Network Management, Optimization and Security with Machine Learning Applications in Wireless Networks

Mariam Nabil
mariam.aboelwafa@aucegypt.edu

The American University in Cairo
الجامعة الأمريكية بالقاهرة

Graduate Studies

Network Management, Optimization and Security with Machine Learning Applications in Wireless Networks

A Thesis Submitted by

Mariam Mohamed Nabil Aboelwafa

to the

Electronics and Communications Engineering Graduate Program

September 8, 2021

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electronics and Communications Engineering

Declaration of Authorship

I, Mariam Mohamed Nabil Aboelwafa, declare that this thesis titled, "Network Management, Optimization and Security with Machine Learning Applications in Wireless Networks" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: Mariam Nabil

Date: September 8th, 2021

Abstract

Wireless communication networks are emerging fast with a lot of challenges and ambitions. Requirements that are expected to be delivered by modern wireless networks are complex, multi-dimensional and sometime contradicting. In this thesis, we investigate several types of emerging wireless networks and tackle some challenges of these various networks. We focus on three main challenges. Those are Resource Optimization, Network Management, and Cyber Security. We present multiple views of these three aspects and propose solutions to probable scenarios. The first challenge (Resource Optimization) is studied in Wireless Powered Communication Networks (WPCNs). WPCNs are considered a very promising approach towards sustainable, self-sufficient wireless sensor networks. We consider a WPCN with Non-Orthogonal Multiple Access (NOMA) and study two decoding schemes aiming for optimizing the performance with and without interference cancellation. This leads to solving convex and non-convex optimization problems. The second challenge (Network Management) is studied for cellular networks and handled using Machine Learning (ML). Two scenarios are considered. First, we target energy conservation. We propose an ML-based approach to turn Multiple Input Multiple Output (MIMO) technology on/off depending on certain criteria. Turning off MIMO can save considerable energy of the total site consumption. To control enabling and disabling MIMO, a Neural Network (NN) based approach is used. It learns some network features and decides whether the site can achieve satisfactory performance with MIMO off or not. In the second scenario, we take a deeper look into the cellular network aiming for more control over the network features. We propose a Reinforcement Learning-based approach to control three features of the network (relative CIOs, transmission power, and MIMO feature). The proposed approach delivers a stable state of the cellular network and enables the network to self-heal after any change or disturbance in the surroundings. In the third challenge

(Cyber Security), we propose an NN-based approach with the target of detecting False Data Injection (FDI) in industrial data. FDI attacks corrupt sensor measurements to deceive the industrial platform. The proposed approach uses an Autoencoder (AE) for FDI detection. In addition, a Denoising AE (DAE) is used to clean the corrupted data for further processing.

Acknowledgements

I would like to give special thanks to my advisor, Prof. Karim Seddik. He is more of a big brother than a supervisor. He has always respected me and given me all the help I need. He is the most understanding supervisor that any student would wish to work with.

I would like to thank my mother for always being supportive. She has always been my guarding angel.

I would give the most gratitude to my husband, Mahmoud, for standing by me through out the whole journey and for believing in me.

I want also to thank my boys who beared the long tiring journey and the mode swings.

I want to dedicate my work to my dear late father who has been my backbone even after he left us.

Contents

List of Figures

List of Abbreviations

AE	**A**uto**E**ncoder
AP	**A**ccess **P**oint
BL	**B**ase**L**ine
CIO	**C**ell **I**ndividual **O**ffset
CQI	**C**hannel **Q**uality **I**ndicator
DAE	**D**enoising **A**uto**E**ncoder
DL	**D**own **L**ink
DQN	**D**eep **Q**-**N**etwork
DDQN	**D**ouble **D**eep **Q**-**N**etwork
EH	**E**nergy **H**arvesting
ER	**E**nergy **R**ich
FDI	**F**alse **D**ata **I**njection
IoT	**I**nternet **o**f **T**hings
IIoT	**I**ndustrial **I**nternet **o**f **T**hings
LCD	**L**ow **C**omplexity **D**ecoding
M2M	**M**achine **t**o **M**achine
Mbps	**M**ega **B**it **P**er **S**econd
MDP	**M**arkov **D**ecision **P**rocess
MIMO	**M**ultiple **I**nput **M**ultiple **O**utput
ML	**M**achine **L**earning
MLP	**M**ulti-**L**ayer **P**erceptron

NN	**N**eural **N**etwork
NOMA	**N**on-**O**rthogonal **M**ultiple **A**ccess
QoE	**Q**uality **o**f **E**xperience
QoS	**Q**uality **o**f **S**ervice
RF	**R**adio **F**requency
RL	**R**einforcement **L**earning
RNN	**R**ecurrent **N**eural **N**etwork
SICD	**S**uccessive **I**nterference **C**ancellation **D**ecoding
SISO	**S**ingle **I**nput **S**ingle **O**utput
SMux	**S**patial **M**ultiplexing
SVM	**S**upport **V**ector **M**achine
SWIPT	**S**imultaneous **W**ireless **I**nformation and **P**ower **T**ransfer
TD3	**T**win **D**elayed **D**eep **D**eterministic Policy Gradient
TDMA	**T**ime **D**evision **M**ultiple **A**ccess
TxD	**T**ransmit **D**iversity
UL	**U**p **L**ink
WET	**W**ireless **E**nergy **T**ransfer
WIT	**W**ireless **I**nformation **T**ransmission
WPCN	**W**ireless **P**owered **C**ommunication **N**etwork
WSN	**W**ireless **S**ensor **N**etwork

List of Symbols

$\mathbf{a}$	The fabricated data vector
α	The learning rate
$A_{aug}(t)$	Augmented action
$A_C(t)$	CIOs action
$A_M(t)$	MIMO feature on/off action
$A_P(t)$	Transmission power action
B_n	The bandwidth of the n^{th} eNB
$Bl(t)$	Resource Block Utilization (RBU)
B_{PRB}	The bandwidth of one PRB
$C(t)$	Number of active users of each cell in cellular network
d_{ij}	Inter-node distance in cellular network
d_{U_i-AP}	Distance between user and AP
d_{U_i-ER}	Distance between user and ER source
η	Hyper-parameter for blocked users penalty
f_c	Central Frequency
$g_{i,t}$	UL channel gain in WPCN
G_r	Receiver Antenna Gain
$h_{i,t}$	DL channel gain in WPCN
$K_{i,n}$	The number of Physical Resource Blocks that serve the i^{th} user in the n^{th} eNB
λ	Discount factor
m_n	Decision of turning MIMO feature ON/OFF for each eNB

$M(t)$	Modulation and Coding Scheme (MCS) Matrix
μ	Hyper-parameter for energy penalty
ϕ_u	The channel quality indicator (CQI) of the u^{th} UE
P_B	Average Transmit Power by ER in WPCN
π	Policy
P_{max}	Maximum eNB transmission power
P_{min}	Minimum eNB transmission power
P_n	eNB transmission power in cellular network
$r(t)$	Reward function
ρ_n	eNB utilization
σ^2	The noise power at the AP
S_i^{th}	SINR practical decoding threshold
$S_{aug}(t)$	Augmented state
$S(t)$	State of the RL environment
θ_{i-j}	The CIO value of the eNB i with respect to eNB j
θ_{j-i}	The CIO value of the eNB j with respect to eNB i
θ_{ij}	Relative CIO values between every two neighboring cells
θ_{max}	Maximum CIO value
$\tau_{0,t}$	Energy Harvesting time at time slot t in WPCN
$\boldsymbol{\tau}_0$	A vector whose elements are the harvesting time duration
v_u	Constant velocity of the u^{th} UE
$\mathbf{z}_a$	The recorded readings vector that may have some false data
$\mathbf{z}$	The clean measurements vector

Chapter 1

Introduction

Wireless communications have become a necessity in all fields of modern life. We need wireless communication in personal lives, education, industry, and luxury. Communication systems are everywhere around us. For instance, Wireless Sensor Networks (WSNs) are used in industry, surveillance, health care, smart homes, etc. Additionally, mobile networks are used for cell phones, smartphones, smart TVs, outdoor internet, etc. Moreover, WiFi is used to create small networks and indoor internet connection. From another perspective, Internet of Things (IoT) is used in home automation, transportation control, agriculture, industry, etc. Many more communication systems can be found serving all fields of life.

The applications of wireless communications are endless. However, a lot of challenges threaten the success of the wireless communication process between two nodes. From these challenges are the fading nature of the wireless channel, the undeterministic complexion of the Transmitter/Receiver connection, the interference from other nodes, and so on.

Moreover, as the applications of wireless communications grow bigger and more vital, new challenges emerge. For instance, energy conservation has become a global interest, and reducing the carbon footprint is of great necessity. This leads to more innovative ideas, such as Radio Frequency Energy Harvesting (RF EH) for WSNs or energy management in mobile networks. These ideas have created a whole new world of research use cases and optimization problems. One more challenge is the human control of wireless networks. This might have worked previously. However, with the proliferation of wireless communications applications and the intense data rates, coverage and connection requirements, human control will be too error-prone to fulfill the expected performance in near future. This leads to the necessity of self-managed networks.

Nevertheless, mobile networks have undergone a great evolution from voice-only services into complex, interconnected, multiple services providing high-speed access for a huge number of subscribers and machines [1]. The significant growth in wireless traffic has had a huge impact on mobile network architectures. Such architectures are required to deliver perfect connectivity among an increasing number of connected devices and extremely high data rates with reasonable cost, and optimized energy consumption [1–4].

From another perspective, the evolving IoT technologies have created absolute connectivity and turned common objects into connected devices (Machine to Machine (M2M)). This infrastructure has been very attractive not only to consumers but also to the industrial domain due to its flexibility and efficiency. With the IoT making its way through industrial applications, we find the emerging paradigm of Industrial IoT (IIoT). However, IIoT is facing many challenges regarding energy efficiency, real-time performance, and security[5].

One promising approach to solve the challenges of modern wireless communication systems is Machine Learning (ML). Most of the above challenges can not be dealt with using classical approaches. The scale of these challenges is huge and it is very hard to device good analytical models to capture systems' dynamics. To address these complex, multidimensional challenges, ML comes into rescue. ML is a design methodology that substitutes the human knowledge and expertise of the problem under consideration with a large number of training data. It is mainly a data-driven design methodology that can solve model-free problems. The main idea is to use historical data to train the machine to enable it to extract features from the data and take decisions based on the extracted information. ML can be very useful in communication systems in problems that lack a clear defined model. For instance, ML can be used in proactive resource allocation in WSNs or IoT [6], [7]. It can also be used for channel detection and decoding at the receiving end [8], [9]. Moreover, ML can be very useful in cognitive radio applications such as scheduling and channel status prediction [10], [11], [12]. Nevertheless, it can be used in content cashing to predict the content of high demand at certain times [13], [14]. Furthermore, the need for self-managed networks - as mentioned earlier- creates great opportunities for ML to deliver this requirement. Other applications can be found in literature as well [15], [16].

In this thesis, our objective is to study different types of networks/technologies (WPCNs, cellular networks and IIoT) to show how diverse wireless communication applications are. We present some approaches to tackle some of the challenges that wireless communications face in various recent applications. We tackle three main axes: Resource optimization, Network Management, and Security Requirements.

1.1 Resource Optimization Challenge

To study resource optimization, we consider a Wireless Powered Communication Network (WPCN) with Non-Orthogonal Multiple Access (NOMA) [17]. WPCNs depend on RF EH for energy management. NOMA provides a multiple access technique in the power domain, where all nodes use the same time slot and the same frequency band. RF EH occurs in a certain time slot, while data transmission of all nodes occurs in a separate time slot.

We aim for reaching an optimal time and power allocation for successful energy harvesting and data transmission. We target two different formulations; in the first one (max-sum), the resource allocation is optimized such that the sum-throughput of all users is maximized. In the second one (max-min), and targeting fairness among users, we consider resource optimization aiming for maximizing the min-throughput of all users. Under the above two formulations, two NOMA decoding schemes are studied, namely, Low Complexity Decoding (LCD) and Successive Interference Cancellation Decoding (SICD).

1.2 Network Management Challenge

To study this challenge, we focus on cellular networks. We use ML to gain control over the cellular network features. We target management of the network from two perspectives:

- Energy conservation by controlling the scheme of Multiple Input Multiple Output (MIMO) [18]. As mentioned earlier, energy conservation is a global direction towards a sustainable future. In cellular networks, one of the features that consumes a lot of energy is MIMO although it has a great influence on enhancing the network performance. We consider the problem of energy optimization in mobile networks by enabling the MIMO feature only when necessary. We employ ML-based approaches to decide on whether a Single Input Single Output (SISO) scheme can achieve the required Quality of Experience (QoE). If SISO can satisfy the target QoE, the base station can decide to switch the MIMO feature off which can result in considerable energy savings.

- Delivering a self-healing network by using Reinforcement Learning (RL) to gain control over the network features [19]. In this work, we propose a framework to fulfill the cellular network needs to sustain a balanced performance while facing continuous growth and endless changes. The motivation behind this is to switch from the manual management of the network which is costly, time-consuming and error-prone. We suggest a ML-based system that manages the cellular network to reach a stable state and gain more immunity towards everyday changes. The presented approach is a deep RL scheme that enables the network manager to learn a policy that delivers a stable state of the network such that the network sum throughput is maximized and the number of blocked users and the consumed energy are minimized. Moreover, the presented algorithm deals with hybrid action space in a layered fashion.

1.3 Security Challenge

As mentioned above, IIot faces cybersecurity challenges. In this thesis, a specific attack which is the False Data Injection (FDI) is the main focus [20]. The FDI attacks aim to mislead the industrial platforms by falsifying their sensor measurements. We present a novel method of FDI attack detection using Autoencoders (AEs). We exploit the sensor data correlation in time and space, which in turn can help identify the falsified data. Moreover, the falsified data are cleaned using the denoising AEs (DAEs).

1.4 Thesis Contribution

In this thesis, we study diverse types of wireless networks and try to solve various use-cases that represent major challenges in these networks/paradigms. The main contributions of this thesis are listed as follows:

- Presenting an approach for the allocation of time and power in WPCNs with NOMA such that the performance is optimized. Two decoding schemes are studied (LCD and SICD). Maximizing the sum-throughput of the network is characterized (max-sum) to find the optimal transmission durations and powers for both decoding schemes. The fairness aspect is also studied by characterizing the maximization of the minimum throughput of the network (max-min) for LCD and SICD [17].

- Applying ML-based approaches in cellular network management, specifically energy conservation. The target is saving the energy consumed by the MIMO feature if it is unnecessary. We create a Neural Network (NN) and train it using historical data drawn from realistic SISO cells. This training results in a network that can emulate the behavior of a SISO site. When the training phase is complete, the machine is subjected to MIMO site features and it emulates the performance of the SISO scheme to decide whether SISO can achieve a satisfactory QoE. If the machine decides that SISO is enough, then MIMO can be turned off to save energy [18].

- Applying an RL framework for cellular network management. The presented framework aims to control some network features. The target is learning a policy that maximizes the network sum throughput and delivers a stable state of the network. The features under control are relative CIOs, transmission powers (both are selected from a continuous action space) and MIMO feature on/off (selected from a discrete action space). Thus, the presented RL framework is a layered approach that enables the agent to take decisions extracted from a hybrid action space [19].

- Proposing the use of AEs as classifiers to detect false data injection attacks in

IIoT. AEs can learn hidden correlation structures in the data that enable them to detect attacks by assessing how far the correlation structure of the corrupted data is from the expected correlation structure. Our proposed scheme has the potential of detecting other types of attacks (not only FDI) without the need for any modifications to it. Additionally, we propose the use of DAEs to retrieve clean data from the corrupted data, by recovering the expected correlation structure [20].

1.5 Thesis Organization

- In Chapter 2, we present a brief technical background for the concepts, algorithms, and methodologies utilized in this thesis.
- In Chapter 3, we study the WPCN with NOMA and present an approach towards optimal resource allocation. We study two decoding schemes (LCD and SICD) that optimize the performance of the WPCN.
- In Chapter 4, we switch to cellular networks and present two ML-based approaches to control the MIMO scheme aiming for energy conservation. We use NN to emulate the behavior of cellular site to take the decision of turning MIMO on/off.
- Chapter 5 focuses also on cellular networks. We present a deep RL-based network management scheme. A deep RL framework is proposed to learn a policy that can control some features of the cellular network and a novel algorithm is proposed to handle hybrid action spaces in a hierarchical fashion.
- In Chapter 6, we migrate to the cybersecurity challenge in IIoT. An ML-based approach is presented, which can detect FDI attacks and recover the corrupt data.

We use Autoencoders (AEs) as classifiers based on the correlation among data to detect corrupt data.

- Chapter 7 draws the thesis conclusions and presents some directions for future work.

Chapter 2

Background

In this chapter, we present a brief background of the methodologies utilized in this thesis. In Chapter 4, we use two types of Neural Networks (NN), namely Multi-Layer Perceptron (MLP) and Recurrent Neural Network (RNN). An overview of these two NNs is presented in Section 2.1 and Section 2.2. In Chapter 5, we propose a Reinforcement Learning (RL) framework for cellular network management. Section 2.3 presents a brief explanation of RL. In Chapter 6, we use Autoencoders (AEs) and Denoising Autoencoders (DAEs) for cyber-attacks detection and data recovery, respectively. We also compare the performance of AEs to Support Vector Machines (SVMs). An overview of AEs, DAEs, and SVMs can be found in Sections 2.4, 2.5, and 2.6.

2.1 Multi-Layer Perceptron (MLP)

One of the Machine Learning (ML) mechanisms utilized in this work is the MLP. It is a type of feed-forward artificial NN [21]. As shown in Fig. 2.1, an MLP consists of at least three layers of nodes: an input layer, a hidden layer(s), and an output layer. Each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called back-propagation for training [22].

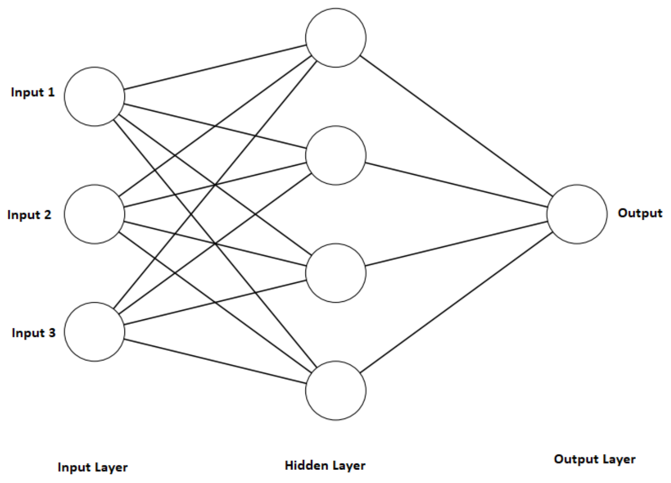

FIGURE 2.1: An Example of a Simple Multi-layer Perceptron

Generally, the output of the neural network can be represented as:

$$\mathbf{y} = h_{\boldsymbol{\theta}}(\mathbf{x}), \tag{2.1}$$

where $\mathbf{x} = [x_1 \ x_2 \ \cdots \ x_n]^T$ is the input vector, $h_{\boldsymbol{\theta}}$ is the feedforward propagation function and $\boldsymbol{\theta}$ is the set of weights and biases to be learned by the network during the training phase. The training phase is done by setting the output to be the target values provided by the training data. Therefore, for a certain $h_{\boldsymbol{\theta}}$, the only unknown in the equation is $\boldsymbol{\theta}$.

The steps of the learning process (back-propagation) can be summarized as follows:

1. Initialize weights and biases (usually random).

2. Use the feed-forward direction (from input to output) to calculate the output.

3. Calculate the error function (a measure of the difference between the actual output and the target output). This error is a function of all the contributing errors from all connected neurons.

4. Use the error function, in the backward direction, to update the weights and biases using the equation:

$$\Delta\theta_i = -\alpha * \frac{\delta E}{\delta\theta_i}, \tag{2.2}$$

where $\Delta\theta_i$ is the update of the i^{th} weight/bias, α is the learning rate and E is the cumulative error function from all the previous contributing layers.

5. Repeat until the output converges.

2.2 Recurrent Neural Network (RNN)

Recurrent Neural Networks (RNNs), as shown in Fig. 2.2, are distinguished from feedforward networks by a feedback connection to their past decisions, taking their own outputs moment after moment as inputs. In other words, RNNs have memory. Adding memory to NNs allows to track the information in the sequence itself, e.g. time correlations, trends, etc.

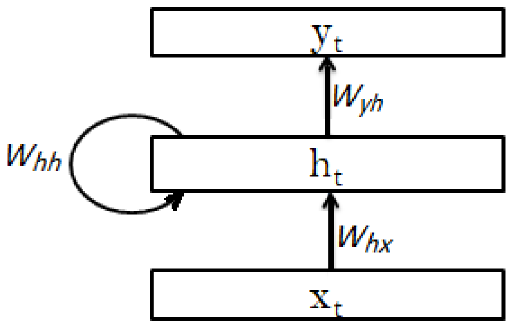

FIGURE 2.2: An Illustration of a Simple Recurrent Neural Network

Generally, the output of an RNN can be given as:

$$y_t = f(W_{yh} \cdot h_t), \tag{2.3}$$

where y_t is the output at time t, f is the activation function of the output layer, W_{yh} is the set of weights and biases between the hidden layer and the output layer, and h_t is the hidden layer output. This part is not different from the feed-forward NN. However, the output of the hidden layer can be given as:

$$h_t = g(W_{hx} \cdot x_t + W_{hh} \cdot h_{t-1}), \tag{2.4}$$

where h_t is the output of the hidden layer at time t, g is the activation function of the hidden layer, W_{hx} is the set of weights and biases between the input layer and the hidden layer, x_t is the input at time t, W_{hh} is the set of weights in the feedback loop of the hidden layer and h_{t-1} is the output of the hidden layer at time $t-1$.

The training phase is done in a similar way to the feedforward networks. However, the calculation of the cumulative error and its gradient is done differently. Details for the interested reader can be found in [23].

2.3 Reinforcement Learning (RL)

Reinforcement learning (RL) is a process in which an agent learns to make decisions (apply actions), observes the impact of its decisions on the surrounding environment, and adjusts its strategy -based on its observation- to achieve a certain goal in the long run (maximize a certain specified reward) [24]. RL can be modeled as a Markov decision process (MDP) in which, at each time step $t = 0, 1, 2, ...$, the agent receives some representation of the environment (a state, $S(t)$, that belongs to a state space, $\boldsymbol{S}$). Depending on that state, the agent selects an action, ($A(t)$ from a predetermined set of actions, $\boldsymbol{A}$). Next, the agent receives the consequence of its action. That is: a numerical reward $r(t+1)$ and a new state $S(t+1)$ [25]. MDP formulation is suitable for the tasks in which the agent knows the long-run goal but doesn't know how exactly to reach

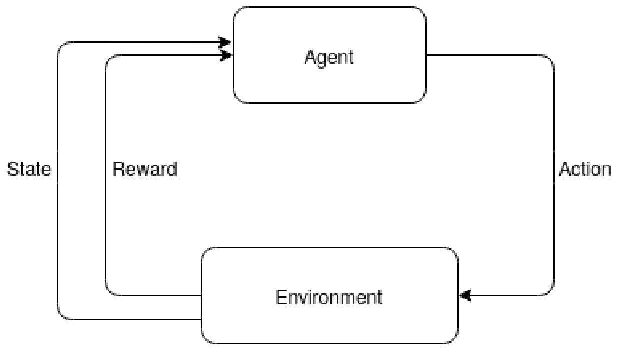

FIGURE 2.3: The Process of Reinforcement Learning

that goal (goal-oriented tasks). Furthermore, the agent doesn't have training data. The process is illustrated in Fig. 2.3.

We can say that the objective of the decision-maker (agent) is to reach the sequence of actions (policy) that maximizes the expected reward function eventually [26]. This *objective* can be characterized as:

$$\max_{\pi} \lim_{L \to \infty} \sum_{t=0}^{L} E[\lambda^t r(t)], \tag{2.5}$$

where, $r(t)$ is the reward function, λ is the discount factor that determines the significance of the reward future expected values, and π is the policy to be learned.

The nature of the action space can be:

- Discrete Action Space: Actions are selected from a finite countable set of actions.
- Continuous Action Space: Actions are selected from a bounded interval.
- Hybrid Action Space: Some actions are taken from continuous space while the others are taken from discrete space.

To solve the aforementioned MDP using RL, there are several variants of RL techniques [25]. There is the basic Q-learning technique that works with a discrete set of actions. Here, the agent constructs a table of the Q-values corresponding to each state-action mapping pair. The Q-value can be defined as a measure for the quality of the decision at a certain state. These Q-values are updated based on the interaction with the environment. There is also the Deep Q-Network (DQN) technique in which a deep neural network can be used to approximate the Q-values. The Double-DQN (DDQN) technique is an extension of DQN in which two different neural networks are implemented for action selection and action evaluation [27]. More techniques exist in literature as well.

For a continuous set of actions, various RL techniques can be used. One way is to use Soft-Actor-Critic methods which are explained in detail in [28]. Another way is to use Policy Gradient methods (and their extensions like TD3) which can be understood from [29].

For hybrid action spaces, there are some parametrized approaches in literature to handle this mixed type of action spaces [19, 30–32].

2.4 Autoencoder (AE)

An AE is a fully connected, unsupervised NN algorithm that applies back-propagation.

As can be noticed from Fig. 2.4, an AE consists of an input layer, one or more hidden layers, and one output layer. Generally, by letting bold letters represent vectors, the output of the neural network is represented as:

$$\mathbf{y} = h_{\boldsymbol{\theta}}(\mathbf{x}), \tag{2.6}$$

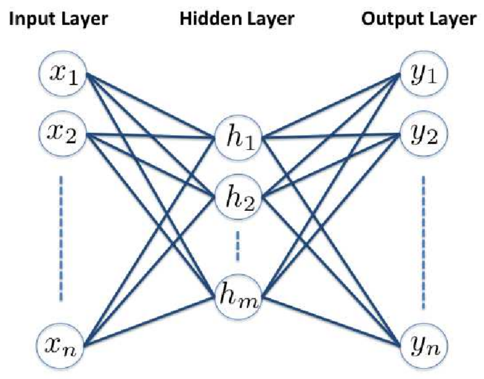

FIGURE 2.4: Architecture of an Autoencoder with One Hidden Layer

where $\mathbf{x} = [x_1 \; x_2 \; \cdots \; x_n]^T$ is the input vector, $h_{\boldsymbol{\theta}}$ is the full-forward propagation function and $\boldsymbol{\theta}$ is the set of weights and biases to be learned by the network during the training phase. This learning process is done by setting the target values at the output to be the same as the input values (i.e., $\mathbf{y} = \mathbf{x}$):

$$h_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{x}. \tag{2.7}$$

AE works as an unsupervised learning algorithm that applies back-propagation. *Unsupervised learning* means that during the training phase, only raw features are required with no labeling. Meanwhile, *back-propagation* means that the error between the predicted output and the target output propagates backward from the output to each neuron in the network to update the weights based on some learning rate (α):

$$\theta_{i+1} = \theta_i + \Delta\theta_i, \tag{2.8}$$

where i is the index of the training epochs and $\Delta\theta_i$ is the weights update, and is calculated as

$$\Delta\theta_i = -\alpha\frac{\partial E}{\partial \theta_i}, \tag{2.9}$$

and E is the cost function.

AEs can *learn* the structure of the data and the relationship among entries. Originally, they were used as a data-compression model. They encode a given input into a representation of smaller dimension. A decoder is then used to reconstruct the input back from the encoded compressed version. The main idea behind the compression and decompression capabilities of AEs is the fact that AEs can exploit the correlation between data entries. The more the correlated data entries are, the more compressible they become. To achieve this compression/decompression, the number of neurons in the hidden layer is constrained to be less than the number of neurons in the input/output layer. The output of the hidden layer is the *compressed* (encoded) version of the input, while the final output is the *retrieved* version.

Correlation among data entries is not only used for data compression but it can also be used for the detection of cyber-attacks that would disrupt the correlation model of the data entries and hence AEs will declare an attack.

