laptop. The streaming operation involves sending requests for video and audio content as uplink data and in response, downloading video content from multimedia servers as downlink data. For the streaming process of online meetings, the process involves transmitting audio and video data as uplink information while receiving incoming video and audio streams as downlink data from the conference servers.

2. **Data Collection:** this section presents an overview of our dataset and the process by which we generated it. We collected VR and Non-VR traffic data in Desktop and laptop, respectively, as shown in Fig.5.1. In our study, we utilized Wireshark, a network protocol analyzer, to gather raw traffic data and to save as .pcap files. The collected traffic data are organized into distinct network flows based on their five-tuple information, comprising {source IP address, source port number, destination IP address, destination port number, and transport protocol (i.e., UDP or TCP)}. During the data collection process, we took measurements to minimize interference from non-targeted traffic by shutting down other applications. Before describing traffic traces, we should explain briefly some used terms:

   - Traces: Traces refer to the recorded sequence of data transmissions or receptions over a network during a specific period. In the context of VR traffic data, traces represent the individual instances of data exchange between devices, such as VR headsets and servers.
   - Batches: Batches are collections of related data items grouped together for processing or transmission. In VR traffic data, batches typically contain multiple frames or fragments of data, often representing a cohesive unit of information, such as a video stream or a sequence of user interactions.

- Frames: Frames are discrete units of data transmitted over a network, typically containing information such as video frames, audio samples, or control signals. In VR traffic data, frames may represent individual frames of a video stream or other multimedia content.
- Fragments: In the context of network communication, frames are sometimes divided into smaller segments known as fragments. When a frame is too large to be transmitted in one piece, it is fragmented into smaller fragments for transmission over the network. These fragments are then reassembled into the original frame at the receiving end. Each fragment may contain portions of multiple packets, depending on the size and structure of the data being transmitted.
- Packets: Packets are the fundamental units of data transmission in network communication. Each packet contains a portion of the data being transmitted, along with additional information such as source and destination addresses, error-checking codes, and sequencing information.
- Bytes: Bytes are the basic units of digital information storage and transmission. In the context of VR traffic data, bytes represent the smallest units of data exchanged between devices over the network.

To provide a more clear understanding of the traffic traces collected within the VR and Non-VR contents, we briefly outline the traffic traces[4] in the following:

- **VR Traffic Traces Description:** for better understanding of the Interactive VR traffic data traces, a plot of SteamVR Home (120fps, 40Mbps) for a 50ms time interval, is presented in the upper plot of Fig. 5.2. Most uplink traces are around 254 bytes, which are related to user tracking data. The other smaller uplink values of traces can be related to the statistics of HMD. On the downlink side, fragments with traces 1490 bytes are common, representing the larger data packets associated with video batches. In the plot, six video batches are depicted, each containing fragments of 1490 bytes. One batch is zoomed in for clarity to highlight packets with this similar size. Besides, within each batch, there is a smaller-size packet, representing the residual packet resulting from frame fragmentation, which is smaller than a full fragment.

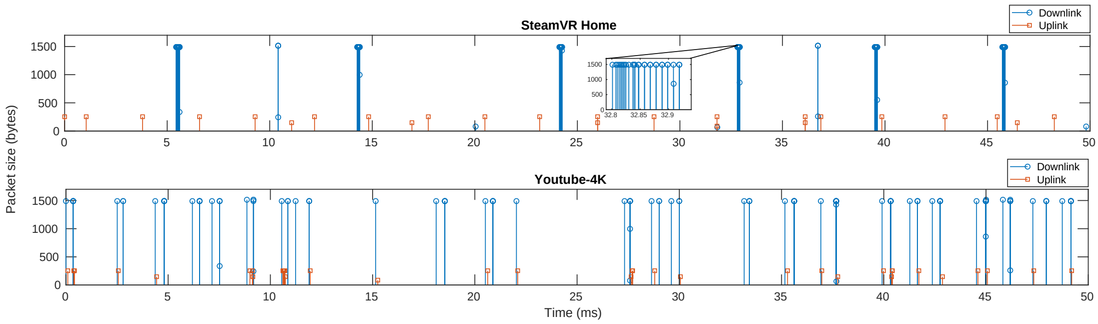


Fig. 5.2. Traffic traces for VR (SteamVR Home) and Non-VR (Youtube-4K) over 50ms. In order to highlight the presence of multiple packets in each batch, zoomed-in view of one VR downlink batch has been illustrated.

[4] An example of Wireshark packet traces for both VR and Non-VR is presented in Appendix. A

- **Non-VR Traffic Traces Description:** A plot of the Non-VR data traces in YouTube (as an example) for a 50ms time interval, is depicted in the lower plot of Fig. 5.2. YouTube primarily streams pre-recorded video content, which is relatively consistent in terms of data size, driven by video streaming requirements and it is not interactive. As shown in the figure, uplink packets are around 80 bytes, representing client video requests, while downlink packets are around 1290 bytes, related to video streaming that is sent to the client.

3. **Feature Engineering:** in this section, feature extraction and feature selection are employed to create features that are useful for the subsequent classification process, as discussed below:

   - **Feature Extraction:** the dataset contains Wireshark traces with information about the exchanged data packets. In the initial step of our analysis, we define some short periods of time. Each set of packets within a period is called a 'sample'. For each sample, first, we separate the downlink and uplink data, and then compute Number of Packets, Total Bytes and statistics such as Minimum, Maximum, Mean, and Standard Deviation for Packet Size and Packet Inter-arrival Time, for both downlink and uplink, separately. In general, VR services follow some common and repetitive patterns in the downlink and uplink directions. Therefore, incorporating features that consider this distinct relationship or correlation between downlink and uplink can contribute to a more robust analysis and accurate identification of VR and Non-VR traffic. Therefore, we also compute three additional features considering this correlation. The first one is the Ratio of Number of Packets (derived from the division of Number of Packets in downlink by Number of Packets in uplink). The Ratio of Total Bytes (calculated from the division of Total Bytes in downlink by Total Bytes in uplink) is considered as a second feature in each sample for further analysis. Furthermore, in each sample we incorporate the cross-correlation among total bytes in a group of downlink and uplink traffic as a third feature. To compute the cross-correlation for each sample, we divide the duration of each sample into a certain number of sub-samples, resulting in τ, according to the next expression (5.1):

$$\tau = \frac{\omega}{N} \ , \tag{5.1}$$

where ω designates the duration of the sample and N denoted to number of sub-samples. Fig. 5.3 illustrates the partitioning of a sample within a duration ω into a designated set of N sub-samples. The total number of

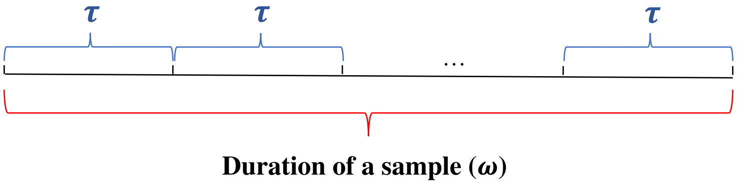


Fig. 5.3. The division of ω into a specified N number of sub-samples.

bytes per sub-sample (ρ_n) can be obtained by the next expression (5.2):

$$\rho_n = \sum_{p=1}^{P} S_p \tag{5.2}$$

where ρ_n denotes total sum of packet sizes S_p (in bytes) considered in the duration of τ, where $n = \{1, 2, \ldots, N\}$ and $p = \{1, 2, \ldots, P\}$ being P the total number of packets in a duration of τ. The expression 5.2 can be used for obtaining ρ_n for both downlink and uplink. These sub-samples can be gathered in two vectors, designated as $D = \{\rho_n\}$ for downlink and $U = \{\rho_n\}$ for uplink. Therefore, the Pearson correlation [267] is calculated according to the expression 5.3:

$$C_{D,U} = \frac{N(\sum_{j=1}^{N} D_j U_j) - (\sum_{j=1}^{N} D_j)(\sum_{j=1}^{N} U_j)}{\sqrt{[N \sum_{j=1}^{N} D_j^2 - (\sum_{j=1}^{N} D_j)^2][N \sum_{j=1}^{N} U_j^2 - (\sum_{j=1}^{N} U_j)^2]}} \tag{5.3}$$

The $C_{D,U}$ is the cross correlation between D and U considered as a feature for each sample. We obtain the correlation values for all samples in each dataset.

Table 5.2 shows the considered symbols for the 23 features extracted for each sample and a brief description of each. It is worth noting that, for the sake of simplicity in the work, we have assigned a symbol to each feature, which can be found in Table 5.2 for reference.

In addition to extracted features, a binary label is also included in the dataset to identify VR from Non-VR (i.e., assigning 1 to VR and 0 to Non-VR).

- **Feature Selection:** This technique is a critical step in SL aimed at identifying the most relevant features for predicting the target variable. There are several feature selection techniques in supervised learning, like “filter” that selects feature subsets based on their relationship with the target variable by statistical methods or feature importance methods, “wrapper” that searches for effective feature subsets [189, 268], “intrinsic” that selects by algorithms like Decision Trees to perform automatic feature selection during training [268] or some novel “hybrid models”, combining them [260]. We use filter selection using a feature importance method, called “permutation importance” that is both model-agnostic and easily comprehensible. Permutation importance stands out as it possesses both of these qualities. It assesses feature importance by quantifying the impact on model error (e.g., Mean Absolute Error (MAE), r-squared, accuracy) when the values of a single feature are permuted, providing an intuitive and straightforward measure [269]. Permutation importance is a technique used to estimate the importance of features in a model by assessing the change in model performance when the feature values are randomly shuffled. The process begins by calculating the baseline performance of the trained model on a validation dataset. For each feature, the values are permuted, breaking the association between the feature and the target variable, and the model's performance is then evaluated on this modified dataset. The importance of a feature is determined by the difference between the baseline performance and the performance on the permuted dataset: a significant drop in performance indicates a highly important feature, while a small change suggests a less important feature. This procedure is repeated for all features. Sometimes, importance values are normalized to sum to 1, but this is not always the case, especially for non-parametric models like kNN, where the importance values may not sum to 1 due to their inherent characteristics and the

Table 5.2. The Description of Symbols for Features Extracted. Ten similar features are considered for downlink and uplink separately, distinguished by 'DL' or 'UL' at the end of their respective symbols.

Feature Symbol	Description (per ω)
NoPDL NoPUL	Number of packets
TBDL TBUL	Total Bytes
MinPSDL MinPSUL	Min Packet Size
MaxPSDL MaxPSUL	Max Packet Size
MeanPSDL MeanPSUL	Mean Packet Size
StdPSDL StdPSUL	Standard Deviation Packet Size
MinPIATDL MinPIATUL	Min Packet inter-arrival time
MaxPIATDL MaxPIATUL	Max Packet inter-arrival time
MeanPIATDL MeanPIATUL	Mean Packet inter-arrival time
StdPIATDL StdPIATUL	Standard Deviation Packet inter-arrival time
RoNoP	Ratio of Number of Packets (DL/UL)
RoTB	Ratio of Total Bytes (DL/UL)
CC	Cross Correlation (i.e. $C_{D,U}$)

nature of the permutation test. This method provides an insightful way to understand feature significance by observing the impact of disrupting each feature's relationship with the target variable. Based on permutation importance feature selection technique, the importance of each feature has been obtained in each specific dataset with a single sample duration, specific correlation sub-sample and for each ML classification algorithm. For most of these algorithms, it is clear that using all the features is crucial to achieve high performance. Nevertheless, there are instances where certain features have been disregarded by the algorithms, such as the DT classifier in all varied settings.

5.3.2 Traffic Classification

Traffic classification methods are essential for managing and optimizing network performance by identifying the types of traffic flowing through the network. In this section, we first present a background of existing traffic classification methods, highlighting the comparison among them. Then, we describe how we apply ML to classify traffic in our dataset.

5.3.2.1 Traffic Classification Background

Different kinds of traffic classification methods can be found in the literature. Traditional methods are based on port-based or payload-based (DPI). On the other hand, other traffic classification methods are based on AI/ML. Table 5.3 compares these three methods according to several aspects including complexity, computational time, handling encrypted traffic, and adaptability. The traditional methods require manual tuning, whereas the ML-based approach works seamlessly after training. Moreover, port-based classification is simple to implement, requires minimal computing resources, and offers high-speed classification. However, it involves accessing the packet's header and examining the utilized port number, providing a simple method with low computational demands. Payload-based (DPI) achieves higher classification accuracy by analyzing the content of packets within the monitored flow. Furthermore, DPI methods are ineffective in classifying encrypted network traffic because encryption obscures the payload, making it unreadable without the decryption keys, which are not accessible due to privacy and security reasons. This issue reduces their utility as many applications utilize encryption. Moreover, DPI's processing speed is slower, particularly when handling aggregated packets. In contrast, ML-based, specifically Supervised Learning (SL)-based classification methods, provide high accuracy by learning from data patterns and moderate computational time, handle encrypted traffic well by analyzing metadata and traffic patterns, and offer fine-grained detection without needing to access the content of network traffic, thus making them highly adaptable to new scenarios without manual intervention. ML-based techniques require moderate complexity for feature extraction and model training [189].

Table 5.3. Comparison of Traffic Classification Methods.

Aspect	Port-based	Payload-based	ML-based
Complexity	Low	High	Moderate
Computational Time	Low	High	Moderate
Handling Encrypted Traffic	Yes	No	Yes
Adaptability	Low	Low	High

5.3.2.2 ML-based Traffic Classification

We particularize the framework for the AI/ML control loop depicted in Fig. 2.3 to focus only on the objective of traffic classification. The framework designed for this specific objective is illustrated in Fig. 5.4. As shown in Fig. 5.4, traffic traces data collected at the AP are sent to data preparation step to preprocess and to extract features. Then, with this data, an offline training is run in order to extract knowledge from the collected data. This offline training is done with certain periodicity in order to keep updated the obtained model according to recent collected measurements. Then, the traffic classification phase to predict label is done online and consists on making use of new measurements collected at the APs together with the binary classification model that were obtained in the training phase. Note that the first sample of a session (flow) is only classified because for instance, once the model identifies that this session corresponds to VR traffic, the prioritization is carried out without needing to classify the remaining samples of the current VR session. The obtained results in the

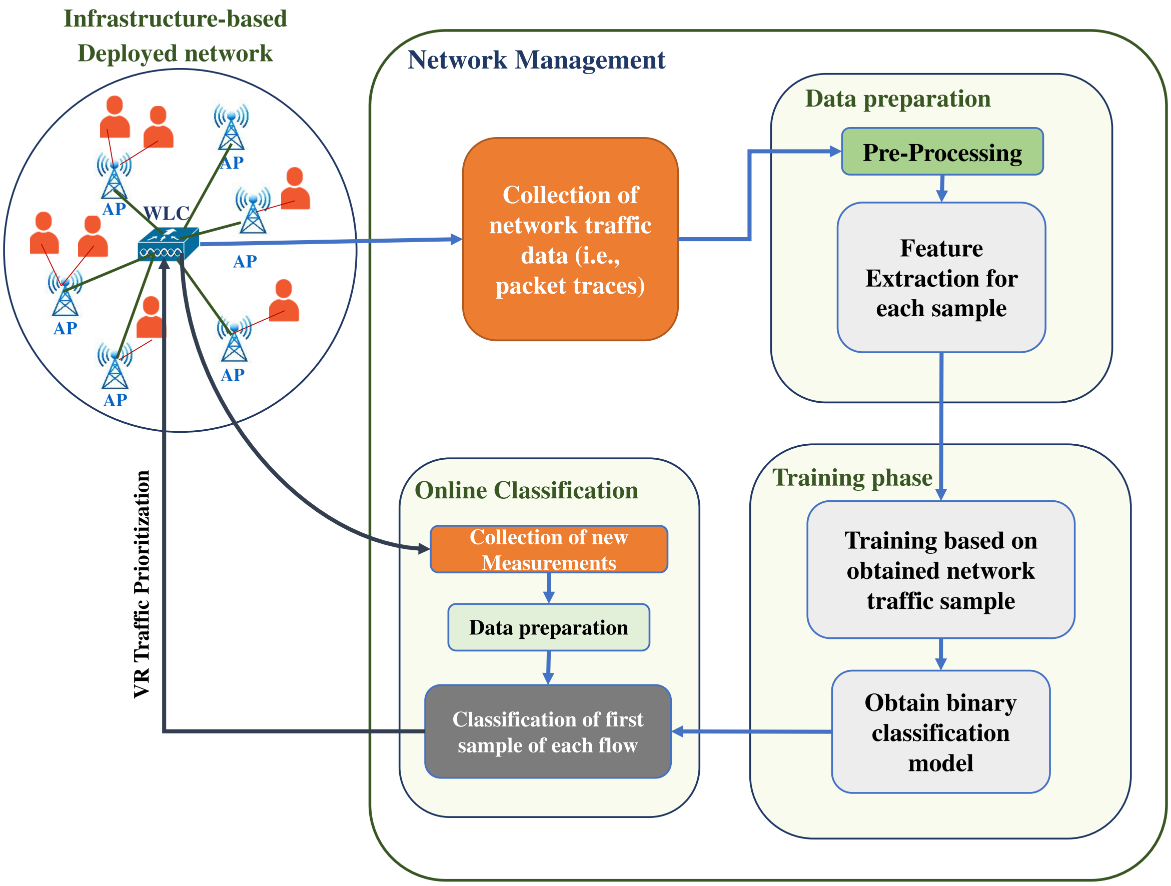

Infrastructure-based
Deployed network
AP
WLC
AP
AP
AP
AP
AP
VR Traffic Prioritization
Network Management
Collection of network traffic data (i.e., packet traces)
Data preparation
Pre-Processing
Feature Extraction for each sample
Training phase
Training based on obtained network traffic sample
Obtain binary classification model
Online Classification
Collection of new Measurements
Data preparation
Classification of first sample of each flow

Fig. 5.4. General framework of traffic classification.

prediction process may be useful prioritization of VR traffic to satisfy the requirements at expenses of a slight degradation in the quality of other kinds of traffic.

