**People's Democratic Republic of Algeria**

**Ministry of Higher Education and Scientific Research**

**IBN KHALDOUN UNIVERSITY OF TIARET**

# Dissertation

Presented to:

FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
DEPARTEMENT OF COMPUTER SCIENCE

In order to obtain the degree of:

**MASTER**

Specialty:

Artificial intelligence and digitalization

Presented by:

**NASRI Sonia**

On the theme:

# Proposed Approach for an Intelligent Energy Management System in Smart Cities

Defended publicly on / 05 /2025 in Tiaret in front the jury composed of: :

| | | | |
|---|---|---|---|
| Mr. Dr. MOSTEFAOUI Kada | MCB | Tiaret University | Chairman |
| Mr. Dr. DAOUD Mohamed Amine | MCB | Tiaret University | Supervisor |
| Mr ALEM Abdelkader | MAA | Tiaret University | Examiner |
| Mr. Dr. MOSTEFAOUI Sid Ahmed Mokhtar | MCA | Tiaret University | Guest |

2024-2025

# *Dedication*

**الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ، وَالشُّكْرُ لَهُ عَلَى نِعْمَتِهِ وَتَوْفِيقِهِ**

*With deep gratitude to Allah for granting me strength, patience, and the ability to complete this work.*
*I dedicate this modest effort to those who have walked beside me with love, care, and encouragement.*

*To my dear parents — your unconditional love, sacrifices, and prayers have been the foundation of my journey. You are my strength and my peace.*

*To my precious sisters, **Melissa**, **Amira**, and **Lidia** — thank you for your constant support and for believing in me.*

*To my cousins, **Souhila** and **Nesrine** — your kindness and encouragement were always a comfort to my heart.*

*To my loyal friends — your presence gave me joy, strength, and motivation when I needed it most.*

*To my whole family — your love surrounded me and helped me move forward.*

*To every person who supported me — in words, in actions, or even silently — I am truly thankful from the bottom of my heart.*

# *Acknowledgements*

*This work would not have been possible without the support and encouragement of many people.*

*First, I thank God for giving me the strength and perseverance to carry out this project.*

*I would like to express my sincere gratitude to my supervisor, Mr. Dr.* ***DAOUD Mohamed Amine****, for his continuous guidance, valuable feedback, and encouragement throughout the development of this work.*

*My thanks also go to the members of the jury, Mr.* ***ALEM Abdelkader****, Mr. Dr.* ***MOSTEFAOUI Kadda****, and Mr. Dr.* ***Mostefaoui Sid Ahmed Mokhtar*** *for taking the time to evaluate my work and for their constructive remarks.*

*I am truly grateful to all the professors of the Computer Science Department at* ***Ibn Khaldoun University****. A special thank you goes to the Professors of the Artificial Intelligence and digitalization specialty for their dedication and the knowledge they shared with us.*

*To my family, the* ***NASRI*** *family, I owe my deepest appreciation. Your constant belief in me, your support, and your love have been a pillar of strength from the beginning.*

*Lastly, I want to thank everyone — friends, classmates, and all those who contributed in any way — for their encouragement and kindness throughout this experience.*

# Abstract


***The increasing complexity of urban environments and the growing demand for sustainable development have accelerated the adoption of smart city paradigms. Smart cities integrate advanced technologies to enable real-time monitoring and intelligent management of critical resources such as electricity gas, and water. These systems aim to enhance operational efficiency, reduce environmental impact, and support long-term sustainability. However, the implementation of such systems is not without challenges. Issues related to interoperability, scalability, data overload, and the energy requirements of IoT infrastructure could hinder the effectiveness and reliability of smart city applications. In particular, energy management within smart buildings remains a critical area requiring robust predictive models capable of handling uncertainty and complex consumption patterns.***

***A novel approach is presented to building energy consumption forecasting using the Building Data Genome Project 2 dataset. The proposed methodology combines fuzzification techniques with deep learning models within an ensemble learning meta-model, designed to enhance prediction accuracy in the presence of uncertain and heterogeneous data. The experimental evaluation demonstrates high prediction performance across three consumption types: 82.66% accuracy for electricity, 99.87% for gas, and 99.12% for water. These results validate the robustness and applicability of the proposed approach in the context of intelligent energy management systems.***