2.5 Denoising Autoencoder (DAE)

Another type of AEs is the *Denoising Autoencoder (DAE)*. As introduced in [33], it has the same architecture as AEs. However, the principle behind DAEs is to be able to reconstruct data from an input of corrupted data. We train a DAE by corrupting the data sets and feeding them into the neural network. During the training phase, the target values at the output are set to be the original data, while inputs are the corrupted version of the data:

$$\text{target output of } h_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}) = \mathbf{x}, \tag{2.10}$$

where $\tilde{\mathbf{x}}$ is the corrupted set of input readings and $h_{\boldsymbol{\theta}}(\tilde{\mathbf{x}})$ is the DAE output. DAE minimizes the cost function $E(\mathbf{x}, h_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}))$, where $E(.,.)$ is some error measure. A DAE must undo the corruption rather than simply replicating their input at the output, and in doing so it captures only the most significant features of the training data. This training enables the DAE to recover the correlation among input readings.

2.6 Support Vector Machine (SVM)

An SVM is a classification technique that defines decision planes to separate data points from different classes. This can be visualized in Fig. 2.5 which shows an illustration of a 2-D example (for simplicity of illustration). The objective of the SVM is to maximize the width of the "*street*" separating the two classes. H_0 represents the boundary (hyperplane) that divides the street into two halves; H_1 and H_2 are the two planes (parallel to H_0) that touch the nearest points from each class to the boundary. These planes are given by

$$\begin{aligned} H_0 &: \mathbf{w}^T\mathbf{x} + b = 0, \\ H_1 &: \mathbf{w}^T\mathbf{x} + b = 1, \\ H_2 &: \mathbf{w}^T\mathbf{x} + b = -1, \end{aligned} \tag{2.11}$$

where $\mathbf{w}$ is the weight vector, $\mathbf{x}$ is input vector and b is the bias. The distance between the planes H_0 and H_1 is given by $|\mathbf{w} \cdot \mathbf{x}|/||\mathbf{w}||= \frac{1}{||\mathbf{w}||}$. Hence, the distance between H_1 and H_2 is $\frac{2}{||\mathbf{w}||}$.

To maximize the margin, we, therefore, need to minimize $||\mathbf{w}||$, with the condition that there are no data points between the planes H_1 and H_2. This results in a constrained

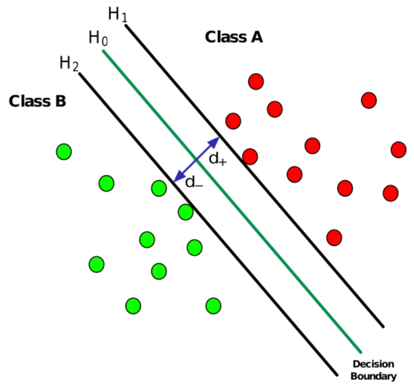

FIGURE 2.5: Support Vector Machine Illustration

optimization problem that can be formulated as

$$\begin{gathered}\min_{\mathbf{w}} \quad \frac{1}{2}||\mathbf{w}||^2 \\ \text{subject to :} \\ y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geq 0,\end{gathered} \tag{2.12}$$

where y_i is the label of each data point $\mathbf{x}_i$ (y_i is +1 for one class and -1 for the other class) and i is the data point iterator. Note that the constraint can be split into:

$$\begin{aligned}\mathbf{w} \cdot \mathbf{x}_i + b \geq +1 \text{ when } y_i = +1 \ \ \text{(class A)} \\ \mathbf{w} \cdot \mathbf{x}_i + b \leq -1 \text{ when } y_i = -1 \ \ \text{(class B).}\end{aligned} \tag{2.13}$$

The above optimization problem is a quadratic convex optimization problem with linear constraints, which can be solved using any quadratic programming solvers [34, Chapter 4].

SVMs can be extended to achieve a nonlinear classification by applying a kernel to the input data, mapping the input to some high-dimension feature space.

As opposed to AEs, SVMs are supervised learning-based algorithms. This means that the label of every point must be provided. Given a set of training points, each belonging to a class, an SVM training algorithm builds a non-probabilistic model that classifies new data points to one of the two classes. Based on the optimization problem described above, an SVM model represents data entries as points in a high-dimension space and finds a hyperplane to separate the classes such that the distance between the nearest data points belonging to each class to the decision boundary is maximized.

Chapter 3

Towards Optimal Resource Allocation in Wireless Powered Communication Networks with Non-Orthogonal Multiple Access

In this chapter, we study the resource optimization challenge in wireless networks. Here, the optimal allocation of time and energy resources is characterized in a Wireless Powered Communication Network (WPCN) with Non-Orthogonal Multiple Access (NOMA).

In WPCN, the users harvest wireless energy from a dedicated Energy Rich (ER) source in the Downlink (DL), and then use it in the Uplink (UL) to send data to the Access Point (AP) [35]. This work considers the ER and AP as separate entities to accommodate a general setting, in contrast to a large body of the literature where they coincide. Time and power allocations are jointly optimized over a finite horizon of $T > 1$ slots. NOMA exploits an approach of user multiplexing in the power domain [36], [37].

The main contributions of the work in this chapter [17] can be summarized as follows:

- Two decoding schemes are studied, namely: Low Complexity Decoding (LCD) and Successive Interference Cancellation Decoding (SICD). The two schemes aim at optimizing the performance of a WPCN with and without interference cancellation.
- Maximizing the sum throughput of the network is studied (max-sum). Since LCD leads to a non-convex problem, an iterative approach is introduced to solve two sub-problems in an alternating manner. On the other hand, the convexity of the max-sum problem with SICD is established and the problem is characterized to find the optimal transmission durations and powers.
- The fairness aspect is also studied and the optimization problem to maximize the minimum throughput of the network (for LCD and SICD) is characterized (max-min). Again, the problem is shown to be non-convex and an iterative algorithm is introduced to find an approximate solution that is close to the global optimum.

3.1 Literature Review

There has been a growing interest, recently, in studying new technologies for prolonging the lifetime of mobile devices [38]. RF Energy Harvesting (EH) is considered a promising solution towards an unlimited power supply for wireless networks. However, it adds more complexity to system design and optimization [39], [40].

There are two main paradigms in RF EH [41]; Simultaneous Wireless Information and Power Transfer (SWIPT) and Wireless Powered Communication Networks (WPCN).

In SWIPT, Wireless Energy Transfer (WET) and Wireless Information Transmission (WIT) occur simultaneously, in which energy and information are transmitted in the same signal [42]. In [43], Boshkovska designs a resource allocation algorithm for SWIPT systems. The algorithm design is formulated as a non-convex optimization problem for the maximization of the total harvested power at the EH receivers subject to the Quality of Service (QoS) constraints. In [44], Ng and Schober study a resource allocation algorithm design for secure information and energy transfer to mobile receivers. In [45], multiple source-destination pairs communicate through their dedicated energy harvesting relays. A power splitting framework using game theory was developed to derive a profile of relays' power splitting ratios. Additionally, to overcome the problem that energy harvesting circuits are unable to harvest energy and decode information simultaneously, there are two proposed receiver designs in [46]: time switching and power splitting. By using the time switching setting, the receiving antenna periodically switches between energy harvesting and information decoding phases. On the other hand, under the power splitting, the received signal is split into two streams; one for the energy harvesting circuitry and the other is for information decoding. The application of SWIPT to Non-Orthogonal Multiple Access (NOMA) networks is investigated in [47], where Liu, et al. propose a new cooperative SWIPT-NOMA protocol in which users close to the source act as relays for far users' transmission.

WPCN has been studied in various works. A cooperative technique was studied in [48] and [49] to overcome the doubly near-far phenomenon. A WPCN with heterogeneous nodes (nodes with and without energy harvesting capabilities) was studied in [50] and it was shown how the presence of non-harvesting nodes can enhance the sum throughput. [51] departed from the strong assumption adopted in [35] - [50], where the energy harvested in a slot is used completely in that slot, and, hence, embraces a long-term optimization framework. Additionally, [52] extended the long-term maximization of the half-duplex case in [51] to the full-duplex scenario. Conventional TDMA wireless networks were generalized in [53] to a new type of wireless networks: generalized-WPCNs (g-WPCNs), where nodes are equipped with RF energy harvesting circuitries along with energy supplies. It was shown that both conventional TDMA wireless networks and WPCNs with only RF energy harvesting nodes constitute lower bounds on the performance of g-WPCNs in terms of the max-sum throughput and max-min throughput.

NOMA was introduced in WPCNs in [54] to enhance the power-bandwidth efficiency. It was shown in [55] that NOMA improves the spectral efficiency relative to orthogonal multiple access schemes. In [54], optimizing the time allocations was the main concern to maximize the sum throughput of the slot-oriented case (all the harvested energy in a slot is also consumed in the same slot). Hence, Diamantoulakis, et al. introduced a sub-optimal policy for time allocations. Yuan and Ding investigated, in [56], the application of NOMA for the uplink (UL) of WPCNs. They maximize the sum rate by jointly designing the time allocation, the downlink (DL) energy beamforming, and the receiver beamforming. In [57], two NOMA-based decoding schemes were introduced to maximize the sum throughput of the network. Due to the difficulty of solving the optimization problem, an approximate iterative approach was proposed to solve a sub-problem and reach a sub-optimal solution. Chingoska and Nikoloska tackled, in [58], the doubly near-far effect in WPCNs by setting the decoding order signals received at the base station (BS) to be the inverse of the distances between the users and the BS.

3.2 System Model

Consider a WPCN composed of one AP, one ER node, and K users[1] $U_i, i = 1, 2, ..., K$ (see Figure 3.1). All nodes in the network are equipped with single antennas and operate over the same frequency. Only the ER node is equipped with a constant energy supply. It broadcasts DL wireless energy to the K users in the network. Users receive energy and use the accumulated energy to send UL data to the AP.

Slotted time is considered and the slot duration is assumed to be normalized to one (without loss of generality). Each slot $t = 1, 2, ..., T$ is divided into two phases: $\tau_{0,t}$ during which the ER broadcasts wireless energy on the DL to recharge the batteries of the devices and $1 - \tau_{0,t}$ during which all users transmit data to the AP independently and simultaneously over the UL. Note that all radios are half-duplex.

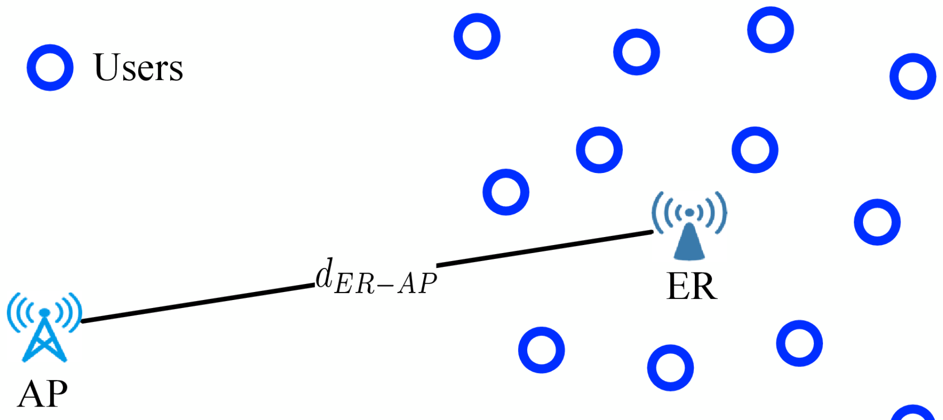

FIGURE 3.1: System Model

Users are randomly distributed around ER. d_{U_i-ER} is the distance between U_i and ER, and d_{U_i-AP} is the distance between U_i and AP. Locations of all nodes are assumed to be known a priori, and therefore their average channel gains can be estimated. The DL channel power gain from ER to U_i and the UL channel power gain from U_i to AP,

[1]The symbol that represents the number of nodes in a wireless network might be different from one chapter to another.

during time slot t, are denoted by $h_{i,t}$ and $g_{i,t}$, respectively. Hence, the harvested energy by U_i in the DL phase can be expressed as [57]:

$$E_{i,t} = \eta_i h_{i,t} P_B \tau_{0,t} = \gamma_{i,t} \tau_{0,t}, \tag{3.1}$$

where η_i denotes the energy harvesting circuitry efficiency [59], P_B is the average transmit power by ER within $\tau_{0,t}$ and $\gamma_{i,t} \overset{\text{def}}{=} \eta_i h_{i,t} P_B$.

3.3 Max-Sum Throughput Optimization

In this section, the time and power allocation constrained optimization problem is formulated such that the sum UL throughput of the network is maximized. Two NOMA decoding schemes are studied, namely, LCD and SICD. The main objective is to formulate the problem of maximizing the achievable sum throughput over a finite horizon of T time slots subject to some constraints.

3.3.1 Max-Sum Problem Formulation with Low Complexity Decoding

Under the LCD scheme, the AP uses a single-user decoder to detect the signals received from all users without performing interference cancellation. Each signal suffers from interference from all other users. Thus, interference from other users is treated essentially as noise.

The achievable throughput of user U_i, in time slot t, can be expressed as:

$$R_{i,t} = (1 - \tau_{0,t}) \log_2(1 + x_{i,t}), \tag{3.2}$$

where $x_{i,t}$ is the average SINR at the AP for U_i in time slot t.

In LCD, $x_{i,t}$ is given by:

$$x_{i,t} = \frac{g_{i,t} E_{i,t}}{\sigma^2 (1 - \tau_{0,t}) + \sum_{j=1, j \neq i}^{K} g_{j,t} E_{j,t}}, \tag{3.3}$$

where $E_{i,t}$ is the amount of energy used by U_i in time slot t and σ^2 is the noise power at the AP. It is worth noting that, under LCD, the interference term (i.e., the summation) in the denominator includes interference from all users other than user U_i.

The sum throughput maximization problem can be formulated as:

$$\begin{aligned}
\mathbf{P1_{LCD}}: \quad & \max_{\boldsymbol{\tau}_0,\mathbf{E},\mathbf{x}} \quad \sum_{t=1}^{T}\sum_{i=1}^{K} R_{i,t}, \\
\text{subject to:} \quad & Eq.(3.3), \\
& \sum_{n=1}^{t} E_{i,n} \leq \sum_{n=1}^{t} \gamma_{i,n}\tau_{0,n}, \quad \forall t \quad \text{(energy causality constraints)}, \\
& x_{i,t} \geq S_i^{th}, \quad \forall i, \forall t \quad \text{(decoding constraints)}, \\
& 0 \leq \tau_{0,t} \leq 1, \quad \forall t, \\
& E_{i,t} \geq 0 \quad \forall i, \forall t,
\end{aligned}$$

where $\boldsymbol{\tau}_0$, $\mathbf{E}$, and $\mathbf{x}$ are vectors whose elements are the harvesting time duration, the consumed energy by each user, and the average SINR at the AP for each user over the finite horizon of T slots, respectively. The role of the energy causality constraints is to guarantee that, in slot t, only the energy harvested in slots $\leq t$ can be used. The decoding constraints highlight the fact that if the SINR $x_{i,t}$ falls under a predefined threshold S_i^{th}, decoding will not be possible.

The objective function (sum throughput) of $\mathbf{P1_{LCD}}$ is non-convex because it has a Hessian matrix that is not positive semidefinite. Hence, the problem is a non-convex optimization problem [60]. However, we exploit the problem structure and propose an efficient approach for solving this rather complex problem via splitting it into two separate subproblems, $\mathbf{P1_{LCD}}$(given $\boldsymbol{\tau}_0$) and $\mathbf{P1_{LCD}}$(given $\mathbf{E}$), that can be solved iteratively. The first is for a given harvesting slot duration $\boldsymbol{\tau}_0$ and the latter is for a given energy allocation vector $\mathbf{E}$. The proposed solution in this section is to solve these two sub-problems iteratively, alternating between a given $\boldsymbol{\tau}_0$ to get optimum energy allocation vector for the first sub-problem and then, for the obtained energy allocation vector, get the optimum harvesting slot duration in the second sub-problem, and so on alternating back and forth between the two sub-problems until convergence is attained.

The First Sub-problem:

The first sub-problem given $\boldsymbol{\tau}_0$ is formulated as:

$$
\begin{aligned}
\mathbf{P1_{LCD}}(\text{given } \boldsymbol{\tau}_0): \quad & \max_{\mathbf{E},\mathbf{x}} \quad \sum_{t=1}^{T}\sum_{i=1}^{K}(1-\tau_{0,t})\log_2(1+x_{i,t}), \\
\text{subject to:} \quad & Eq.(3.3), \\
& \sum_{n=1}^{t} E_{i,n} \le \sum_{n=1}^{t} \gamma_{i,n}\tau_{0,n}, \quad \forall t \quad \text{(energy causality constraints)}, \\
& x_{i,t} \ge S_i^{th}, \quad \forall i, \forall t \quad \text{(decoding constraints)}, \\
& E_{i,t} \ge 0 \quad \forall i, \forall t.
\end{aligned}
$$

The objective function can be expressed as:

$$\sum_{t=1}^{T}\sum_{i=1}^{K}\log_2(1+x_{i,t})^{(1-\tau_{0,t})}. \tag{3.4}$$

Maximizing (3.4) is equivalent to maximizing the expression:

$$\prod_{t=1}^{T}\prod_{i=1}^{K}(1+x_{i,t})^{(1-\tau_{0,t})}, \tag{3.5}$$

or to minimizing the expression:

$$\frac{1}{\prod_{t=1}^{T}\prod_{i=1}^{K}(1+x_{i,t})^{(1-\tau_{0,t})}}. \tag{3.6}$$

.

Hence, the first sub-problem can now be written as:

$$
\begin{aligned}
\mathbf{P1_{LCD}}(\text{given } \boldsymbol{\tau}_0): \quad & \min_{\mathbf{E},\mathbf{x}} \quad \frac{1}{\prod_{t=1}^{T}\prod_{i=1}^{K}(1+x_{i,t})^{(1-\tau_{0,t})}}, \\
\text{subject to:} \quad & x_{i,t} \times \frac{\sigma^2(1-\tau_{0,t}) + \sum_{j=1,j\neq i}^{K} g_{j,t}E_{j,t}}{g_{i,t}E_{i,t}} \leq 1, \quad \forall i, \forall t, \\
& \frac{\sum_{n=1}^{t} E_{i,n}}{\sum_{n=1}^{t} \gamma_{i,n}\tau_{0,n}} \leq 1, \quad \forall i, \forall t, \\
& S_i^{th} x_{i,t}^{-1} \leq 1, \quad \forall i, \forall t, \\
& E_{i,t} \geq 0 \quad \forall i, \forall t.
\end{aligned}
$$

All constraints are expressed in the standard form for a Geometric Program (GP) except the last one ($E_{i,t} \geq 0 \quad \forall i, \forall t$). The objective function of $\mathbf{P1_{LCD}}(\text{given } \boldsymbol{\tau}_0)$ is a ratio between two posynomials, thus, $\mathbf{P1_{LCD}}(\text{given } \boldsymbol{\tau}_0)$ is a nonconvex complementary GP [60]. Solving complementary GPs directly is NP-hard. Therefore, an approximate approach is used [57]. The denominator of the objective function, denoted by *f(x)*, is approximated with a monomial function $\tilde{f}(x)$. In this case, the new approximate optimization problem becomes a standard GP that can be solved iteratively using standard techniques [34, chapter 4].

$\tilde{f}(x)$ is chosen to be:

$$\tilde{f}(x) = c\prod_{t=1}^{T}\prod_{i=1}^{K}(x_{i,t})^{y_{i,t}(1-\tau_{0,t})}, \tag{3.7}$$

where,

$$c = \frac{\prod_{t=1}^{T}\prod_{i=1}^{K}(1+\overline{x}_{i,t})^{y_{i,t}(1-\tau_{0,t})}}{\prod_{t=1}^{T}\prod_{i=1}^{K}(\overline{x}_{i,t})^{y_{i,t}(1-\tau_{0,t})}}, \tag{3.8}$$

$\overline{x}$ is the solution of the approximate GP in the previous iteration, and

$$y_{i,t} = \frac{\overline{x}_{i,t}}{1+\overline{x}_{i,t}}, \quad \forall i, \forall t. \tag{3.9}$$

Starting with an initial $\overline{x}$, we can obtain c and $y_{i,t}$ from (3.8) and (3.9), respectively. With these values, we solve the approximate geometric program. The obtained solution can be used to get new values of c and $y_{i,t}$. The procedure is repeated until the sum throughput converges to a pre-specified accuracy. It is worth mentioning that the used approximation satisfies the Karush-Kuhn-Tucker (KKT) conditions of $\mathbf{P1_{LCD}}(\text{given } \boldsymbol{\tau}_0)$ and thus is guaranteed to be a local optimal solution for $\mathbf{P1_{LCD}}(\text{given } \boldsymbol{\tau}_0)$. Moreover, satisfying the KKT conditions guarantees that $x_{i,t} \geq 0$ and this directly means that $E_{i,t} \geq 0$ (last constraint) [57].

The Second Sub-problem:

The second sub-problem given $\mathbf{E}$ is formulated as:

$$\begin{aligned}
\mathbf{P1_{LCD}}(\text{given } \mathbf{E}) : \qquad & \max_{\boldsymbol{\tau}_0, \mathbf{x}} \quad \sum_{t=1}^{T} \sum_{i=1}^{K} (1 - \tau_{0,t}) \log_2(1 + x_{i,t}), \\
\text{subject to:} \quad & Eq.(3.3), \\
& \sum_{n=1}^{t} E_{i,n} \leq \sum_{n=1}^{t} \gamma_{i,n} \tau_{0,n}, \quad \forall t \quad \text{(energy causality constraints)}, \\
& x_{i,t} \geq S_i^{th}, \quad \forall i, \forall t \quad \text{(decoding constraints)}, \\
& 0 \leq \tau_{0,t} \leq 1, \quad \forall t.
\end{aligned}$$

Note that, by using (3.3), $x_{i,t}$ can be substituted and removed from the problem. All constraints are affine and it can be verified that the objective function has a Hessian matrix that is **positive semidefinite**. Therefore, the problem $\mathbf{P1_{LCD}}(\text{given } \mathbf{E})$ is convex and, hence, can be solved using standard convex optimization tools.

The algorithm to solve $\mathbf{P1_{LCD}}$ iteratively, using the two sub-problems discussed above, is given in Algorithm 1.

Algorithm 1 Solving $\mathbf{P1_{LCD}}$

```
1:  repeat
2:      procedure SOLVE P1(GIVEN $\boldsymbol{\tau}_0$)
3:          Initialize $\overline{x}$
4:          Compute $c$ and $y_{i,t}$ using (3.8) and (3.9)
5:          repeat
6:              Solve the approximate P1(given $\boldsymbol{\tau}_0$)
7:              Update $c$ and $y_{i,t}$ using (3.8) and (3.9)
8:          until Sum throughput converges
9:          Find sub-optimum $R_{sum}$ and $\mathbf{E}$.
10:     end procedure
11:     procedure SOLVE P1(GIVEN $\mathbf{E}$)
12:         Use standard convex optimization solver: SDPT3.
13:         Find sub-optimum $R_{sum}$ and $\boldsymbol{\tau}_0$.
14:     end procedure
15: until The maximized sum converges to a pre-specified accuracy
```

3.3.2 Max-Sum Problem Formulation with Successive Interference Cancellation Decoding

The LCD scheme, despite its highly complex computations (since it yields a non-convex optimization problem), has a simple implementation because interference is simply treated as noise. However, the performance is modest due to ignoring the structure of interference. This leads to a lower sum throughput because each user suffers from interference from all other users. This motivates us to look at more sophisticated interference cancellation techniques to enhance the sum throughput performance. In this sub-section, a Successive Interference Cancellation Decoding (SICD) scheme is introduced whereby interference can be partially canceled out. The associated sum throughput optimization problem is also formulated.

The SINR of U_i in time slot t, after interference cancellation, can be expressed as:

$$x_{i,t} = \frac{g_{i,t}E_{i,t}}{\sigma^2(1-\tau_{0,t}) + \sum_{j=i+1}^{K} g_{j,t}E_{j,t}}, \quad \forall i. \tag{3.10}$$

The ability of the SICD scheme to better capture interference compared to the LCD scheme hinges on its ability to remove the interference from the already decoded users' signals on the remaining signals. Therefore, in (10), the interference term in the denominator includes interference from all users that will be decoded after the signal of user U_i. As a result, the first user's signal to be decoded will still suffer from interference from all other users' signals. However, successively, this will be improved until decoding the last user signal, which will not suffer from any interference from any other user (as all interference should have been successively canceled at this stage). The achievable throughput for U_i can be expressed as in (3.2) while substituting for the SINR $x_{i,t}$, from (3.10). Therefore, the achievable sum throughput over all K users under SICD, which is independent of the users decoding order, can be derived as follows:

$$\begin{aligned}
R_{sum} &= (1-\tau_0)\Bigg(\sum_{i=1}^{K-1}\log_2\left(1+\frac{g_iP_i}{\sigma^2+\sum_{j=i+1}^{K} g_jP_j}\right)+\log_2\left(1+\frac{g_KP_K}{\sigma^2}\right)\Bigg)\\
&= (1-\tau_0)\Bigg(\sum_{i=1}^{K-1}\log_2\left(\frac{\sigma^2+\sum_{j=i}^{K} g_iP_i}{\sigma^2+\sum_{j=i+1}^{K} g_jP_j}\right)+\log_2\left(\frac{\sigma^2+g_KP_K}{\sigma^2}\right)\Bigg)\\
&= (1-\tau_0)\log_2\left(\prod_{i=1}^{K-1}\frac{\sigma^2+\sum_{j=i}^{K} g_iP_i}{\sigma^2+\sum_{j=i+1}^{K} g_jP_j}\times\frac{\sigma^2+g_KP_K}{\sigma^2}\right)
\end{aligned}$$

$$
\begin{aligned}
&= (1-\tau_0)\log_2\left(\frac{\sigma^2+\sum_{j=1}^{K} g_iP_i}{\sigma^2+\sum_{j=2}^{K} g_jP_j}\times\frac{\sigma^2+\sum_{j=2}^{K} g_iP_i}{\sigma^2+\sum_{j=3}^{K} g_jP_j}\times\ldots\times\frac{\sigma^2+\sum_{j=K-1}^{K} g_iP_i}{\sigma^2+g_KP_K}\times\frac{\sigma^2+g_KP_K}{\sigma^2}\right)\\
&= (1-\tau_0)\log_2\left(\frac{\sigma^2+\sum_{j=1}^{K} g_iP_i}{\sigma^2}\right)\\
&= (1-\tau_0)\log_2\left(1+\frac{\sum_{j=1}^{K} g_iP_i}{\sigma^2}\right)\\
&= (1-\tau_0)\log_2\left(1+\frac{\sum_{j=1}^{K} g_iE_i}{\sigma^2(1-\tau_0)}\right).
\end{aligned}
$$

The SICD sum throughput optimization problem can then be formulated as:

$$
\begin{aligned}
\mathbf{P1_{SICD}}: \quad & \max_{\boldsymbol{\tau}_0,\mathbf{E},\mathbf{x}} \quad \sum_{t=1}^{T}\sum_{i=1}^{K} R_{i,t},\\
\text{subject to:} \quad & Eq.(3.10),\\
& \sum_{n=1}^{t} E_{i,n} \le \sum_{n=1}^{t} \gamma_{i,n}\tau_{0,n}, \quad \forall t \quad \text{(energy causality constraints)},\\
& x_{i,t} \ge S_i^{th}, \quad \forall i, \forall t \quad \text{(decoding constraints)},\\
& 0 \le \tau_{0,t} \le 1, \quad \forall t,\\
& E_{i,t} \ge 0 \quad \forall i, \forall t.
\end{aligned}
$$

Recall that for a general function, $f(x)$, which is concave, its perspective function $g(x,t) = tf(\frac{x}{t})$ would also be concave [34]. By using $t = 1-\tau_0^{(t)}$, the sum throughput $R_{sum}^{(t)}$ is the perspective function of the concave function $\log_2\left(1+\frac{\sum_{i=1}^{K} g_i^{(t)}E_i^{(t)}}{\sigma^2}\right)$. Therefore, $R_{sum}^{(t)}$ is a concave function in $[\tau_0^{(t)}, E_1^{(t)}, ..., E_K^{(t)}]$. Note that a non-negative weighted sum of concave functions is also concave, then the objective function of $\mathbf{P1_{SICD}}$ which is the non-negative weighted summation of $R_{sum}^{(t)},\ \forall t$, is a concave function in $(\boldsymbol{\tau}_0, \mathbf{E})$. In addition, all constraints of $\mathbf{P1_{SICD}}$ are affine in $(\boldsymbol{\tau}_0, \mathbf{E})$. As a result, $\mathbf{P1_{SICD}}$ is a convex optimization problem, and, hence can be solved efficiently using standard convex optimization tools.