For traffic classification, we aim to identify simple classifiers that can be later easily implemented in the AP. As such, six common and distinct ML classifiers, namely LR, SVM, kNN, DT, RF, and NB [187] are selected. A brief background on these classifiers has been described in Chapter 2, Section 2.3. In our case, we apply these supervised learning classifiers for binary classification, assigning labels as either "VR" or "Non-VR" traffic. All classification models are constructed using the scikit-learn [269] ML library. During the classification process, we perform hyperparameter tuning for ML models using GridSearchCV [187, 269] for each classifier. GridSearch is employed to identify the optimal hyperparameters for each classifier. Below, we provide brief descriptions of the common hyperparameters considered for each classifier:

- **LR and SVM:** in both LR and SVM, a key hyperparameter is "C", representing the regularization strength. "C" controls the degree of regularization applied to the model and inversely influences its level of regularization. However, there are specific hyperparameters unique to each algorithm. In LR, we have the "solver" hyperparameter, which dictates the optimization algorithm used during model training. Meanwhile, SVM introduces the "kernel" hyperparameter, determining the kernel function applied to the input data to facilitate the discovery of non-linear decision boundaries. This transformation is vital for SVMs to capture complex relationships within the data.

- **kNN:** in kNN, two essential hyperparameters are "n_neighbors" and "weights." "n_neighbors" determines the number of nearest neighbors that the algorithm considers when making a prediction for a new data point and "weights" defines the weight assigned to each neighbor when making a prediction. The "weights" hyperparameter in kNN can take on one of two common values: "uniform," which assigns equal weight to all neighbors in the prediction, or "distance", which assigns weights inversely proportional to each neighbor's distance from the data point being predicted.

- **DT and RF:** for DT and RF, one of the hyperparameters is "max_depth" that signifies the maximum depth of a tree. When left unspecified, the tree expands until each leaf node contains only one value. Therefore, by reducing this parameter, we can prevent the tree from learning all training samples, thus mitigating the risk of overfitting. In addition, among the other hyperparameters in DT and RF, such as min_samples_leaf, max_leaf_nodes, and min_impurity_decrease, which allow for the development of asymmetric trees and impose constraints at the leaf or node level, we consider the impact of "min_samples_split". In the case of only RF, the hyperparameter "n_estimator" is also taken into account to restrict the number of decision trees in the ensemble.

- **NB:** the "var_smoothing" hyperparameter in NB algorithms, controls the amount of smoothing applied to the variance of numerical features. Smoothing helps prevent issues when calculating probabilities, especially for features that have zero variance in the training data.

5.4 Results

This section first describes the obtained results for the extracted features and the corresponding files generated based on them, which serve as input for the classification process. Second, it presents the results of traffic classification.

5.4.1 Feature Extraction

We define ω with values of 1000ms (1 second), 500ms, 200ms, 100ms, and 50ms. From initial packet traces, for different values of sample duration (ω), we extract features, sample by sample, and store in a single dataset, resulting in a total of five datasets. We consider only samples that contain a set of non-zero packets. Table 5.4 illustrates the count of obtained VR samples for VR, Non-VR samples and total number of samples when the sample duration (ω) takes different values. In the training phase of the classification process, it is essential to strive for a balanced distribution of data across labels. We can observe inconsistencies between the number of VR and Non-VR samples as we consider smaller values of ω. VR applications demand rapid updates and real-time interactions, leading to more frequent transmission of smaller data chunks and therefore, even for low ω, the number of samples with zero packets is low. On the other hand, Non-VR data show more samples with zero packet traces than VR due to their own nature. To address this, we removed the last samples of VR to balance the data distribution across VR and Non-VR labels. This removal process was applied to all dataset except dataset with $\omega = 1sec$, and in the case of $\omega = 500ms$, fewer number of samples were removed. As we approach about $\omega = 50ms$, a larger number of samples needs to be removed to maintain balance.

In order to calculate the CC feature, we define the values of 5, 10, and 20 for N. Therefore, for each specific values of ω and N (referred to as a setting in this study),

Table 5.4. The count of obtained VR samples, Non-VR samples and total number of samples for different values of sample duration (ω). Note that shorter sample duration leads to higher number of samples.

ω	VR Traffic samples	Non-VR Traffic samples	Total samples
1sec	790	841	1631
500ms	1359	1291	2650
200ms	2421	2435	4856
100ms	4176	4237	8413
50ms	7694	7790	15484

we store the obtained features in separate dataset, resulting in a total of 15 datasets including both VR and Non-VR samples for each of dataset. Table 5.5 displays the average CC obtained from VR samples and Non-VR samples for different values of ω and N (i.e. settings).

Table 5.5. Average CC obtained from VR samples and Non-VR samples for different values of sample duration (ω) and number of sub-samples (N) (i.e. different settings).

ω	Type of samples (No. of samples)	Average of CC Values per N		
		5	10	20
1sec	VR (790)	0.96889	0.95294	0.92365
	Non-VR (841)	0.84653	0.80525	0.71961
500ms	VR (1359)	0.96321	0.93112	0.84515
	Non-VR (1291)	0.83223	0.75383	0.66058
200ms	VR (2421)	0.93459	0.86528	0.76849
	Non-VR (2435)	0.78340	0.63742	0.48915
100ms	VR (4176)	0.85400	0.75078	0.55645
	Non-VR (4237)	0.66973	0.50797	0.40811
50ms	VR (7694)	0.74553	0.55422	0.33449
	Non-VR (7790)	0.53264	0.42503	0.33813

In Table 5.5, it is observed that the average correlation of VR traffic is consistently higher than that of Non-VR traffic across all settings with the exception of the $\omega = 50ms$ and $N = 20$, where they exhibit nearly identical levels of correlation. As we decrease the ω, the correlation values tend to decrease. Moreover, as the number of sub-samples used to calculate the correlation increases, the correlation values decrease. As Table 5.5 reveals, within $\omega = 50ms$ and $N = 20$, a narrowing gap exists between the correlations for VR and Non-VR traffic.

5.4.2 Traffic Classification Results

In this process, we evaluate all six classifiers for each of the 15 different input datasets, one per setting. Before the classification process, the dataset is split into 70% for training and 30% for validation. Within the classification phase, hyperparameter tuning and feature selection are conducted as follows:

1. **Hyperparameter Tuning:** The considered hyperparameters by using GridSearchCV framework, which incorporates a robust cross-validation approach with a fixed count of three, are listed as below:

- "LR_parameters": ["solver": ['liblinear', 'saga'], 'C': [0.1, 1]]
- "SVM_parameters": ['kernel': ['rbf','sigmoid'], 'C': [0.1, 1]]
- "kNN_parameters": ['n_neighbors': [5,10], 'weights': ['uniform', 'distance']]
- "DT_parameters": ['min_samples_split': [5,8], 'max_depth': [5,10]]
- "RF_parameters": ['n_estimators': [5,20,50], 'min_samples_split': [5,8], 'max_depth': [5,10]]
- "NB_parameters": ['var_smoothing': np.logspace(0,-9, num=100)]

2. **Permutation Importance Feature Selection:** using this technique, we select the most crucial features to influence the traffic classification process according to the specific ML classification algorithm under consideration. Features with an importance rate of zero are excluded from participation in the classification process.

By incorporating these two processes, we perform classification method. Four commonly used metrics, namely Accuracy, Precision, Recall, and F1-Score, are employed to evaluate the effectiveness of the built classifiers in a supervised classification problem [189, 270]. Accuracy indicates the overall performance of the model by measuring the proportion of correctly classified samples out of the total number of samples in a dataset. It provides a general sense of how well the model is performing across all classes. Accuracy is calculated as the ratio of the sum of true positives (correctly predicted positive cases) and true negatives (correctly predicted negative cases) to the total number of samples, which includes true positives, true negatives, false positives (incorrectly predicted positive cases), and false negatives (incorrectly predicted negative cases). Precision measures the proportion of true positive results among all the positive results predicted by the model. It indicates how many of the predicted positive cases were actually correct. Precision is important when the cost of false positives is high. It is calculated as the ratio of true positives (correctly predicted positive cases) to the sum of true positives and false positives (incorrectly predicted positive cases). Recall, also known as sensitivity or true positive rate, measures the proportion of true positive results among all the actual positive cases in the dataset. It reflects the model's ability to identify all relevant instances and is crucial when the cost of false negatives is high. Recall is calculated as the ratio of true positives to the sum of true positives and false negatives (actual positive cases that were incorrectly predicted as negative). The F1-Score is a single metric that balances both precision and recall by taking their harmonic mean. It is useful when you need to balance the trade-off between precision and recall, especially in situations where the class distribution is imbalanced. The F1-Score provides a comprehensive measure by considering both false positives and false negatives. It is calculated as 2 times the product of precision and recall divided by the sum of precision and recall [189]. In light of this, we present a comparison of the accuracy results of classifiers during the validation phase referred to as the validation score in the study, for each specific ω and N (i.e. each of 15 settings) in Table 5.6. In the context of correlation analysis, as depicted in Table 5.5, a subset of 5 sub-samples exhibited superior correlation outcomes than 10 or 20 sub-samples, while as indicated in Table 5.6, the best validation score pertains to sample duration of 500ms (i.e. $\omega = 500ms$) and the utilization of 20 sub-samples (i.e. $N = 20$) for the correlation analysis across five distinct classifiers, namely SVM, kNN, DT, RF, and NB. The most favorable outcome is attributed to the RF classifier, yielding a validation score of 0.99245. However, when evaluating LR, a slightly better performance is observed with sample duration of 500ms and the 20 number of sub-samples dataset, although the differences in accuracy between these values are minimal. It is important to note that while the average correlation values (as presented in Table 5.5) are higher for the 5 and 10 considered sub-samples compared to the 20 number of sub-sample dataset, the results indicate that classifiers exhibit a better ability to distinguish between VR and Non-VR in the 20 number of sub-sample dataset.

Table 5.6. Validation scores of six classifiers for all considered setting (i.e. different values of ω and N). The highest accuracies are highlighted in bold.

N	ω	LR	SVM	kNN	DT	RF	NB
20	1sec	0.9388	0.9326	0.9306	0.9265	0.9347	0.8939
	500ms	0.9597	**0.9811**	**0.9624**	**0.9874**	**0.9924**	**0.9459**
	200ms	0.9547	0.9547	0.9588	0.9547	0.9554	0.8881
	100ms	0.9616	0.9596	0.9612	0.9592	0.9592	0.8918
	50ms	**0.9621**	0.9578	0.9587	0.9597	0.9632	0.9053
10	1sec	0.9224	0.9326	0.9265	0.9286	0.9306	0.8918
	500ms	0.9572	0.9597	0.9610	0.9572	0.9648	0.9270
	200ms	0.9526	0.9561	0.9554	0.9513	0.9513	0.8964
	100ms	0.9544	0.9564	0.9552	0.9576	0.9584	0.8930
	50ms	0.9537	0.9529	0.9518	0.9507	0.9546	0.9029
5	1sec	0.9286	0.9245	0.9265	0.9245	0.9265	0.9122
	500ms	0.9610	0.9535	0.9459	0.9522	0.9535	0.9157
	200ms	0.9554	0.9574	0.9581	09574	0.9581	0.8888
	100ms	0.9481	0.9485	0.9497	0.9473	0.9509	0.8902
	50ms	0.9546	0.9565	0.9574	0.9582	0.9593	0.8969

In Table 5.7, we provide the validation report for each classifier under the best-performing setting, which corresponds to sample duration of 500ms and number of sub-samples of 20 (i.e. $\omega = 500$ms and $N = 20$). In the support column, indicating the actual occurrences of each class in the specified dataset, it is evident that 795 samples are included in the validation process to assess the training data, comprising 392 for Non-VR and 403 for VR. Regarding the Precision, Recall, and F1-Score metrics for the top three classifiers (kNN, DT, and RF), all values stand at 0.98 and 0.99, with an increase in the precision of VR in the top-performing classifier, RF, which is higher than 0.995 and rounds to 1.00.

Table 5.7. Validation report for the setting of $\omega = 500ms$ and $N = 20$.

Classifier	Type of Traffic	Precision	Recall	F1-Score	Support
LR	Non-VR	0.94	0.98	0.96	392
	VR	0.98	0.94	0.96	403
SVM	Non-VR	0.97	0.99	0.98	392
	VR	0.99	0.98	0.98	403
kNN	Non-VR	0.98	0.98	0.98	392
	VR	0.98	0.99	0.98	403
DT	Non-VR	0.99	0.98	0.99	392
	VR	0.99	0.99	0.99	403
RF	Non-VR	0.99	0.99	0.99	392
	VR	1.00	0.99	0.99	403
NB	Non-VR	0.99	0.90	0.94	392
	VR	0.91	0.99	0.95	403

Table 5.8 displays the confusion matrices depicting true labels and predicted labels for all classifiers in the best-performing setting. In the case of the top-performing classifier, RF, it correctly labels 390 out of 392 Non-VR samples, with only 2 misclassified as VR. For the 403 VR samples, it accurately predicts 399 as true VR and misclassifies 4 of them. Conversely, in the case of NB, which exhibits lower accuracy metrics compared to the others, we observe that this performance is primarily influenced by the misclassification of 39 Non-VR traffic instances as VR.

Table 5.8. Confusion matrices of validation data for the setting of $\omega = 500ms$ and $N = 20$.