**Keywords**: Smart cities, energy management, deep learning, Fuzzification, Ensemble learning, Energy forecasting.

# *Résumé*

***La complexité croissante des environnements urbains et la demande grandissante en matière de développement durable ont accéléré l'adoption des paradigmes de ville intelligente. Les villes intelligentes intègrent des technologies avancées pour permettre une surveillance en temps réel et une gestion intelligente des ressources critiques telles que l'électricité, le gaz et l'eau. Ces systèmes visent à améliorer l'efficacité opérationnelle, à réduire l'impact environnemental et à soutenir la durabilité à long terme. Cependant, la mise en œuvre de tels systèmes présente plusieurs défis. Des problèmes liés à l'interopérabilité, à la scalabilité, à la surcharge de données et aux besoins énergétiques de l'infrastructure IoT peuvent nuire à l'efficacité et à la fiabilité des applications des villes intelligentes. En particulier, la gestion de l'énergie dans les bâtiments intelligents reste un domaine critique nécessitant des modèles prédictifs robustes capables de gérer l'incertitude et les schémas de consommation complexes.***

***Une nouvelle approche est proposée pour la prévision de la consommation énergétique des bâtiments à l'aide du dataset Building Data Genome Project 2. La méthodologie combine des techniques de fuzzification avec des modèles d'apprentissage profond au sein d'un méta-modèle d'apprentissage ensembliste, conçu pour améliorer la précision des prédictions face à des données incertaines et hétérogènes. L'évaluation expérimentale démontre une performance de prédiction élevée pour les trois types de consommation : 82,66 % pour l'électricité, 99,87 % pour le gaz, et 99,12 % pour l'eau. Ces résultats valident la robustesse et la pertinence de l'approche proposée dans le contexte des systèmes intelligents de gestion de l'énergie.***

**Mots clés :** Villes intelligentes, Gestion de l'énergie, apprentissage profond, Fuzzification, Apprentissage ensembliste, prévision de la consommation énergétique.

# الملخص:

***أدى التعقيد المتزايد في البيئات الحضرية والطلب المتنامي على التنمية المستدامة الى تسريع تبني نماذج المدن الذكية. تقوم المدن الذكية على دمج تقنيات متقدمة تتيح المراقبة في الوقت الحقيقي والإدارة الذكية للموارد الحيوية مثل الكهرباء، الغاز، والمياه. تهدف هذه الأنظمة الى تعزيز الكفاءة التشغيلية، تقليل الأثر البيئي، ودعم الاستدامة على المدى الطويل.***

***ومع ذلك، فان تنفيذ هذه الأنظمة لا يخلو من التحديات، إذ تواجه صعوبات تتعلق بقابلية التشغيل البيني، وقابلية التوسع، وتدفق البيانات الهائل، بالإضافة إلى متطلبات الطاقة العالية للبنية التحتية لإنترنت الأشياء، مما قد يؤثر على فعالية وموثوقية تطبيقات المدن الذكية. وتُعد إدارة الطاقة داخل المباني الذكية مجالًا حيويًا يتطلب نماذج تنبؤيه قوية قادرة على التعامل مع عدم اليقين وتعقيد أنماط الاستهلاك.***

***في هذا الإطار، تم تقديم منهجية جديدة للتنبؤ باستهلاك الطاقة في المباني، باستخدام قاعدة بيانات.*** *Building Data Genome Project 2*

***تعتمد المنهجية المقترحة على دمج تقنيات التمويج (الفَزّية) مع نماذج التعلم العميق ضمن إطار تعلّم تجميعي بهدف تحسين دقة التنبؤ في ظل البيانات غير المؤكدة والمتباينة. وقد أظهرت التقييمات التجريبية أداءً عالياً في التنبؤ بأنواع الاستهلاك الثلاثة: 82.66 للكهرباء, 99,87 للغاز، و99.12 للمياه. تؤكد هذه النتائج قوة وملاءمة المنهجية المقترحة في سياق أنظمة إدارة الطاقة الذكية.***