An efficient, yet simple, way to solve a constrained optimization problem, is to find its Lagrangian and solve the dual problem. The Lagrangian of $\mathbf{P1_{SICD}}$ is given by:

$$\mathcal{L}(\mathbf{E}, \boldsymbol{\tau}_0, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \sum_{t=1}^{T}\sum_{i=1}^{K} R_{i,t} + \sum_{i=1}^{K}\sum_{n=1}^{T} \lambda_{i,n}\left(\sum_{t=1}^{n}(\gamma_{i,t}\tau_{0,t} - E_{i,t})\right) + \sum_{i=1}^{K}\sum_{t=1}^{T} \mu_{i,t}(x_{i,t} - S_i^{th}),$$

where $\lambda_{i,t}$, and $\mu_{i,t}$ are the dual variables associated with the energy causality and practical decoding constraints. Now, we need to solve the following optimization problem, namely $\mathbf{D}_{SICD}$, to get the dual function, denoted by, $G(\boldsymbol{\lambda}, \boldsymbol{\mu})$:

$$\begin{aligned} \mathbf{D}_{SICD}: \quad & \max_{\boldsymbol{\tau}_0, \mathbf{E}} \quad \mathcal{L}(\mathbf{E}, \boldsymbol{\tau}_0, \boldsymbol{\lambda}, \boldsymbol{\mu}), \\ \text{subject to:} \quad & 0 \le \tau_{0,t} \le 1, \quad \forall t, \\ & E_{i,t} \ge 0 \quad \forall i, \forall t. \end{aligned}$$

Consequently, the dual problem will be: $\min\limits_{\boldsymbol{\lambda}, \boldsymbol{\mu} \ge 0} G(\boldsymbol{\lambda}, \boldsymbol{\mu})$.
To solve the dual problem, an algorithm is presented next.

Lemma: Given $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$, the optimal time and energy allocations of $\mathbf{D_{SICD}}$ are given by:

$$\tau_{0,t}^* = \min\left[\left(1 - \frac{\sum_{i=1}^{K} g_{i,t}E_{i,t}}{z_t^* \sigma^2}\right)^+, 1\right]. \tag{3.11}$$

$$E_{i,t}^* = \left(\frac{(1-\tau_{0,t})(g_{i,t} - \sigma^2 a_{i,t})}{a_{i,t} g_{i,t}} - \frac{1}{g_{i,t}}\sum_{j \ge i}^{K} g_{i,t}E_{i,t}\right)^+. \tag{3.12}$$

The variable $a_{i,t}$ is defined as:

$$a_{i,t} \overset{\text{def}}{=} \ln(2)\left(\sum_{n=t}^{T} \lambda_{i,n} + g_{i,t}\chi\{i \ge 2\}\sum_{j=1}^{i-1} \mu_{j,t} S_j^{th} - \mu_{i,t} g_{i,t}\right), \tag{3.13}$$

where $\chi\{.\}$ is the indicator function and $(.)^+ \overset{\text{def}}{=} \max\{0, .\}$ and z_t^* is the unique solution of $f(z_t) = b^{(t)}$, where $f(z)$ and $b^{(t)}$ are given by:

$$f(z_t) = \ln(1 + z_t) - \frac{z_t}{1 + z_t}. \tag{3.14}$$

$$b^{(t)} = \ln(2)\Big(\sigma^2 \sum_{i=1}^{K} \mu_{i,t} S_i^{th} + \sum_{i=1}^{K}\sum_{n=t}^{T} \lambda_{i,n}\gamma_{i,t}\Big). \tag{3.15}$$

Proof: It can be verified that there exists $\boldsymbol{\tau}_0$ and $\mathbf{E}$ that strictly satisfy all the constraints of $\mathbf{D_{SICD}}$. Hence, strong duality holds for this problem [34]; therefore, the KKT conditions given below are necessary and sufficient for the global optimality:

$$\frac{\delta}{\delta \tau_{0,t}}\mathcal{L} = \ln\left(1 + \frac{\sum_{j=1}^{K} g_{j,t}E_{j,t}}{\sigma^2(1-\tau_{0,t}}\right) - \frac{\sum_{j=1}^{K} g_{j,t}E_{j,t}}{\sigma^2(1-\tau_{0,t}) + \sum_{j=1}^{K} g_{j,t}E_{j,t}} - a^{(t)} = 0. \tag{3.16}$$

$$\frac{\delta}{\delta E_{i,t}}\mathcal{L} = \frac{\frac{g_{i,t}}{\sigma^2}}{1 + \frac{\sum_{i=1}^{K} g_{i,t}E_{i,t}}{\sigma^2(1-\tau_{0,t})}} - b_{i,t} = 0, \tag{3.17}$$

$\forall i$ and t, where $a_{i,t}$ and $b^{(t)}$ are given by (3.13) and (3.15), respectively.

By defining $z_t = \frac{\sum_{j=1}^{K} g_{i,t}E_{j,t}}{\sigma^2(1-\tau_{0,t})}$, (3.16) can be reformulated as $f(z_t) = b^{(t)}$, where $f(z_t)$ is given in (3.14). Since $f(z_t)$ can be verified to be a monotonically increasing function of $z_t \geq 0$, where $f(0) = 0$, then there exists a unique solution z_t^* that satisfies $f(z_t^*) = b^{(t)}$ and, hence, $\tau_{0,t}^*$ can be expressed as in (3.11) and by using (3.17), $E_{i,t}^*$ can be expressed as in (3.12).

In summary, we can say that there is a trade-off between implementation complexity and performance. When the simple implementation is of interest, despite the resulted modest performance, then the LCD scheme presents a better choice. On the other hand, when the system performance is the main objective, then SICD scheme would be the better option, at the expense of increased system complexity.

3.4 Max-Min Throughput Optimization

In this section, we address the potential unfairness typically exhibited by sum throughput optimal policies. Under the max-sum throughput formulation, some nodes are likely to be allocated very little, or no, resources (time and power) in some scenarios, such that they achieve almost zero throughput. This can be a serious problem for some applications. In wireless sensor networks, for instance, in which all sensors need to periodically send their sensing data to the AP at the same rate, fairness is a necessity. Nevertheless, in most applications that depend on WSNs (or WPCNs in particular), delay is one of the most important concerns. Minimizing the delay is directly related to maximizing the service rate which is, in turn, directly related to maximizing the common throughput among nodes or maximizing the minimum individual throughput. To address these fairness and delay issues in our problem context, we adopt the max-min optimization [61], [62]. Next, the problem formulation and solution approach for maximizing the minimum UL throughput in LCD and SICD is discussed in detail.

3.4.1 Max-Min Problem Formulation with Low Complexity Decoding

The max-min UL throughput problem, with low complexity decoding, can be formulated as follows:

$$
\begin{aligned}
\mathbf{P2_{LCD}}: \quad & \max_{\boldsymbol{\tau}_0,\mathbf{E},\mathbf{x},\overline{\mathbf{R}}} \quad \overline{R}, \\
\text{subject to:} \quad & R_{i,t} \geq \overline{R}, \\
& Eq.(3.3), \\
& \sum_{n=1}^{t} E_{i,n} \leq \sum_{n=1}^{t} \gamma_{i,n}\tau_{0,n}, \quad \forall t \quad \text{(energy causality constraints)}, \\
& x_{i,t} \geq S_i^{th}, \quad \forall i, \forall t \quad \text{(decoding constraints)}, \\
& 0 \leq \tau_{0,t} \leq 1, \quad \forall t, \\
& E_{i,t} \geq 0 \quad \forall i, \forall t,
\end{aligned}
$$

where $\overline{R}$ is the minimum throughput to be maximized, $R_{i,t}$ is the achievable UL throughput of U_i in time slot t and is expressed in (3.2) and $x_{i,t}$ is the average SINR at the AP for U_i in time slot t.

This problem differs from the two max-sum problems, studied in Section 3.3, in the objective function and the first constraint. The objective function is affine but the first constraint is not convex. Hence, $\mathbf{P2_{LCD}}$ is a non-convex problem.

In order to circumvent the non-convexity hurdle of $\mathbf{P2_{LCD}}$, we adopt an iterative solution approach similar to the one followed in Section 3.3.1 to solve $\mathbf{P1_{LCD}}$. To this end, we split $\mathbf{P2_{LCD}}$ into two sub-problems $\mathbf{P2_{LCD}}(\text{given } \boldsymbol{\tau}_0)$ and $\mathbf{P2_{LCD}}(\text{given } \mathbf{E})$. The first sub-problem is still non-convex and will be solved using an approximate iterative method. On the other hand, the second sub-problem is convex and can be solved using standard convex optimization tools.

The First Sub-problem:

The first sub-problem given $\boldsymbol{\tau}_0$ is formulated as follows:

$$\begin{aligned}
\mathbf{P2_{LCD}}(\text{given } \boldsymbol{\tau}_0): \quad & \max_{\mathbf{E},\mathbf{x},\overline{\mathbf{R}}} \quad \overline{R}, \\
\text{subject to:} \quad & R_{i,t} \geq \overline{R} \quad \forall i, \forall t, \\
& Eq.(3.3), \\
& \sum_{n=1}^{t} E_{i,n} \leq \sum_{n=1}^{t} \gamma_{i,n}\tau_{0,n}, \quad \forall t \quad \text{(energy causality constraints)}, \\
& x_{i,t} \geq S_i^{th}, \quad \forall i, \forall t \quad \text{(decoding constraints)}, \\
& E_{i,t} \geq 0 \quad \forall i, \forall t.
\end{aligned}$$

The objective function and the constraints are affine, except for the first constraint which is non-convex. To solve this problem, an approximate iterative approach is adopted [63].

The approach used to solve $\mathbf{P2_{LCD}}(\text{given } \boldsymbol{\tau}_0)$ is based on solving a sequence of strongly convex inner approximations of the problem until a stationary solution of $\mathbf{P2_{LCD}}(\text{given } \boldsymbol{\tau}_0)$ is reached. This solution guarantees, based on the proof and assumptions in [63], the feasibility of the solutions in every iteration.

The approach in [63] is based on replacing a non-convex objective function (say $U(\mathbf{x})$) by a strongly convex and simple function ($\tilde{U}(\mathbf{x};\mathbf{y})$) and constraints ($g_m(\mathbf{x})$, where m is the non-convex constraints index) with convex upper estimates ($\tilde{g}_m(\mathbf{x};\mathbf{y})$) to create a sub-problem $\mathbf{P_y}$. The sub-problem $\mathbf{P_y}$ is strongly convex and has a unique solution $\hat{\mathbf{x}}(\mathbf{y})$ (a function of $\mathbf{y}$). By starting from a feasible point $\mathbf{y}^{(0)}$, the proposed method, iteratively, computes the solution of the sub-problem $\mathbf{P_y}$, which is $\hat{\mathbf{x}}(\mathbf{y})$ and then takes a step ($\zeta^n \in (0,1]$, where n is the iteration index) from $\mathbf{y}$ towards $\hat{\mathbf{x}}(\mathbf{y})$.

Note that the point $\mathbf{y}$ generated by the algorithm in every iteration is always feasible for the original problem $\mathbf{P2_{LCD}}(\text{given } \boldsymbol{\tau}_0)$. Convergence is guaranteed under mild assumptions that offer a lot of flexibility in the choice of the approximation functions and free parameters.

The main problem that affects this approach is the *affine* objective function. To check the stationarity of every iteration, we need the objective function to be a function of $\mathbf{y}$ to study its gradient until a stationary solution is reached. To solve this problem, the objective function $\overline{R}$ is replaced by a single user throughput (the one that had the minimum value at the initial search point: $R_{l,t}$) and an equality constraint is added to attain the same target. The objective function must be modified every iteration based on the minimum achieved value at the new initials (the optimum point of the previous iteration).

After this modification, problem $P2_{LCD}(\text{given } \boldsymbol{\tau}_0)$ can be formulated as follows:

$$\begin{aligned}
\mathbf{P2_{LCD}}(\text{given } \boldsymbol{\tau}_0): \quad & \max_{\mathbf{E},\mathbf{x},\overline{\mathbf{R}}} \quad R_{l,t}, \\
\text{subject to:} \quad & R_{l,t} = \overline{R} \quad \forall t, \\
& R_{i,t} \geq \overline{R} \quad \forall i = 1,2,3,...,K, i \neq l, \forall t, \\
& Eq.(3.3), \\
& \sum_{n=1}^{t} E_{i,n} \leq \sum_{n=1}^{t} \gamma_{i,n}\tau_{0,n}, \quad \forall t \quad \text{(energy causality constraints)}, \\
& x_{i,t} \geq S_i^{th}, \quad \forall i, \forall t \quad \text{(decoding constraints)}, \\
& E_{i,t} \geq 0 \quad \forall i, \forall t.
\end{aligned}$$

The chosen approximation is given by (3.18) [63]:

$$\tilde{U}(x;y) = \nabla_x U(y)^T (x-y) + \frac{1}{2}|x-y|^2. \tag{3.18}$$

This approximation mimics proximal gradient methods. It can be used if no convexity whatsoever is present. It is also used for the first two constraints, but we must check that the approximation $\tilde{g}_m(\mathbf{x};\mathbf{y})$ is an upper approximate function if the non-convex constraint is $g_m(\mathbf{x}) \leq C$, where C is a constant and a lower approximate function if $g_m(\mathbf{x}) \geq C$.

The Second Sub-problem:

The second sub-problem given $\mathbf{E}$ is formulated as follows:

$$
\begin{aligned}
\mathbf{P2_{LCD}}(given\mathbf{E}): \quad & \max_{\boldsymbol{\tau}_0,\mathbf{x},\overline{\mathbf{R}}} \quad \overline{R}, \\
\text{subject to:} \quad & R_{i,t} \geq \overline{R}, \\
& Eq.(3.3), \\
& \sum_{n=1}^{t} E_{i,n} \leq \sum_{n=1}^{t} \gamma_{i,n}\tau_{0,n}, \quad \forall t \quad \text{(energy causality constraints)}, \\
& x_{i,t} \geq S_i^{th}, \quad \forall i, \forall t \quad \text{(decoding constraints)}, \\
& 0 \leq \tau_{0,t} \leq 1, \quad \forall t.
\end{aligned}
$$

The objective function in addition to all constraints is affine except for the first constraint. It can be verified that the left-hand side of the first constraint has a Hessian matrix that is **positive semidefinite**, hence, it is convex. Therefore, problem $\mathbf{P2_{LCD}}$(given $\mathbf{E}$) is convex and can be solved efficiently using standard convex optimization tools.

3.4.2 Max-Min Problem Formulation with Successive Interference Cancellation Decoding

The max-min optimization problem with SICD is formulated in a similar manner as LCD except for minor differences:

$$
\begin{aligned}
\mathbf{P2_{SICD}}: \quad & \max_{\boldsymbol{\tau}_0,\mathbf{E},\mathbf{x},\overline{\mathbf{R}}} \quad \overline{R}, \\
\text{subject to:} \quad & R_{i,t} \geq \overline{R}, \\
& Eq.(3.10), \\
& \sum_{n=1}^{t} E_{i,n} \leq \sum_{n=1}^{t} \gamma_{i,n}\tau_{0,n}, \quad \forall t \quad \text{(energy causality constraints)}, \\
& x_{i,t} \geq S_i^{th}, \quad \forall i, \forall t \quad \text{(decoding constraints)}, \\
& 0 \leq \tau_{0,t} \leq 1, \quad \forall t, \\
& E_{i,t} \geq 0 \quad \forall i, \forall t.
\end{aligned}
$$

Notice that the only difference between $\mathbf{P2_{LCD}}$ and $\mathbf{P2_{SICD}}$ is the definition of the SINR which is denoted by x. As mentioned earlier, in LCD, the interference term in (3.3) includes signals from all other users. On the other hand, in SICD, it includes only the signals which have not been decoded yet and therefore are not canceled from interference, as given in (3.10).

Under the SICD scheme, it is important to notice that although the decoding order, that is, the order by which the users' signals are decoded at the AP, doesn't affect the sum throughput of the network as shown in [64], it will certainly affect the fairness aspect. To achieve the highest fairness, it can be easily proven that the optimal decoding order w.r.t fairness is based on the received signal strength that is directly affected by the channel gain; from the strongest to the weakest signal [58].

The approach used to solve $\mathbf{P2_{SICD}}$ is the same as the one used to solve $\mathbf{P2_{LCD}}$. It is worth mentioning that, for $\mathbf{P2_{LCD}}$ and $\mathbf{P2_{SICD}}$, if the optimum solution can be reached, the max-min throughput would have been the common throughput achieved by all users. However, since an approximate approach is used, the users do not reach a common throughput and some variations can be noticed between the throughputs achieved by individual users.

The formal description of the approach to solving $\mathbf{P2_{LCD}}$ and $\mathbf{P2_{SICD}}$ is given in Algorithm 2.

Algorithm 2 Solving $\mathbf{P2_{LCD}}$ and $\mathbf{P2_{SICD}}$

```
1: repeat
2:     procedure SOLVE P2(GIVEN $\boldsymbol{\tau}_0$)
3:         Initialize $n = 0, \zeta^{(n)} \in (0, 1], y^{(n)} \in$ feasible set.
4:         repeat
5:             Choose the objective function to be the user throughput with minimum value at
   initial point.
6:             Approximate non-convex function and constraints.
7:             Compute $\hat{\mathbf{x}}(\mathbf{y^{(n)}})$, the solution of the sub-problem $\mathbf{P_y}$.
8:             Set $y^{(n+1)} = y^{(n)} + \zeta^{(n)}(\hat{x}(y^{(n)}) - y^{(n)})$.
9:             $n \leftarrow n + 1$.
10:        until Stationary solution of P2($\boldsymbol{\tau}_0$) is reached
11:        Find sub-optimum $R_{min}$ and $\mathbf{E}$.
12:    end procedure
13:    procedure SOLVE P2(GIVEN $\mathbf{E}$)
14:        Use standard convex optimization solver: SDPT3.
15:        Find sub-optimum $R_{min}$ and $\boldsymbol{\tau}_0$.
16:    end procedure
17: until The maximized throughput converges to a pre-specified accuracy
```

3.5 Performance Evaluation

3.5.1 Simulation Setup

We consider K single-antenna nodes, where K ranges from 2 to 20. Nodes are distributed randomly around the ER node in a circular area of radius of 10 meters. A horizon of T timeslots, ranging from 1 to 10, is studied. Nodes receive DL power from the ER node and send UL data to the AP which is located d_{ER-AP} meters from the

ER. For the UL transmission, the path loss model is $g_{i,t} = 10^{-3} d_{U_i-AP}^{-2}$. For the DL, we use the parameters of the P2110 device [59]. The system parameters, used to generate numerical results, are listed in Table 6.1.

TABLE 3.1: Simulation Parameters

Parameters	Definition	Values
d_{ER-AP}	Distance between ER and AP (meters)	0:20:120
σ^2	Noise Power	-155 dBm/Hz
BW	Bandwidth	1 MHz
P_B	ER Node Transmission Power	3 W
f_c	Central Frequency	915 MHz
G_r	Receiver Antenna Gain	6 dB
η_i	Harvesting Efficiency	0.49
S_i^{th}	Decoding Threshold (dB)	-10:1:0

3.5.2 Numerical Results

Simulations were carried out using the optimization toolbox in MATLAB. The performance results presented next revolve around three main thrusts, namely max-sum optimization, max-min optimization, and the fundamental throughput-fairness trade-off within our problem context.

Max-Sum Performance Results

Recall that, under the max-sum problem formulation, the objective function to be maximized is the sum UL throughput. The performance of the presented approaches, namely LCD and SICD, are presented next.

We compare three solutions for the max-sum problem under LCD ($\mathbf{P1_{LCD}}$): i) Solving a sub-problem given $\boldsymbol{\tau}_0$ as in [57], ii) Our proposed near-optimal solution in Section 3.3.1 which solves $\mathbf{P1_{LCD}}$ iteratively, given $\boldsymbol{\tau}_0$ (Using the approximation in [57]) and given $\mathbf{E}$, and iii) The same approach in Section 3.3.1 except for solving $\mathbf{P1_{LCD}}$(given $\boldsymbol{\tau}_0$) using the approximation in [63]. This is shown in Figure 3.2 in which the max-sum throughput is plotted vs. the distance between the AP and ER (d_{ER-AP}) for $K = 5$ and $T = 2$. The figure shows the superior performance of our proposed near-optimal solution.

In Figure 3.3, the max-sum throughput for different values of K at $T = 2$ is plotted vs. the distance between the AP and ER (d_{ER-AP}). As noticed, the sum throughput decreases as the distance increases due to the path loss. As the number of users increases, the sum throughput increases as well due to the existence of more nodes contributing to the sum throughput. The figure shows that SICD outperforms LCD, which is expected due to the merits of SICD successively decoding interfering signals and canceling them out from the interference to other signals. On the other hand, SICD adds up more computational complexity to the system.

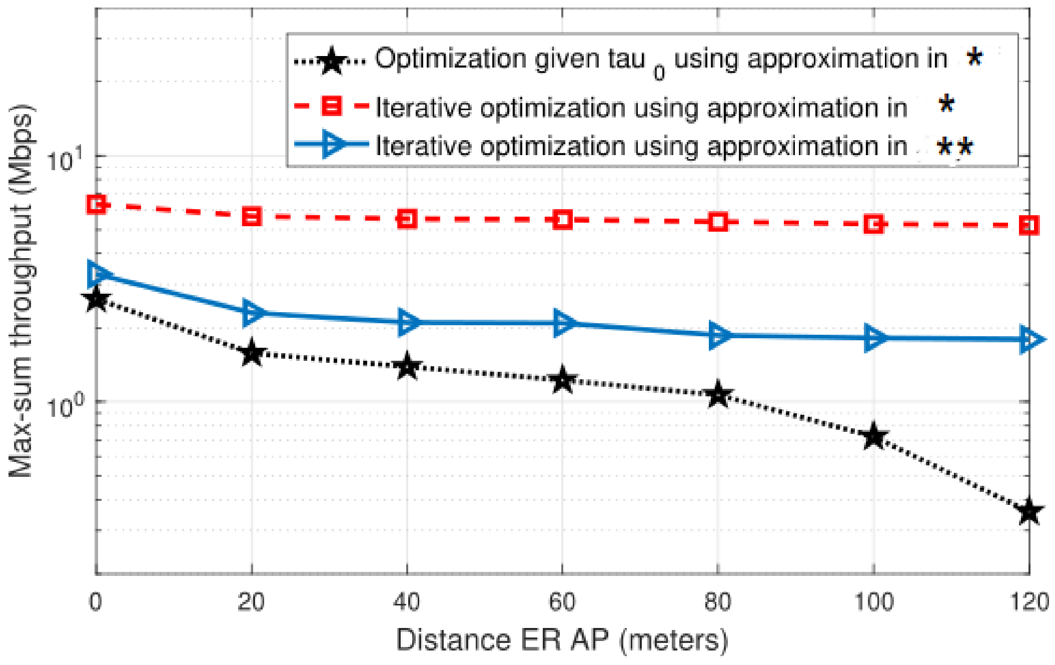

FIGURE 3.2: Max-sum at $K = 5$ and $T = 2$, where $*$ is [57] and $**$ is [63]

Moreover, in Figure 3.4, the max-sum throughput for different values of T at $K = 5$ is plotted vs. the distance between the AP and ER. An increase is observed in the max-sum throughput, as the time horizon, T, is extended.

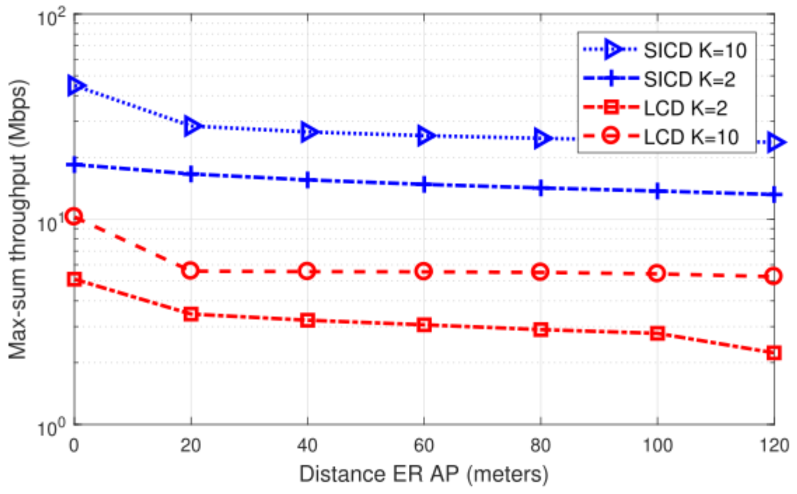

FIGURE 3.3: Max-sum at $T = 2$

At a d_{ER-AP} of 100 meters, the effect of the decodability threshold, S_i^{th} is studied in Figure 3.5. $\mathbf{P1_{LCD}}$ has a solution only for very low values (i.e. low SINR requirements) of the decodability threshold; that's why LCD curves stop early in Figure 3.5. Moreover, the higher the number of users, the sooner $\mathbf{P1_{LCD}}$ becomes infeasible, due to elevated levels of interference. On the other hand, $\mathbf{P1_{SICD}}$ is not affected by the decodability threshold. This is because, with SICD, the average SINR of every user is always greater than the threshold due to interference cancellation.

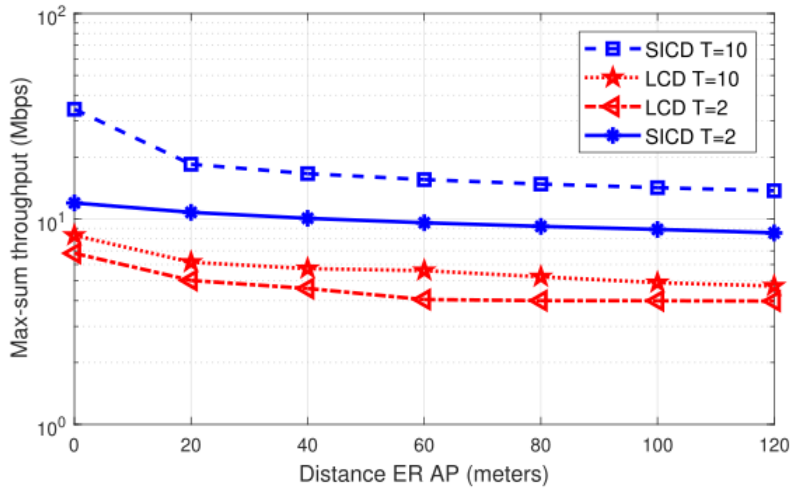

FIGURE 3.4: Max-sum at $K = 5$

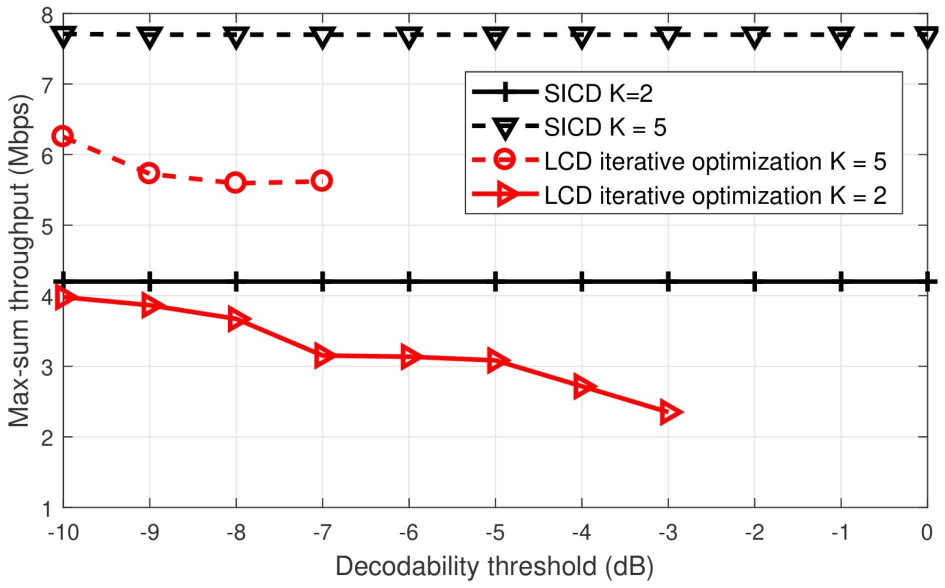

FIGURE 3.5: Max-sum at $d_{ER-AP} = 100\ m$

Max-Min Performance Results

In an attempt to enhance fairness across users, the objective function to be maximized, under max-min, is the minimum user's throughput. The performance of the presented approach and comparison to the optimum using exhaustive search are presented next.