	Classifier	Type of Traffic	Predicted Label	
			Non-VR	VR
True Label	LR	Non-VR	384	8
		VR	24	376
	SVM	Non-VR	387	5
		VR	10	393
	kNN	Non-VR	384	8
		VR	6	397
	DT	Non-VR	386	6
		VR	4	399
	RF	Non-VR	390	2
		VR	4	399
	NB	Non-VR	353	39
		VR	4	399

Upon reviewing all the results, we assessed the importance of the features for the top three classifiers in the best-performing setting. This was done to gauge the significance of each feature in the process of permutation importance feature selection. Table 5.9 provides a breakdown of the importance of each feature for each classifier in sequential order. It is worth noting that symbols representing the features are used in this table to simplify and streamline the presentation. In the best-performing setting for these three classifiers, we can observe the order of feature importance that yielded the best results. In the case of kNN, all features are utilized in the classification process, whereas the DT classifier employs only 12 features. Moreover, in DT, the first feature ranked in importance is RoNoP, with a high value (i.e., 0.7751) which can present the significance of the correlation between downlink and uplink to distinguish VR from Non-VR. In RF, all features except for "MinPIATDL" are utilized in the classification process.

Regarding the hyperparameters selected among all those considered during GridSearchCV for the three best classifiers, we observe that setting the following values leads to stabilized accuracy metrics:

- For kNN, configuring n_neighbors as 5 and employing 'distance' as the weighting scheme.
- For DT, setting max_depth to 10 and min_samples_split to 5.
- For RF, setting max_depth as 10, min_samples_split as 8, and employing 50 estimators (n_estimators).

Table 5.9. Importance of the features in three top-performing classifiers for the best-performing setting (i.e. $\omega = 500ms$ and $N = 20$).

No.	Features ranked importance	Importance values for kNN	No.	Features ranked importance	Importance values for DT	No.	Features ranked importance	Importance values for RF
0	NoPDL	0.0930	0	RoNoP	0.7751	0	MaxPSUL	0.2155
1	CC	0.0894	1	MaxPSUL	0.1484	1	MaxPIATDL	0.1380
2	TBDL	0.0809	2	NoPUL	0.0344	2	RoNoP	0.1199
3	NoPUL	0.0445	3	MinPIATUL	0.0137	3	TBDL	0.1107
4	MeanPSDL	0.0323	4	TBDL	0.0064	4	MeanPSUL	0.1000
5	TBUL	0.0216	5	MaxPIATUL	0.0051	5	MeanPIATDL	0.0796
6	MeanPSUL	0.0163	6	StdPIATUL	0.0048	6	TBUL	0.0473
7	MaxPSDL	0.0079	7	MinPSUL	0.0038	7	StdPIATDL	0.0374
8	RoNoP	0.0069	8	MaxPIATDL	0.0024	8	MinPSDL	0.0348
9	MaxPSUL	0.0066	9	CC	0.0021	9	MeanPIATUL	0.0313
10	RoTB	0.0054	10	NoPDL	0.0020	10	RoTB	0.0299
11	StdPSDL	0.0036	11	TBUL	0.0017	11	NoPUL	0.0202
12	MaxPIATDL	0.0036	12	MeanPSDL	0.00003	12	MaxPSDL	0.0177
13	StdPSUL	0.0030	13	StdPIATDL	0.0000	13	NoPDL	0.0040
14	MinPSDL	0.0027	14	MeanPIATDL	0.0000	14	MinPIATUL	0.0037
15	MaxPIATUL	0.0022	15	MinPIATDL	0.0000	15	MinPSUL	0.0033
16	MeanPIATDL	0.0013	16	MeanPSUL	0.0000	16	MaxPIATU	0.0015
17	StdPIATDL	0.0012	17	StdPSUL	0.0000	17	StdPSDL	0.0014
18	StdPIATUL	0.0012	18	StdPSDL	0.0000	18	StdPIATUL	0.0013
19	MinPSUL	0.0009	19	MeanPIATU	0.0000	19	StdPSUL	0.0012
20	MinPIATDL	0.0005	20	MaxPSDL	0.0000	20	MeanPSDL	0.0011
21	MeanPIATUL	0.0002	21	MinPSDL	0.0000	21	CC	0.0002
22	MinPIATUL	0.0002	22	RoTB	0.0000	22	MinPIATDL	0.0000

5.5 Evaluating the Interactive VR Traffic Identification model

5.5.1 Model testing

In this section, we evaluate the classification model using the packet traces from: (i) a multi-user VR session using ALVR, and (ii) a single-user session using Steam Link[5].

First, we test the classification model with packet traces of three users playing VR games in a multi-user experimental setup [244]. As shown in Fig. 5.5, each computer is connected to the AP using an Ethernet cable, and each HMD is connected to the AP over Wi-Fi 6. Each HMD client is connected to a server running on a computer. During the training process, we used single-user traffic traces. The evaluation aims to ensure that the model performs effectively on multi-user data that was not part of training process. To achieve this, we evaluate the model for each user, separately. This process is illustrated in the Fig. 5.5. For each user, in traffic classification, first we apply the feature extraction model on packet arrivals to obtain samples in which the features are computed and then, the classification model to obtain whether the model can correctly predict its type or not. We apply two top-performing classifiers (i.e., DT and RF), for best-performing setting (i.e., ω = 500ms, N = 20). From 128 samples of the first user (i.e., Client A) that was playing a VR game, 127 samples are correctly classified (i.e., VR), and only one is misclassified when applying the RF classifier. Therefore, the accuracy result for the test phase (referred to as test

[5] https://store.steampowered.com/app/353380/Steam_Link/

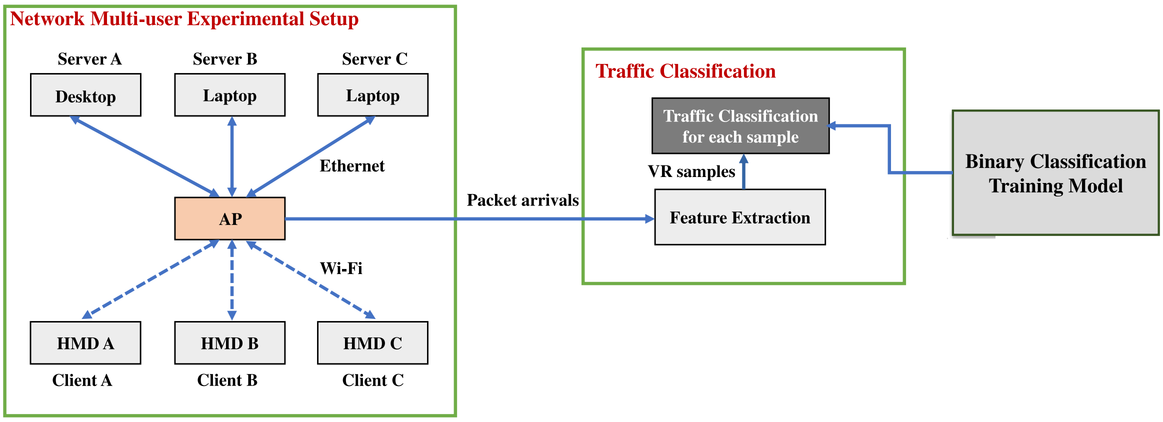


Fig. 5.5. Testing the Interactive VR Traffic Identification Model in a Multi-user Experimental Setup.

score) is: 0.9922, while with DT, 126 samples are predicted in a correct class and 2 misclassified and the test score is: 0.9844. For the Client B and Client C, both DT and RF classifiers achieve consistent test scores of 0.9925 for Client B and 0.9921 for Client C. Only one VR sample is misclassified by both classifiers, with 134 samples for Client B and 127 samples for Client C. Additionally, we assessed the computational time exclusively for Client A. In each sample, it takes approximately 0.7670 seconds for feature extraction and 0.0131 seconds for classification using a DT classifier. The computational time increases slightly to 0.0240 seconds when using a RF classifier. Thus, the total computational time (i.e. both feature extraction and classification phases) remains below 1 second for each sample. Consequently, when a user starts a game, the AP can promptly detect whether the traffic corresponds to VR or not.

Secondly, we test the classification model with packet traces from Steam Link in a single-user setup. Steam Link is the streaming solution of Valve that has been recently made available for VR. This evaluation aims to ensure that the model performs effectively with a source that was not used in the training process. Note that in the training process we only used packet traces from ALVR and Unity Render Streaming. For the traffic of a single user, we first use the feature extraction model on the packet arrivals to obtain samples with computed features. Then, we apply the two top-performing classifiers (i.e., DT and RF), for best-performing setting (i.e., ω = 500ms, N = 20). From 123 samples of the user playing a VR game, 121 samples are correctly classified as VR, with only two misclassified by both the DT and RF classifiers. Therefore, the test phase accuracy (test score) is 0.9837. Furthermore, we evaluated the computational time, too. For each sample, feature extraction takes about 0.8710 seconds, and classification with a DT classifier takes 0.0137 seconds. When using an RF classifier, the classification time increases slightly to 0.0251 seconds. Therefore, the total computational time, including both feature extraction and classification, remains less than 1 second per sample.

5.5.2 Enhancing Wi-Fi QoS through the Prioritization of VR traffic

In this section we use a network simulator to analyze the impact that VR traffic identification can have in current Wi-Fi networks. We employ an extended version of the C++ simulator used in [271], which allows us to simulate Wi-Fi 6 networks reproducing ALVR traces. For this particular simulation we set the position of the nodes, the channels that they use and their bandwidth, as well as their transmission power, which is 20 dBm, a standard. Then we setup the traffic generation, which can be modified to transmit lower volumes (like 50 MBps for instance), and the frame rate is also variable for VR. In addition, we consider a Cloud Edge VR scenario that consists of an AP and two STAs. The following MCS are used: 1024-QAM 5/6 for the first STA, and 256-QAM 5/6 for the second one, both use 80 MHz channels on the 5 GHz band. The first STA is using ALVR to play a VR game at 100 Mbps and 90 fps. The second STA is receiving non-VR background (BG) ON/OFF traffic with a load ranging from 200 to 400 Mbps. The ON/OFF BG traffic alternates the ON and OFF states (exponentially distributed) for 70 ms and 30 ms in average respectively. During the ON period, packets arrive to the AP following a Poisson distribution. To showcase the impact that traffic identification can have in Wi-Fi QoS, we allow the AP to detect whether the traffic is VR or BG, and prioritize the first one accordingly, ignoring all BG packets until all VR packets are transmitted. For best-performing setting (i.e. $\omega = 500$ms and $N = 20$), there are between 280 to 300 packets in each VR sample of 500 ms. After applying the classification model and determining that the first sample of an VR session corresponds to VR traffic, all arriving packets with the same source and destination IP will receive high priority from the AP and will be transmitted. Note that the classifier is activated once enough packets are gathered for a 500ms sample to determine the label of incoming traffic in the sample. To better illustrate the AP prioritization process, a traffic scenario for a few packets is depicted in Fig. 5.6.

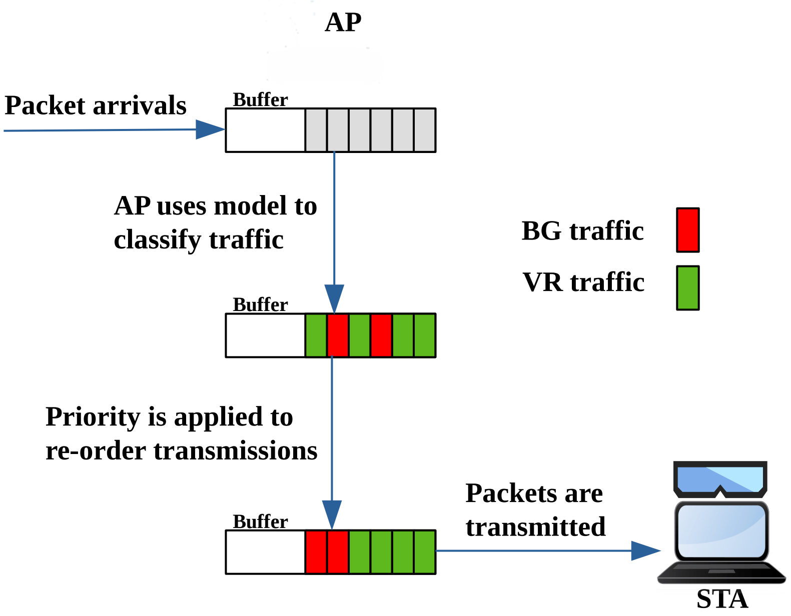


Fig. 5.6. System operation example: VR traffic classification and prioritization.

Fig. 5.7 shows the median and worst-case delay (99th percentile) for both VR traffic and BG traffic with and without VR prioritization active in the downlink. Blue and yellow bars show default First In First Out (FIFO) operation (i.e., no prioritization), and red and purple bars show the delay once VR prioritization is active. The 5 ms and 10 ms delay thresholds have been highlighted with dashed lines. VR requires very low latency, 5 ms or under would lead to the optimal experience, while delays above 10 ms would lead to a loss of video frames and an uneven experience that is noticeable to the player. Unprioritized VR traffic remains under 5 ms of worst-case delay for 200 Mbps, and 10 ms for 300 Mbps, maintaining a stable experience for the end user.

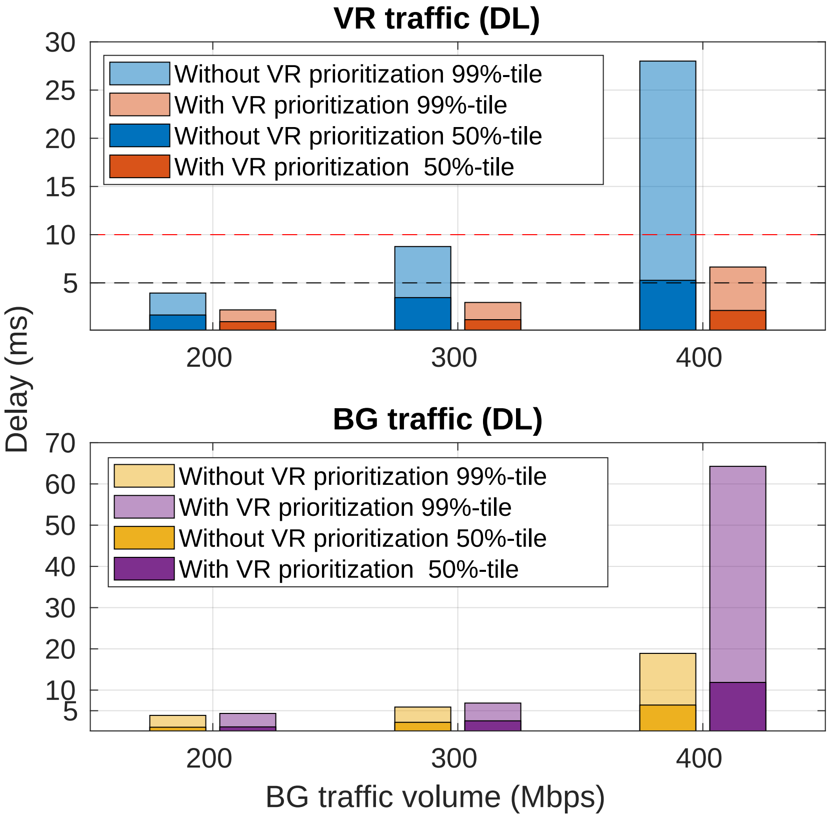


Fig. 5.7. A comparison of traffic packet delay for VR and BG traffic (downlink) in both medium and worse delay scenarios, with and without VR prioritization.