**الكلمات المفتاحية:** المدن الذكية، إدارة الطاقة، التعلم العميق، التمويج، التعلّم التجميعي، التنبؤ باستهلاك الطاقة.

# *TABLE OF CONTENTS*

## *FIGURE TABLE*

## TABLES LIST

## *Abbreviations and acronyms*

<AI> <Artificial intelligence>

<ANN> <Artificial Neural Networks>

<AR> <AutoRegressive>

<ARMA> <AutoRegressive Integrated Moving Average>

<ARMA> < AutoRegressive Moving Average>

<CNN> <Convolutional Neural Network>

<DT> <Decision Trees>

<GRU> <Gated Recurrent Unit>

<ICT> <Information and Communications Technology>

<IoT> <Internet of Things>

<IT> <Information Technology>

<LED> <Light-Emitting Diode>

<LPWAN> <Low Power Wide Area Network>

<LSTM> <Long Short-Term Memory>

<MA> <Moving Average>

<MAE> <Mean Absolute Error>

<MAPE> <Mean Absolute Percentage Error>

<MF> <Membership Function>

<MLP> <Multi-Layer Perceptron>

<MSE> <Mean Squared Error>

<$R^2$> <R-Squared>

<RMSE> <Root Mean Squared Error>

<RNN> <Recurrent Neural Network>

<SVM> <Support Vector Machine>

<TSA> <Time Series Analysis>

# *General Introduction*

# General Introduction

In recent years, the concept of *smart cities* has emerged as a response to the growing challenges of urbanization, sustainability, and efficient resource utilization. Smart cities leverage advanced technologies such as sensors, the Internet of Things (IoT), and data analytics to manage essential resources like energy, water, and waste in real-time. These technologies enable city infrastructures to monitor usage patterns, reduce inefficiencies, and optimize system operations.

For example, smart grids intelligently distribute electricity according to demand, minimizing energy waste and enhancing the stability of the electrical network. Similarly, smart water systems employ real-time leak detection mechanisms to prevent water loss, ensuring both efficiency and sustainability.

Effective resource management is at the core of sustainable urban development. By reducing environmental impacts and promoting long-term availability of resources, smart resource management plays a vital role in achieving global sustainability goals.

Despite these advantages, smart city systems face several challenges. The integration of diverse devices and platforms often leads to interoperability and scalability issues. Ensuring seamless communication between heterogeneous systems is critical; otherwise, communication breakdowns and data loss can compromise the reliability and efficiency of smart infrastructures. Moreover, the deployment of IoT-based systems generates an enormous volume of data. Managing, storing, and analyzing this data requires advanced computational infrastructures, and inadequate data management can cause system slowdowns or information loss. In addition, IoT devices, including sensors, typically require continuous power, raising concerns about power consumption and battery life maintenance.

To address these challenges within the energy domain, this thesis proposes an approach for analyzing building energy consumption using the **Building Data Genome Project 2** dataset [65]. The primary objective is to forecast energy usage by applying advanced deep learning techniques. Our methodology is designed to cover the full pipeline — from data preparation and preprocessing to classification and result analysis.

More specifically, this work integrates **fuzzification techniques** and **ensemble learning** strategies to improve the classification and prediction of energy consumption patterns.

The combination of these techniques allows the proposed model to handle uncertainty in the data and enhance prediction performance, ultimately contributing to smarter, more energy-efficient urban infrastructures.

The effectiveness of the proposed approach has been validated through experiments across three types of resource consumption: electricity, gas, and water. The model achieved an accuracy of **82.66%** for electricity, **99.87%** for gas, and **99.12%** for water consumption prediction. These promising results demonstrate the reliability and robustness of the approach in supporting intelligent energy management within smart buildings.

This thesis is organized into three main chapters, each addressing a key aspect of the research.