In Figure 3.6, the max-min throughput is plotted vs. d_{ER-AP} for both LCD and SICD. The minimum throughput is noticed to decrease as the number of nodes increases. This is because the same resources are distributed among a larger number of users. The performance of SICD is superior to LCD due to higher SINRs as a result of interference cancellation.

To compare the performance of the presented schemes to the global optimum, we resort to exhaustive search since solving for the global optimum is prohibitively complex due to problem non-convexity. Since carrying out an exhaustive search is very complex and time consuming, it is not applicable to carry it out for a large network. As a result, we carried out an exhaustive search for a small network scenario which consists of two nodes and 1 time slot ($K = 2$ and $T = 1$). The results of this search are shown in Figure 3.7.

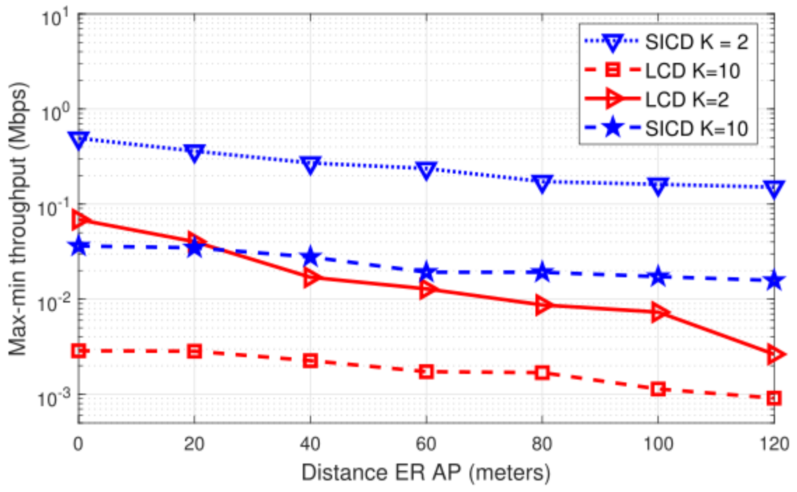

FIGURE 3.6: Max-min at $T = 2$

Throughput-Fairness Trade-off

There is a fundamental trade-off between the max-sum and the max-min problems. A network with max-sum throughput objective will allocate more resources to nodes with high channel gains (near nodes) to add up to the sum throughput. On the other hand, a network with max-min throughput objective will balance the resource allocation to accommodate users with low channel gains (far nodes) to enhance fairness.

To show this behavior, the sum throughput achieved by the optimal max-min problem is compared to the sum throughput achieved by the max-sum problem. Figure 3.8 confirms the expected behavior whereby the max-min formulation enhances fairness (as shown in Figures 3.9 and 3.10) at the expense of reduced sum throughput, compared to the max-sum formulation. On the other hand, and in order to complete the picture, the minimum user throughput is compared, under both formulations, in Figure 3.9 showing that the max-min formulation is superior to max-sum, due to its fairness merits.

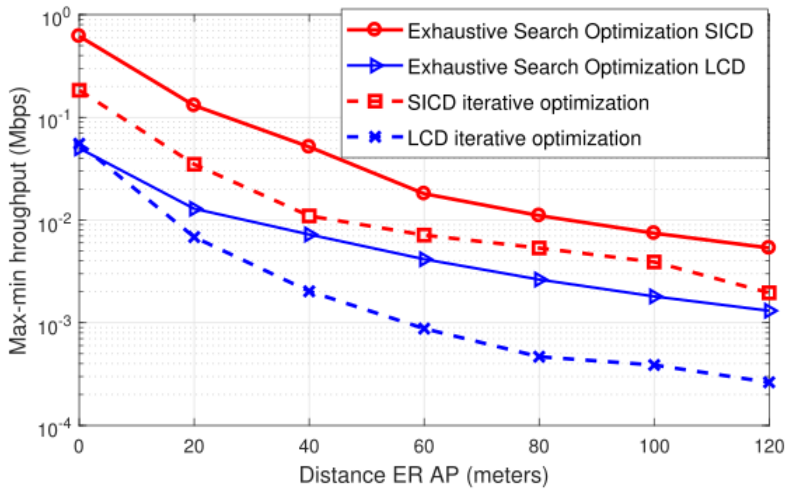

FIGURE 3.7: Max-min at K = 2 and T = 1

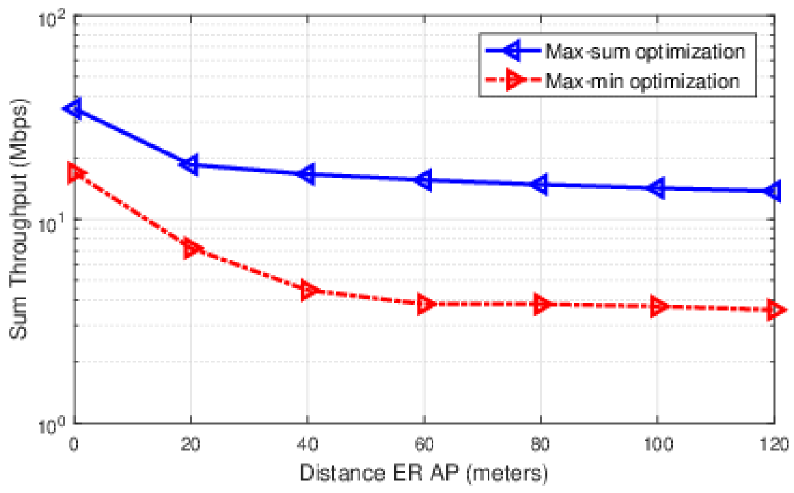

FIGURE 3.8: Sum Throughput in max-min and max-sum Problems

Moreover, Jain's index (a measure of fairness between nodes) is plotted for the max-sum and max-min optimal policies to show to what extent fairness is accommodated in each formulation.

Jain's index [65] is defined as:

$$\jmath(x_i) = \frac{\left(\sum_{i=1}^{K} x_i\right)^2}{K \cdot \sum_{i=1}^{K} x_i^2}, \tag{3.19}$$

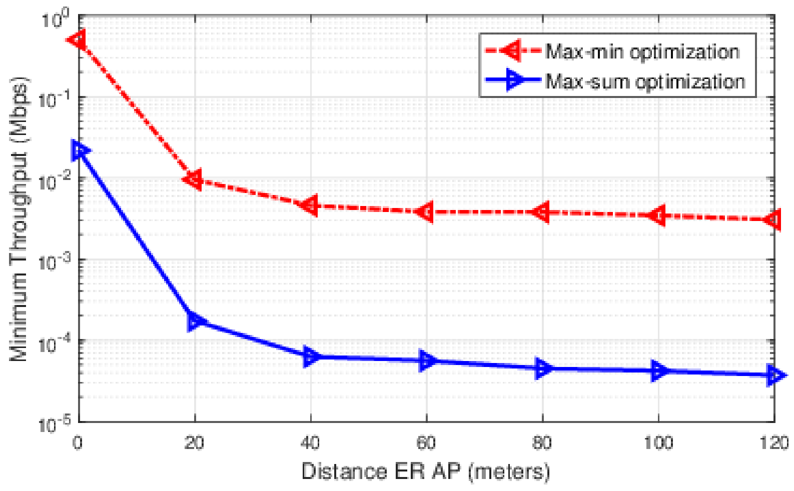

FIGURE 3.9: Minimum Throughput in max-min and max-sum Problems

where, x_i is the throughput for user i, i ranges from 1 to K. Jain's index ranges from $\frac{1}{K}$ (worst fairness) to 1 (best fairness), and it is maximum when all users receive the same allocation.

In Figure 3.10, Jain's index is plotted vs. the number of users. It is noticed how Jain's index in the max-sum problem has the worst-case value ($\frac{1}{K}$), while it is better for max-min schemes.

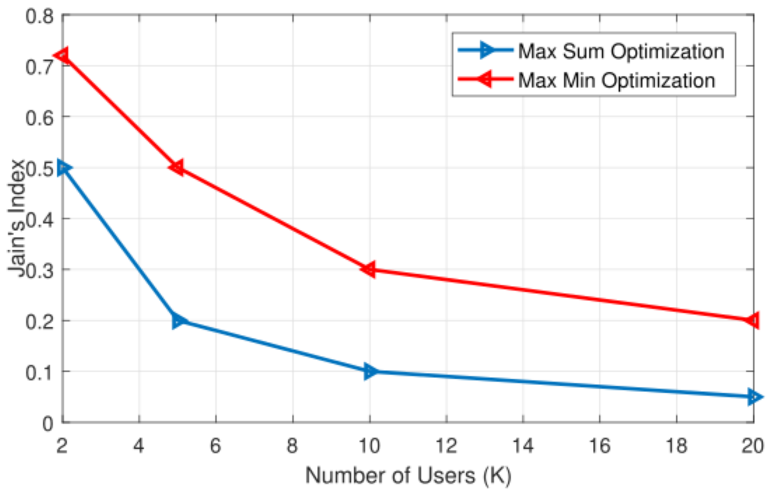

FIGURE 3.10: Jain's Index (SICD Scheme)

Finally, it worth noting that Figures 3.8, 3.9 and 3.10 are plotted for SICD. Similar results can be obtained for LCD.

3.6 Conclusion

In this chapter, we investigate the problem of optimal resource (time and power) allocation in WPCNs using NOMA. Two different optimization problem formulations are considered; in the first one, the sum throughput (max-sum) of all users is maximized. In the second one, the min-throughput (max-min) of all users is maximized. Under these two formulations, two NOMA decoding schemes are studied, namely, LCD and SICD. Due to the non-convex nature of the max-sum and max-min optimization problems, we propose an approximate solution approach, in which the non-convex optimization problem is approximated by a convex optimization problem, which satisfies all the constraints of the original problem. The approximate convex optimization problem can be then solved iteratively. The results show a trade-off between maximizing the sum throughout and achieving fairness via maximizing the minimum throughput.

Chapter 4

Machine Learning-Based MIMO Enabling Techniques for Energy Optimization in Cellular Networks

This chapter and the next, discuss the cellular network management challenge. This chapter focuses on conserving energy by the management of turning the MIMO feature on/off. The next chapter adopts a wider perspective and deeper control of the cellular network (as will be explained later).

In mobile communications technology, the component with the greatest share of energy consumption in mobile networks is the Base-station (or eNodeB in LTE standards) with a share greater than 50% [66]. Therefore, this work focuses on reducing energy consumption at the eNodeB in mobile networks.

In 4G networks, one of the most energy-consuming features is the use of Multiple Input Multiple Output (MIMO) scheme. Unlike the Single Input Single Output (SISO) scheme, the main idea behind MIMO is to use multiple antennas at the transmitter and/or the receiver simultaneously. This allows for higher throughput and guarantees faster downloads and higher spectral efficiency [67].

MIMO, despite its multiple advantages, consumes a large amount of energy that might not be always necessary. If a certain level of Quality of Experience (QoE) is achieved by SISO, a lot of energy can be saved by turning off the MIMO scheme at the eNodeB. Currently, mobile operators do this manually based on some user-defined schedules. An automated method to turn on/off the MIMO capability, based on the network performance, is certainly needed to reduce the amount of used energy.

To clarify how controlling MIMO can be effective to save energy, Vodafone Egypt provided information about the average energy consumption before and after turning off MIMO. For the MIMO sites, from our test dataset that will be presented later, turning off MIMO can cause energy saving that ranges from 2.5% to 8.6% of the total site energy (an average of 5.5% energy saving) resulting from only controlling the radio unit of the base station. This variation of the energy savings is because sites lie in different locations and are subject to different conditions, therefore, the total consumed energy is affected by conditions other than the radio unit.

The main objective of this study [18] is to use Machine Learning, specifically Neural Networks (NN), to learn some features of the network and decide whether SISO is sufficient to achieve a satisfactory level of QoE. Based on this, the mobile operator can decide whether to enable the MIMO capability.

NNs have been gaining a lot of interest recently in the wireless communications literature [68], [16]. NNs can learn the features of a certain system even if it has no model to represent it. By subjecting the NN to the data drawn from the system, it will be able to extract the common features and learn the performance of the system.

The main contribution of the work in this chapter is to use two types of NNs: 1) Fully Connected NN (also called Multi-Layer Perceptron (MLP)) to learn the features of the SISO scheme 2) Recurrent NN to track any trends in the data history. Both machines are trained using historical data drawn from realistic SISO cells[1]. This data includes several

[1]All the data used in this work are provided by one of the mobile operators in Egypt.

network features recorded around the clock. When the training phase is complete, the machine is subjected to some cell features of a MIMO site and it emulates the performance of the SISO scheme to decide whether SISO can achieve a satisfactory QoE. To make this decision, the average DownLink (DL) user throughput is predicted and monitored as a measure of QoE. If the machine decides that SISO is enough, then MIMO can be turned off to save energy. Otherwise, the MIMO is kept on to achieve an acceptable QoE performance.

4.1 Literature Review

Reducing the carbon footprint of mobile networks has been of interest in the literature recently. For example, [69] discusses the possibility of powering mobile networks with green energy. It presents an overview of the design and challenges of green energy-enabled mobile networks.

The authors in [70], [71], and [72] discuss possible techniques to reduce the power consumption in base stations (BSs) since it is the entity with the highest power consumption in the whole mobile network [66]. They focus on optimizing air conditioning power consumption and minimizing feeder losses.

The exploitation of Machine Learning (ML) techniques has been of interest in many research works in cellular networks. In [68], the authors investigate the possible use cases of machine learning in future cellular networks. They review the basic concepts of machine learning and propose their use in 5G networks, including cognitive radios, massive MIMOs, femto/small cells, heterogeneous networks, smart grid, energy harvesting, and device-to-device communications. On the other hand, in [73], the authors are interested in cache content optimization in small base stations based on learning algorithms.

Using ML for energy saving in base stations has been investigated in the literature as well. In [74], the authors present a Reinforcement Learning (RL) approach for resource allocation in wireless networks. The presented algorithm learns the utility of performing various tasks over time and uses the application constraints for task management by optimizing energy usage and network lifetime. From another perspective, the authors in [75] adopt machine learning in energy harvesting. They propose a strategy learning algorithm that exploits the expected energy and adapts spectrum selection strategies to maximize the network's performance. The learning algorithm addresses how multiple users discover available channels and harvest energy over the network.

The novelty of our proposed approach lies in focusing on the energy optimization of the *Radio* part of the BS and not air conditioning or feeder losses. It targets the power amplifiers which are responsible for 65% of the total BS energy consumption [76]. Additionally, and to the best of our knowledge, this is the first work to address machine learning as means for the enabling decision of the MIMO scheme(s) based on energy saving purposes. The current practice in mobile networks is to have some user-defined schedules or some other "experience-based" manual approach. These practices suffer from the fact that they are not real-time and that they are human-driven rather than data-driven. Our approach allows for real-time data-driven seamless operation (i.e. it learns features and their relationships, which is not the case in the human-driven approaches).

4.2 Problem Description and Proposed Approach

As mentioned earlier, the main objective of this work is to exploit Machine Learning (ML) approaches to reduce energy consumption in the radio part of the base stations. This is achieved by turning off the MIMO feature *if and only if* SISO can achieve a minimum satisfactory Quality of Experience (QoE), which is measured in our work by a minimum average DownLink (DL) user throughput[2]. Our models will follow regression models to estimate the expected user DL average throughput for a SISO scheme based on some network parameters as will be explained later. It is worth mentioning that our algorithm does not provide any service guarantees for any specific UE. MIMO feature is turned on/off based on the expected average throughput per user.

4.2.1 Data

The data used for training, validation, and testing are real data shared by Vodafone Egypt. It includes Key Performance Indicators (KPIs) from 145 **SISO** sites calculated at the cell level every hour through a whole duration of two weeks. Based on the operator expertise, the KPIs that are most relevant (directly or indirectly) to the user DL average throughput are selected. The KPIs are listed next.

DL Physical Resource Block (PRB) utilization

A PRB consists of 12 consecutive subcarriers for one slot (0.5 ms), and is the smallest element of resource allocation assigned by the eNodeB. PRB utilization is a measure of the cell utilization (congestion). As the PRB usage ratio increases, the resources may not be allocated in a timely and reliable manner to the users of the cell.

Average Channel Quality Indicator (CQI) per cell

This is a measure of the average channel quality in the whole cell. The CQI reported value is a number between 0 and 15; this indicates the level of modulation and coding

[2]Our approach presented in this work can be readily generalized to any other measure of users' QoE.

the UE can operate at. The CQI value is computed by considering the SINR values using a precalculated lookup table of the CQI index versus SINR as a reference.

DL Traffic Volume

This is measured in GBytes. It is the total traffic of all users for each cell along the measuring step (1 hour).

Average number of User Equipment (UE)

That is how many UEs attach to this cell on the average. This is an indicator of the cell utilization (along with PRB utilization).

Maximum number of UEs

This KPI indicates how many UEs attach to this cell at most. This parameter is required to avoid underestimation of worst case scenarios.

User DL average throughput

That is the performance metric that the operator adopts to indicate the QoE (that is the current practice in mobile networks). A QoE is considered satisfactory only if the user DL average throughput is ≥ 5 Mbps.

MLP takes only the current measurements to predict the DL user average throughput if SISO is used. On the other hand, RNN tracks the historical trend of the data to predict the user DL average throughput.

The total number of data points in our data set is 382,456. During our experiments, 70% of the available data was used for training, 20% was used for validation and 10% was used for testing.

4.2.2 ML Models Architectures

We now present the models that we use in the two NN methods that we use in this study as follows.

Multi-Layer Perceptron Architecture

The used MLP consists of:

- 1 input layer containing 12 neurons (ReLu activation function).
- 1 hidden layer containing 8 neurons (ReLu activation function).
- 1 output layer containing 1 neuron (DL average user throughput) (Linear activation function).

The used optimizer is ADAM, with a learning rate of 0.001 and batch size of 50.

It should be noted that the number of hidden layers, neurons, batch size, optimizer and learning rate are all hyper-parameters that are selected after several trials.

A layout of the used network architecture can be found in Fig. 4.1

Recurrent Neural Network

The used RNN consists of:

- 1 input layer containing 50 neurons (ReLu activation function).
- 1 hidden layer containing 100 neurons (ReLu activation function).
- 1 output layer containing 1 neuron (DL average user throughput) (Linear activation function).
- Feedback connection from the previous hidden layer output to the current hidden layer input to track the historical features of the data.

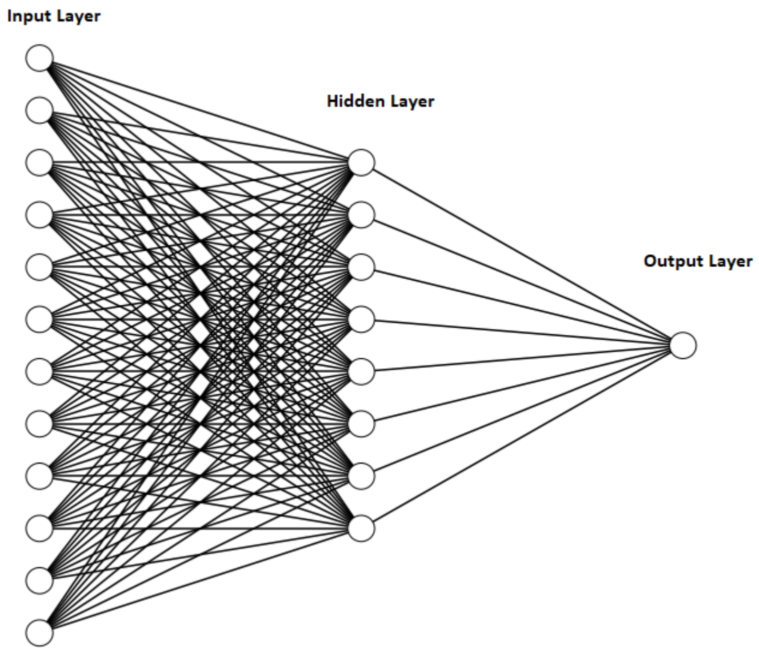

FIGURE 4.1: Neural Network Architecture

The used optimizer is ADAM, with a learning rate of 0.001 and batch size of 72.

It should be noted that the available data are KPIs measures at every hour from different cells. To use these data to train the RNN, we assume that the historical trend of data does not differ from one site to another. Therefore, we separate the data from different cells, make sure they are arranged from oldest to newest, and use these sequences to train the machine cell after cell. After each sequence of data, the trained machine is used as a starting point for the next sequence drawn from the next mobile network cell.

4.2.3 The Algorithm

The proposed scheme is a regression model that takes the available KPIs as inputs to the Neural Network (NN) and the DL user average throughput as the output. The main idea is to create an NN that is trained via the SISO data. This results in a network that has learned the performance and behavior of SISO sites. Eventually, when the network is fully trained, it can take the KPIs of the MIMO site under consideration and use its KPIs to predict the DL average throughput using the SISO behavior learned by the network. If the predicted output is acceptable (i.e. achieves a satisfactory users' QoE), which means that SISO is good enough to handle the network traffic, then MIMO can be turned off (temporarily) to save energy. Otherwise, energy must be expended to keep the MIMO on to maintain a satisfactory users' QoE.

In our work, we define a satisfactory user QoE as achieving a user average DL throughput that is not less than 5 Mbps. Therefore, the output of the ML network is compared to a threshold of 5 Mbps to decide whether SISO can provide an acceptable QoE. There is a trade-off between the threshold that represents the satisfactory QoE and the amount of saved energy.

Our proposed algorithm is described in Algorithm 3. This description applies to both the MLP and RNN architectures.

4.3 Performance Evaluation

In this section, the performance evaluation results of the proposed schemes are presented.

Algorithm 3 Using MLP and RNN to save energy in 4G cellular networks by optimizing MIMO usage: algorithm description

```
1: procedure PREPARE DATA SET
2:     Normalization of the data to the range from 0 to 1.
3:     Divide data set into: $70\%$ training, $20\%$ validation and $10\%$ testing.
4: end procedure
5: procedure TRAINING
6:     Initialize: Random weights.
7:     Input: 5 KPIs provided by the training dataset.
8:     Output: DL average user throughput.
9:     Back-Propagation Learning Technique:
10:    while Validation error is decreasing do
11:        for Each training epoch do
12:            Monitor the output (DL average user throughput).
13:            Compute MAE based on current weights and current input.
14:            Update Weights based on the MAE and learning rate.
15:            Monitor validation error to avoid overfitting.
16:        end for
17:    end while
18:    Result: A machine that has learned the features (the historical trend in case of RNN) of
       SISO scheme.
19: end procedure
20: procedure TESTING
21:    Input: 5 KPIs from the testing data.
22:    Output: DL average user throughput.
23:    RUN NN in feed-forward direction.
24:    Compare predicted throughput to actual throughput.
25:    while Results are not satisfying do
26:        Change hyper parameters.
27:        Repeat training phase.
28:    end while
29: end procedure
30: procedure APPLICATION
31:    Use the fully trained machine to use the KPIs of a certain MIMO cell as inputs and apply
       regression to predict the output (DL average user throughput) if SISO scheme is used.
32:    If output is satisfying ($> 5Mbps$), MIMO is turned OFF to save energy. Otherwise,
       MIMO is used.
33: end procedure
```

4.3.1 Training Phase

The progress of the training and validation error during the training phase is shown in Fig. 4.2. It can be seen that the training error is slightly lower than the validation error. Both errors exhibit a decreasing trend. An early halt of the training process occurs when the validation error starts to increase while the training error is still decreasing (to avoid overfitting).

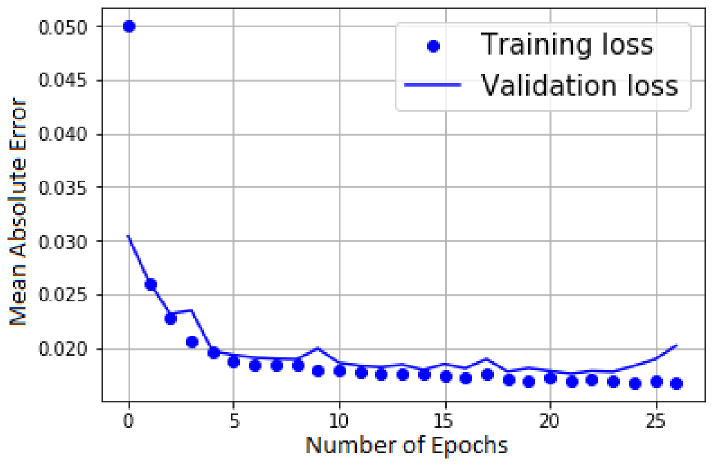

FIGURE 4.2: Training and Validation Errors (in Mbps) During the Training Process of MLP

At the end of the training phase, the performance of the machine is measured via the loss function. The Mean Absolute Error (MAE) is the chosen loss function during the training. Since we aim to build a regression model to estimate the user DL average throughput, it is reasonable to consider MAE as our performance measure. The results are stated in Table 4.1. The results show that our trained models can achieve very small MAE errors and that the RNN architecture was able to achieve smaller errors in general.

TABLE 4.1: Mean Absolute Error for Data Normalized from 0 to 1

Machine	Training Phase	Validation Phase	Testing Phase
MLP	0.0962	0.0985	0.1702
RNN	0.0289	0.0304	0.0993

4.3.2 Testing Phase

When the machine completes the training phase, it is ready to be tested. In the testing phase, the machine uses new data (different from the training and validation data). MAE of the testing phase is recorded in Table 4.1.

4.3.3 Testing using data from MIMO sites

To test the machine on MIMO sites, a group of MIMO sites is selected for the testing process by the service provider. MIMO is turned off in these sites and the DL average user throughput is recorded. The KPIs of these sites are fed to the proposed machines to predict the DL average user throughput and compared to the recorded actual throughput.

The MAE of this test is 0.31 for MLP and 0.26 for RNN. It should be noted that data were originally normalized to the range from 0 to 1. This is necessary when features have different ranges, like the case here. To have a look at the error function related to the original throughput value, normalization must be inverted. After denormalization, MAE reached the value of 2.75 Mbps for MLP and 1.38 Mbps for RNN.

It is worth mentioning that the sites used for training, validation, and primary testing are SISO sites. On the other hand, the sites used in this second testing phase are MIMO sites which are different from the sites used in the previous processes.

Fig. 4.3 and Fig. 4.4 show how close the predicted throughput (output of the machine) is to the actual throughput (provided by the data). They also show the absolute error at every point.

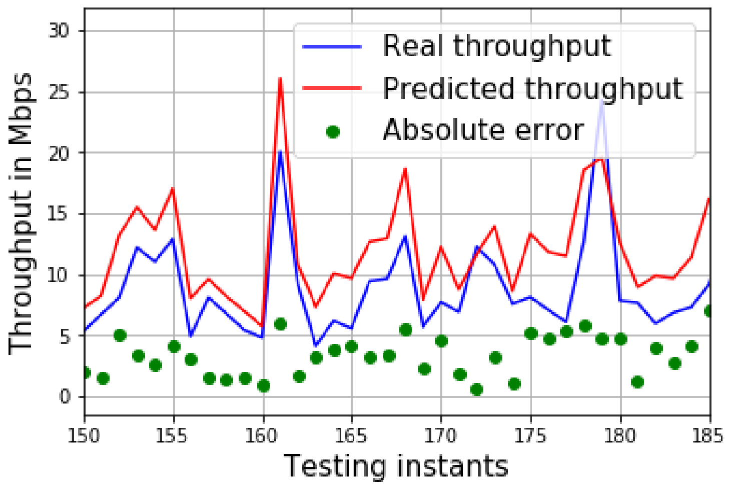

FIGURE 4.3: Actual Throughput, Predicted Throughput and the Absolute Error During a Portion of the Testing Phase for MLP.