At 400 Mbps however, the delay for VR traffic exceeds 25 ms, more than twice the maximum delay we need. Once VR prioritization is active, VR delay decreases in all cases, and for 400 Mbps we can observe a 76.27% decrease, leading to a delay of less than 10 ms, and a smoother experience for the player. For BG traffic on the other hand, prioritization leads to an increase in the delay. This increase is negligible at lower loads, but for 400 Mbps the delay is more than doubled. Indeed, prioritizing one type of traffic will lead to worse delays for other traffic types. However, BG traffic could be a file download or video streaming through Youtube, which either do not have strict latency requirements (former) or can compensate through the use of buffering (latter), which VR traffic cannot use.

In this section, we have shown that Interactive Traffic Identification can be used to significantly improve the performance of VR traffic. By identifying the types of traffic being driven through an AP, the AP can then make smart decisions that prioritize delay-sensitive traffic at the expense of non-sensitive traffic. With our approach, we have achieved VR traffic delays 4.2x lower than without prioritization, with only a 2.3x higher delay in the BG traffic.

VR flows (i.e., VR sessions) can last several minutes. However, the online traffic classification process is run only for the first sample of a VR flow. The total computation time for both feature extraction and classification of a sample is below one second by using a standard personal computer. A similar value can be obtained with an AP which may have dedicated resources to support ML operations. This value is very small compared to the VR session duration. Therefore, when the first sample of the VR flow is correctly classified as VR traffic, we can assume that the rest of the packets of the VR flow corresponds to VR traffic, without the necessity of running the classification process for the rest of the samples. Then, once the flow is identified as VR traffic, the prioritization process can be done without having an impact in terms of latency. Note that the training time – considering training is done offline and before deploying the model – is not evaluated since it is not relevant for real-time operations.

5.6 Conclusion

Interactive XR/VR traffic identification over Wi-Fi can be helpful to decrease latency for XR users, improving network QoS and user experience. In this chapter, we initially extracted statistical features from network characteristics across 15 settings, employing specific duration of sample (i.e. each of 1sec, 500ms, 200ms, 100ms, and 50ms) and a certain number of sub-samples (5, 10, and 20). We also included three features related to the correlation between downlink and uplink (i.e., RoNoP, RoTB and CC). Specifically, CC was computed based on a number of sub-samples (5, 10 and 20). Subsequently, in the classification phase, six ML methods were compared across all settings to determine the best-performing setting. This phase included feature selection (utilizing permutation importance) and hyperparameter tuning (via GridSearchCV). Evaluation metrics such as Accuracy, Precision, Recall, and F1-Score were employed for assessing the performance of the classification methods. The setting with $\omega = 500$ms and $N = 20$ emerged as the best-performing one, where two top-performing ML methods, DT and RF, achieved accuracies of 0.9874 and 0.9924, respectively.

To evaluate the interactive VR traffic identification model obtained from the training phase, two distinct processes were executed. Firstly, we evaluated the model by using (i) three datasets, each associated with a different user in a multi-user experimental setup, and (ii) a dataset collected when using SteamLink in single-user setup. These datasets were not part of the training dataset. In the multi-user dataset, using the top-performing classifiers (DT and RF) with the best setting ($\omega = 500$ms and $N = 20$), we achieved accuracies above 0.992 for all users, except for one instance where DT accuracy was 0.984. The total computational time for Client A was under 1 second per sample for both classifiers. In the single-user SteamLink dataset, both DT and RF also achieved high accuracy (0.984) with a computational time under 1 second per sample. The results indicate that our model performs well across these three streaming solutions. Therefore, it is likely that the classification model can work with other technologies, regardless of the source, as long as the traffic is encapsulated in transport protocols.

In general, this chapter demonstrates that the ML-based traffic classification method offers a compelling balance between accuracy and computational efficiency. In our testing phase using the DT or RF classifier, we achieved an accuracy higher than 0.984 with a computational time under 1 second per sample. This indicates that the computational time is moderate, making it suitable for long-lasting real-time applications.

Secondly, we utilized a network simulator to investigate the potential impact of VR traffic identification in existing Wi-Fi networks. In a scenario involving an AP and two STAs, the first STA engaged in playing a VR game using ALVR, while the second STA received non-VR background traffic. The AP was configured to detect whether the traffic is VR or BG, and prioritize VR traffic, disregarding BG packets until all VR packets were transmitted. With this prioritization approach, we achieved VR traffic delays 4.2 times lower than without prioritization, accompanied by only a 2.3 times higher delay in the BG traffic.

Chapter 6

Conclusion and Future lines of Work

6.1 Conclusion

The increasing number of Wi-Fi users and the emergence of bandwidth-intensive applications have led to a greater demand for denser AP deployment, resulting in more intricate network management requirements. Additionally, advancements in data monitoring and analytics technologies provide opportunities to glean valuable insights into network behavior, thereby enhancing network management efficiency. Utilizing machine learning techniques offers a promising avenue to tackle these challenges and enhance Wi-Fi network performance. This thesis has proposed a general framework to contribute to enhancing Wi-Fi network management using machine learning techniques, focusing on three core elements: Next user connectivity prediction, Wi-Fi metric prediction (with an emphasis on traffic prediction), and Traffic classification.

Firstly, this thesis introduces a method for predicting future Wi-Fi APs to which users will connect (Chapter 3). This method utilizes supervised learning, leveraging historical user connections to develop a prediction model. Various approaches are delineated based on the historical data used. Generally, the PBTP (Prediction Based on Time-Period Patterns) approach, which predicts based on the most recent user connections, yields the highest prediction accuracies. However, PBDP (Prediction Based on Daily Patterns) or PBWP (Prediction Based on Weekly Patterns) may perform better for users exhibiting daily or weekly behavioral patterns. Notably, a combined approach (JBP) demonstrates superior prediction accuracy compared to individual approaches, with relatively low computation time per user. The impact of training set size on JBP accuracy and computation time is explored, revealing that a larger number of days in the training set enhances prediction accuracy at the expense of increased computation time, particularly for Neural Networks. Additionally, the influence of sliding window size is assessed, highlighting the trade-off between detecting user behavior patterns and ensuring an adequate number of training samples. Results suggest that Neural Network-based prediction achieves higher accuracy than Random Forest but requires increased computational time.

Secondly, we presented a methodology for predicting future values of specific network metrics such as traffic load and transmission failures (Chapter 4). This predictive capability enabled proactive network management actions like load balancing, resource allocation, and congestion control, ultimately enhancing network performance. Our approach leveraged historical measurements and spatial correlations among neighboring APs. We began with a spatial correlation analysis among APs in a given region, followed by either temporal prediction based solely on historical measurements of the target AP or spatio-temporal prediction incorporating data from both the target AP and its highly correlated neighbors. Our prediction methodology relied on Neural Networks and operated in two steps: offline model training using historical data and real-time prediction using newly collected data. We evaluated various DL methods including SimpleRNN, CNN, GRU, LSTM, and Transformer for both temporal and spatio-temporal prediction tasks. Additionally, we proposed hybrid DL algorithms combining CNN for spatial correlation extraction and RNN for temporal correlation exploitation. Evaluation with real Wi-Fi network data from a university campus demonstrated high prediction accuracy, accompanied by relatively small Training Computational Time and Prediction Computational Time. Exploiting spatial correlations yielded higher accuracy in spatio-temporal predictions, albeit with a slight increase in TCT. Evaluation also highlighted improvements in prediction accuracy when considering the spatial domain for APs with highly correlated neighbors. Implementation aspects were discussed, including comparison of ML techniques based on key performance indicators and considerations for scalability and model retraining in real network environments. As a result, the RNN (often LSTM) exhibited better performance than other ML methods in temporal prediction. In spatio-temporal prediction, the CNN-RNN (often CNN-LSTM) showed superior performance compared to other DL methods and even outperformed only temporal prediction. Regarding the combined methodology of temporal and spatio-temporal prediction in a scenario with 100 APs, LSTM (for temporal)+CNN-LSTM (for spatio-temporal) demonstrated slightly better prediction accuracy than LSTM+Single LSTM and only temporal prediction with LSTM. Specifically, incorporating spatial correlations among neighboring APs notably enhanced prediction accuracy in spatio-temporal predictions compared to temporal predictions alone, albeit with a slight increase in TCT. Nonetheless, PCT values for both temporal and spatio-temporal prediction NN models remained relatively small.

Finally, a ML-based approach was presented for identifying interactive VR traffic within a Cloud-Edge VR environment, aimed at mitigating latency for XR users and enhancing network QoS and user experience by means of VR traffic prioritization (Chapter 5). Initially, statistical features were extracted from network characteristics across various settings, varying sample duration (ranging from 1 second to 50 milliseconds) and sub-sample sizes (5, 10, and 20). Additionally, three features pertaining to the correlation between Downlink and Uplink were incorporated. Subsequently, in the classification phase, six ML methods were compared across all settings to determine the best-performing setting. The setting with a sample duration (ω) of 500 milliseconds and a number of sub-samples of $N = 20$ emerged as the most effective, with DT and RF achieving high accuracies. The interactive VR traffic identification model was evaluated using three-user VR traffic traces from a multi-user VR scenario and single-user traces from a VR framework not included in the training data, yielding consistently high accuracies for all users. Moreover, the Computational cost analysis revealed processing times under 1 second for each sample for both classifiers. Furthermore, a network simulator was employed to evaluate the impact of VR traffic identification in existing Wi-Fi networks, demonstrating a significant reduction in VR traffic delays with prioritization compared to scenarios without prioritization, while incurring only a modest increase in delay for background (BG) traffic.

6.2 Topics for Further Research

Our research contributions introduce new possibilities for future investigations. Some of the possible interesting directions for future research on the issues that our work has not yet covered are discussed as follows.

- One promising avenue for further research lies in the prediction of various aspects of user behavior and network activity. This includes forecasting user traffic demands, understanding user interests to tailor network services, and predicting user traffic activity patterns, such as when a user will be active or inactive on the network. By delving into these areas, user needs are anticipated and the network performance is developed.

- For the case of next AP prediction (see chapter 3) different approaches have been considered to collect the user patterns at different time (hourly, daily of weekly) scales, leading to PBTP, PBDP, PBWP approaches, respectively. However, for the case of traffic prediction (Chapter 4), an approach similar to the PBTP has been considered, in which the prediction is based on the recent values of AP traffic in the previous time periods. An extension of this traffic prediction methodology to extract also daily and weekly patterns, or even a joint approach similar to JBP presented in Chapter 3, would be another future line of work.

- In our real university dataset scenario, accommodating new circumstances may involve adjusting the model's parameters or conducting a complete retraining. For instance, in the context of a University Campus, changes in classroom setups or the introduction of new academic programs necessitate modifications to the predictive model. This adaptation process entails fine-tuning the model's parameters or conducting a training from scratch to better align with the evolving circumstances. While the current model remains valid throughout the available dataset period, future retraining might be necessary. Assessing the frequency of retraining typically requires access to additional data, which is highlighted as a component of future research endeavors.

- Another potential area for future research in our university's real dataset pertains to energy saving efforts. This involves identifying periods of low or zero traffic in specific APs, allowing them to be switched off to save energy. This can be done by means of ML prediction/classification methods. Additionally, it is imperative to verify that the network continues to work effectively during periods where APs are switched off. In particular, it is important to guarantee that UEs entering in the coverage area of a switched-off AP should have the possibility to connect to neighbour APs or other wireless networks (e.g. cellular networks).

- Finally, in Chapter 5, considering the significance of network traffic classification, particularly in identifying VR traffic, the future line of work involves expanding the classification scope to multi-classification scenarios. While the current work focuses on binary classification for VR traffic detection, extending the classification framework to encompass multiple classes of network traffic would enhance its applicability and utility. This extension could involve the development of novel ML models tailored for multi-class classification tasks, leveraging insights gained from the existing binary classification approach. Additionally, exploring techniques to address challenges specific to multi-class classification, such as class imbalance and overlapping feature distributions, would be essential. By broadening the classification scope to encompass various types of network traffic beyond VR, the proposed model's versatility and effectiveness in network management and optimization can be further augmented. On the other hand, in the context of VR services within Wi-Fi networks, in order to consider additional network effects, such as delays and packet losses, which can introduce variability in the received data stream at the AP and impact the reliability of VR applications. Investigating novel Wi-Fi mechanisms tailored specifically for VR traffic is another crucial area of focus, as these mechanisms have the potential to significantly enhance the overall performance of VR experiences. This may involve making adjustments to existing Wi-Fi standards to better accommodate the unique demands of VR applications and ensure optimal user experiences.

6.3 List of Publications

The three objectives of the thesis have been published and/or submitted for publication in journals and conference. In the context of second journal paper, after the first round of review, the editor and reviewers request to carry out ”moderate” (not major) changes. These activities are listed in the following:

6.3.1 Journals

- **Seyedeh Soheila Shaabanzadeh** and Juan Sánchez-González, "A spatio-temporal prediction methodology based on deep learning and real Wi-Fi measurements", *Computer Networks*, Elsevier, June 2024, ISSN 1389-1286, `https://doi.org/10.1016/j.comnet.2024.110569`.

- **Seyedeh Soheila Shaabanzadeh**, Marc Carrascosa-Zamacois, Juan Sánchez-González, Costas Michaelides, and Boris Bellalta. "Virtual Reality Traffic Prioritization for Wi-Fi Quality of Service Improvement using Machine Learning Classification Techniques". In *Journal of Network and Computer Applications, Elsevier* (Under Review).

6.3.2 Conference

- **Seyedeh Soheila Shaabanzadeh** and Juan Sánchez-González. "On the prediction of future user connections based on historical records in wireless networks". In: *Artificial Intelligence Applications and Innovations. AIAI 2020 IFIP WG 12.5 International Workshops: MHDW 2020 and 5G-PINE 2020*, Neos Marmaras, Greece, June 5–7, 2020, Proceedings 16. Springer. 2020, pp. 84–94.

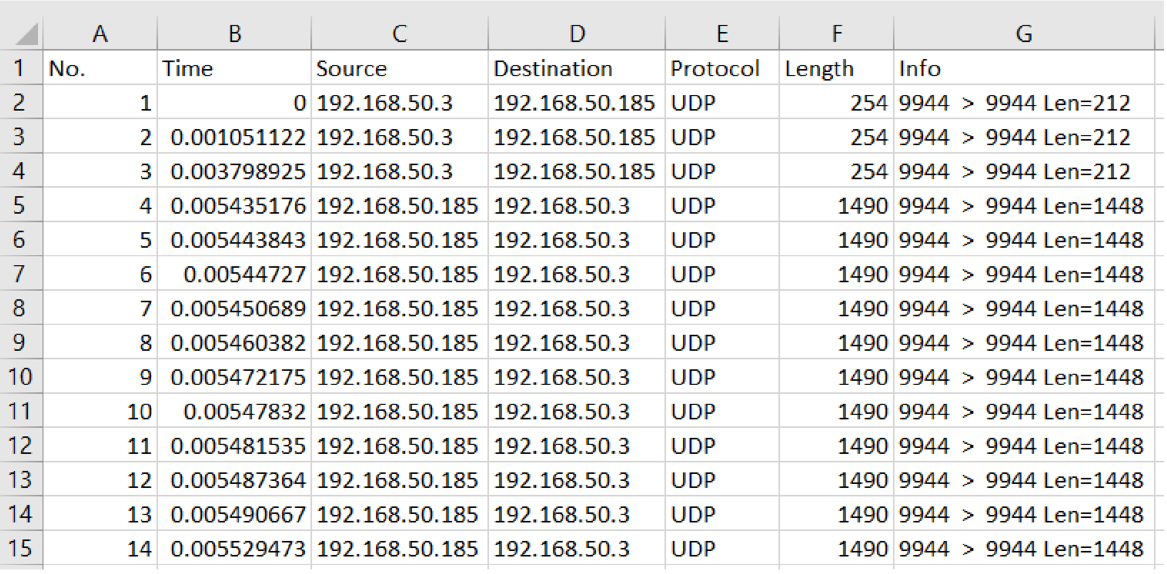
6.4 Collaboration

I have been doing a nine-month research stay at Universitat Pompeu Fabra (UPF), and I collaborated with the "Wireless Networking research group" in "Department of Engineering". Our research carried out is related to the applicability of machine learning classification techniques to improve Wi-Fi QoS through the prioritization of Virtual Reality traffic.