- **Chapter 1** provides an overview of the smart city concept, focusing on the role of advanced technologies in improving urban sustainability. It explores the importance of intelligent resource management, with particular emphasis on energy systems.
- **Chapter 2** introduces deep learning and highlights its relevance and usability in the context of smart cities, especially for energy consumption forecasting and resource optimization. This chapter also reviews related research and state-of-the-art approaches.
- **Chapter 3** presents the proposed approach for building energy consumption analysis, which combines fuzzification techniques and deep learning within an ensemble learning meta-model. The chapter details the methodology, experimental results, and analysis.
- Finally, the thesis concludes with a **general conclusion** that summarizes the contributions of the work and outlines potential future research directions.

# *Chapter 01*

## Smart Cities and Energy Management

# Chapter 01: Smart Cities and Energy Management

## 1. Introduction

The rise in urban expansion as well as the ever-growing demands of energy requires innovative solutions to manage resources efficiently. Smart cities, driven by advanced technologies like the Internet of Thing (IoT) and Artificial intelligence (AI), where the goal of urban development is focused on the increased quality of life for citizens in a sustainable way. In this context, this chapter examines the basic principles of smart cities and their importance in energy management. It investigates the challenges and dimensions of smart cities, integration with energy management systems, as well as the underlying technologies that provided their functioning. By understanding these elements, we can highlight the importance of intelligent systems in solving the urban energy challenges and enabling future sustainable urban development.

## 2. Smart Cities and Energy Management

Today's cities are facing many challenges related to energy management, due to increasing urbanization and technological advancements. With the continuous growth of the urban population and the resulting energy demand, it is crucial to find solutions to optimize the use of resources. Concepts such as **smart cities** and **smart energy grids** are emerging to address these issues, making energy management more efficient and improving the quality of life for residents.

### 2.1 Smart city

In this subsection, we are going to cover the basic principles associated with smart cities, as well as their definition, characteristics, challenges and dimensions.

#### 2.1.1 Definitions

**a) Definition 01**

A smart city is a city where most urban objects are connected. A city that connects elements of IT infrastructure, social infrastructure, physical infrastructure, and commercial infrastructure to achieve the collective intelligence of the city. [1]

**b) Definition 02**

A smart city is an urban area where technology and data collection help improve quality of life as well as the sustainability and efficiency of city operations. Smart city technologies used by local governments include information and communication technologies (ICT) and the Internet of Things (IoT). [2]

### 2.1.2 Dimensions of a smart city

Smart cities are built on several key dimensions that work together to create efficient, sustainable, and livable urban environments. These dimensions cover various aspects of city life, forming the foundation of smart urban development:

#### a) Smart Governance:

Smart governance is about using technology to facilitate and support better planning and decision-making. It is about improving democratic processes and transforming the ways that public services are delivered. It is a new way of governance relying on information and communication technologies and it is citizen centric, data driven and performance focused [3].

- E-government services
- Citizen participation
- Transparent decision-making [4]

#### b) Smart Economy:

Smart Economy is an economy based on technological innovation, resource efficiency, sustainability, and high social welfare. Smart Economy adopts innovations, new entrepreneurial initiatives, increases productivity and competitiveness, with the overall goal of improving the quality of life of all citizens [5].

- Innovation and entrepreneurship
- Productivity
- Flexible labor market [4]

#### c) Smart Mobility:

Smart mobility refers to the use of advanced technologies to enhance the efficiency, sustainability, and convenience of transportation systems. It utilizes data-driven solutions and information technologies to optimize the movement of people and goods [6].

- Integrated transport systems
- Local and international accessibility
- Sustainable ICT infrastructure [4]

d) **Smart Environment:**

A smart environment is an ecosystem of communicating objects that have the capacity to coordinate, provide services, and control complicated information (e.g., sensors, actuators, information devices, and other network-connected devices). The physical smart environment is intelligent in design and benefits from the interface of various devices and computer systems, aimed at enhancing services to people [7].

- Sustainable resource management
- Pollution reduction
- Environmental protection [4]

e) **Smart People:**

It is defined as possessing e-skills, working in Information and Communications Technology (ICT)-enabled environments, having access to education and training, human resources, and capacity management. As a feature, it can also allow individuals and groups to enter, utilize, modify and customize data on their own, for instance by using dashboards and data analytics tools that are suitable for making decisions and developing goods and services [8].