It should be noticed that the considered KPIs are not the only factors that affect the throughput. For example, the weather conditions, the location of the site, the interference, and surrounding construction can all affect the achieved throughput. Considering more features is left for extending the current work.

As mentioned earlier, the threshold for acceptable QoE is chosen to be 5 Mbps. The percentage of correct decision (to turn MIMO on/off) for both the MLP and RNN architectures is presented in Table 4.2. This result is important because the main objective of this work is to take a correct decision to turn MIMO on/off not to predict the exact DL user average throughput.

TABLE 4.2: Percentage of Correct and Erroneous Decisions

Decision	MLP	RNN
Correct Decision to turn MIMO on/off	88.21%	90.17%
Erroneous Decision to turn off MIMO	3.61%	2.55%
Erroneous Decision to keep MIMO on	8.18%	7.28%

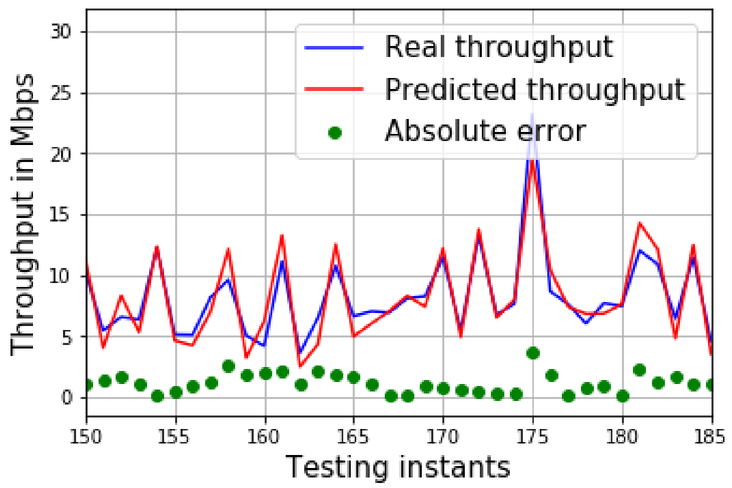

FIGURE 4.4: Actual Throughput, Predicted Throughput and the Absolute Error During a Portion of the Testing Phase for RNN.

4.4 Conclusion

In this chapter, we consider the use of ML-based approaches for energy saving in mobile networks. More specifically, these approaches help in deciding on the disabling MIMO schemes if the SISO mode can achieve a network user desired QoE. We propose two different architectures for ML networks to estimate the user DL average throughput under SISO base stations, namely, multi-layer perceptron (MLP) and recurrent neural network (RNN). We compare the estimated SISO user DL average throughput to a predefined threshold that represents the desired QoE. If SISO can achieve the target QoE, MIMO schemes are turned OFF, otherwise, MIMO schemes are enabled. We train our models based on real mobile network data. Results reveal the efficiency and effectiveness of proposed approaches as they allow for real-time, data-driven solutions. These are unique features of our proposed solutions as compared to the current practices in mobile networks.

Chapter 5

Deep Reinforcement Learning-Based Management Technique for Cellular Networks

Cellular networks have been facing a lot of rapid changes lately [77]. For instance, the number of users varies drastically from time to time depending on the current events. The coverage requirement varies as well due to fast urban changes. Moreover, energy conservation has become a universal need in the past decade [78], [79]. Cellular networks need to be adaptive enough to cope with the rapid ongoing changes and meet the new requirements. Furthermore, a certain criterion must be set to make sure that these rapid network changes will not cause disturbance in the network. In other words, a network equilibrium is a must meet requirement. This will give the network immunity towards the surrounding environment changes and allow it to self-heal from any expected problems.

In this chapter, the main target is to reach a *stable state* in the cellular network after any variation in the surrounding environment. We measure this stable state by meeting five main targets (which can be contradicting):

1. Maximize sum throughput of the network.

2. Achieve load balancing among cells.
3. Minimize numbers of blocked users.
4. Satisfy Quality of Experience (QoE) for users.
5. Minimize energy consumption.

We try to reach this stable state by controlling three parameters/features:

1. Relative Cell Individual Offset (CIO) between eNBs (will be denoted next by CIOs) which will affect load balancing and user blocking.
2. The eNodeB (eNB) transmission power (will be denoted next by P_n). This parameter has a direct effect on all the requirements mentioned above.
3. MIMO feature on/off. This will affect sum throughput, QoE, and energy consumption.

To control these parameters, we use a Reinforcement Learning (RL) based approach [25]. RL consists mainly of an agent (decision maker) and a surrounding environment. The process depends mainly on trial and error. The agent selects a decision from a pre-defined action space and based on the effect of this action on the environment it updates its decision-making policy. The target of the agent is to explore the action space as much as possible and to maximize the reward (a pre-determined function) of its decision in the long run.

To this end, we face two main issues. The first one is the fact that *trial and error* is almost impossible in a live cellular network since it will directly impact the QoE of the end-user. The second issue is the hybrid action space (having a mix of continuous and discrete control parameters). P_n and CIOs values are selected from a continuous action space while MIMO on/off action is selected from a discrete action space.

To solve the first issue, we use a modified version of the simulated cellular network in [80] using NS3 simulator which is a very powerful tool to simulate a network with the capability of extracting KPIs. This simulated network will be the environment for the RL agent (decision maker). The agent will apply its decision to the simulated cellular network and we can extract KPIs to calculate the reward as feedback to the agent.

To solve the second issue, we adopt two existing RL algorithms in a layered fashion, as will be explained later. The first one is the Double Deep Q-Network (DDQN) [27] which is used for discrete action spaces (MIMO on/off in our case). The second one is Twin Delayed Deep Deterministic Policy Gradient (TD3) [29] which is used for continuous action space (P_n and CIOs in our case).

The main contribution of the work in this chapter can be summarized as follows:

- Joint optimization of eNBs' transmission power, relative CIOS and MIMO scheme using an RL-based framework to automate network management and train the cellular network to reach a stable state of the network such that the sum throughput is maximized.
- Present a layered approach to handling hybrid action space. We give a detailed algorithm to enable the agent to take decisions extracted from discrete and continuous action spaces and apply them to the environment.

5.1 Literature Review

As mentioned earlier, the main target of this work is to train the cellular network to be able to reach a stable state from five perspectives: Maximized sum throughput, Load Balancing, Minimum number of blocked users, User Equipment (UE) QoE, and Energy

Consumption. This will enable the network to be *Self Healing* and more *immune* to the changes that occur occasionally in the network.

As far as came to our knowledge, this is the first work to address three network management controls simultaneously. These are: Transmission Power, CIOs, and MIMO scheme. However, other works in literature have addressed these controls separately or two of them together.

Power control has been discussed in earlier works in literature. In [81], the authors discuss a dynamic operation of cellular base stations. That is: switching off redundant base stations during periods of low traffic. This dynamic operation was proven to provide significant energy savings in cellular networks. Also, in [82], the authors aim to design an efficient online scheduling algorithm to minimize energy consumption while meeting an acceptable end-user experience. Moreover, in [18], the problem of energy optimization in mobile networks is considered by enabling the MIMO feature (the most energy-consuming feature) only when necessary. This is done by using a neural network-based algorithm to learn the behavior of SISO networks and decide whether or not it will be sufficient to reach a satisfactory user Quality of Experience (QoE).

Load balancing has been covered in previous works as well. In [83] and [84], the authors design a machine learning framework for optimizing cell parameters that can achieve more balance in the traffic. Their target is controlling the CIO of neighboring cells to force handover from congested cells into lighter load cells. In [80] and [26], the authors aim to control energy and CIOs simultaneously to achieve load-balancing such that the DL sum throughput is maximized while the number of uncovered users is minimized.

Reinforcement learning (RL) has been applied in cellular networks in many previous works. A survey of the applications of deep RL in cellular networks as an emerging tool to address various challenges can be found in [24]. In [85], the authors present an RL-based algorithm to solve a non-convex constrained SINR optimization problem aiming to enhance the performance in practical cellular environments. The authors in [86] present an RL-based scheduler that aims to optimally schedule IoT traffic to dynamically adapt to traffic variation. Moreover, a deep RL approach is considered in [87] for power control in multi-user wireless communication cellular networks aiming for maximizing the sum rate of the network by coordinating the inter-cell interference. In [26, 80, 83, 84], RL was adopted aiming to reach load balancing in the network such that the sum throughput is maximized.

From another perspective, as mentioned earlier, the action space in this work is a mix of continuous and discrete sets. Handling a hybrid action space exists in literature in different settings. For instance, in [30] , the authors are interested in the hybrid action space of video games. They adopt the Soft-Actor-Critic (SAC) algorithm in a parameterized way. They duplicate the continuous components that depend on other discrete components (one for each discrete component) as long as the model dimension remains reasonable. Another example is in [31] in which hybrid control in Robotics is the main concern. The authors build on top of the algorithm that is called Maximum aposteriori Policy Optimisation (MPO). Also in [32], the authors deploy a multi-agent algorithm with a hybrid action space as an extension of parameterized DQN.

The **novelty** of this work lies in two main points. First, we aim for controlling three different parameters/features -simultaneously- to reach a state of "equilibrium" in the network. Those three features, as mentioned above, are the transmission power, CIOs, and MIMO status. Most work in literature aims to control one or two features at most. We, on the other hand, aim to have more holistic control over the network.

The second point is the hybrid nature of the action space. CIO control and Transmission power control belong to a continuous action space. However, turning MIMO on/off belongs to a discrete action space. In this work, the hybrid action space is handled in a hierarchical approach. We propose a new RL architecture that allows the agent to take its decision in two successive stages. This is lacking in the RL literature to the best of our knowledge. The presented scheme is simple, yet efficient and one of its main advantages is that it requires no modification in the core of the used techniques (DDQN and TD3) as will be explained later.

5.2 Problem Description and Proposed Approach

5.2.1 System Model

The model adopted in this study is close to that in [80]. Consider an LTE cellular network that consists of N eNBs and U User Equipment (UEs)[1]. It is worth mentioning that this work is considered as a starting point for using RL to jointly optimize discrete and continuous network parameters. It can be extended to real network as the scenario adopted in [84] in which authors used real eNBs locations and real users mobility model.

eNodeBs:

Each eNB sends its transmission in the DownLink (DL) with a power level $P_n \in [P_{min}, P_{max}] dBm$, where $n = 1, 2, .., N$. At time $t = 0, 1, 2, 3, ...$, each UE measures the Signal-to-Interference-plus-Noise-Ratio (SINR) of near eNBs and attaches to the cell that results in the highest one.

[1]Although this work is applied to LTE cellular networks, our model is general and can be applied to 5G cellular networks and beyond.

An eNB can be over-utilized or under-utilized. This is determined according to the value of the eNB utilization ρ_n:

$$\rho_n = \frac{\sum_{i=1}^{U_n} K_{i,n}}{B_n / B_{PRB}}, \tag{5.1}$$

where, U_n is the number of UEs served by the n^{th} eNB, $K_{i,n}$ is the number of Physical Resource Blocks (PRBs) that serves the i^{th} user in the n^{th} eNB, B_n is the bandwidth of the n^{th} eNB and B_{PRB} is the bandwidth of one PRB (=180 KHz in LTE). Note that ρ_n is the ratio of the total number of required PRBs of n^{th} eNB (to serve the attached users) to the maximum number of PRBs that it can offer. Therefore, $\rho_n < 1$ means that the eNB is under-utilized while $\rho_n > 1$ means that the eNB is over-utilized. Under-utilization allows the eNB to serve all its attached users with satisfying rates while this will not happen in case of over-utilization.

Every eNB can have the MIMO feature turned ON or OFF (depending on the network manager decision). Turning the MIMO feature ON has a big effect on the gain received at the receiving end, thus, a better Quality of Experience (QoE) can be achieved at the UE. However, MIMO is one of the most energy-consuming features in the eNB. When the MIMO feature is turned ON, a *scheduler* decides whether to use the multiple antennas to apply Spacial Multiplexing or Transmit Diversity depending on the CQI of the UE. Nodes with higher CQIs will be assigned Spatial Multiplexing to give them more enhancement. On the other hand, nodes with lower CQIs (far nodes) will be assigned Transmit Diversity to increase their SINR and offer them a satisfying performance.

UEs:

The u^{th} UE moves with a constant velocity v_u. It regularly searches for a better cell (according to the higher SINR) and attaches to the better cell if found. Moreover, The channel quality indicator (CQI) ϕ_u of the u^{th} UE is reported to the associated cell periodically. ϕ_u is based on the perceived SINR and Block Error Rate (BLER) at the UE. The CQI is a discrete measure that represents the quality of the channel. $\phi_u \in 0, 1, , 15$. When $\phi_u = 0$, this means that the u^{th} UE is out of coverage (blocked) [88]. A higher CQI value corresponds to higher channel quality [89].

When a certain UE is attached to cell i, it might require handover to another neighboring cell j if [88]:

$$RSRP_j + \theta_{j-i} > Hys + RSRP_i + \theta + i - j, \tag{5.2}$$

where, $RSRP_i$ and $RSRP_j$ are the measured Reference Signal Received Power from eNBs i and j respectively. θ_{i-j} is the CIO value of the eNB i with respect to eNB j and θ_{j-i} is the CIO value of the eNB j with respect to eNB i. Hys is a hysteresis value to minimize repeated handover requests that might occur due to minor signal quality fluctuations.

CIO (θ_{i-j}) is a cell-specific parameter that has a direct effect on handover between cells. If it is set to zero, the UE attaches to a certain cell depending only on RSRP. If it is set to a different value, it makes the measured RSRP of cell i appear stronger (or weaker) when compared with the measured RSRP of cell j.

5.2.2 Reinforcement Learning Framework

Assume that the network has a central network manager that takes the role of the RL agent. The interaction between the agent and the network occurs at discrete time steps with constant durations. At each time step, the agent observes the state of the environment and decides to be applied to the environment aiming to maximize a certain reward function. As a result of the agent decision, feedback from the environment gives the agent a hint about the effect of the taken decision. This feedback is the reward function and a new state of the environment. This process is repeated until the reward function converges, which means that the agent has reached a policy that maximizes the predetermined reward function.

The mapping of the RL algorithm to the proposed problem can be explained in brief as follows:

- Agent: Central network manager.
- Environment: Cellular Network under consideration (simulated).
- State: A subset of the network KPIs which are:
  - Resource Block Utilization (RBU) $(Bl(t))$: The fraction of used PRB blocks that serve the users of each cell. It is an N-length vector. It is a representation of how congested each cell is.
  - A vector of total DL throughput of each cell $(R(t))$: It is a representation of the eNB performance. Each element can be expressed as $R_n(t) = \sum_{u_n=1}^{U_n} R_{u_n}(t)$, where $R_{u_n}(t)$ is the measured throughput of user u_n in the n^{th} eNB.
  - Number of active users of each cell $(C(t))$: This measures the number of users that are not idle in a certain time step.
  - Modulation and Coding Scheme (MCS) Matrix $(M(t))$: It is a matrix that gives the fraction of users with a certain MCS. It is considered a representation of the quality of channels.

  Finally, we can say that the state is the concatenation of the above vectors (after reshaping $M(t)$):

$$S(t) = [Bl(t)^T R(t)^T C(t)^T M(t)^T]. \tag{5.3}$$

- Action: The features that the agent has control over:

  - Relative CIO values between every two neighboring cells. $\theta_{ij} = -\theta_{ji} = \theta_{i-j} - \theta_{j-i}$. This action belongs to a continuous actions space $[-\theta_{max}, \theta_{max}]$.
  - Transmission Power of each eNB P_n. It belongs to a continuous action space. The agent chooses a value from the set $[-P_{max}, P_{max}]$ to add to a constant value of 30 dBm.
  - Turning MIMO feature ON/OFF for each eNB m_n. The whole MIMO action is $[m_1...m_N]$. This action is selected from a discrete set of size 2^N since each eNB has a decision of $m_n = 0$ (MIMO OFF) or $m_n = 1$ (MIMO ON).

- Reward Function: The main target of the RL agent is to reach the policy that maximizes the expected reward function on the long run [26]. That is:

$$\max_{\pi} \lim_{L \to \infty} \sum_{t=0}^{L} E[\lambda^t r(t)], \tag{5.4}$$

where, $r(t)$ is the reward function, λ is the discount factor that determines the significance of the reward future expected values and π is the policy. The selected reward function in this work is:

$$r(t) = \sum_{n=1}^{N} R_n(t) - \eta \overline{R}(t) \sum_{u=1}^{U} \mathbf{1}(\phi_u = 0) - \mu \sum_{n=1}^{N} m_n, \tag{5.5}$$

where, η and μ are hyper-parameters that are determined by trial and $\overline{R}(t)$ is the average user throughput at time instant t. This reward function consists of three terms. The first term is the sum throughput of the network (to be maximized). The second term is the sum throughput of the blocked users scaled by a hyper-parameter. This term is to be minimized (a penalty) that's why it is subtracted. This penalty is scaled by η to control how significant it is to the agent. The last term is also a penalty. It is the number of eNBs that have the MIMO feature turned ON. It is also scaled by another hyper-parameter μ.

Thus, we can say that the objective of the agent is to reach the policy (sequence of action) that maximizes the total network throughput, minimize the number of out-of-coverage users and minimize the consumed extra energy due to turning MIMO ON.

Choosing the reward function to be the total throughput with no penalties might let the agent choose to keep only the users with high rates and alter CIOs or reduce power levels to cause handover of the edge users to a poorer performance cell. That's why putting a penalty on the number of out-of-coverage users is important. The other penalty is put to limit the consumed energy due to turning MIMO ON. Without this penalty, the agent would just choose to turn MIMO ON all the time with no regard to its energy consumption.

The aforementioned RL framework enables us to control power levels, CIOs, and MIMO feature. These controls have a direct and simultaneous effect on handover processes, coverage, QoE, and energy consumption. By using the proposed algorithm, explained next, we can take the cellular network to a stable state: Load balanced, minimum out-of-coverage users, satisfying QoE and minimum energy consumption such that the sum throughput of the network is maximized. The main advantage of this stable state is that it is *learned*, which means that it can be re-gained after any network change. This means a more immune cellular network.

5.2.3 Proposed Algorithm

In this chapter, we propose an RL framework that consists of an environment (cellular network) and an agent (central network manager). The agent observes a state (a subset of network KPIs) and consequently, it takes a decision (action) to control some network features.

The main issue is having a hybrid action space. That is: two actions belong to a continuous set (relative CIOs and power levels), while one action belongs to a discrete set (MIMO feature on/off). To solve this issue, the agent takes the decision in two stages.

- **First Stage:** The agent observes the state and takes the action of MIMO on/off based on the DDQN technique [27]. The action is taken from the discrete set $m_n \in /0, 1/$ (for each eNB). Note that the action of the first stage is not applied to the environment until the end of the second stage.
- **Second Stage:** The first stage action is augmented with the state being observed and then it takes the CIO and power level actions based on the TD3 technique. They are selected from the continuous intervals $[-\theta_{max}, \theta_{max}]$ and $[-P_{max}, P_{max}]$ respectively.

After the two stages, the augmented action $A_{aug}(t) = [A_C(t), A_P(t), A_M(t)]$ is applied to the environment. Where: $A_C(t) = \theta_{ij}$ (for all neighboring cells), $A_P(t) = [P_0, P_1, ..., P_N])$ and $A_M(t) = [m_0, m_1, ..., m_N]$. Note that as the agent explores the whole action space, therefore, the effect of the different combinations of the first stage action (MIMO on/off) and second stage action (Relative CIOs and Power Levels) is learned.

An overview of the proposed scheme can be seen in Fig. 5.1 and it can be described in more detail in Algorithm 4.

It is worth mentioning that turning on the MIMO feature for a certain eNB doesn't mean that spatial multiplexing will be applied for all users attached to this eNB. There is a *scheduler* applied for each eNB that decides which users to apply Spatial Multiplexing (SMux) and which to use Transmit Diversity (TxD) depending on the channel of each

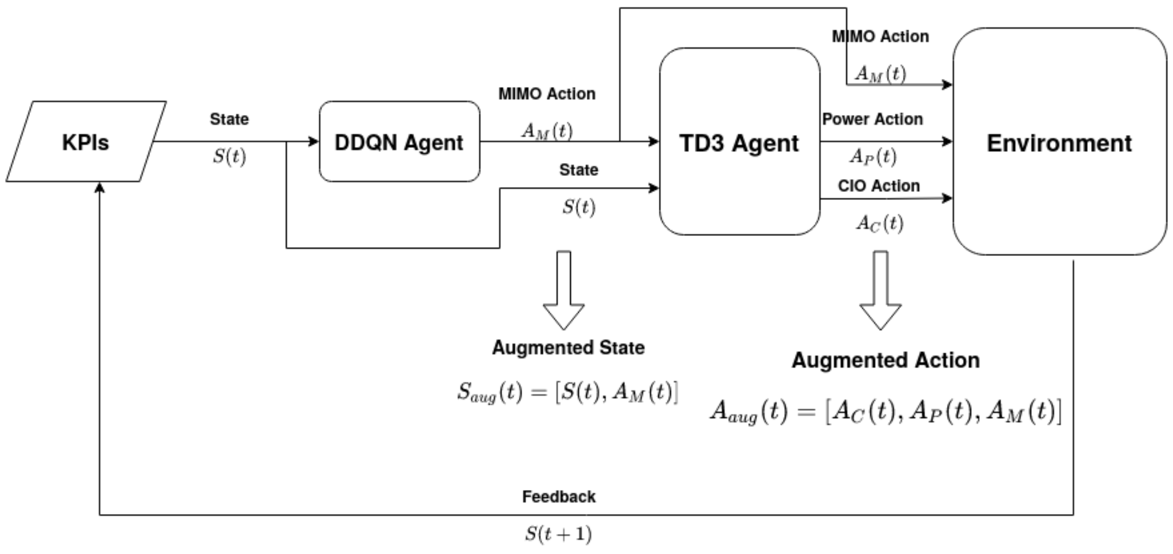

FIGURE 5.1: An Overview of The Decision Making Process.

user. NS3 has SISO as the default running scheme and it leaves absolute liberty to the user to turn MIMO modes (SMux or TxD) on or off (i.e. Scheduler doesn't have a role in selecting the suitable MIMO mode). The main concern is that turning SMux for users with low CQI will only make things worse. TxD will be more suitable here to enhance the channel of less fortunate users. To solve this problem, we created a simple scheduler that applies SISO when the agent chooses $m_n = 0$ but when the agent chooses $m_n = 1$, the CQI of each attached user is studied to select SMux for users that have $CQI \geq 7$ and select TxD otherwise. The CQI threshold is determined such that TxD is applied for users that use QPSK and SMux for users that have higher MCS.

5.3 Performance Evaluation

Consider a cellular network consisting of 3 eNBs and 42 UEs as in Fig. 5.2. Each cell has 10 users centralized around the eNB. On the edges between every two nodes. there are 4 edge users. The inter-node distance is 500 meters. UEs have a random mobility

Algorithm 4 Proposed RL Framework

```
1: Determine Reward Function.
2: Reset all values.
3: repeat
4:     procedure STAGE ONE
5:         Observe State ($S(t)$).
6:         Select MIMO feature decision (DDQN) ($A_M(t)$).
7:         Create a new augmented state ($S_{aug}(t) = [S(t), A_M(t)]$).
8:     end procedure
9:     procedure STAGE TWO
10:        Observe state ($S_{aug}(t)$).
11:        Select relative CIO and power level actions (TD3) ($[A_C(t), A_P(t)]$)
12:        Apply augmented action to the network $A_{aug} = [A_C(t), A_P(t), A_M(t)]$.
13:    end procedure
14:    Calculate Reward.
15:    Calculate next state.
16: until Reward Function Converges
```

pattern in boxes around their starting points. UEs are assumed to be active all the time.

The environment, which is represented by the described cellular network, is simulated using NS3 simulator (LENA module) [90]. The agent is simulated using Python [91]. The interface between the agent and the environment is implemented using NS3gym [92]. This interface is responsible for:

1. Applying agent actions to the environment.

2. Returning the reward and new environment state to the agent.

After applying the agent action, the environment is simulated using the control parameters sent by the agent action. The reward is calculated using (5.5).

Simulation parameters are summarized in Table 5.1.

Next, two scenarios are evaluated. The first one is a cellular network consisting of 3 identical cells. Each contains 10 users close to the eNB (center users U_c). Each edge between two cells contains 4 users (edge users U_e). The second scenario is a cellular network consisting of 3 unidentical cells. First cell has $U_c = 18$ (congested), second cell has $U_c = 9$ (medium utilization) and third cell has $U_c = 3$ (uncongested).

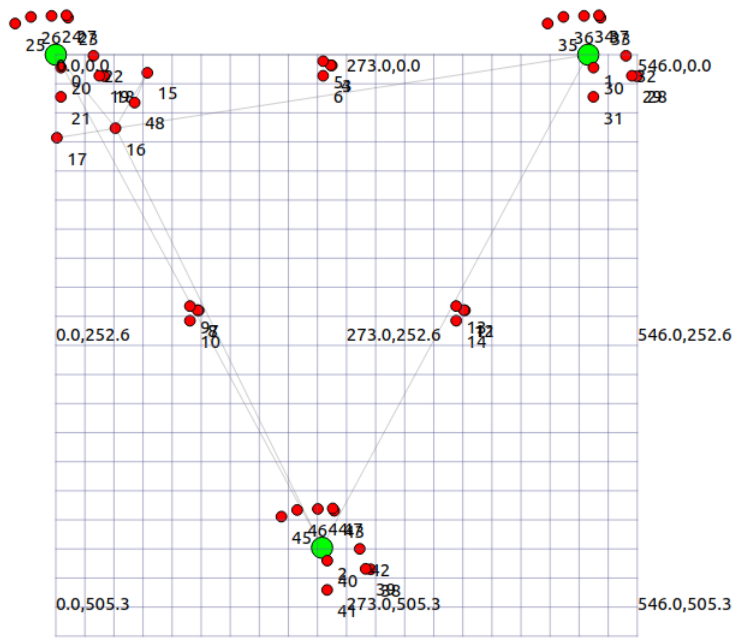

FIGURE 5.2: Cellular Network Topology (Green Nodes are eNBs and Red Nodes are UEs)

TABLE 5.1: Simulation Parameters

Parameter	Value
Number of eNBs (N)	3
Inter-eNB distance (d_{ij})	500 m
eNB antenna height	30 m
eNB antenna Pattern	Omni-directional
UE mobility model	Random walk (3 m/s)
UE antenna height	1.5 $\sim$ 2 m
Handover hysteresis	3 dB
System bandwidth	20 MHz
Pathloss model	COST Hata
UE traffic model	CBR (1 Mbps)
Basic eNB transmission power (P_{n_0})	30 dBm
Penalty on blocked users (η)	$\in [0, 2]$
Penalty on applying MIMO (μ)	$\in [0, 10]$
Training steps	8,000
Steps per episode	255
Step time	200 ms

For both scenarios, the target is to find the optimum policy to control relative CIOs, eNBs transmission power, and MIMO feature such that the network sum throughput is maximized.

5.3.1 First Scenario: 3 Identical Cells

In Fig. 5.3, the network sum throughput (in Mbps) is plotted against the number of episodes of the training phase. We compare 2 different settings: RL agent and Baseline (BL). The RL agent is the proposed approach in this paper. It is evaluated for different values of the hyper-parameter μ which scales the MIMO energy penalty. It is also evaluated with MIMO always on and with MIMO always off. BL is the setting in which the agent has no control over any network feature. It is evaluated with MIMO always on and with MIMO always off as well. We observe that the sum throughput increases during the training phase until it converges. It is noticed that as μ increases, the agent's desire to turn MIMO on decreases (since it negatively affects the reward function). Therefore, the sum throughput for smaller μ values is higher. However, reducing μ means increasing the consumed energy as a result of turning the MIMO feature on.