Appendix A

Traces Samples for Traffic Classification

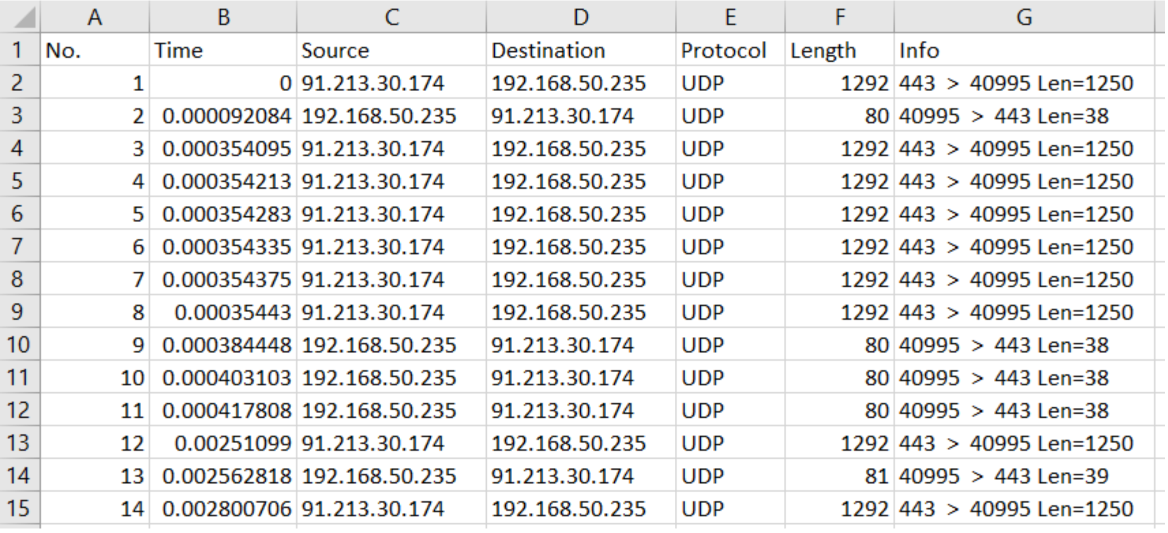
As mentioned, packet traces were captured using Wireshark. In the following, we present samples of raw traffic traces for SteamVR Home (120fps, 40Mbps) representing VR (A.1) and Youtube-4K representing Non-VR (A.2), both captured by Wireshark. Note that the plots for these traces have been depicted in 5.2 over a period of 50ms.



	A	B	C	D	E	F	G
1	No.	Time	Source	Destination	Protocol	Length	Info
2	1	0	192.168.50.3	192.168.50.185	UDP	254	9944 > 9944 Len=212
3	2	0.001051122	192.168.50.3	192.168.50.185	UDP	254	9944 > 9944 Len=212
4	3	0.003798925	192.168.50.3	192.168.50.185	UDP	254	9944 > 9944 Len=212
5	4	0.005435176	192.168.50.185	192.168.50.3	UDP	1490	9944 > 9944 Len=1448
6	5	0.005443843	192.168.50.185	192.168.50.3	UDP	1490	9944 > 9944 Len=1448
7	6	0.00544727	192.168.50.185	192.168.50.3	UDP	1490	9944 > 9944 Len=1448
8	7	0.005450689	192.168.50.185	192.168.50.3	UDP	1490	9944 > 9944 Len=1448
9	8	0.005460382	192.168.50.185	192.168.50.3	UDP	1490	9944 > 9944 Len=1448
10	9	0.005472175	192.168.50.185	192.168.50.3	UDP	1490	9944 > 9944 Len=1448
11	10	0.00547832	192.168.50.185	192.168.50.3	UDP	1490	9944 > 9944 Len=1448
12	11	0.005481535	192.168.50.185	192.168.50.3	UDP	1490	9944 > 9944 Len=1448
13	12	0.005487364	192.168.50.185	192.168.50.3	UDP	1490	9944 > 9944 Len=1448
14	13	0.005490667	192.168.50.185	192.168.50.3	UDP	1490	9944 > 9944 Len=1448
15	14	0.005529473	192.168.50.185	192.168.50.3	UDP	1490	9944 > 9944 Len=1448

Fig. A.1. Some samples of traffic traces for VR (SteamVR Home) captured by Wireshark

Firstly, features are extracted from these raw packet traces using a script of 'Feature Extraction'. To distinguish between DL and UL and extract features specific to each, as well as three features computed from both DL and UL (i.e. RoNoP, RoTB, CC), we utilize the Client IP addresses, namely '192.168.50.3' for VR and '192.168.50.235' for Non-VR, as illustrated in A.1 and A.2. The 'Feature Extraction' script generates CSV output files, which serve as inputs for the 'Classification Model' script. Secondly, the 'Classification Model' script is utilized to conduct binary classification on the inputs for training and testing obtained from 'Feature Extraction' phase. The raw packet traces files, the input files and Python scripts for both the feature extraction and traffic classification are available on Zenodo [265].



	A	B	C	D	E	F	G
1	No.	Time	Source	Destination	Protocol	Length	Info
2	1	0	91.213.30.174	192.168.50.235	UDP	1292	443 > 40995 Len=1250
3	2	0.000092084	192.168.50.235	91.213.30.174	UDP	80	40995 > 443 Len=38
4	3	0.000354095	91.213.30.174	192.168.50.235	UDP	1292	443 > 40995 Len=1250
5	4	0.000354213	91.213.30.174	192.168.50.235	UDP	1292	443 > 40995 Len=1250
6	5	0.000354283	91.213.30.174	192.168.50.235	UDP	1292	443 > 40995 Len=1250
7	6	0.000354335	91.213.30.174	192.168.50.235	UDP	1292	443 > 40995 Len=1250
8	7	0.000354375	91.213.30.174	192.168.50.235	UDP	1292	443 > 40995 Len=1250
9	8	0.00035443	91.213.30.174	192.168.50.235	UDP	1292	443 > 40995 Len=1250
10	9	0.000384448	192.168.50.235	91.213.30.174	UDP	80	40995 > 443 Len=38
11	10	0.000403103	192.168.50.235	91.213.30.174	UDP	80	40995 > 443 Len=38
12	11	0.000417808	192.168.50.235	91.213.30.174	UDP	80	40995 > 443 Len=38
13	12	0.00251099	91.213.30.174	192.168.50.235	UDP	1292	443 > 40995 Len=1250
14	13	0.002562818	192.168.50.235	91.213.30.174	UDP	81	40995 > 443 Len=39
15	14	0.002800706	91.213.30.174	192.168.50.235	UDP	1292	443 > 40995 Len=1250

Fig. A.2. Some samples of traffic traces for Non-VR (Youtube-4K) captured by Wireshark

References

[1] "IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks–Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications". In: *IEEE Std 802.11-2020 (Revision of IEEE Std 802.11-2016)* (2021), pp. 1–4379. DOI: 10.1109/IEEESTD.2021.9363693.

[2] Cisco. "Cisco annual internet report (2018–2023) white paper". In: *Cisco: San Jose, CA, USA* 10.1 (2020), pp. 1–35.

[3] International Data Corporation (IDC). *Worldwide Wi-Fi Technology Forecast, 2023-2027*. Tech. rep. San Mateo, CA, USA: International Data Corporation (IDC), 2023.

[4] Wi-Fi Alliance and Telecom Advisory Services. *Global Economic Value of Wi-Fi 2021-2025*. Tech. rep. Austin, Texas, USA: Wi-Fi Alliance, 2021.

[5] Lorenzo Galati Giordano et al. "What Will Wi-Fi 8 Be? A Primer on IEEE 802.11 bn Ultra High Reliability". In: *arXiv preprint arXiv:2303.10442* (2023).

[6] IEEE 802 LMSC. *IEEE 802 LAN/MAN Standards Committee (LMSC)*. Accessed 7.19.23. 2023. URL: https://www.ieee802.org/.

[7] IEEE 802.11. *IEEE 802.11, The Working Group Setting the Standards for Wireless LANs*. Accessed 7.19.23. 2023. URL: https://www.ieee802.org/11/.

[8] Wi-Fi Alliance. *The worldwide network of companies that brings you Wi-Fi®*. Accessed: 7.19.23. 2023. URL: https://www.wi-fi.org/.

[9] Oughton, Edward and Geraci, Giovanni and Polese, Michele and Shah, Vijay and Bubley, Dean and Blue, Scott. "Prospective Evaluation of Wireless Broadband in the Peak Smartphone Era: 6G versus Wi-Fi 7 and 8". In: *Available at SSRN 4632857* (2023).

[10] "IEEE Standard for Information technology–Telecommunications and information exchange between systems—Local and metropolitan area networks–Specific requirements–Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications–Amendment 4: Enhancements for Very High Throughput for Operation in Bands below 6 GHz." In: *IEEE Std 802.11ac(TM)-2013 (Amendment to IEEE Std 802.11-2012, as amended by IEEE Std 802.11ae-2012, IEEE Std 802.11aa-2012, and IEEE Std 802.11ad-2012)* (2013), pp. 1–425. DOI: 10.1109/IEEESTD.2013.7797535.

[11] "IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks–Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 1: Enhancements for High-Efficiency WLAN". In: *IEEE Std 802.11ax-2021 (Amendment to IEEE Std 802.11-2020)* (2021), pp. 1–767. DOI: 10.1109/IEEESTD.2021.9442429.

[12] "IEEE Draft Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 8: Enhancements for Extremely High Throughput (EHT)". In: *IEEE P802.11be/D4.0, July 2023* (2023), pp. 1–1031.

[13] Garcia-Rodriguez, Adrian and López-Pérez, David and Galati-Giordano, Lorenzo and Geraci, Giovanni. "IEEE 802.11 be: Wi-Fi 7 strikes back". In: *IEEE Communications Magazine* 59.4 (2021), pp. 102–108.

[14] Chen, Hui and Sarieddeen, Hadi and Ballal, Tarig and Wymeersch, Henk and Alouini, Mohamed-Slim and Al-Naffouri, Tareq Y. "A tutorial on terahertz-band localization for 6G communication systems". In: *IEEE Communications Surveys & Tutorials* 24.3 (2022), pp. 1780–1815.

[15] Au, Edward. "New Standards Initiatives on Wi-Fi [Standards]". In: *IEEE Vehicular Technology Magazine* 18.2 (2023), pp. 122–123. DOI: 10.1109/MVT.2023.3254123.

[16] Chen, Cheng and Chen, Xiaogang and Das, Dibakar and Akhmetov, Dmitry and Cordeiro, Carlos. "Overview and performance evaluation of Wi-Fi 7". In: *IEEE Communications Standards Magazine* 6.2 (2022), pp. 12–18.

[17] Korolev, Nikolay and Levitsky, Ilya and Startsev, Ivan and Bellalta, Boris and Khorov, Evgeny. "Study of multi-link channel access without simultaneous transmit and receive in IEEE 802.11 be networks". In: *IEEE Access* 10 (2022), pp. 126339–126351.

[18] Carrascosa-Zamacois, Marc and Geraci, Giovanni and Knightly, Edward and Bellalta, Boris. "Wi-Fi Multi-Link Operation: An Experimental Study of Latency and Throughput". In: *IEEE/ACM Transactions on Networking* (2023).

[19] Bellalta, Boris and Carrascosa, Marc and Galati-Giordano, Lorenzo and Geraci, Giovanni. "Delay Analysis of IEEE 802.11 be multi-link operation under finite load". In: *IEEE Wireless Communications Letters* 12.4 (2023), pp. 595–599.

[20] Cavalcanti, Dave and Cordeiro, Carlos and Smith, Malcolm and Regev, Alon. "WiFi TSN: Enabling Deterministic Wireless Connectivity over 802.11". In: *IEEE Communications Standards Magazine* 6.4 (2022), pp. 22–29.

[21] IEEE Standards Association and others. "P802. 11bn". In: *Institute of Electrical and Electronics Engineers (IEEE) Standards Association, Piscataway, NJ, USA* (2023).

[22] Reshef, Ehud and Cordeiro, Carlos. "Future directions for Wi-Fi 8 and beyond". In: *IEEE Communications Magazine* 60.10 (2022), pp. 50–55.

[23] James F Kurose and Keith W Ross. *Computer Networking: a top-down approach: 8th edition.* Pearson, 2020.

[24] Haris Gacanin and Amir Ligata. "Wi-Fi self-organizing networks: challenges and use cases". In: *IEEE Communications Magazine* 55.7 (2017), pp. 158–164.

[25] Mirza Golam Kibria et al. "Big data analytics, machine learning, and artificial intelligence in next-generation wireless networks". In: *IEEE access* 6 (2018), pp. 32328–32338.

[26] I Chih-Lin et al. "On big data analytics for greener and softer RAN". In: *IEEE Access* 3 (2015), pp. 3068–3075.

[27] Robert Nisbet, John Elder, and Gary D Miner. *Handbook of statistical analysis and data mining applications*. Academic press, 2009.

[28] Jingjing Wang et al. "Thirty years of machine learning: The road to Pareto-optimal wireless networks". In: *IEEE Communications Surveys & Tutorials* 22.3 (2020), pp. 1472–1514.

[29] IEEE 802.11 AIML Topic Interest Group. *Artificial Intelligence Machine Learning Topic Interest Group's Proposed Technical Report Text for AIML Model Sharing Use Case*. Accessed 9.29.23. 2023. URL: `https://mentor.ieee.org/802.11/documents?is_dcn=50&is_group=aiml&is_year=2023`.

[30] Yaohua Sun et al. "Application of machine learning in wireless networks: Key techniques and open issues". In: *IEEE Communications Surveys & Tutorials* 21.4 (2019), pp. 3072–3108.

[31] Merima Kulin et al. "A survey on machine learning-based performance improvement of wireless networks: PHY, MAC and network layer". In: *Electronics* 10.3 (2021), p. 318.

[32] Nguyen Cong Luong et al. "Applications of deep reinforcement learning in communications and networking: A survey". In: *IEEE Communications Surveys & Tutorials* 21.4 (2019), pp. 3133–3174.

[33] Szymon Szott et al. "Wi-Fi meets ML: A survey on improving IEEE 802.11 performance with machine learning". In: *IEEE Communications Surveys & Tutorials* 24.3 (2022), pp. 1843–1893.

[34] Muhammad Asif Khan et al. "ML-Based Handover Prediction and AP Selection in Cognitive Wi-Fi Networks". In: *Journal of Network and Systems Management* 30.4 (2022), p. 72.

[35] Raja Karmakar, Samiran Chattopadhyay, and Sandip Chakraborty. "Dynamic link adaptation in IEEE 802.11 ac: A distributed learning based approach". In: *2016 IEEE 41st Conference on Local Computer Networks (LCN)*. IEEE. 2016, pp. 87–94.

[36] Perahia, Eldad and Gong, Michelle X. "Gigabit wireless LANs: an overview of IEEE 802.11 ac and 802.11 ad". In: *ACM SIGMOBILE mobile computing and communications review* 15.3 (2011), pp. 23–33.

[37] Raja Karmakar, Samiran Chattopadhyay, and Sandip Chakraborty. "Impact of IEEE 802.11 n/ac PHY/MAC high throughput enhancements on transport and application protocols—A survey". In: *IEEE Communications Surveys & Tutorials* 19.4 (2017), pp. 2050–2091.