- Education level
- Lifelong Learning [4]

f) **Smart Living:**

Smart Living is a trend encompassing advancements that give people the opportunity to benefit from new ways of living. It involves original and innovative solutions aimed at making life more efficient, more controllable, economical, productive, integrated and sustainable [9].

- Health
- Safety
- Housing quality
- Cultural facilities [4]

These dimensions work together to create a comprehensive framework for urban development, aiming to enhance the overall quality of life for citizens while optimizing the use of urban resources. [4]

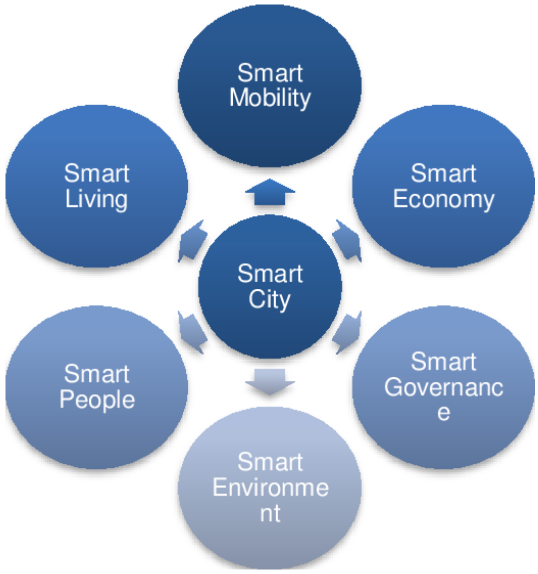



*Figure 1 Dimensions of a smart city [10]*

### 2.1.3 Challenges and benefits of Smart Cities

In this subsection, we are going to see the challenges and benefits of smart cities by showing the obstacles they face and the advantages they offer.

**a) Challenges:**

- **Data privacy and security concerns:** massive data collection poses risks to citizen privacy and cyber security.
- **High initial implementation costs:** Substantial upfront investment required for smart city technologies strains municipal budgets.
- **Technology obsolescence and upgrade challenges:** Rapid technological advancements necessitate frequent, costly system updates.

**b) Benefits:**

- **Greater connection and efficiency:** Seamless integration of urban systems improves service delivery and resource utilization.

- **More intelligent use of data:** Advanced analytics enable data-driven decision-making and personalized urban services. [11]

#### 2.1.4 Examples of smart cities worldwide

- **Oslo, Norway:** focused on sustainable development. Oslo uses digitalization and new technologies to improve the daily lives of its residents (intelligent lighting, child safety applications, etc.).
- **Barcelona, Spain:** the global connected Smart City. It is fully equipped with LED sensors that monitor urban traffic, air quality, pedestrian activity and street noise.
- **Seoul, South Korea:** Advanced in digital infrastructure and e-government. [12]

### 2.2 Smart cities and sustainability

Smart cities and sustainability are closely connected because smart cities use innovative technologies to reduce environmental impacts, through the integration of innovative solutions such as the internet of Things (IoT), artificial intelligence (AI), they can manage energy, water, transportation, and waste in a smarter way. Meanwhile, the development of smart cities aims at enhancing people's living conditions, upgrading the level of public services, and easing daily life. By working towards sustainability, they create cities that are good for both current and future generations, balancing economic, social, and environmental needs.

### 2.3 Smart cities and resource management

Smart cities use advances technologies to manage resources like energy, water, and waste more efficiently. With tools like sensors, the internet of Things (IoT), and data analytics, they can monitor resource usage in real time, reduce waste and optimize systems. For example, smart grids save energy by distributing electricity based on demand, and smart water systems detect leaks to prevent water loss.

Resource management is a component of sustainability. Managing resources effectively contributes to sustainability by reducing environmental impacts and ensuring that resources remain available for the future.

## 3. Energy management systems in smart cities

Energy management systems are among the integral constituents of a smart city, integrating intelligent technologies for optimal energy use and distribution. These systems integrate smart grids and IoT sensors in the process of monitoring and controlling energy consumption with the aim of helping cities reduce waste and cut costs by promoting sustainable energy behavior. This section explores the current state of these systems, their key features, and their limitations.