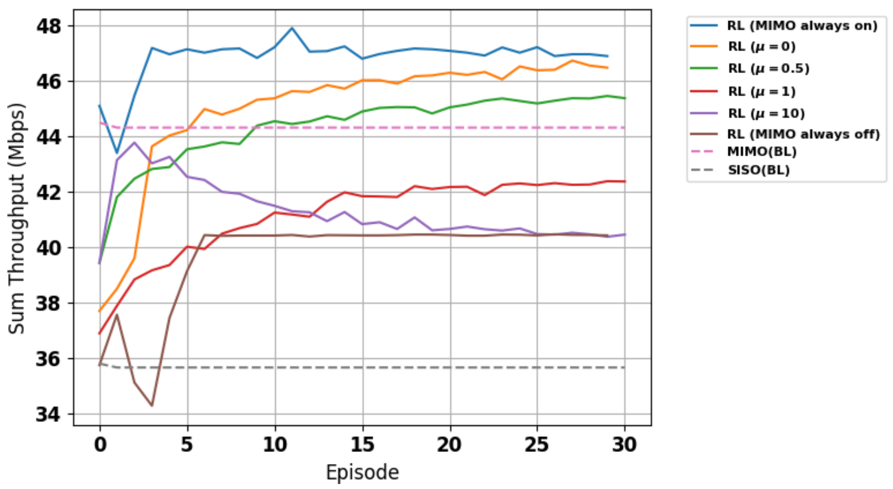

FIGURE 5.3: Effect of MIMO Energy Penalty (μ) on Sum Throughput

We can see also that for all values of (μ), the sum throughput is higher than the Baseline (BL) methodology in which the agent has no control over the network features. We also note that the RL agent that has control over the MIMO feature outperforms the RL agent that has MIMO turned off by default (SISO). It also approaches a close performance to the RL agent that has MIMO turned on by default (but with less energy as we can see later).

Furthermore, we can see in Fig. 5.3 that turning on MIMO with no other control on any network feature (BL) gives a higher sum throughput than turning MIMO off (with and without control over other features). However, our proposed scheme with $\mu = 0, 0.5$ outperforms BL with MIMO on but it consumes less MIMO energy.

It is worth mentioning that at $\mu = 10$, the sum throughput curve increases at first and decreases afterward. This is because the first portion of the learning phase is mainly for the exploration of the action space. This means that the agent is trying most of the possible actions to examine their effect on the environment. Turning MIMO on in this case ($\mu = 10$) will probably increase the sum throughput but it will also increase the penalty (which is scaled by 10). Therefore, the reward will decrease, and eventually, the agent will learn that turning MIMO off is more beneficial (in this case) to the reward (penalized sum throughput). Since we plot only part of the reward (sum throughput), the behavior shown in the curve is logical.

We can see from Fig. 5.4 the percentage of time MIMO is turned on for different values of (μ) after convergence. Please note that these percentages are considered a measure of the percentage of consumed energy compared to turning MIMO on as a default setting.

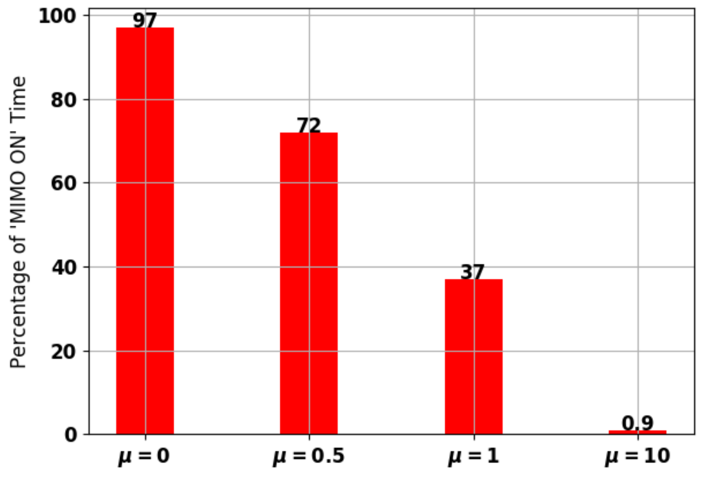

FIGURE 5.4: Effect of MIMO Energy Penalty (μ) on Ratio of Time MIMO is Turned on

We can give more focus on the two extreme cases ($\mu = 0$) and ($\mu = 10$). The first one is having no energy penalty. The sum throughput approaches the MIMO curve and MIMO is turned on 97% of time. The second one approaches the SISO curve and MIMO is turned on 0.9% of time.

From another perspective, we needed to report the minimum throughput and the number of unsatisfied users to make sure that the proposed scheme works in their favor as well and not only from a collective point of view. We can see in Fig. 5.5 and Fig. 5.6 that our system outperforms BL scheme regarding the less fortunate users. It is worth mentioning that the unsatisfied users in Fig. 5.6 are determined as the users with individual throughput ≤ 1 Mbps. This threshold is determined based on the nature of the network, the scenario, the Bandwidth used, and the number of users. It might vary depending on the criterion determined by the operator.

We need to point out that the fluctuations in the minimum throughput curves are since this is the worst case, not the sum of a group of users. This worst-case might vary by a very considerable amount which causes these fluctuations.

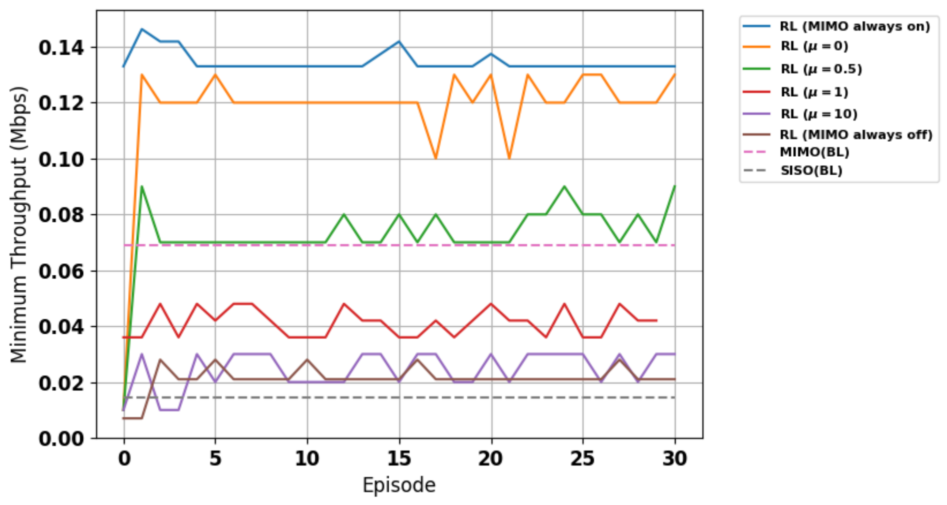

FIGURE 5.5: Effect of MIMO Energy Penalty (μ) on Minimum Throughput

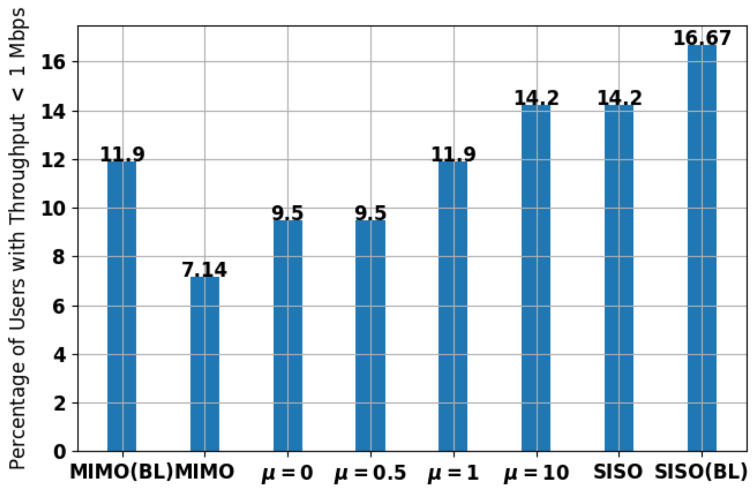

FIGURE 5.6: Effect of MIMO Energy Penalty (μ) on Percentage of Unsatisfied Users

To study the effect of the blocked users penalty, we varied the hyper-parameter that scales it (η) and plotted the percentage of unblocked users in the network and the sum throughput in Fig. 5.7 and Fig. 5.8. We notice here that there is a trade-off between the sum throughput and the percentage of unblocked users. As (η) increases, the percentage of unblocked users increases but the sum throughput decreases. This is because trying to achieve a higher sum throughput might lead to block users with low CQIs to other cells (because they will consume resources and will not contribute much in the sum throughput) and give more resources to users with high CQIs (which will benefit the sum throughput of the network).

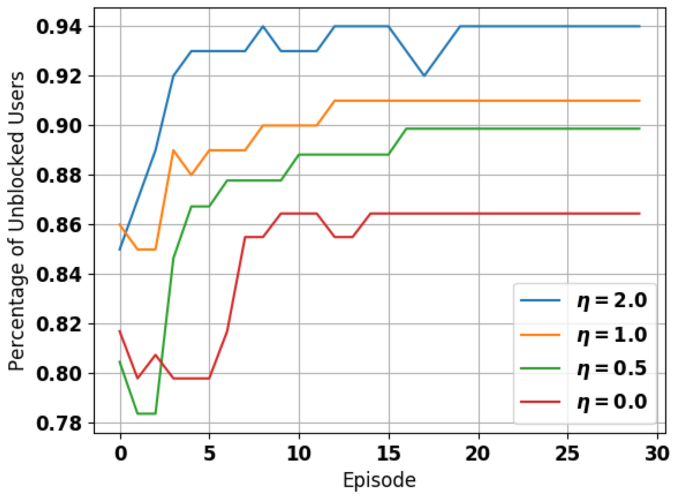

FIGURE 5.7: Effect of Blocked Users Penalty (η) on Percentage of Unblocked Users

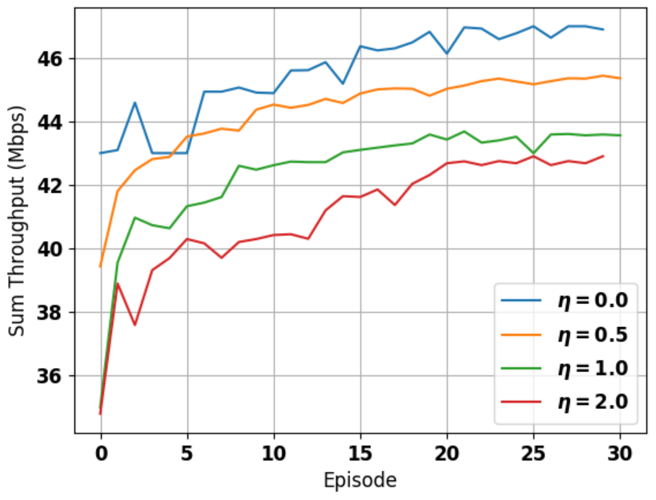

FIGURE 5.8: Effect of Blocked Users Penalty (η) on Sum Throughput

5.3.2 Second Scenario: 3 Unidentical Cells

Here, we check the performance of the proposed algorithm in another type of network which consists of 3 Unidentical cells (congested ($U_c = 18$), uncongested ($U_c = 3$) and medium utilization ($U_c = 9$)). Moreover, $U_e = 2$ at each edge. We can consider this second scenario as a change in the number of users in the network due to an event for instance. The proposed algorithm performs similarly to the first scenario.

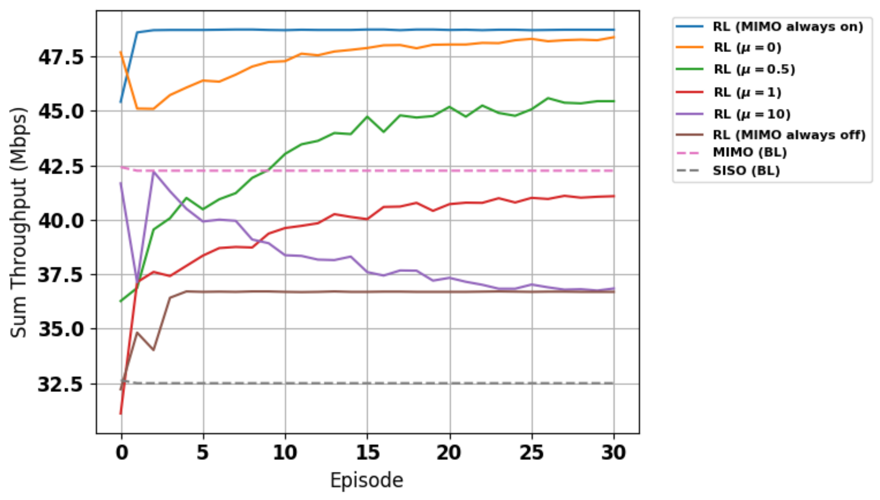

FIGURE 5.9: Effect of MIMO Energy Penalty (μ) on Sum Throughput (Unidentical Cells)

In Fig. 5.9, the sum throughput is plotted vs the training episodes. As expected, the sum throughput increases during the training phase until it converges. We can see the effect of the MIMO energy penalty (μ). In this scenario as well, the larger the penalty, the less the sum throughput (due to less freedom to turn MIMO on). Moreover, we notice here also the anomaly of the curve at ($\mu = 10$) which increases at first and decreases afterward. This replicates the case of the first scenario and it is expected due to the special nature of this extreme case as explained earlier.

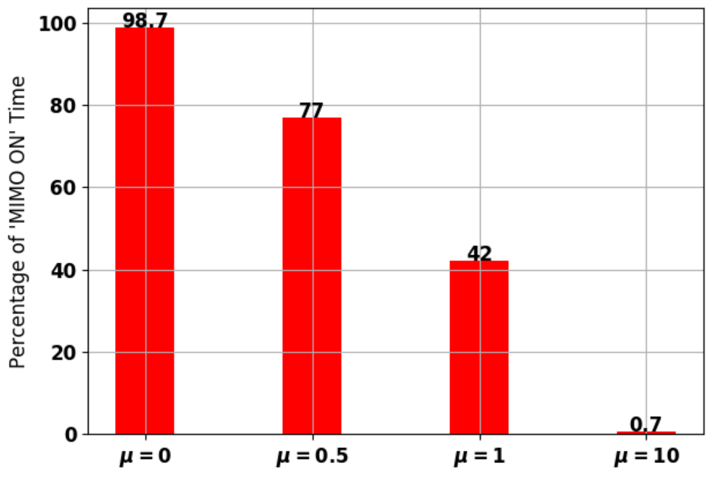

FIGURE 5.10: Effect of MIMO Energy Penalty (μ) on Ratio of Time MIMO is Turned on (Unidentical Cells)

The ratio of time MIMO is turned on for each value of μ can be studied in Fig. 5.10. It follows a similar pattern as the first scenario. More energy can be saved by increasing μ but this leads to less sum throughput.

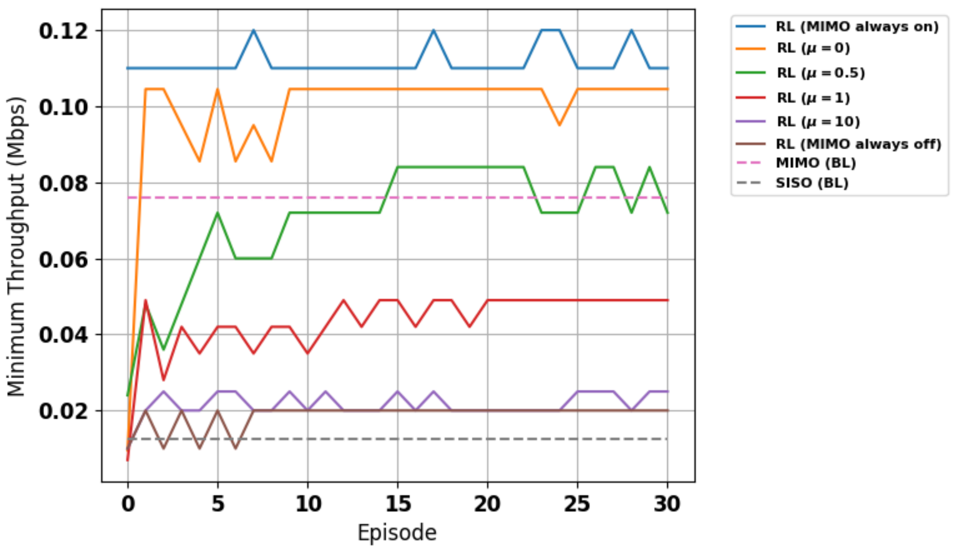

FIGURE 5.11: Effect of MIMO Energy Penalty (μ) on Minimum Throughput (Unidentical Cells)

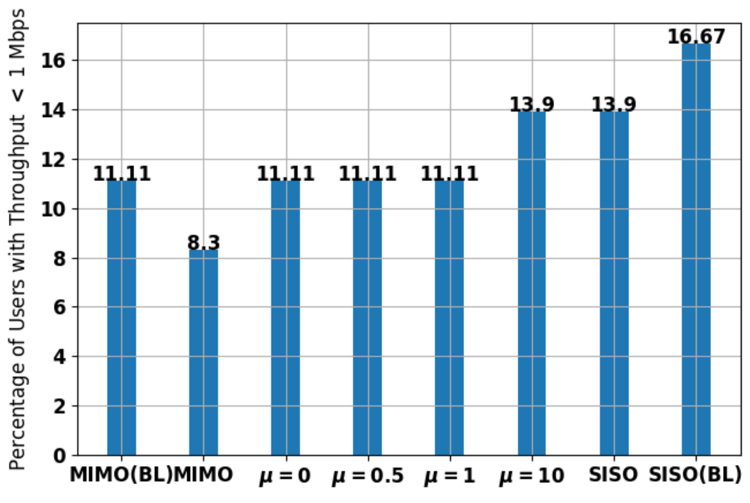

FIGURE 5.12: Effect of MIMO Energy Penalty (μ) the Percentage of Unsatisfied Users (Unidentical Cells)

Reporting the minimum throughput of the network and the percentage of unsatisfied users for each value of μ can be seen in Fig. 5.11 and Fig. 5.12. The RL agent outperforms the BL with the SISO feature for all values of μ and it outperforms the BL with the MIMO feature for smaller values of μ.

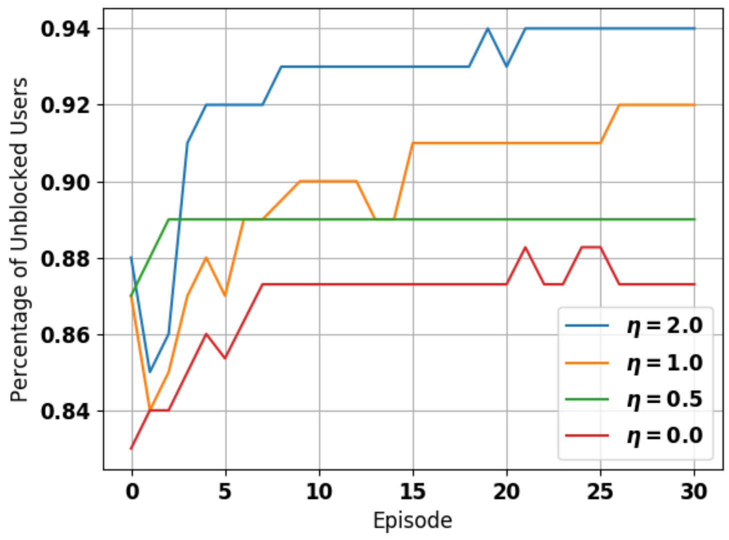

FIGURE 5.13: Effect of Unblocked Users Penalty (η) on the Percentage of Unblocked Users (Unidentical Cells)

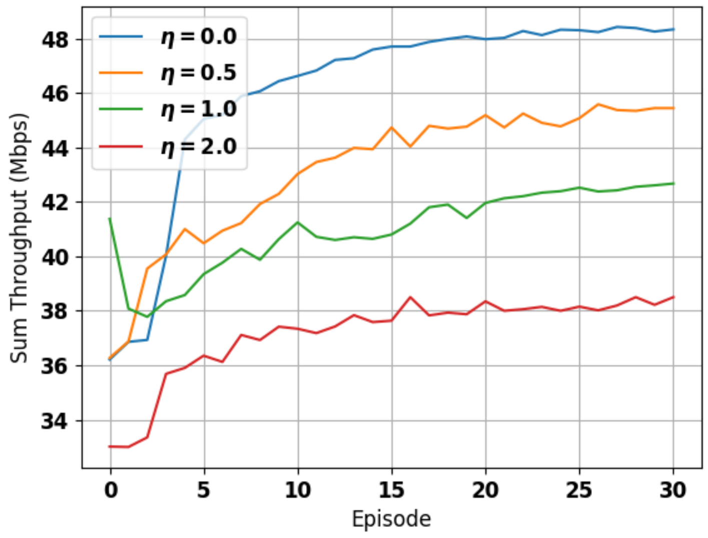

FIGURE 5.14: Effect of Unblocked Users Penalty (η) on Sum Throughput (Unidentical Cells)

In Fig. 5.13 and Fig. 5.14 we study the effect of the unblocked users penalty (η). As expected, η has a positive effect on the percentage of unblocked users but a negative effect on the sum throughput of the network. This behavior is because focusing on covering as many users as possible will probably lead to giving resources to less fortunate users (low CQIs) which will not contribute to the sum throughput with a considerable amount.

The results of the second scenario are proof that the proposed scheme delivers the network to a stable state with a maximized sum throughput after changes that occur in the network. The proposed scheme has the advantage of reaching a *learned* policy to control the network. Learning means that the stable state can be regained again after probable occurrences.

5.4 Conclusion

In this chapter, we present an RL framework to gain control over three main features in cellular networks. These are: Relative CIOs, Transmitted Power Levels, and MIMO scheme. This control allows the network to reach a stable state from the perspective of load balancing, the number of covered users, QoE, and energy consumption such that the network sum throughput is maximized. The advantage of this work is that it enables the network to regain this stable state, without the operator interference, after any change in the environment. Moreover, this work creates a layered approach that enables the framework to handle hybrid action space in a relatively easy yet efficient methodology. The results show that the proposed scheme can achieve higher sum throughput than the Baseline system (no RL control). Also, larger percentage of satisfied users is achieved while energy consumption is optimized.

Chapter 6

Machine Learning-Based Technique for False Data Injection Attacks Detection in Industrial IoT

The rapid proliferation of the Internet of Things (IoT) technology in industrial enterprises has exposed the vulnerability of critical infrastructure to serious cyber-attacks. Industrial IoT (IIoT) has helped solve many intractable issues in the industry by allowing system self-controlling and offering real-time response systems. To ensure the successful roll-out of IIoT applications, security issues need to be well-studied and addressed for such applications.

Monitoring of industrial systems such as hydraulic, oil, and gas stations against cyber-attacks has received great attention recently [93]. One of the major types of attacks that can affect such systems is termed the "False Data Injection" (FDI) attack. FDI attacks present a serious form of cyber-attacks against industrial infrastructures. They corrupt sensor measurements to deceive the attacked industrial platform [94].

Machine Learning (ML) has been used in many applications to address security threats. In this work, we propose to use Autoencoders (AEs) as our ML tool for the detection of FDI attacks. Recently, AEs have gained a lot of interest to address many security issues and cyber-attacks in communication networks. AEs are neural networks that tend to learn a latent feature representation of the input, normally in a smaller dimension space. Another attractive feature of AEs is that they do not need "labeled" data for their training.

Since sensor networks are generally resource-constrained, the use of ML-based techniques for security purposes provides the additional advantage of optimizing the resource utilization for this purpose which can extend the network lifetime. Therefore, in this work, we propose the use of an ML-based technique that uses AEs for the detection of FDI attacks. The main benefit behind the used ML-based approach is its ability to deal with data that has no structure and no model can represent it. ML-based approaches can extract features from raw sensor measurements. As mentioned earlier, one of the main attractive features of AEs is the fact that they can be trained in an unsupervised fashion to learn important features of the data. The AEs can learn latent correlation structures in the data samples. In our work, we exploit the correlation in two dimensions, namely, time and space (i.e. "time" correlation between the same sensor data at different times and "spatial" correlations across the sensors). AEs can learn efficient, reduced-size feature representation of the data. With this feature representation, attacked data samples will show high dissimilarity between the AE input and the corresponding output.

The main contributions of the work in this chapter [20], can therefore be summarized as follows:

- We propose the use of AEs as a classification approach to detect false data injection attacks. AEs can learn hidden correlation structures in the data in an unsupervised manner that would allow them to detect corrupted data by assessing how far the corrupted data correlation structure is from the expected, "learned" correlation structure. Our proposed AE learns correlation in two dimensions: the time (same sensor data at different times) and the spatial (across sensors) dimensions. This would allow for a better representation of the correlation model at the hidden layers. To the best of our knowledge, this work is the first to propose the use of an AE-based scheme to detect SDFI attacks. Additionally, our proposed scheme has the potential of detecting other types of attacks without the need for any modifications to it.

- We propose the use of Denoising Autoencoders (DAEs) to clean the corrupted data, by recovering the expected correlation structure. Our results show the efficacy of our proposed data cleaning DAEs in recovering clean data from corrupted data. To the best of our knowledge, this is the first work to propose the use of denoising AEs to clean corrupted (attacked) data.

6.1 Literature Review

IIoT sensors' readings at pivotal industrial areas are critical subjects whose loss or malfunction due to attacks or otherwise can lead to considerable losses that may include the loss of human lives [95]. For instance, the Stuxnet attack on the Iranian nuclear system [96] has led to large losses and interruptions.

False Data Injection has been studied in various works in literature. In [97], the study demonstrates that an intruder can build an FDI attack without being detected by classical approaches such as state estimation (SE) and bad data detection (BDD) [98]. State estimation is one of the basic solutions in detecting FDI attacks in critical infrastructures. Relying on state estimation to achieve comprehensive sensing accuracy leads to measurements' alteration/fabrication [99]. The study in [97] highlights this kind of weaknesses. If an intruder is aware of the measurements' figures, he/she can falsify the devices' readings. To address this critical vulnerability, two recent studies

[100, 101], introduced machine learning (ML) using support vector machine (SVM) to efficiently detect FDI attacks. The study in [100] investigates the performance of supervised and semi-supervised machine learning approaches based on SVMs against several attacks. In their experimentation, they observed that the SVM solution performs better than the regular solutions that employ the state estimation (SE) approach for the detection of both observable and non-observable attacks. In addition, their results showed that the semi-supervised learning approaches are stronger to deal with the different data sparsity degrees than the fully-supervised learning approaches.

Using Machine Learning (ML) in security-related applications exists in literature in multiple works. For example, in [102], the authors present an analysis of ML techniques applied to the detection of cyber threats such as intrusion, malware, and spam. The goal is to study the current maturity of these solutions and to identify their main limitations. In [103], the authors present an ML-based technique for detecting cyber threats. The system focuses on differentiating between true positive and false positive alerts thus helping to rapidly respond to cyber threats. Another security threat, SQL injection, is studied in [104]. The authors in this survey study various machine learning algorithms used for the detection of SQL injection threats. An experimental analysis is performed in [105] of the ML methods for the Botnet DDoS attack detection. The studied methods are Support Vector Machine (SVM), Artificial Neural Network (ANN), Naïve Bayes (NB), Decision Tree (DT), and Unsupervised Learning (USML). In [106], the authors cover existing security threats on ML techniques and give a survey on them from the aspects of the training phase and the testing phase. They also categorize the defensive techniques of machine learning into security assessment mechanisms, countermeasures, data security, and privacy.

In [101], the study presents two methods to detect false data injection. The first method uses supervised learning over labeled data and a trained SVM. The other method does not need to train data to detect the measurements' deviation. In both schemes, they applied principal component analysis (PCA) to protect the data on a low-dimensional space based on the observation that normal data and attacked data tend to be separated after projection. Finally, they concluded that SVM performs better than the classical attacks detection methods, such as state estimation (SE) and bad data detection (BDD) classical approaches, once it has an appropriate number of trained data sets.

In [107], the study proposes using AEs to learn latent features and to reduce the size of the feature vector to be fed to some machine-learning-based classifier. The work in [107] considers two applications, namely, network anomaly intrusion detection and malware classification. In [108], the study proposes different anomaly detection models based on different deep neural network structures, including AEs, convolutional neural networks, and recurrent neural networks. In [109], sparse AEs are used to learn a new feature vector, of reduced size, to be fed to an SVM used for detecting network intrusion attacks. AEs are used to efficiently reduce the size of the feature vector in an unsupervised fashion. In [110], a DAE feeding a compact multi-layer perceptron (MLP) is used for network intrusion detection. DAEs are also used in [33] to extract some reduced size robust features of data. The same authors in [111] use stacked AEs to enhance the classification capability of neural networks. AEs are again used to learn important features to be fed to the MLP stage of the detection algorithm.