[38] Rashid Ali et al. "Deep reinforcement learning paradigm for performance optimization of channel observation–based MAC protocols in dense WLANs". In: *IEEE Access* 7 (2018), pp. 3500–3511.

[39] Abhishek Kumar et al. "Adaptive contention window design using deep Q-learning". In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2021, pp. 4950–4954.

[40] Pingan Zhu et al. "Achieving quality of service with adaptation-based programming for medium access protocols". In: *2012 IEEE Global Communications Conference (GLOBECOM)*. IEEE. 2012, pp. 1932–1937.

[41] Amir Hossein Yazdani Abyaneh, Mohammed Hirzallah, and Marwan Krunz. "Intelligent-CW: AI-based framework for controlling contention window in WLANs". In: *2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*. IEEE. 2019, pp. 1–10.

[42] Yalda Edalat and Katia Obraczka. "Dynamically Tuning IEEE 802.11's Contention Window Using Machine Learning". In: *Proceedings of the 22nd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. 2019, pp. 19–26.

[43] Lyutianyang Zhang et al. "Enhancing WiFi multiple access performance with federated deep reinforcement learning". In: *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*. IEEE. 2020, pp. 1–6.

[44] Witold Wydmański and Szymon Szott. "Contention window optimization in IEEE 802.11 ax networks with deep reinforcement learning". In: *2021 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE. 2021, pp. 1–6.

[45] SaiDhiraj Amuru et al. "To send or not to send-learning MAC contention". In: *2015 IEEE Global Communications Conference (GLOBECOM)*. IEEE. 2015, pp. 1–6.

[46] Estefanía Coronado, José Villalón, and Antonio Garrido. "Improvements to multimedia content delivery over IEEE 802.11 networks". In: *NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium*. IEEE. 2020, pp. 1–6.

[47] Rashid Ali et al. "Performance optimization of QoS-supported dense WLANs using machine-learning-enabled enhanced distributed channel access (MEDCA) mechanism". In: *Neural Computing and Applications* 32 (2020), pp. 13107–13115.

[48] Chang Kyu Lee and Seung Hyong Rhee. "Collision avoidance in IEEE 802.11 DCF using a reinforcement learning method". In: *2020 International conference on information and communication technology convergence (ICTC)*. IEEE. 2020, pp. 898–901.

[49] Yuto Kihira et al. "Adversarial reinforcement learning-based robust access point coordination against uncoordinated interference". In: *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*. IEEE. 2020, pp. 1–5.

[50] Oscar Punal, Hanzhi Zhang, and James Gross. "RFRA: random forests rate adaptation for vehicular networks". In: *2013 IEEE 14th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*. IEEE. 2013, pp. 1–10.

[51] Chi-Yu Li et al. "Practical machine learning-based rate adaptation solution for Wi-Fi NICs: IEEE 802.11 ac as a case study". In: *IEEE Transactions on Vehicular Technology* 69.9 (2020), pp. 10264–10277.

[52] Tarun Joshi et al. “SARA: Stochastic automata rate adaptation for IEEE 802.11 networks”. In: *IEEE Transactions on Parallel and Distributed Systems* 19.11 (2008), pp. 1579–1590.

[53] Chiapin Wang. “Dynamic ARF for throughput improvement in 802.11 WLAN via a machine-learning approach”. In: *Journal of Network and Computer Applications* 36.2 (2013), pp. 667–676.

[54] Soohyun Cho. “Reinforcement learning for rate adaptation in csma/ca wireless networks”. In: *Advances in Computer Science and Ubiquitous Computing: CSA-CUTE 2019*. Springer. 2021, pp. 175–181.

[55] Ernest Kurniawan et al. “Machine learning-based channel-type identification for IEEE 802.11 ac link adaptation”. In: *2018 24th Asia-Pacific Conference on Communications (APCC)*. IEEE. 2018, pp. 51–56.

[56] Raja Karmakar et al. “S2-GI: Intelligent selection of guard interval in high throughput WLANs”. In: *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. IEEE. 2020, pp. 1–7.

[57] Raja Karmakar, Samiran Chattopadhyay, and Sandip Chakraborty. “A deep probabilistic control machinery for auto-configuration of WiFi link parameters”. In: *IEEE Transactions on Wireless Communications* 19.12 (2020), pp. 8330–8340.

[58] Pochiang Lin and Tsungnan Lin. “Machine-learning-based adaptive approach for frame-size optimization in wireless LAN environments”. In: *IEEE transactions on vehicular technology* 58.9 (2009), pp. 5060–5073.

[59] Estefanía Coronado, Abin Thomas, and Roberto Riggio. “Adaptive ML-based frame length optimisation in enterprise SD-WLANs”. In: *Journal of Network and Systems Management* 28.4 (2020), pp. 850–881.

[60] Estefanía Coronado et al. “aios: An intelligence layer for sd-wlans”. In: *NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium*. IEEE. 2020, pp. 1–9.

[61] Raja Karmakar. “Online learning-based energy-efficient frame aggregation in high throughput WLANs”. In: *IEEE Communications Letters* 23.4 (2019), pp. 712–715.

[62] Shervin Khastoo, Tim Brecht, and Ali Abedi. “Neura: Using neural networks to improve wifi rate adaptation”. In: *Proceedings of the 23rd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. 2020, pp. 161–170.

[63] Anand Kashyap, Utpal Paul, and Samir R Das. “Deconstructing interference relations in WiFi networks”. In: *2010 7th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*. IEEE. 2010, pp. 1–9.

[64] Suchul Lee and Chong-Kwon Kim. “D-Fi: A diversity-aware Wi-Fi using an OFDM-based Bloom filter”. In: *2012 20th IEEE International Conference on Network Protocols (ICNP)*. IEEE. 2012, pp. 1–10.

[65] Suchul Lee et al. “Frequency diversity-aware Wi-Fi using OFDM-based Bloom filters”. In: *IEEE Transactions on Mobile Computing* 14.3 (2014), pp. 525–537.

[66] Chiranjib Saha and Harpreet S Dhillon. “Interference characterization in wireless networks: A determinantal learning approach”. In: *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE. 2019, pp. 1–6.

[67] Ebtesam Almazrouei et al. “A deep learning approach to radio signal denoising”. In: *2019 IEEE Wireless Communications and Networking Conference Workshop (WCNCW)*. IEEE. 2019, pp. 1–8.

[68] Yonghwi Kim, Sanghyun An, and Jungmin So. “Identifying signal source using channel state information in wireless LANs”. In: *2018 International Conference on Information Networking (ICOIN)*. IEEE. 2018, pp. 616–621.

[69] J Dinal Herath, Anand Seetharam, and Arti Ramesh. “A deep learning model for wireless channel quality prediction”. In: *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE. 2019, pp. 1–6.

[70] Julien Herzen, Henrik Lundgren, and Nidhi Hegde. “Learning wi-fi performance”. In: *2015 12th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE. 2015, pp. 118–126.

[71] Se-Young Yun, Yung Yi, Jinwoo Shin, et al. “Optimal CSMA: a survey”. In: *2012 IEEE international conference on communication systems (ICCS)*. IEEE. 2012, pp. 199–204.

[72] Sangwoo Moon et al. “Neuro-DCF: Design of wireless MAC via multi-agent reinforcement learning approach”. In: *Proceedings of the Twenty-second International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*. 2021, pp. 141–150.

[73] Jie Hui and Michael Devetsikiotis. “A unified model for the performance analysis of IEEE 802.11 e EDCA”. In: *IEEE Transactions on Communications* 53.9 (2005), pp. 1498–1510.

[74] Szymon Szott, Marek Natkaniec, and AndrzejR Pach. “An IEEE 802.11 EDCA model with support for analysing networks with misbehaving nodes”. In: *EURASIP Journal on Wireless Communications and Networking* 2010 (2010), pp. 1–13.

[75] Syuan-Cheng Chen, Chi-Yu Li, and Chui-Hao Chiu. “An experience driven design for IEEE 802.11 ac rate adaptation based on reinforcement learning”. In: *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE. 2021, pp. 1–10.

[76] Raja Karmakar, Samiran Chattopadhyay, and Sandip Chakraborty. “An online learning approach for auto link-Configuration in IEEE 802.11 ac wireless networks”. In: *Computer Networks* 181 (2020), p. 107426.

[77] Raja Karmakar, Samiran Chattopadhyay, and Sandip Chakraborty. “IEEE 802.11 ac link adaptation under mobility”. In: *2017 IEEE 42nd Conference on Local Computer Networks (LCN)*. IEEE. 2017, pp. 392–400.

[78] Raja Karmakar, Samiran Chattopadhyay, and Sandip Chakraborty. “SmartLA: Reinforcement learning-based link adaptation for high throughput wireless access networks”. In: *Computer Communications* 110 (2017), pp. 1–25.

[79] Dionysios Skordoulis et al. “IEEE 802.11 n MAC frame aggregation mechanisms for next-generation high-throughput WLANs”. In: *IEEE Wireless Communications* 15.1 (2008), pp. 40–47.

[80] Evgeny Khorov et al. “A tutorial on IEEE 802.11 ax high efficiency WLANs”. In: *IEEE Communications Surveys & Tutorials* 21.1 (2018), pp. 197–216.

[81] Katarzyna Kosek-Szott and Norbert Rapacz. “Tuning wi-fi traffic differentiation by combining frame aggregation with TXOP limits”. In: *IEEE Communications Letters* 24.3 (2019), pp. 700–703.

[82] Hamid Hassani, Francesco Gringoli, and Douglas J Leith. “Quick & plenty: Achieving low delay & high rate in 802.11 ac edge networks”. In: *Computer Networks* 187 (2021), p. 107820.

[83] Guofeng Zhao et al. “Joint power control and channel allocation for interference mitigation based on reinforcement learning”. In: *IEEE Access* 7 (2019), pp. 177254–177265.

[84] Boris Bellalta and Katarzyna Kosek-Szott. “AP-initiated multi-user transmissions in IEEE 802.11 ax WLANs”. In: *Ad Hoc Networks* 85 (2019), pp. 145–159.

[85] Sergio Barrachina-Muñoz, Boris Bellalta, and Edward W Knightly. “Wi-Fi channel bonding: An all-channel system and experimental study from urban hotspots to a sold-out stadium”. In: *IEEE/ACM Transactions on Networking* 29.5 (2021), pp. 2101–2114.

[86] Álvaro López-Raventós and Boris Bellalta. “Multi-link operation in IEEE 802.11 be WLANs”. In: *IEEE Wireless Communications* 29.4 (2022), pp. 94–100.

[87] Gaurang Naik et al. “Next generation Wi-Fi and 5G NR-U in the 6 GHz bands: Opportunities and challenges”. In: *IEEE Access* 8 (2020), pp. 153027–153056.

[88] Marc Carrascosa et al. “An experimental study of latency for IEEE 802.11 be multi-link operation”. In: *ICC 2022-IEEE International Conference on Communications*. IEEE. 2022, pp. 2507–2512.

[89] Francesc Wilhelmi et al. “Spatial reuse in IEEE 802.11 ax WLANs”. In: *Computer Communications* 170 (2021), pp. 65–83.

[90] Cailian Deng et al. “IEEE 802.11 be Wi-Fi 7: New challenges and opportunities”. In: *IEEE Communications Surveys & Tutorials* 22.4 (2020), pp. 2136–2166.

[91] David Nunez et al. “TXOP sharing with coordinated spatial reuse in multi-AP cooperative IEEE 802.11 be WLANs”. In: *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE. 2022, pp. 864–870.

[92] Khalid Aldubaikhy et al. “mmWave IEEE 802.11 ay for 5G fixed wireless access”. In: *IEEE Wireless Communications* 27.2 (2020), pp. 88–95.

[93] Ting-Wei Chang, Li-Hsiang Shen, and Kai-Ten Feng. “Learning-based beam training algorithms for IEEE802. 11ad/ay networks”. In: *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*. IEEE. 2019, pp. 1–5.

[94] Li-Hsiang Shen et al. “Design and implementation for deep learning based adjustable beamforming training for millimeter wave communication systems”. In: *IEEE Transactions on Vehicular Technology* 70.3 (2021), pp. 2413–2427.

[95] Michele Polese, Francesco Restuccia, and Tommaso Melodia. “DeepBeam: Deep waveform learning for coordination-free beam management in mmWave networks”. In: *Proceedings of the Twenty-second International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*. 2021, pp. 61–70.

[96] Batool Salehi et al. “Machine learning on camera images for fast mmWave beamforming”. In: *2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS).* IEEE. 2020, pp. 338–346.

[97] Takayuki Nishio et al. “Proactive received power prediction using machine learning and depth images for mmWave networks”. In: *IEEE Journal on Selected Areas in Communications* 37.11 (2019), pp. 2413–2427.

[98] Yusuke Koda et al. “Handover management for mmWave networks with proactive performance prediction using camera images and deep reinforcement learning”. In: *IEEE Transactions on Cognitive Communications and Networking* 6.2 (2019), pp. 802–816.

[99] Ehab Mahmoud Mohamed, Kei Sakaguchi, and Seiichi Sampei. “Wi-Fi coordinated WiGig concurrent transmissions in random access scenarios”. In: *IEEE Transactions on Vehicular Technology* 66.11 (2017), pp. 10357–10371.

[100] Pei Zhou et al. “Deep learning-based beam management and interference coordination in dense mmWave networks”. In: *IEEE Transactions on Vehicular Technology* 68.1 (2018), pp. 592–603.

[101] Thi Ha Ly Dinh et al. “Deep reinforcement learning-based user association in sub6GHz/mmWave integrated networks”. In: *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC).* IEEE. 2021, pp. 1–7.

[102] Ernest Kurniawan, Lin Zhiwei, and Sumei Sun. “Machine learning-based channel classification and its application to IEEE 802.11 ad communications”. In: *GLOBECOM 2017-2017 IEEE Global Communications Conference.* IEEE. 2017, pp. 1–6.

[103] Lu Bai et al. “Predicting wireless mmWave massive MIMO channel characteristics using machine learning algorithms”. In: *Wireless Communications and Mobile Computing* 2018 (2018), pp. 1–12.

[104] Shivang Aggarwal et al. “An experimental study of rate and beam adaptation in 60 GHz WLANs”. In: *Proceedings of the 23rd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems.* 2020, pp. 171–180.

[105] Shivang Aggarwal et al. “LiBRA: learning-based link adaptation leveraging PHY layer information in 60 GHz WLANs”. In: *Proceedings of the 16th International Conference on emerging Networking EXperiments and Technologies.* 2020, pp. 245–260.

[106] Xintong Lin, Lin Zhang, and Yuan Jiang. “Location aided intelligent deep learning channel estimation for millimeter wave communications”. In: *2020 IEEE/CIC International Conference on Communications in China (ICCC).* IEEE. 2020, pp. 489–494.

[107] Raja Karmakar, Samiran Chattopadhyay, and Sandip Chakraborty. “Intelligent MU-MIMO user selection with dynamic link adaptation in IEEE 802.11 ax”. In: *IEEE Transactions on Wireless Communications* 18.2 (2019), pp. 1155–1165.

[108] Alberto Rico-Alvarino and Robert W Heath. “Learning-based adaptive transmission for limited feedback multiuser MIMO-OFDM”. In: *IEEE transactions on wireless communications* 13.7 (2014), pp. 3806–3820.

[109] Pedram Kheirkhah Sangdeh et al. “LB-SciFi: Online learning-based channel feedback for MU-MIMO in wireless LANs”. In: *2020 IEEE 28th International Conference on Network Protocols (ICNP)*. IEEE. 2020, pp. 1–11.

[110] Shi Su et al. “Client pre-screening for MU-MIMO in commodity 802.11 ac networks via online learning”. In: *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE. 2019, pp. 649–657.