## 3.1 Current energy management system

Energy management systems in smart cities optimize energy generation, distribution, and consumption through smart grids and real-time monitoring. These systems reduce energy waste, lower costs, and promote the use of renewable energy sources.

### 3.1.1 Smart grid

The smart grid is an innovation that has the potential to revolutionize the transmission, distribution and conservation of energy. Actually, the current electric power delivery system is almost entirely a mechanical system, with only limited use of sensors, minimal electronic communication and almost no electronic control. On the contrary, smart grid employs digital technology to improve transparency and to increase reliability as well as efficiency. ICTs and especially sensors and sensor networks play a major role in turning traditional grids into smart grids. [13]

Therefore, the definition of Smart Grid in E.U.'s viewpoint is:

"A smart grid is an electricity network that can intelligently integrate the actions of all users connected to it – generators, consumers and those that do both – in order to efficiently deliver sustainable, economic and secure electricity supplies." [13]

Smart grids can be defined by four key characteristics:

- **Flexibility**: They allow for more precise management of the balance between production and consumption.
- **Reliability**: They enhance the efficiency and security of the networks.
- **Accessibility**: They promote the integration of renewable energy sources across the entire network.
- **Economy**: Through better system management, they provide energy savings and cost reductions in both production and consumption. [14]

### 3.1.2 IoT sensors

Different sensing devices have been deployed in many city applications, e.g., traditional physical sensors (fixed and mobile) have been widely used to measure physical phenomenon (e.g., temperature, air quality, light and traffic flow); wearable sensors and smartphones equipped with accelerometers, magnetometer, and gyroscopes, have been used to recognize human activities and behavior. Sensors in different applications have very different sampling rates: while Electroencephalogram (EEG) sensors used in healthcare offer millisecond-range temporal resolution, air quality sensors report in minutes. Sensor measurement data is usually associated with metadata, e.g., spatial information (where the measurement happens) and temporal information (when), which is important for sensor data representation and analysis [15].

IoT sensors gather information so devices can be used remotely and data can be shared in real time. The data gathered by IoT sensors and sent to the cloud is analyzed by software that can make sense of the information and then sent to users. This data is used to track trends and gather insights about everything from efficiency and energy use in factories to athletic performance and a user's health. [16]

## 3.2 Limitations of Existing Systems

Smart Grids and IoT sensors are critical for modern energy management, they face notable limitations:

- **High initial Costs:** Building and deploying *a smart grid and IoT sensors* can be a significant investment, and many utilities may be hesitant to make such a large investment without a clear return on investment.
- **Cyber security and Data Privacy Risks:** With a vast number of interconnected devices, businesses face increased vulnerability to cyber threats, such as hacking, data breaches, and unauthorized access. Protecting sensitive data requires robust security measures, including encryption, authentication protocols, and regular updates
- **Interoperability and scalability issues:** With so many different devices and systems involved, it can be difficult to ensure that they all work together seamlessly. This can result in communication breakdowns, *data loss*, and other issues that can compromise the reliability and efficiency of the grid.
- **Data overload and management:** Both smart grids and IoT sensors generate vast amounts of data. Managing, storing and analyzing this data requires advanced infrastructure, and poor management can lead to data loss and system slowdowns.
- **Power consumption:** IoT sensors require continuous power and maintaining battery life. [17] [18]

# 4. Underlying Technologies

This section presents the key technologies enabling modern energy management. The IoT connects devices to monitor and optimize energy usage, while AI enhances decision-making through data analysis and predictive modeling. Together, they push forward efficiency and sustainability in energy systems.