Our approach differs from those using AEs for similar purposes (e.g. in [107–110]) in that we propose to use AEs as classifiers not just to learn some reduced size features. This has the benefit of simplifying the training process since there is no need for labeled data for training. Another important feature of the use of AEs as classifiers is their ability to detect other attacks since AEs are trained with unlabeled clean data. Any attack that distorts the data correlation structure would be identified by an AE-based detector. Other conventional detectors (classifiers) must be trained with labeled data sets for every attack that they have been designed to detect, and they are not guaranteed to detect any other attacks for which they were not trained. This, in addition to simplifying the training of AE-based detectors, enhances the performance of AE-based schemes in comparison to other attack detection algorithms. Moreover, it is more natural to detect false data injection attacks using AEs as they can learn latent correlation structures in the data. Dissimilarity between the expected correlation model and the observed correlation model can lead to a more natural classification compared to, for example, SVM, where it is assumed that the two classes can be separated by a hyperplane in some feature space.

6.2 Problem Description

In this section, a description of the problem at hand is presented including the adopted model, the nature of the attacks, and finally the proposed detection technique.

6.2.1 System Setup

The system that is used in this work is the hydraulic system presented in [93]. The data set is a group of sensor readings scattered for the hydraulic system monitoring purpose. The data set was experimentally obtained (i.e. not simulated) with a hydraulic test rig. The system cyclically repeats constant load cycles (duration of 60 seconds) and measures process values while the condition of four hydraulic components (cooler, valve, pump, and accumulator) is quantitatively varied. The data set includes readings from temperature, pressure, volume flow, vibration, motor power, and cooling power sensors. The total number of sensors is 15 (6 pressure sensors, 4 temperature sensors, 2 volume flow sensors, 1 motor power sensor, 1 vibration sensor, and 1 cooling power sensor). The data set contains raw process sensor data (i.e. without feature extraction). The number of readings gathered from each sensor ranges from 132300 to 1323000. We use the least number of readings to avoid empty incomplete input vectors.

6.2.2 False Data Injection (FDI) Attack Description

The attack that this work aims to detect is the False Data Injection (FDI) attack, which is one of the most common cyber-attacks in IIoT. To adapt the use of our technique to the needs and nature of IIoT systems, we adopt the attack model presented in [112]. In this model, the intruder can alter and/or falsely inject single or several sensors' data at any time to have these false data within a valid range of authentic measurements.

The integrity attack on measured data is defined as follows. Let's define $\mathbf{z}_a$ as the recorded readings vector that may have some false data. Consequently, $\mathbf{z}_a$ can be described as $\mathbf{z}_a = \mathbf{z} + \mathbf{a}$, where $\mathbf{z} = [z_1, \ldots, z_m]^T$ is the clean measurements vector, and $\mathbf{a} = [a_1, \ldots, a_m]^T$ is the added/fabricated data vector. Therefore, the vector $\mathbf{a}$ represents the vector of the attack where the i-th element of $\mathbf{a}$, a_i, is of a nonzero value that reflects the intruder's ability to falsify the i-th reading and to substitute its corresponding authentic value with an alternative false value.

6.2.3 Proposed Attack Detection Algorithm

As mentioned earlier, in this study, an AE is used for attack detection due to its capability of capturing the structure of the data and learning the correlation between readings. Then, a DAE is used for cleaning the corrupted data previously detected by the AE. The process is shown in Fig. 6.1 which is a flowchart of the proposed scheme.

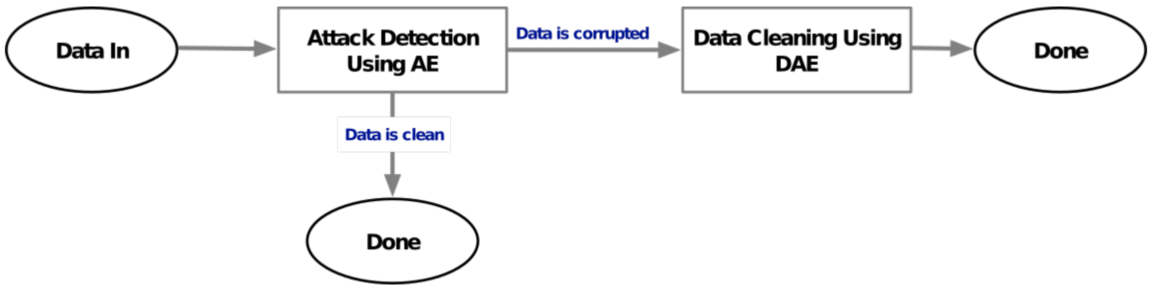

FIGURE 6.1: Flowchart of the AE-based Proposed Solution.

Autoencoder-based Detection Scheme

During the training phase, the AE is fed with the original data taken from the sensors. Target values are set to be the same data. By the end of this phase, the network becomes able to exploit the inter-correlation between entries to compress and then decompress back the input vector. Once the training phase is complete and the weights are set, the AE can then be used for false data detection.

When false data are fed to the AE, the network tries to compress and decompress the input vector. However, due to the lack of the expected correlation structure in the false data, the Mean Square Error (MSE) is larger than expected. To detect false data, the MSE is compared to a pre-determined threshold which is chosen as the mean of the validation MSE. Once an MSE value exceeds the mentioned threshold, an attack is declared. It is worth mentioning that validation is necessary to avoid overfitting which means that the machine learns the training data too well that it is unable to investigate new data. This happens when the training data keep decreasing but the validation error starts to increase. Therefore, by monitoring the validation error, early stopping of the training process occurs once it starts to increase.

Denoising Autoencoder-based Data Cleaning

When an attack is detected, the corrupted data are then fed into a DAE. DAEs are trained using the corrupted data as inputs and the original data as target outputs. As a result, the DAE has the capability of recovering the correlation among inputs. When falsified data are fed into the DAE, the output is a clean version of the corrupted input. This version can then be used in further processing of the system at hand as a substitute during the time of fixing the sensor under attack.

A pseudo-code of the whole process is given in Algorithm 5.

6.3 Performance Evaluation

In this section, the performance evaluation results of the proposed scheme are presented. This includes comparing it to another SVM-based scheme from the literature.

6.3.1 Simulation Setup and Parameters

Simulations were carried out using the data set of [93]. The AE of the proposed scheme is trained using 60% of the data, its validation is done using 20% of the data, and its testing is done using the remaining 20%. As mentioned earlier, during the training phase, target values are set equal to input values and weights are updated, one epoch after another, such that the MSE is minimized. During validation and testing, no target output is set. The output is calculated based on the weights adjusted by the training and finally, the MSE is calculated. The validation error is used for two purposes. First, it is monitored to avoid overfitting. Second, it is used to calculate the threshold which, if exceeded by the MSE, an alarm (detected attack) is set. Testing error is compared (for every input) with the threshold to decide whether the input data are attacked. It is worth noting that AEs need to be exposed to neither the false data during the training phase nor the labels for the input training data.

To exploit both the inter-correlation between the sensors and the autocorrelation of the readings of a single sensor in the time domain, the input to the AE is set to include

Algorithm 5 False Detection Using Autoencoder

```
1: procedure PREPARE DATA SET
2:     Divide data set into: $60\%$ training, $20\%$ validation and $20\%$ testing.
3:     Prepare a falsified copy of the testing data by replacing readings with another random numbers drawn from the normal distribution having the same variance.
4: end procedure
5: procedure TRAINING
6:     Initialize: Random weights.
7:     Input: $N_t$ readings from each sensor.
8:     procedure BACKPROPAGATION LEARNING TECHNIQUE:
9:         while Validation error is decreasing do
10:            for Each training epoch do
11:                Compute MSE based on current weights and current input: $E = \frac{1}{N}\sum_m(\mathbf{y}_m - \mathbf{x}_m)^2$, where $N$ is the size of the training set, $\mathbf{x}_m$ is the $m$-th input vector, and $\mathbf{y}_m$ is the $m$-th output vector.
12:                Update Weights based on the MSE and learning rate.
13:                Make sure that the validation error is decreasing to avoid overfitting.
14:            end for
15:        end while
16:    end procedure
17: end procedure
18: procedure TESTING
19:     Complexity: $O(d \cdot k)$ for each iteration, where $d$ is the number of dimensions of the input and $k$ denotes the number of dimensions of the encoding.
20:     Input: Testing data and falsified data.
21:     while Results are not satisfying do
22:         Change hyper parameters (e.g. learning rate, batch size, number of hidden layers, optimizer, ... etc.
23:         Repeat training phase.
24:     end while
25: end procedure
26: procedure DENOISING
27:     Training: Use corrupted data as inputs and original data as target outputs.
28:     Testing: Use detected falsified data as inputs.
29:     Output: Clean data.
30: end procedure
```

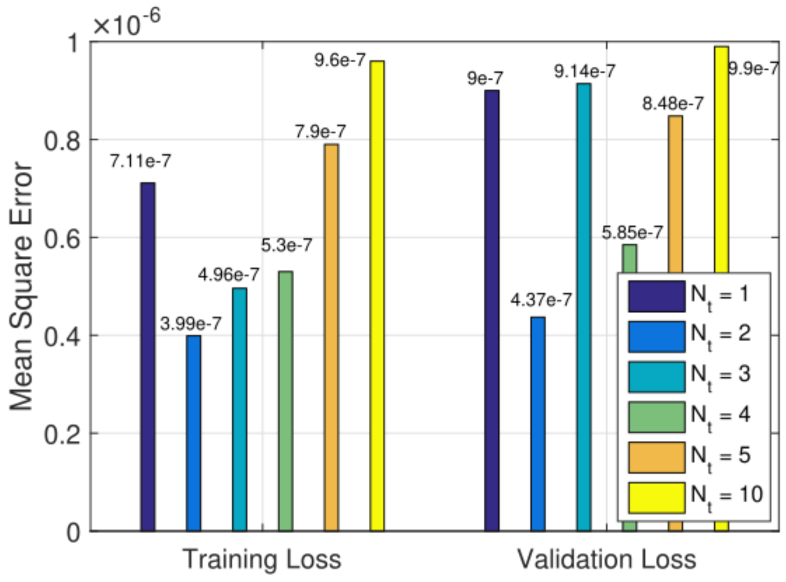

FIGURE 6.2: The Effect of Varying Number of Readings per Sensor on the Training and Validation Losses

more than one reading from each sensor at every iteration rather than a single reading per sensor. To choose the best number of time instants to be considered, the AE was trained and tested for $N_t = 1,\ 2,\ 3,\ 4,\ 5,$ and 10, where N_t is the number of readings (time instants) from each sensor to be fed to the AE per epoch. The training loss for each N_t value is shown in Fig. 6.2. The lowest loss value is at $N_t = 2$, where it is 3.99e-7 for the training loss and 4.37e-7 for the validation loss. The trend of decreasing training and validation losses at $N_t = 2$ can be seen in Fig. 6.3.

The readings from each sensor are fed to the AE in parallel to other sensors. There are 15 sensors in the network and each one feeds the AE with N_t consecutive readings every iteration. This results in a total of $15 * N_t$ neurons in the input layer. As mentioned earlier, the output layer is the same as the input layer. Five hidden layers are used. The compression factor, which is the ratio between the number of inputs and the innermost hidden layer which stands between the encoder and the decoder, is set to 3. The size of the hidden layers between the input (output) layer and the innermost layer decreases linearly according to the compression factor. It is worth mentioning that the number

TABLE 6.1: Autoencoder Simulation Parameters

Parameter	Value
Number of readings per sensor (N_t)	[1 2 3 4 5 10]
Number of neurons in input layer	$N_t * 15$
Number of neurons in output layer	$N_t * 15$
Compression factor	3
Number of hidden layers	5
Number of neurons per hidden layer	Decreases linearly according to the compression factor
Learning Algorithm	Unsupervised learning applying backpropagation
Optimizer	RMSprop
Learning Rate	0.001
Training Data	60% of whole set
Validation Data	20% of whole set
Testing Data	20% of whole set + equal amount of false data
Number of training epochs	20 (determined by the early stopping criterion)
Threshold	mean of validation error

of hidden layers and the compression factor are hyperparameters that are determined after several trials. A summary of the AE simulation parameters is listed in Table 6.1.

When considering the comparison with SVM, on the other hand, we find that SVM must be trained with both types of data (original and false) along with labels for each class. This requires some extra effort for preparing the false (attacked) data for training and labeling each set of sensors' data. During the training phase, the machine creates the optimization model (based on Lagrange multipliers). Then, during the testing phase, the model investigates new inputs and classifies each of the classes (false data class or clean data class).

Falsified data used for testing and also for SVM training are obtained by two methods

1. Add a random number to the entries of all sensors. The random number is drawn from a uniform distribution in the range of $+/-10\%$ of the original value - Referred to as "Case 1" in the rest of this section.

2. Alter the entries of only one sensor (to simulate the case of only one attacked

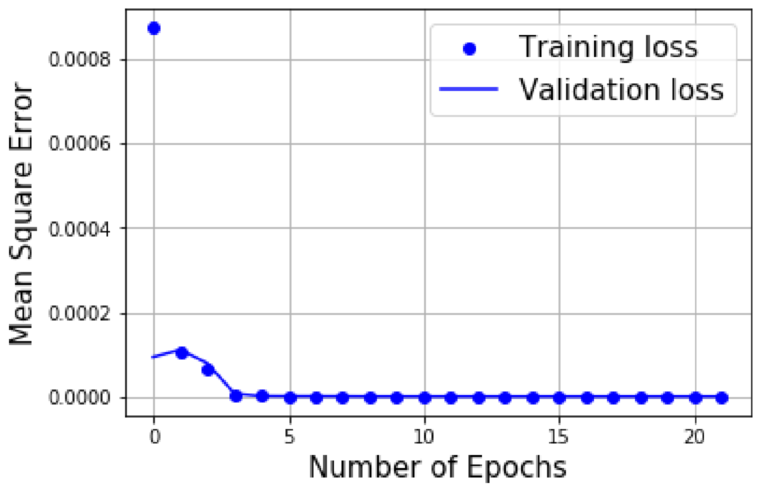

FIGURE 6.3: Training and Validation Losses in the Training Phase.

sensor). The entries of one sensor are replaced with another value drawn from a normal distribution with mean and variance that are equal to those of original data (To create a scenario more difficult to be detected). This case is referred to as "Case 2" in the rest of this section

Both machines (AE and SVM) were tested under the two scenarios.

As mentioned earlier, the AE was tested for $N_t = 1,\ 2,\ 3,\ 4,\ 5,$ and 10. For the sake of further confirmation of the best choice of N_t, an actual test under "Case 1" and "Case 2" is carried out for each value of N_t. Furthermore, the test was done for the SVM as well to make a fair comparison. Results can be seen in Fig. 6.4 in which the percentage of accurate decisions is plotted. In the figure, the *Accuracy Rate* can be defined as the percentage of the number of entries that were correctly decided by the machine (regardless of whether an attack or clean data were determined) to the whole number of entries. From the figure, we can confirm that $N_t = 2$ is the best choice for the proposed AE-based scheme. However, for the SVM, $N_t = 3$ seems to give better results. For this reason, further comparisons will take place using the best value of N_t for the corresponding scheme. It can also be seen from Fig. 6.4 that the accuracy in

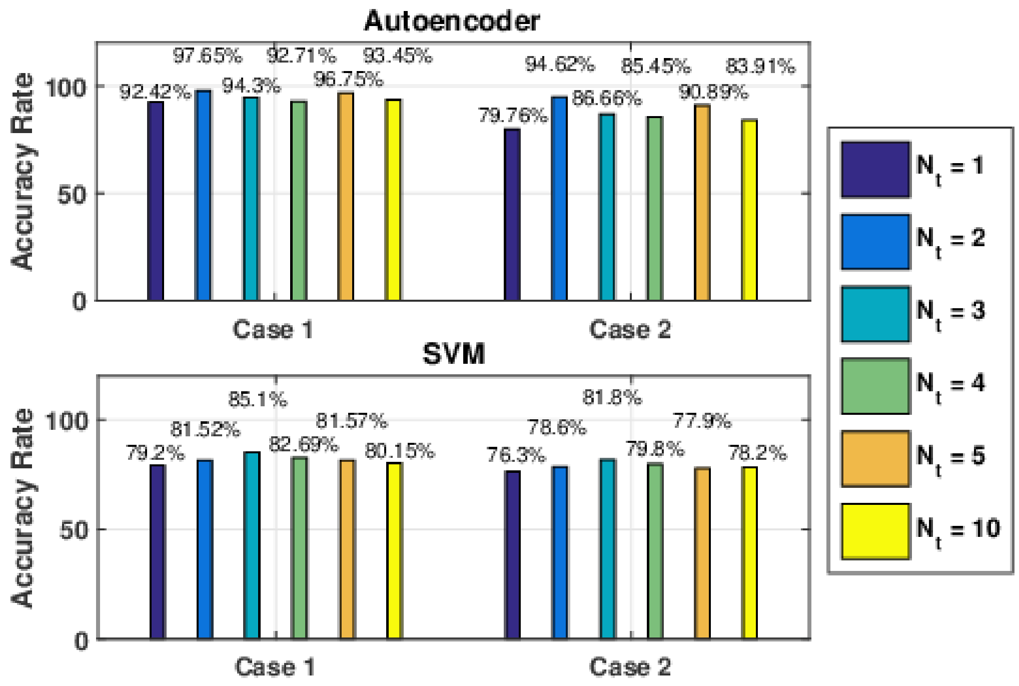

FIGURE 6.4: The Effect of Varying Number of Readings per Sensor on the Decision Accuracy

"Case 1" (for both machines) is higher than that of "Case 2". This is due to the difficulty of detecting only one malfunctioning (attacked) sensor. Moreover, the type of attack is even trickier to be detected. This is because replacing the sensor readings with values from the normal distribution with the same mean and variance will slightly affect the inter-correlation between readings. It is also worth mentioning that the accuracy of the proposed scheme is higher than that of the SVM-based scheme in all cases. However, this will be discussed in more detail later.

6.3.2 Testing and Comparing the Final Decision Accuracy

A comparison between the two machines from the perspective of the percentage of correct attack detection and the percentage of false alarm is shown in Fig. 6.5. In the figure, the *Rate of Detection* can be defined as the percentage of the number of false entries that were correctly detected by the machine to the whole number of entries.

Additionally, the *Rate of False Alarm* can be defined as the percentage of the number of clean entries that were mistakenly decided by the machine as false data to the whole number of entries. Other famous kernels (other than the linear kernel) were tested for the SVM (e.g., RBF and Gaussian). The AE has proven to be more reliable and outperforms the SVM in both cases. This can be justified by the capability of AEs to learn the hidden complex correlation structures of the data. Anomalies in input data that cause the difference between the expected data structure and the observed one can lead to a more robust classification compared to SVM where it is assumed that the two classes are separable by a hyperplane in some space.

A higher percentage of detection and a lower percentage of false alarm are recorded. Furthermore, the percentage of detection of the AE-based scheme reaches 100% in "Case 1". SVM, with a linear kernel, gives better performance than with the other two kernels. It is worth mentioning that the AE gives equal percentages of false alarm in both cases. This is because during the training phase, the machine was subjected to original (clean) data only and the same machine is used for both cases. However, the SVM must be subjected to false data in the training phase. That is why each case needs a new trained machine. This is also another major advantage of AE-based approaches, which is that they can detect other attacks since AEs are not trained to classify any "specific" attack. Rather, AEs are trained with clean data. An AE-based attack detector can detect any attack that causes a significant distortion in the data correlation model. However, to detect other attacks in the case of SVM, the machine must be trained with labeled data for every possible attack and there is no guarantee that an SVM trained to classify a specific attack will be able to detect other attacks.

To make a better visualization of the results, Fig. 6.6 gives a snapshot of a portion of the time series of the test data and the decision of both machines. As explained earlier, the ability of the AE to learn the data structure results in better performance and higher

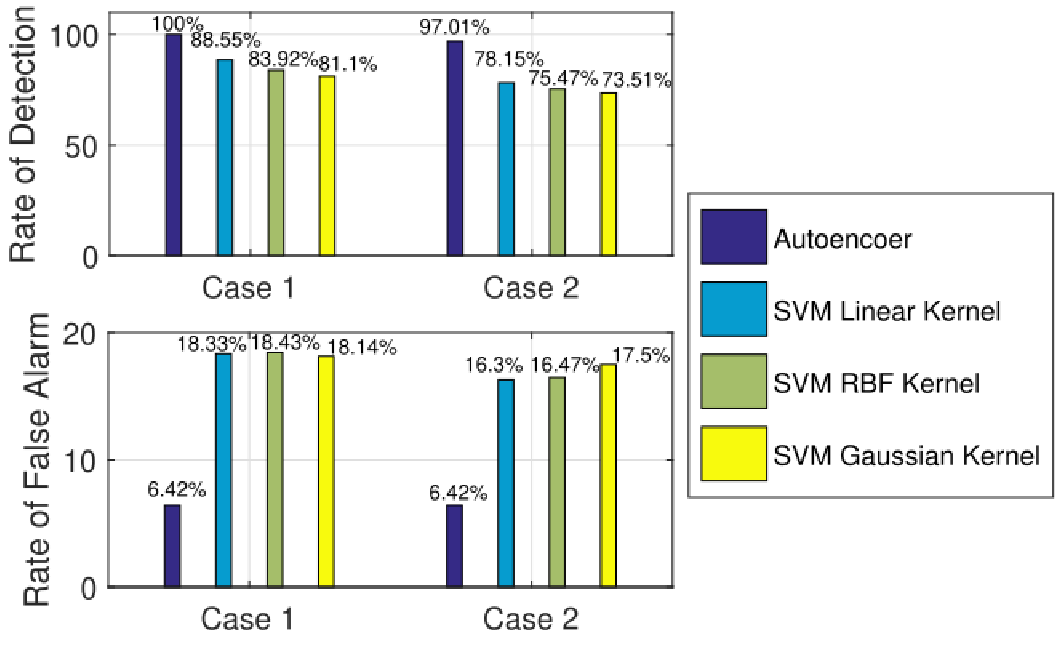

FIGURE 6.5: A Comparison w.r.t Detection Percentage and False Alarm Percentage

accuracy. Therefore, Fig. 6.6 shows fewer miss-detections and false alarms in the case of the AE-based scheme. It is worth mentioning that the AE is not better than the SVM at every instant. For example, at instant 175, SVM was able to detect the attack while AE was not. However, AE still gives better average performance and a higher percentage of detection.

The Receiver Operating Characteristics (ROC) plot for both machines is shown in Fig. 6.7. The ROC is the relation between the rate of false alarm and the rate of detection. When comparing two methodologies, a better scheme is the one that achieves a higher detection rate for the same false alarm rate. It is noticeable from Fig. 6.7 that the AE-based scheme outperforms the SVM-based scheme in terms of ROC as well.

6.3.3 Complexity Analysis

The training complexity of SVM can be approximated to $O(N^3)$, where N is the number of data entries [113]. On the other hand, the training complexity of AEs is $O(d \cdot k)$ for

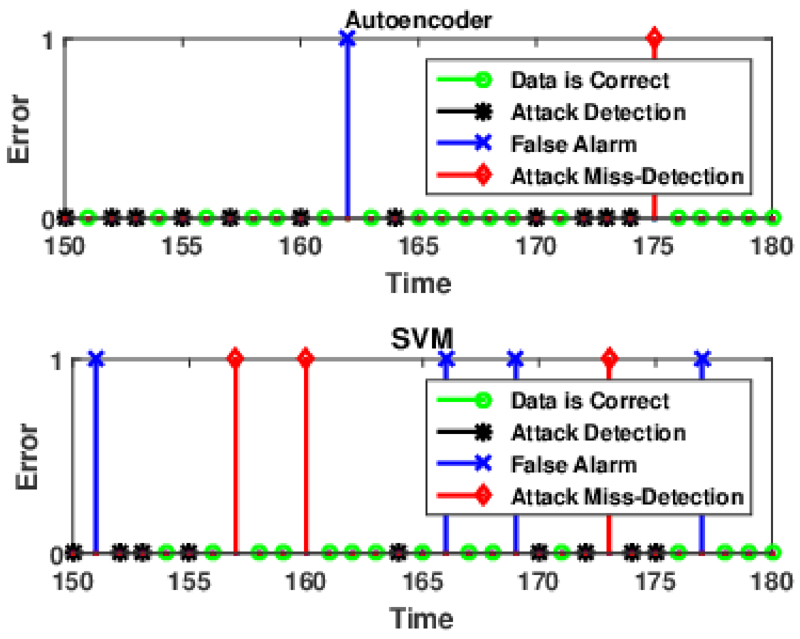

FIGURE 6.6: A Visualization of a Time Series Snap Shot

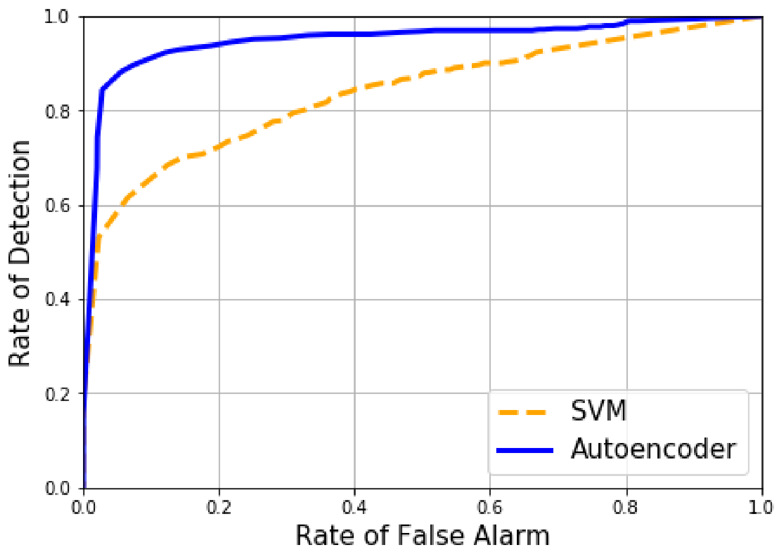

FIGURE 6.7: Receiver Operating Characteristics Plot

each iteration, where d is the number of dimensions of the input and k denotes the number of dimensions of the encoding [114]. To apply this to the case at hand, the number of readings taken from each sensor is 132300. Only 60% is used for training; this gives 79380. By taking $N_t = 3$ readings from each sensor every iteration, the total number of entries is $\frac{79380}{3}(= 26460)$ for each sensor. With 15 sensors under consideration, this results in $N = 396900$ data entries. This gives a training complexity, approximately equals $O(6e16)$ for the SVM. On the other hand, for the AE, the dimension of input vector $d = N_t * 15 = 30$, while the dimension of the encoded vector $k = 10$ (since the compression factor is chosen to be 3). It is also necessary to consider the number of training epochs, which is chosen to be 20 in our case. This results in a training complexity $O(30 * 10 * 396900 * 20) = O(2e9)$. The complexity of the AE is a lot less than that of the SVM. The training was carried out on a SAMSUNG laptop that uses a Windows 10 Pro operating system, an Intel 2.40 GHz Core i7 processor, and 8.00 GB RAM. The AE needed less than one minute to train while the SVM (linear kernel) needed 15 minutes to train.