[111] Shi Su et al. “Data-driven mode and group selection for downlink MU-MIMO with implementation in commodity 802.11 ac network”. In: *IEEE Transactions on Communications* 69.3 (2021), pp. 1620–1634.

[112] Ravikumar Balakrishnan et al. “Deep reinforcement learning based traffic-and channel-aware OFDMA resource allocation”. In: *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE. 2019, pp. 1–6.

[113] Dheeraj Kotagiri, Koichi Nihei, and Tansheng Li. “Multi-user distributed spectrum access method for 802.11 ax stations”. In: *2020 29th International Conference on Computer Communications and Networks (ICCCN)*. IEEE. 2020, pp. 1–2.

[114] Dheeraj Kotagiri, Koichi Nihei, and Tansheng Li. “Distributed convolutional deep reinforcement learning based OFDMA MAC for 802.11 ax”. In: *ICC 2021-IEEE International Conference on Communications*. IEEE. 2021, pp. 1–6.

[115] Shunnosuke Kotera et al. “Lyapunov optimization-based latency-bounded allocation using deep deterministic policy gradient for 11ax spatial reuse”. In: *IEEE Access* 9 (2021), pp. 162337–162347.

[116] Pedram Kheirkhah Sangdeh and Huacheng Zeng. “DeepMux: Deep-learning-based channel sounding and resource allocation for IEEE 802.11 ax”. In: *IEEE Journal on Selected Areas in Communications* 39.8 (2021), pp. 2333–2346.

[117] Michael Timmers et al. “A spatial learning algorithm for IEEE 802.11 networks”. In: *2009 IEEE International Conference on Communications*. IEEE. 2009, pp. 1–6.

[118] Bo Yin et al. “Learning-based spatial reuse for WLANs with early identification of interfering transmitters”. In: *IEEE Transactions on Cognitive Communications and Networking* 6.1 (2019), pp. 151–164.

[119] Danh H Nguyen et al. “Enhancing indoor spatial reuse through adaptive antenna beamsteering”. In: *Proceedings of the Tenth ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization*. 2016, pp. 81–82.

[120] Francesc Wilhelmi et al. “Implications of decentralized Q-learning resource allocation in wireless networks”. In: *2017 ieee 28th annual international symposium on personal, indoor, and mobile radio communications (pimrc)*. IEEE. 2017, pp. 1–5.

[121] Francesc Wilhelmi et al. “Potential and pitfalls of multi-armed bandits for decentralized spatial reuse in WLANs”. In: *Journal of Network and Computer Applications* 127 (2019), pp. 26–42.

[122] Francesc Wilhelmi et al. “Collaborative spatial reuse in wireless networks via selfish multi-armed bandits”. In: *Ad Hoc Networks* 88 (2019), pp. 129–141.

[123] Álvaro López-Raventós et al. "Combining software defined networks and machine learning to enable self organizing WLANs". In: *2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE. 2019, pp. 1–8.

[124] Elif Ak and Berk Canberk. "FSC: Two-scale AI-driven fair sensitivity control for 802.11 ax networks". In: *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE. 2020, pp. 1–6.

[125] Anthony Bardou, Thomas Begin, and Anthony Busson. "Improving the spatial reuse in ieee 802.11 ax wlans: A multi-armed bandit approach". In: *Proceedings of the 24th International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. 2021, pp. 135–144.

[126] Sergio Barrachina-Muñoz, Alessandro Chiumento, and Boris Bellalta. "Stateless reinforcement learning for multi-agent systems: The case of spectrum allocation in dynamic channel bonding WLANs". In: *2021 Wireless Days (WD)*. IEEE. 2021, pp. 1–5.

[127] Raja Karmakar, Samiran Chattopadhyay, and Sandip Chakraborty. "Learning based adaptive fair QoS in IEEE 802.11 ac access networks". In: *2019 11th International Conference on Communication Systems & Networks (COMSNETS)*. IEEE. 2019, pp. 22–29.

[128] Zaheer Khan and Janne J Lehtomäki. "Interactive trial and error learning method for distributed channel bonding: Model, prototype implementation, and evaluation". In: *IEEE Transactions on Cognitive Communications and Networking* 5.2 (2019), pp. 206–223.

[129] Hiroshi Kanemasa et al. "Dynamic channel bonding using laser chaos decision maker in WLANs". In: *2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*. IEEE. 2021, pp. 078–082.

[130] Hang Qi et al. "On-demand channel bonding in heterogeneous WLANs: A multi-agent deep reinforcement learning approach". In: *Sensors* 20.10 (2020), p. 2789.

[131] Yizhou Luo and Kwan-Wu Chin. "Learning to bond in dense WLANs with random traffic demands". In: *IEEE Transactions on Vehicular Technology* 69.10 (2020), pp. 11868–11879.

[132] Kumar Ayush et al. "Supporting throughput fairness in IEEE 802.11 ac dynamic bandwidth channel access: A hybrid approach". In: *2017 IEEE 42nd Conference on Local Computer Networks (LCN)*. IEEE. 2017, pp. 462–470.

[133] Sergio Barrachina-Muñoz, Alessandro Chiumento, and Boris Bellalta. "Multi-armed bandits for spectrum allocation in multi-agent channel bonding WLANs". In: *IEEE Access* 9 (2021), pp. 133472–133490.

[134] Mengqi Han et al. "A deep reinforcement learning based approach for channel aggregation in IEEE 802.11 ax". In: *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE. 2020, pp. 1–6.

[135] Francesc Wilhelmi et al. "Machine learning for performance prediction of channel bonding in next-generation IEEE 802.11 WLANs". In: *arXiv preprint arXiv:2105.14219* (2021).

[136] Paola Soto et al. "ATARI: A graph convolutional neural network approach for performance prediction in next-generation WLANs". In: *Sensors* 21.13 (2021), p. 4321.

[137] Raja Karmakar, Samiran Chattopadhyay, and Sandip Chakraborty. "Smart-Bond: A deep probabilistic machinery for smart channel bonding in IEEE 802.11 ac". In: *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE. 2020, pp. 2599–2608.

[138] Kazuto Yano et al. "Achievable throughput of multiband wireless LAN using simultaneous transmission over multiple primary channels assisted by idle length prediction based on PNN". In: *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*. IEEE. 2019, pp. 022–027.

[139] Neelakantan Nurani Krishnan et al. "Optimizing throughput performance in distributed MIMO Wi-Fi networks using deep reinforcement learning". In: *IEEE Transactions on Cognitive Communications and Networking* 6.1 (2019), pp. 135–150.

[140] Yirun Zhang, Qirui Wu, and Mohammad Shikh-Bahaei. "A pointer network based deep learning algorithm for user pairing in full-duplex Wi-Fi networks". In: *IEEE Transactions on Vehicular Technology* 69.10 (2020), pp. 12363–12368.

[141] Zhou Wang et al. "Intelligent hybrid automatic repeat request retransmission for multi-band Wi-Fi networks". In: *IET Communications* 15.9 (2021), pp. 1249–1258.

[142] David López-Pérez et al. "IEEE 802.11 be extremely high throughput: The next generation of Wi-Fi technology beyond 802.11 ax". In: *IEEE Communications Magazine* 57.9 (2019), pp. 113–119.

[143] Katarzyna Kosek-Szott and Krzysztof Domino. "An efficient backoff procedure for IEEE 802.11 ax uplink OFDMA-based random access". In: *IEEE Access* 10 (2022), pp. 8855–8863.

[144] Imad Jamil, Laurent Cariou, and Jean-François Hélard. "Novel learning-based spatial reuse optimization in dense WLAN deployments". In: *EURASIP Journal on Wireless Communications and Networking* 2016 (2016), pp. 1–19.

[145] Linyu Huang, Liang Lu, and Wei Hua. "A survey on next-cell prediction in cellular networks: Schemes and applications". In: *IEEE Access* 8 (2020), pp. 201468–201485.

[146] Marc Carrascosa and Boris Bellalta. "Decentralized AP selection using multi-armed bandits: Opportunistic ε-greedy with stickiness". In: *2019 IEEE Symposium on Computers and Communications (ISCC)*. IEEE. 2019, pp. 1–7.

[147] Marc Carrascosa and Boris Bellalta. "Multi-armed bandits for decentralized AP selection in enterprise WLANs". In: *Computer Communications* 159 (2020), pp. 108–123.

[148] Álvaro López-Raventós and Boris Bellalta. "Concurrent decentralized channel allocation and access point selection using multi-armed bandits in multi BSS WLANs". In: *Computer Networks* 180 (2020), p. 107381.

[149] Biljana Bojovic et al. "A supervised learning approach to cognitive access point selection". In: *2011 IEEE GLOBECOM Workshops (GC Wkshps)*. IEEE. 2011, pp. 1100–1105.

[150] Yao Liu et al. "On cognitive network channel selection and the impact on transport layer performance". In: *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*. IEEE. 2010, pp. 1–5.

[151] Thi Ha Ly Dinh et al. “Distributed user-to-multiple access points association through deep learning for beyond 5G”. In: *Computer Networks* 197 (2021), p. 108258.

[152] Mohamed Amine Kafi, Alexandre Mouradian, and Véronique Vèque. “On-line client association scheme based on reinforcement learning for WLAN networks”. In: *2019 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE. 2019, pp. 1–7.

[153] Changhua Pei et al. “Why it takes so long to connect to a WiFi access point”. In: *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE. 2017, pp. 1–9.

[154] Lixing Song and Aaron Striegel. “Leveraging frame aggregation to improve access point selection”. In: *2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE. 2017, pp. 325–330.

[155] Mauro Feltrin and Stefano Tomasin. “A machine-learning-based handover prediction for anticipatory techniques in wi-fi networks”. In: *2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN)*. IEEE. 2018, pp. 341–345.

[156] Ensar Zeljković et al. “ABRAHAM: machine learning backed proactive handover algorithm using SDN”. In: *IEEE Transactions on Network and Service Management* 16.4 (2019), pp. 1522–1536.

[157] Zijun Han et al. “Artificial intelligence-based handoff management for dense WLANs: A deep reinforcement learning approach”. In: *IEEE Access* 7 (2019), pp. 31688–31701.

[158] Jiamei Chen, Lin Ma, Yubin Xu, et al. “Support vector machine based mobility prediction scheme in heterogeneous wireless networks”. In: *Mathematical Problems in Engineering* 2015 (2015).

[159] Jianxi Yang, Chuping Dai, and Zhengguang Ding. “A scheme of terminal mobility prediction of ultra dense network based on SVM”. In: *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)*. IEEE. 2017, pp. 837–842.

[160] Yaxing Qiu et al. “Intelligent user profile prediction in radio access network”. In: *Signal and Information Processing, Networking and Computers: Proceedings of the 5th International Conference on Signal and Information Processing, Networking and Computers (ICSINC)*. Springer. 2019, pp. 437–445.

[161] CP Koushik and P Vetrivelan. “Heuristic relay-node selection in opportunistic network using RNN-LSTM based mobility prediction”. In: *Wireless Personal Communications* 114.3 (2020), pp. 2363–2388.

[162] Sibren De Bast et al. “Deep reinforcement learning for dynamic network slicing in IEEE 802.11 networks”. In: *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE. 2019, pp. 264–269.

[163] Feng Lyu et al. “Large-scale full WiFi coverage: Deployment and management strategy based on user spatio-temporal association analytics”. In: *IEEE Internet of Things Journal* 6.6 (2019), pp. 9386–9398.

[164] Pierluigi Gallo and Domenico Garlisi. “Wi-Dia: Data-driven wireless diagnostic using context recognition”. In: *2018 IEEE 4th International Forum on Research and Technology for Society and Industry (RTSI)*. IEEE. 2018, pp. 1–6.

[165] Ilias Syrigos et al. “On the employment of machine learning techniques for troubleshooting WiFi networks”. In: *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE. 2019, pp. 1–6.

[166] Nikita Trivedi, Bighnaraj Panigrahi, and Hemant Kumar Rath. “WiNetSense: Sensing and analysis model for large-scale wireless networks”. In: *IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE. 2020, pp. 787–792.

[167] Anisa Allahdadi et al. “Hidden Markov models on a self-organizing map for anomaly detection in 802.11 wireless networks”. In: *Neural Computing and Applications* 33.14 (2021), pp. 8777–8794.

[168] Maghsoud Morshedi and Josef Noll. “Estimating PQoS of video streaming on Wi-Fi networks using machine learning”. In: *Sensors* 21.2 (2021), p. 621.

[169] Maghsoud Morshedi and Josef Noll. “Estimating PQoS of video conferencing on Wi-Fi networks using machine learning”. In: *Future Internet* 13.3 (2021), p. 63.

[170] Muhammad Asif Khan et al. “Real-time throughput prediction for cognitive Wi-Fi networks”. In: *Journal of Network and Computer Applications* 150 (2020), p. 102499.

[171] Huifang Feng et al. “SVM-based models for predicting WLAN traffic”. In: *2006 IEEE international conference on communications*. Vol. 2. IEEE. 2006, pp. 597–602.

[172] Alisha Thapaliya, James Schnebly, and Shamik Sengupta. “Predicting congestion level in wireless networks using an integrated approach of supervised and unsupervised learning”. In: *2018 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. IEEE. 2018, pp. 977–982.

[173] Mehmet Karaca and Björn Landfeldt. “Load-aware channel selection for 802.11 WLANs with limited measurement”. In: *2016 23rd International Conference on Telecommunications (ICT)*. IEEE. 2016, pp. 1–5.

[174] Apollinaire Nadembega, Abdelhakim Hafid, and Tarik Taleb. “Mobility-prediction-aware bandwidth reservation scheme for mobile networks”. In: *IEEE Transactions on Vehicular Technology* 64.6 (2014), pp. 2561–2576.

[175] Floriano De Rango, Peppino Fazio, and Salvatore Marano. “Utility-based predictive services for adaptive wireless networks with mobile hosts”. In: *IEEE Transactions on Vehicular Technology* 58.3 (2008), pp. 1415–1428.

[176] Chaoming Song et al. “Limits of predictability in human mobility”. In: *Science* 327.5968 (2010), pp. 1018–1021.

[177] Chih-Wei Huang, Chiu-Ti Chiang, and Qiuhui Li. “A study of deep learning networks on mobile traffic forecasting”. In: *2017 IEEE 28th annual international symposium on personal, indoor, and mobile radio communications (PIMRC)*. IEEE. 2017, pp. 1–6.

[178] Yong Li et al. “Limits of predictability for large-scale urban vehicular mobility”. In: *IEEE Transactions on Intelligent Transportation Systems* 15.6 (2014), pp. 2671–2682.

[179] Arfah Hasbollah et al. “Mobility prediction method for vehicular network using Markov chain”. In: *Jurnal Teknologi* 78.6–2 (2016), pp. 7–13.

[180] Hongbo Si et al. “Mobility prediction in cellular network using hidden markov model”. In: *2010 7th IEEE consumer communications and networking conference*. IEEE. 2010, pp. 1–5.

[181] Shubhajeet Chatterjee et al. “An improved mobility management technique for IEEE 802.11 based WLAN by predicting the direction of the mobile node”. In: *2012 National Conference on Computing and Communication Systems*. IEEE. 2012, pp. 1–5.

[182] Ashish Patro and Suman Banerjee. “Outsourcing coordination and management of home wireless access points through an open API”. In: *2015 IEEE conference on computer communications (INFOCOM)*. IEEE. 2015, pp. 1454–1462.

[183] Eugene Chai et al. “LTE in unlicensed spectrum: Are we there yet?” In: *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. 2016, pp. 135–148.

[184] Fuad M Abinader et al. “Enabling the coexistence of LTE and Wi-Fi in unlicensed bands”. In: *IEEE communications magazine* 52.11 (2014), pp. 54–61.

[185] Gürkan Gür. “Expansive networks: Exploiting spectrum sharing for capacity boost and 6G vision”. In: *Journal of Communications and Networks* 22.6 (2020), pp. 444–454.

[186] Jiawei Han, Jian Pei, and Hanghang Tong. *Data mining: concepts and techniques*. Morgan kaufmann, 2022.

[187] Jason Brownlee. *Machine learning mastery with Python: understand your data, create accurate models, and work projects end-to-end*. Machine Learning Mastery, 2016.

[188] Charu C.. Aggarwal. *Data Classification: Algorithms and Applications*. CRC Press, 2020.

[189] Ahmad Azab et al. “Network traffic classification: Techniques, datasets, and challenges”. In: *Digital Communications and Networks* (2022).

[190] Bayya Yegnanarayana. *Artificial neural networks*. PHI Learning Pvt. Ltd., 2009.

[191] Wayne A Woodward, Bivin Philip Sadler, and Stephen Robertson. *Time series for data science: Analysis and forecasting*. CRC Press, 2022.

[192] Robert H Shumway, David S Stoffer, and David S Stoffer. *Time series analysis and its applications*. Vol. 3. Springer, 2000.

[193] Sachin S Talathi and Aniket Vartak. “Improving performance of recurrent neural network with relu nonlinearity”. In: *arXiv preprint arXiv:1511.03771* (2015).

[194] Jason Brownlee. *Deep learning for time series forecasting: predict the future with MLPs, CNNs and LSTMs in Python*. Machine Learning Mastery, 2018.

[195] Weicong Kong et al. “Short-term residential load forecasting based on LSTM recurrent neural network”. In: *IEEE transactions on smart grid* 10.1 (2017), pp. 841–851.

[196] Mirco Ravanelli et al. “Light gated recurrent units for speech recognition”. In: *IEEE Transactions on Emerging Topics in Computational Intelligence* 2.2 (2018), pp. 92–102.

[197] Anastasia Borovykh, Sander Bohte, and Cornelis W Oosterlee. “Conditional time series forecasting with convolutional neural networks”. In: *arXiv preprint arXiv:1703.04691* (2017).

[198] Jeffrey Donahue et al. “Long-term recurrent convolutional networks for visual recognition and description”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2015, pp. 2625–2634.

[199] Neo Wu et al. “Deep transformer models for time series forecasting: The influenza prevalence case”. In: *arXiv preprint arXiv:2001.08317* (2020).

[200] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in neural information processing systems* 30 (2017).

[201] Shan Jiang, Joseph Ferreira, and Marta C Gonzalez. “Activity-based human mobility patterns inferred from mobile phone data: A case study of Singapore”. In: *IEEE Transactions on Big Data* 3.2 (2017), pp. 208–219.

[202] Barbara Furletti et al. “Identifying users profiles from mobile calls habits”. In: *Proceedings of the ACM SIGKDD international workshop on urban computing.* 2012, pp. 17–24.

[203] Juan Sanchez-Gonzalez et al. “On extracting user-centric knowledge for personalised Quality of Service in 5G networks”. In: *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM).* IEEE. 2017, pp. 971–974.

[204] Amit Kumar and Hari Om. “A secure seamless handover authentication technique for wireless LAN”. In: *2015 International Conference on Information Technology (ICIT).* IEEE. 2015, pp. 43–47.

[205] Chen Yan et al. “CELoF: WiFi dwell time estimation in free environment”. In: *International Conference on Multimedia Modeling.* Springer. 2016, pp. 503–514.

[206] 3GPP. “Technical specification Group services and system Aspects”. In: *Release* 15 (2018).

[207] Meysam Goodarzi et al. “Next-cell prediction based on cell sequence history and intra-cell trajectory”. In: *2019 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN).* IEEE. 2019, pp. 257–263.

[208] Pratap S Prasad and Prathima Agrawal. “Mobility prediction for wireless network resource management”. In: *2009 41st Southeastern Symposium on System Theory.* IEEE. 2009, pp. 98–102.

[209] Khong-Lim Yap and Yung-Wey Chong. “Optimized access point selection with mobility prediction using hidden Markov model for wireless network”. In: *2017 ninth international conference on ubiquitous and future networks (ICUFN).* IEEE. 2017, pp. 38–42.

[210] Ahlam Ben Cheikh et al. “Optimized handoff with mobility prediction scheme using hmm for femtocell networks”. In: *2015 IEEE International Conference on Communications (ICC).* IEEE. 2015, pp. 3448–3453.

[211] Khong-Lim Yap, Yung-Wey Chong, and Weixia Liu. “Enhanced handover mechanism using mobility prediction in wireless networks”. In: *PloS one* 15.1 (2020), e0227982.

[212] Dilranjan S Wickramasuriya et al. “Base station prediction and proactive mobility management in virtual cells using recurrent neural networks”. In: *2017 IEEE 18th Wireless and Microwave Technology Conference (WAMICON).* IEEE. 2017, pp. 1–6.

[213] Marvin Manalastas et al. “Where to go next?: A realistic evaluation of AI-assisted mobility predictors for HetNets”. In: *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE. 2020, pp. 1–6.

[214] Justin Manweiler et al. “Predicting length of stay at wifi hotspots”. In: *2013 Proceedings IEEE INFOCOM*. IEEE. 2013, pp. 3102–3110.

[215] Smita Skrivanek. “The use of dummy variables in regression analysis”. In: *More Steam, LLC* (2009).

[216] Cisco Prime Infrastructure. *3.5 Administrator Guide*.

[217] *RapidMiner Studio*. `http://www.rapidminer.com`. Accessed: Online.

[218] Anurag Kumar, D Manjunath, and Joy Kuri. *Wireless networking*. Elsevier, 2008.

[219] Kaisa Zhang et al. “A new method for traffic forecasting in urban wireless communication network”. In: *EURASIP Journal on Wireless Communications and Networking* 2019 (2019), pp. 1–12.

[220] Jing Wang et al. “Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach”. In: *IEEE INFOCOM 2017-IEEE conference on computer communications*. IEEE. 2017, pp. 1–9.

[221] Wenxiong Chen et al. “Flag: Flexible, accurate, and long-time user load prediction in large-scale WiFi system using deep RNN”. In: *IEEE Internet of Things Journal* 8.22 (2021), pp. 16510–16521.

[222] Su P Sone, Janne J Lehtomäki, and Zaheer Khan. “Wireless traffic usage forecasting using real enterprise network data: Analysis and methods”. In: *IEEE Open Journal of the Communications Society* 1 (2020), pp. 777–797.

[223] Su Pyae Sone et al. “Forecasting wireless network traffic and channel utilization using real network/physical layer data”. In: *2021 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*. IEEE. 2021, pp. 31–36.

[224] Tao Deng et al. “Short term prediction of wireless traffic based on tensor decomposition and recurrent neural network”. In: *SN Applied Sciences* 3.9 (2021), p. 779.

[225] Yue Xu et al. “High-accuracy wireless traffic prediction: A GP-based machine learning approach”. In: *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE. 2017, pp. 1–6.

[226] Laisen Nie et al. “Network traffic prediction based on deep belief network in wireless mesh backbone networks”. In: *2017 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE. 2017, pp. 1–5.

[227] Yu Li et al. “Prophet model and Gaussian process regression based user traffic prediction in wireless networks”. In: *Science China Information Sciences* 63 (2020), pp. 1–8.

[228] Min Wei and Keecheon Kim. “Intrusion detection scheme using traffic prediction for wireless industrial networks”. In: *Journal of Communications and Networks* 14.3 (2012), pp. 310–318.

[229] Chuanting Zhang et al. “Citywide cellular traffic prediction based on densely connected convolutional neural networks”. In: *IEEE Communications Letters* 22.8 (2018), pp. 1656–1659.

[230] Ming Li et al. "A deep learning method based on an attention mechanism for wireless network traffic prediction". In: *Ad Hoc Networks* 107 (2020), p. 102258.

[231] Juan Wen et al. "Assisting intelligent wireless networks with traffic prediction: Exploring and exploiting predictive causality in wireless traffic". In: *IEEE Communications Magazine* 58.6 (2020), pp. 26–31.

[232] Guanyao Li et al. "A Lightweight and Accurate Spatial-Temporal Transformer for Traffic Forecasting". In: *IEEE Transactions on Knowledge and Data Engineering* (2022).

[233] Yahui Hu et al. "Citywide Mobile Traffic Forecasting Using Spatial-Temporal Downsampling Transformer Neural Networks". In: *IEEE Transactions on Network and Service Management* 20.1 (2022), pp. 152–165.

[234] Bo Gu et al. "A Spatial-Temporal Transformer Network for City-Level Cellular Traffic Analysis and Prediction". In: *IEEE Transactions on Wireless Communications* (2023).

[235] Shang Yimeng et al. "A Prediction Method of 5G Base Station Cell Traffic Based on Improved Transformer Model". In: *2022 IEEE 4th International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*. IEEE. 2022, pp. 40–45.

[236] Cisco. *Radio Resource Management*. White Paper. 2016.

[237] Wenyu Zhang et al. "Toward intelligent network optimization in wireless networking: An auto-learning framework". In: *IEEE Wireless Communications* 26.3 (2019), pp. 76–82.

[238] Haoyi Zhou et al. "Informer: Beyond efficient transformer for long sequence time-series forecasting". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 35. 12. 2021, pp. 11106–11115.

[239] Jason Brownlee. *Introduction to time series forecasting with python: how to prepare data and develop models to predict the future*. Machine Learning Mastery, 2017.

[240] Klenilmar Lopes Dias et al. "An innovative approach for real-time network traffic classification". In: *Computer networks* 158 (2019), pp. 143–157.

[241] *XR*. Wi-Fi Alliance. https://www.wi-fi.org/discover-wi-fi/xr. 2022.

[242] ReportLinker. *Extended Reality Market Size & Share Analysis – Growth Trends & Forecasts (2023-2028)*. https://www.reportlinker.com/p06484014/Extended-Reality-Market-Size-Share-Analysis-Growth-Trends-Forecasts.html?utm_source=GNW. Accessed on August, 2023.

[243] Dang Tai Tan, Sanghyun Kim, and Ji-Hoon Yun. "Enhancement of Motion Feedback Latency for Wireless Virtual Reality in IEEE 802.11 WLANs". In: *2019 IEEE Global Communications Conference (GLOBECOM)*. 2019, pp. 1–6. DOI: 10.1109/GLOBECOM38437.2019.9013507.

[244] Costas Michaelides et al. "Is Wi-Fi 6 Ready for Virtual Reality Mayhem? A Case Study Using One AP and Three HMDs". In: (2023).

[245] Wi-Fi Alliance. *Wi-Fi CERTIFIED QoS Management*. https://www.wi-fi.org/file/wi-fi-certified-qos-managementtm-highlights. 2021.

[246] Internet Assigned Numbers Authority (IANA). *Service Name and Transport Protocol Port Number Registry*. https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml. Accessed on: July 1, 2020. 2020.

[247] Andrew W Moore and Konstantina Papagiannaki. “Toward the accurate identification of network applications”. In: *International workshop on passive and active network measurement*. Springer. 2005, pp. 41–54.

[248] Subhabrata Sen, Oliver Spatscheck, and Dongmei Wang. “Accurate, scalable in-network identification of p2p traffic using application signatures”. In: *Proceedings of the 13th international conference on World Wide Web*. 2004, pp. 512–521.

[249] Pratibha Khandait, Neminath Hubballi, and Bodhisatwa Mazumdar. “Efficient keyword matching for deep packet inspection based network traffic classification”. In: *2020 International Conference on COMmunication Systems & NETworkS (COMSNETS)*. IEEE. 2020, pp. 567–570.

[250] Stênio Fernandes et al. “Slimming down deep packet inspection systems”. In: *IEEE INFOCOM Workshops 2009*. IEEE. 2009, pp. 1–6.

[251] Neminath Hubballi and Mayank Swarnkar. “*Bitcoding*: Network traffic classification through encoded bit level signatures”. In: *IEEE/ACM Transactions on Networking* 26.5 (2018), pp. 2334–2346.

[252] Daniel Gibert, Carles Mateu, and Jordi Planes. “The rise of machine learning for detection and classification of malware: Research developments, trends and challenges”. In: *Journal of Network and Computer Applications* 153 (2020), p. 102526.

[253] Gianni D’Angelo and Francesco Palmieri. “Network traffic classification using deep convolutional recurrent autoencoder neural networks for spatial–temporal features extraction”. In: *Journal of Network and Computer Applications* 173 (2021), p. 102890.

[254] Licheng Zhang et al. “Video traffic identification with novel feature extraction and selection method”. In: *arXiv preprint arXiv:2303.03949* (2023).

[255] Philippe Graff et al. “Efficient Identification of Cloud Gaming Traffic at the Edge”. In: *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*. 2023, pp. 1–10. DOI: 10.1109/NOMS56928.2023.10154417.

[256] Zhong Fan and Ran Liu. “Investigation of machine learning based network traffic classification”. In: *2017 International Symposium on Wireless Communication Systems (ISWCS)*. 2017, pp. 1–6. DOI: 10.1109/ISWCS.2017.8108090.

[257] Ruixi Yuan et al. “An SVM-based machine learning method for accurate internet traffic classification”. In: *Information Systems Frontiers* 12 (2010), pp. 149–156.

[258] A Jenefa and M BalaSingh Moses. “An Upgraded C5. 0 Algorithm for Network Application Identification”. In: *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*. IEEE. 2018, pp. 789–794.

[259] Guanglu Sun et al. “Internet traffic classification based on incremental support vector machines”. In: *Mobile Networks and Applications* 23 (2018), pp. 789–796.

[260] Jie Cao et al. “An improved network traffic classification model based on a support vector machine”. In: *Symmetry* 12.2 (2020), p. 301.

[261] Ali Safari Khatouni and Nur Zincir-Heywood. “Integrating machine learning with off-the-shelf traffic flow features for http/https traffic classification”. In: *2019 IEEE Symposium on Computers and Communications (ISCC)*. IEEE. 2019, pp. 1–7.

[262] Shi Dong. “Multi class SVM algorithm with active learning for network traffic classification”. In: *Expert Systems with Applications* 176 (2021), p. 114885.

[263] Afeez Ajani Afuwape et al. “Performance evaluation of secured network traffic classification using a machine learning approach”. In: *Computer Standards & Interfaces* 78 (2021), p. 103545.

[264] Elaiyasuriyan Ganesan et al. “SDN-enabled FiWi-IoT smart environment network traffic classification using supervised ML models”. In: *Photonics*. Vol. 8. 6. MDPI. 2021, p. 201.

[265] S.S. Shaabanzadeh, C. Michaelides, M. Casasnovas, D. Marchitelli, J. Sánchez-González, B. Bellalta. *Virtual Reality Traces for Traffic Classification*. `https://doi.org/10.5281/zenodo.10401241`. 2023.

[266] Boris Bellalta. “IEEE 802.11 ax: High-efficiency WLANs”. In: *IEEE wireless communications* 23.1 (2016), pp. 38–46.

[267] Jason Brownlee. *Statistical methods for machine learning: Discover how to transform data into knowledge with Python*. Machine Learning Mastery, 2018.

[268] Max Kuhn, Kjell Johnson, et al. *Applied predictive modeling*. Vol. 26. Springer, 2013.

[269] Fabian Pedregosa et al. “Scikit-learn: Machine learning in Python”. In: *the Journal of machine Learning research* 12 (2011), pp. 2825–2830.

[270] Peter Christen and Peter Christen. “Evaluation of matching quality and complexity”. In: *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection* (2012), pp. 163–184.

[271] Marc Carrascosa-Zamacois et al. “Understanding Multi-link Operation in Wi-Fi 7: Performance, Anomalies, and Solutions”. In: *2023 IEEE 34th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. IEEE. 2023, pp. 1–6.