6.3.4 Denoising Autoencoder

In this experiment, the DAE is used to clean up the corrupted data, detected by the original AE. The DAE is trained and tested using corrupted data from both cases (Case 1 and Case 2). As mentioned earlier, when an attack is detected (by the AE), the falsified data are fed to the DAE to obtain a clean version as close as possible to the original data. To prove that the output of the DAE can regain the correlation model of the data, the Mean Square Error (MSE) of the corrupted data (both cases) is plotted before and after the DAE in Fig. 6.8. This *error* represents the difference between the corrupted data and the original data with and without the DAE. In Fig. 6.8, the MSE is noticed to decrease immensely when DAE is used. This means that the DAE can recover a clean

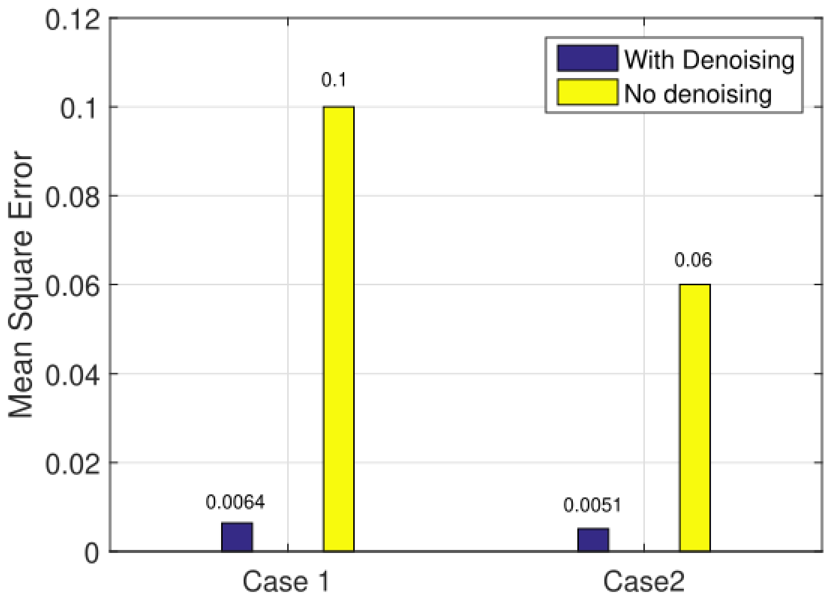

FIGURE 6.8: Mean Square Error of the Corrupted Data before and after the Denoising Autoencoder.

version of the corrupted data by recovering the data "hidden" correlation model. It is also noticeable that the MSE in case 2 is less than that in case 1. This is because the nature of the attack in case 1 causes all sensors to be corrupt, while in case 2, only one sensor is corrupt. Therefore, although the attack on only one sensor is harder to detect if the attack is causing only one sensor to be corrupted it is easier to recover the data if the attack is detected. It is worth mentioning that the input data are normalized, that is why mean square errors are all less than 1.

6.3.5 Limitations

In the following, we highlight the limitations of the proposed algorithm. These limitations can be summarized as follows

- The proposed algorithm can detect attacks that ruin the correlation among sensor readings. If the attack can maintain this correlation, the AE might not be able to detect it. For instance, if the attacker used data from the history of the measurements in which readings are taken from the same cycle, the AE algorithm may not detect this.

- The DAE must be trained for each type of attack, unlike the AE which does not need any labels or supervised learning. This limits the use of the DAE to the specific attacks that it was trained to fix.

6.4 Conclusions

In this chapter, we present a novel method of detecting FDI attacks against a complex hydraulic sensor-based system using AEs. The proposed detection method offers better detection performance compared to support vector machine-based methods (e.g. linear kernel, RBF kernel, and Gaussian kernel). In addition, AEs are easier to train since they do not require labeled data for training. Finally, AEs can detect different attacks since they can learn hidden complex correlation structures in the data. Any attack that will cause significant changes in these correlation structures can be detected by the AE-based attack detection algorithm. However, this is not the case for any other classifier like SVM which is trained to detect a "specific" attack and it is not guaranteed to detect any other attack for which it was not trained with labeled data. We adopt two different scenarios of FDI attacks which we referred to as case 1 and case 2 in the study. In both scenarios, our approach offers the highest probability of attack detection with the lowest rates of false alarm and the lowest execution time. We also present the results of using a DAE to recover from the attack's effects on data. The results demonstrate the high ability of the DAE to recover the data to its original state with very low mean square error values.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this work, three of the main challenges of the fifth-generation mobile network are the main focus. The first challenge is **Resource Allocation**. It is studied for Wireless Powered Communication Networks (WPCNs) with Non-Orthogonal Multiple Access (NOMA). Time and Power are allocated such that a certain objective function is maximized. Two cases are studied. The first is maximizing the network sum throughput and the second is maximizing the minimum throughput. For both cases, two decoding schemes are adopted, namely Low Complexity Decoding and Successive Interference Cancellation Decoding. Due to the non-convex nature of the max-sum and max-min optimization problems, we propose an approximate solution approach, in which the non-convex optimization problem is approximated by a convex optimization problem, which satisfies all the constraints of the original problem. The approximate convex optimization problem can be then solved iteratively. The results show a trade-off between maximizing the sum throughout and achieving fairness via maximizing the minimum throughput.

The second challenge is **Network Management**. Cellular networks are the main focus in this part. We present Machine Learning (ML) based approaches for this purpose. Two use cases are tackled here. The first one is energy conservation by controlling the MIMO feature. We present a Neural Network (NN) machine that emulates the network behavior and turns MIMO on only when necessary to satisfy a predetermined QoE criterion. This approach saves the energy consumed by the MIMO feature when it is not needed. We train our model based on real mobile network data. Results reveal the efficiency and effectiveness of proposed approaches as they allow for real-time, data-driven solutions. These are unique features of our proposed solutions as compared to the current practices of mobile network operators.

The second use case is the need of the network to replace the human control over the features (which is error-prone) with self-control that delivers a stable state of the network regarding load balance, the number of covered users, Quality of Experience (QoE), and consumed energy. We present a Reinforcement Learning (RL) based solution that controls three network features, namely, relative CIOs, transmission power, and MIMO on/off. The advantage of the presented model is that it enables the network to regain this stable state, without the operator interference, after any change in the environment. Moreover, this work proposes a layered approach that enables the framework to handle hybrid action space in a relatively easy yet efficient methodology.

The third challenge is **Cyber Security**, especially in the industrial Internet of Things (IoT). We presented a novel method of detecting False Data Injection (FDI) attacks against a complex hydraulic sensor-based system using Autoencoders (AEs). The proposed detection method offers better detection performance compared to Support Vector Machine based methods. AEs are easier to train since they do not require labeled data for training. Moreover, AEs can detect different attacks since they can learn hidden complex correlation structures in the data. Our approach offered the highest probability of attack detection with the lowest rates of false alarm and the lowest execution time. We also presented the results of using a denoising autoencoder to recover from the attack's effects on data. The results demonstrated the high ability of the denoising autoencoder to recover the data to its original state with very low error values.

7.2 Future Work

In this section, we present some directions for future work as follows:

- Explore the use of ML for resource allocation in WPCN. ML can provide an efficient, real-time resource allocation algorithm as compared to the proposed approach in this thesis based on convex optimization tools.
- Another interesting direction for future work is to extend the RL based approach in mobile networks to manage real scenarios (real eNBs locations and practical users' mobility models).

Bibliography

[1] Naser Al-Falahy and Omar Y Alani. "Technologies for 5G networks: Challenges and opportunities". In: *IT Professional* 19.1 (2017), pp. 12–20.

[2] Ali Ö Ercan, M Oğuz Sunay, and Ian F Akyildiz. "RF energy harvesting and transfer for spectrum sharing cellular IoT communications in 5G systems". In: *IEEE Transactions on Mobile Computing* 17.7 (2017), pp. 1680–1694.

[3] Prodromos-Vasileios Mekikis et al. "Connectivity analysis in clustered wireless sensor networks powered by solar energy". In: *IEEE transactions on wireless communications* 17.4 (2018), pp. 2389–2401.

[4] Li You et al. "Pilot reuse for massive MIMO transmission over spatially correlated Rayleigh fading channels". In: *IEEE Transactions on Wireless Communications* 14.6 (2015), pp. 3352–3366.

[5] Emiliano Sisinni et al. "Industrial internet of things: Challenges, opportunities, and directions". In: *IEEE transactions on industrial informatics* 14.11 (2018), pp. 4724–4734.

[6] Fatima Hussain et al. "Machine learning for resource management in cellular and IoT networks: Potentials, current solutions, and open challenges". In: *IEEE Communications Surveys & Tutorials* 22.2 (2020), pp. 1251–1275.

[7] M Bhanderi and H Shah. "Machine Learning for Wireless Sensor Network: A Review, Challenges and Applications". In: *Adv. Electron. Electr. Eng* 4 (2014), pp. 475–486.

[8] Jingon Joung and Bang Chul Jung. “Machine learning based blind decoding for space–time line code (STLC) systems”. In: *IEEE Transactions on Vehicular Technology* 68.5 (2019), pp. 5154–5158.

[9] Sumitra N Motade and Anju V Kulkarni. “Channel estimation and data detection using machine learning for MIMO 5G communication systems in fading channel”. In: *Technologies* 6.3 (2018), p. 72.

[10] Charles Clancy et al. “Applications of machine learning to cognitive radio networks”. In: *IEEE Wireless Communications* 14.4 (2007), pp. 47–52.

[11] Karaputugala Madushan Thilina et al. “Machine learning techniques for cooperative spectrum sensing in cognitive radio networks”. In: *IEEE Journal on selected areas in communications* 31.11 (2013), pp. 2209–2221.

[12] Jennifer S Raj. “Machine learning implementation in cognitive radio networks with game-theory technique”. In: *Journal: IRO Journal on Sustainable Wireless Systems* 2020.2 (2020), pp. 68–75.

[13] Arvind Narayanan et al. “Deepcache: A deep learning based framework for content caching”. In: *Proceedings of the 2018 Workshop on Network Meets AI & ML*. 2018, pp. 48–53.

[14] Georgios Paschos et al. “Wireless caching: Technical misconceptions and business barriers”. In: *IEEE Communications Magazine* 54.8 (2016), pp. 16–22.

[15] Osvaldo Simeone. “A very brief introduction to machine learning with applications to communication systems”. In: *IEEE Transactions on Cognitive Communications and Networking* 4.4 (2018), pp. 648–664.

[16] Mingzhe Chen et al. “Machine learning for wireless networks with artificial intelligence: A tutorial on neural networks”. In: *arXiv preprint arXiv:1710.02913* (2017).

[17] Mariam MN Aboelwafa et al. "Towards optimal resource allocation in wireless powered communication networks with non-orthogonal multiple access". In: *Ad Hoc Networks* 85 (2019), pp. 1–10.

[18] Mariam Aboelwafa et al. "Machine Learning-Based MIMO Enabling Techniques for Energy Optimization in Cellular Networks". In: *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE. 2020, pp. 1–6.

[19] Mariam Aboelwafa et al. "Deep Reinforcement Learning-Based Management Technique for Cellular Networks". In: *Manuscript submitted for publication*. 2021.

[20] Mariam MN Aboelwafa et al. "A machine-learning-based technique for false data injection attacks detection in industrial IoT". In: *IEEE Internet of Things Journal* 7.9 (2020), pp. 8462–8471.

[21] Matt W Gardner and SR Dorling. "Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences". In: *Atmospheric environment* 32.14-15 (1998), pp. 2627–2636.

[22] Nevio Benvenuto and Francesco Piazza. "On the complex backpropagation algorithm". In: *IEEE Transactions on Signal Processing* 40.4 (1992), pp. 967–969.

[23] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. "On the difficulty of training recurrent neural networks". In: *International conference on machine learning*. 2013, pp. 1310–1318.

[24] 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.

[25] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[26] Ghada Alsuhli et al. "Optimized Power and Cell Individual Offset for Cellular Load Balancing via Reinforcement Learning". In: *2021 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE. 2021, pp. 1–7.

[27] Qi Zhang, Tao Du, and Changzheng Tian. "A Sim2real method based on DDQN for training a self-driving scale car". In: *Mathematical Foundations of Computing* 2.4 (2019), p. 315.

[28] Tuomas Haarnoja et al. "Soft actor-critic algorithms and applications". In: *arXiv preprint arXiv:1812.05905* (2018).

[29] Richard S Sutton et al. "Policy gradient methods for reinforcement learning with function approximation." In: *NIPs*. Vol. 99. Citeseer. 1999, pp. 1057–1063.

[30] Olivier Delalleau et al. "Discrete and continuous action representation for practical rl in video games". In: *arXiv preprint arXiv:1912.11077* (2019).

[31] Michael Neunert et al. "Continuous-discrete reinforcement learning for hybrid control in robotics". In: *Conference on Robot Learning*. PMLR. 2020, pp. 735–751.

[32] Haotian Fu et al. "Deep multi-agent reinforcement learning with discrete-continuous hybrid action spaces". In: *arXiv preprint arXiv:1903.04959* (2019).

[33] Pascal Vincent et al. "Extracting and composing robust features with denoising autoencoders". In: *Proceedings of the 25th international conference on Machine learning*. ACM. 2008, pp. 1096–1103.

[34] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004. ISBN: 0521833787.

[35] Suzhi Bi, Yong Zeng, and Rui Zhang. "Wireless powered communication networks: An overview". In: *IEEE Wireless Communications* 23.2 (2016), pp. 10–18.

[36] David Tse and Pramod Viswanath. *Fundamentals of wireless communication*. Cambridge university press, 2005.

[37] Kenichi Higuchi and Anass Benjebbour. “Non-orthogonal multiple access (NOMA) with successive interference cancellation for future radio access”. In: *IEICE Transactions on Communications* 98.3 (2015), pp. 403–414.

[38] Dusit Niyato et al. *Wireless-Powered Communication Networks*. Cambridge University Press, 2016.

[39] Xiao Lu et al. “Wireless networks with RF energy harvesting: A contemporary survey”. In: *IEEE Communications Surveys & Tutorials* 17.2 (2015), pp. 757–789.

[40] Deep Patel et al. “RF energy harvesting”. In: *International Journal of Engineering Trends and Technology (IJETT)* 16.8 (2014), pp. 382–385.

[41] Suzhi Bi, Chin Ho, and Rui Zhang. “Wireless powered communication: opportunities and challenges”. In: *IEEE Communications Magazine* 53.4 (2015), pp. 117–125.

[42] Ioannis Krikidis et al. “Simultaneous wireless information and power transfer in modern communication systems”. In: *IEEE Communications Magazine* 52.11 (2014), pp. 104–110.

[43] Elena Boshkovska et al. “Practical non-linear energy harvesting model and resource allocation for SWIPT systems”. In: *IEEE Communications Letters* 19.12 (2015), pp. 2082–2085.

[44] Derrick Wing Kwan Ng and Robert Schober. “Secure and green SWIPT in distributed antenna networks with limited backhaul capacity”. In: *IEEE Transactions on Wireless Communications* 14.9 (2015), pp. 5082–5097.

[45] He Henry Chen et al. “Distributed power splitting for SWIPT in relay interference channels using game theory”. In: *IEEE Transactions on Wireless Communications* 14.1 (2015), pp. 410–420.

[46] Rui Zhang and Chin Keong Ho. “MIMO broadcasting for simultaneous wireless information and power transfer”. In: *IEEE Transactions on Wireless Communications* 12.5 (2013), pp. 1989–2001.

[47] Yuanwei Liu et al. “Cooperative non-orthogonal multiple access with simultaneous wireless information and power transfer”. In: *IEEE Journal on Selected Areas in Communications* 34.4 (2016), pp. 938–953.

[48] Hyungsik Ju and Rui Zhang. “User cooperation in wireless powered communication networks”. In: *2014 IEEE Global Communications Conference (GLOBECOM)*, pp. 1430–1435.

[49] Mariam MN Aboelwafa, Karim G Seddik, and Mustafa ElNainay. “Cooperation in multi-user wireless powered communication networks”. In: *2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE. 2017, pp. 1–6.

[50] Mohamed A Abd-Elmagid, Tamer ElBatt, and Karim G Seddik. “Optimization of wireless powered communication networks with heterogeneous nodes”. In: *Global Communications Conference (GLOBECOM), 2015 IEEE*. IEEE. 2015, pp. 1–7.

[51] Alessandro Biason and Michele Zorzi. “Battery-powered devices in WPCNs”. In: *IEEE Transactions on Communications* 65.1 (2017), pp. 216–229.

[52] Mohamed A Abd-Elmagid et al. “On optimal policies in full-duplex wireless powered communication networks”. In: *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), 2016 14th International Symposium on*. IEEE. 2016, pp. 1–7.

[53] Mohamed A Abd-Elmagid, Tamer ElBatt, and Karim G Seddik. “Optimization of energy-constrained wireless powered communication networks with heterogeneous nodes”. In: *Wireless Networks* 23 (2017), pp. 1–18.

[54] Panagiotis D Diamantoulakis et al. “Wireless-powered communications with non-orthogonal multiple access”. In: *IEEE Transactions on Wireless Communications* 15.12 (2016), pp. 8422–8436.

[55] Yuya Saito et al. “System-level performance evaluation of downlink non-orthogonal multiple access (NOMA)”. In: *Personal Indoor and Mobile Radio Communications (PIMRC), 2013 IEEE 24th International Symposium on*. IEEE. 2013, pp. 611–615.

[56] Yi Yuan and Zhiguo Ding. “The application of non-orthogonal multiple access in wireless powered communication networks”. In: *Signal Processing Advances in Wireless Communications (SPAWC), 2016 IEEE 17th International Workshop on*. IEEE. 2016, pp. 1–5.

[57] Mohamed A Abd-Elmagid et al. “Non-orthogonal multiple access schemes in wireless powered communication networks”. In: *2017 IEEE International conference on communications (ICC)*. IEEE. 2017, pp. 1–6.

[58] H. Chingoska et al. “Resource Allocation in Wireless Powered Communication Networks With Non-Orthogonal Multiple Access”. In: *IEEE Wireless Communications Letters* 5.6 (Dec. 2016), pp. 684–687.

[59] Product Datasheet. “P2110-915MHz RF Powerharvester Receiver”. In: *Powercast Corporation* (2010).

[60] Mung Chiang. “Geometric Programming for Communication Systems”. In: *Foundations and Trends® in Communications and Information Theory* 2.1–2 (2005), pp. 1–154.

[61] Ellen L. Hahne. “Round-robin scheduling for max-min fairness in data networks”. In: *IEEE Journal on Selected Areas in communications* 9.7 (1991), pp. 1024–1039.

[62] Leandros Tassiulas and Saswati Sarkar. “Maxmin fair scheduling in wireless ad hoc networks”. In: *IEEE Journal on Selected Areas in Communications* 23.1 (2005), pp. 163–173.

[63] Gesualdo Scutari, Francisco Facchinei, and Lorenzo Lampariello. “Parallel and Distributed Methods for Constrained Nonconvex Optimization-Part I: Theory.” In: *IEEE Trans. Signal Processing* 65.8 (2017), pp. 1929–1944.

[64] Mohsen Mollanoori and Majid Ghaderi. “Fair and efficient scheduling in wireless networks with successive interference cancellation”. In: *Wireless communications and networking conference (WCNC)*. IEEE. 2011, pp. 221–226.

[65] Raj Jain, Dah-Ming Chiu, and William R Hawe. *A quantitative measure of fairness and discrimination for resource allocation in shared computer system*. Vol. 38. Eastern Research Laboratory, Digital Equipment Corporation Hudson, MA, 1984.

[66] Margot Deruyck et al. “Power consumption in wireless access network”. In: *2010 European Wireless Conference (EW)*. IEEE. 2010, pp. 924–931.

[67] Thomas L Marzetta. “Massive MIMO: an introduction”. In: *Bell Labs Technical Journal* 20 (2015), pp. 11–22.

[68] Chunxiao Jiang et al. “Machine learning paradigms for next-generation wireless networks”. In: *IEEE Wireless Communications* 24.2 (2016), pp. 98–105.

[69] Tao Han and Nirwan Ansari. “Powering mobile networks with green energy”. In: *IEEE Wireless Communications* 21.1 (2014), pp. 90–96.

[70] Antonio Spagnuolo et al. “Monitoring and optimization of energy consumption of base transceiver stations”. In: *Energy* 81 (2015), pp. 286–293.

[71] Josip Lorincz, Tonko Garma, and Goran Petrovic. “Measurements and modelling of base station power consumption under real traffic loads”. In: *Sensors* 12.4 (2012), pp. 4281–4310.

[72] N Faruk et al. “Techniques for minimizing power consumption of base transceiver station in mobile cellular systems”. In: *International Journal of Sustainability* 2.1 (2013), pp. 1–11.

[73] Pol Blasco and Deniz Gündüz. “Learning-based optimization of cache content in a small cell base station”. In: *2014 IEEE International Conference on Communications (ICC)*. IEEE. 2014, pp. 1897–1903.

[74] Kunal Shah and Mohan Kumar. “Distributed independent reinforcement learning (DIRL) approach to resource management in wireless sensor networks”. In: *2007 IEEE International Conference on Mobile Adhoc and Sensor Systems*. IEEE. 2007, pp. 1–9.

[75] Yi Liu et al. “Integrated energy and spectrum harvesting for 5G wireless communications”. In: *IEEE Network* 29.3 (2015), pp. 75–81.

[76] Tao Chen, Haesik Kim, and Yang Yang. “Energy efficiency metrics for green wireless communications”. In: *2010 International Conference on Wireless Communications & Signal Processing (WCSP)*. IEEE. 2010, pp. 1–6.

[77] Sumita Mishra and Nidhi Mathur. “Load balancing optimization in LTE/LTE-A cellular networks: a review”. In: *arXiv preprint arXiv:1412.7273* (2014).

[78] Stefano Buzzi et al. “A survey of energy-efficient techniques for 5G networks and challenges ahead”. In: *IEEE Journal on Selected Areas in Communications* 34.4 (2016), pp. 697–709.

[79] Junaid Ahmed Khan, Hassaan Khaliq Qureshi, and Adnan Iqbal. "Energy management in wireless sensor networks: A survey". In: *Computers & Electrical Engineering* 41 (2015), pp. 159–176.

[80] Ghada Alsuhli et al. "Deep Reinforcement Learning-based CIO and Energy Control for LTE Mobility Load Balancing". In: *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE. 2021, pp. 1–6.

[81] Eunsung Oh et al. "Toward dynamic energy-efficient operation of cellular network infrastructure". In: *IEEE Communications Magazine* 49.6 (2011), pp. 56–61.

[82] Yong Cui et al. "Performance-aware energy optimization on mobile devices in cellular network". In: *IEEE Transactions on Mobile Computing* 16.4 (2016), pp. 1073–1089.

[83] Kareem Attiah et al. "Load Balancing in Cellular Networks: A Reinforcement Learning Approach". In: *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE. 2020, pp. 1–6.

[84] G Alsuhli et al. "Mobility Load Management in Cellular Networks: A Deep Reinforcement Learning Approach". In: *Accepted for publication in IEEE Transactions on Mobile Computing* (2021).

[85] Faris B Mismar, Jinseok Choi, and Brian L Evans. "A framework for automated cellular network tuning with reinforcement learning". In: *IEEE Transactions on Communications* 67.10 (2019), pp. 7152–7167.

[86] Sandeep Chinchali et al. "Cellular network traffic scheduling with deep reinforcement learning". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.

[87] Fan Meng et al. "Power allocation in multi-user cellular networks: Deep reinforcement learning approaches". In: *IEEE Transactions on Wireless Communications* 19.10 (2020), pp. 6255–6267.

[88] ETSI LTE. "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures". In: *ETSI TS* (2017), pp. 136–213.

[89] ETSI TSGR. "Lte: Evolved universal terrestrial radio access (e-utra)". In: *Multiplexing and channel coding (3GPP TS 36.212 version 10.3. 0 Release 10) ETSI TS* 136.212 (2011), p. V10.

[90] Nicola Baldo et al. "An open source product-oriented LTE network simulator based on ns-3". In: *Proceedings of the 14th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems*. 2011, pp. 293–298.

[91] Ashley Hill et al. *Stable Baselines*. `https://github.com/hill-a/stable-baselines`. 2018.

[92] Piotr Gawłowicz and Anatolij Zubow. "ns3-gym: Extending openai gym for networking research". In: *arXiv preprint arXiv:1810.03943* (2018).

[93] Nikolai Helwig, Eliseo Pignanelli, and Andreas Schütze. "Condition monitoring of a complex hydraulic system using multivariate statistics". In: *Instrumentation and Measurement Technology Conference (I2MTC), 2015 IEEE International*. IEEE. 2015, pp. 210–215.

[94] Md Ashfaqur Rahman and Hamed Mohsenian-Rad. "False data injection attacks with incomplete information against smart power grids". In: *Global Communications Conference (GLOBECOM), 2012 IEEE*. Citeseer. 2012, pp. 3153–3158.

[95] Stefan Forsström et al. "Challenges of Securing the Industrial Internet of Things Value Chain". In: *2018 Workshop on Metrology for Industry 4.0 and IoT*. IEEE. 2018, pp. 218–223.

[96] Stamatis Karnouskos. "Stuxnet worm impact on industrial cyber-physical system security". In: *IECON 2011-37th Annual Conference on IEEE Industrial Electronics Society*. IEEE. 2011, pp. 4490–4494.

[97] Yao Liu, Peng Ning, and Michael K Reiter. "False data injection attacks against state estimation in electric power grids". In: *ACM Transactions on Information and System Security (TISSEC)* 14.1 (2011), pp. 1–33.

[98] Adnan Anwar, Abdun Naser Mahmood, and Mark Pickering. "Modeling and performance evaluation of stealthy false data injection attacks on smart grid in the presence of corrupted measurements". In: *Journal of Computer and System Sciences* 83.1 (2017), pp. 58–72.

[99] Aditya Ashok, Manimaran Govindarasu, and Venkataramana Ajjarapu. "Online detection of stealthy false data injection attacks in power system state estimation". In: *IEEE Transactions on Smart Grid* 9.3 (2018), pp. 1636–1646.

[100] Mete Ozay et al. "Machine learning methods for attack detection in the smart grid". In: *IEEE transactions on neural networks and learning systems* 27.8 (2016), pp. 1773–1786.

[101] Mohammad Esmalifalak et al. "Detecting stealthy false data injection using machine learning in smart grid". In: *IEEE Systems Journal* 11.3 (2017), pp. 1644–1652.

[102] Giovanni Apruzzese et al. "On the effectiveness of machine and deep learning for cyber security". In: *2018 10th International Conference on Cyber Conflict (CyCon)*. IEEE. 2018, pp. 371–390.

[103] Jonghoon Lee et al. "Cyber Threat Detection Based on Artificial Neural Networks Using Event Profiles". In: *IEEE Access* 7 (2019), pp. 165607–165626.

[104] Tareek Pattewar et al. “Detection of SQL injection using machine learning: A survey”. In: *Int. Res. J. Eng. Technol.(IRJET)* 6.11 (2019), pp. 239–246.

[105] Tong Anh Tuan et al. “Performance evaluation of Botnet DDoS attack detection using machine learning”. In: *Evolutionary Intelligence* 13.2 (2020), pp. 283–294.

[106] Qiang Liu et al. “A survey on security threats and defensive techniques of machine learning: A data driven view”. In: *IEEE access* 6 (2018), pp. 12103–12117.

[107] M. Yousefi-Azar et al. “Autoencoder-based feature learning for cyber security applications”. In: *2017 International Joint Conference on Neural Networks (IJCNN)*. May 2017, pp. 3854–3861. DOI: 10.1109/IJCNN.2017.7966342.

[108] S. Naseer et al. “Enhanced Network Anomaly Detection Based on Deep Neural Networks”. In: *IEEE Access* 6 (2018), pp. 48231–48246. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2863036.

[109] M. Al-Qatf et al. “Deep Learning Approach Combining Sparse Autoencoder With SVM for Network Intrusion Detection”. In: *IEEE Access* 6 (2018), pp. 52843–52856. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2869577.

[110] H. Zhang et al. “An Effective Deep Learning Based Scheme for Network Intrusion Detection”. In: *2018 24th International Conference on Pattern Recognition (ICPR)*. Aug. 2018, pp. 682–687. DOI: 10.1109/ICPR.2018.8546162.

[111] Pascal Vincent et al. “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion.” In: *Journal of machine learning research* 11.12 (2010).

[112] Yong Wang et al. “Srid: State relation based intrusion detection for false data injection attacks in scada”. In: *European symposium on research in computer security*. Springer. 2014, pp. 401–418.

[113] Abdiansah Abdiansah and Retantyo Wardoyo. “Time complexity analysis of support vector machines (SVM) in LibSVM”. In: *International journal computer and application* 128.3 (2015), pp. 28–34.

[114] Ozan Irsoy and Ethem Alpaydin. “Autoencoder trees”. In: *Asian conference on machine learning*. PMLR. 2016, pp. 378–390.