## 4.1 Internet of things

The Internet of Things (IoT) describes the network of physical objects—"things"—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet. These devices range from ordinary household objects to sophisticated industrial tools. [19]

IoT devices are not limited to computers or machinery. The Internet of Things can include anything with a sensor that is assigned a unique identifier (UID). The primary goal of the IoT is to create self-reporting devices that can communicate with each other (and users) in real time. [20]

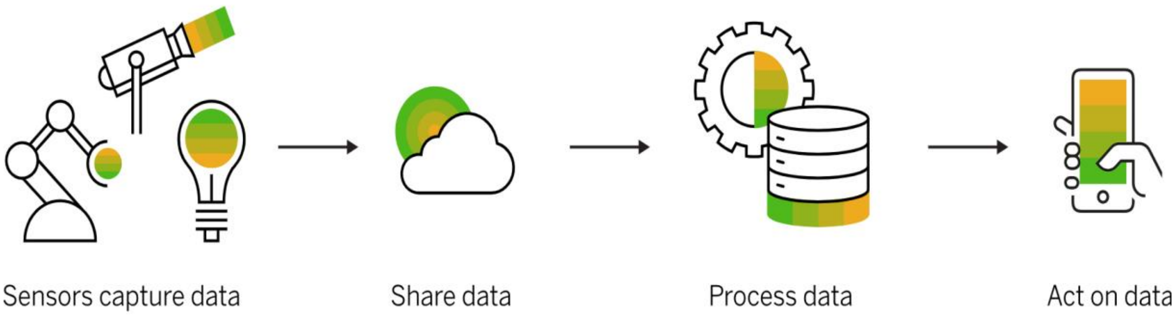



*Figure 2 the four key stages of the Internet of Things [21]*

### 4.1.1 Types of IoT Sensors for Energy Management

- **Temperature Sensors**: ensuring optimal heating and cooling while minimizing energy wastage. [22]
- **Humidity Sensors:** By managing indoor air quality and comfort. [22]

- **Light Sensors**: These devices automate lighting control by adjusting the use of electric lights based on the availability of natural sunlight, significantly reducing energy consumption. [22]
- **Motion sensors:** By detecting presence or absence, motion sensors enable automatic adjustments in lighting, heating, and cooling for unoccupied spaces, saving substantial amounts of energy. [22]
- **Energy Consumption Sensors**: Offering real-time insights into energy usage, these sensors help identify areas of high consumption for targeted energy-saving measures. [22]
- **Liquid Level Sensors and Flow Meters:** These sensors are crucial for managing the consumption and flow of liquids in industrial processes, directly impacting energy usage. [22]
- **Air Quality Sensors:** Monitoring pollutants and the quality of air indoors, these sensors can trigger ventilation systems efficiently, optimizing energy use. [22]
- **Air Flow Sensors (Quantity & Pressure):** these sensors ensure that airflow and pressure are maintained at optimal levels for energy efficiency. [22]

### 4.1.2 IoT Networks

IoT Network refers to the communication technologies Internet of Things (IoT) devices use to share or spread data to other devices or interfaces available within reachable distance. Various types of IoT networks are available for IoT devices/sensors to communicate. Choosing the proper networking protocol for given requirements is critical to collecting real-time data and accessing insights through IoT applications [23]. The internet of things identifies several types of networks used to which connected objects are attached, depending on their functionality and connectivity

a) **Personal networks:**

These are the networks attached to the person, oriented around sport and well-being, leisure, media, social networks and personal effectiveness. These networks are therefore very often linked to smartphones or miniaturized and portable devices. The main networks are:

- Bluetooth Low Energy (BLE), which allows easy connectivity a few meters away and natively available on a large number of devices.
- ANT+, which is a proprietary protocol, with many industrial partners and strongly linked to sport and well-being. [24]

### b) Building networks:

The main networks deployed RFID in buildings are numerous: WIFI, Bluetooth Low Energy (BLE), Zigbee, Z-Wave … [24]

### c) LPWAN networks:

These are the networks specifically designed for connected objects, both outdoors and indoors, to minimize energy consumption. They are ideal for transmitting basic sensor data and location to object management platforms, in a public or private cloud.

The main LPWAN networks are LoRaWAN (open standard supported by the LoRa Alliance) and Sigfox (proprietary protocol). These networks offer:

- Very low energy consumption (autonomy of 5 to 10 years).
- The very long communication distance (several km).
- Good coverage within buildings.
- The very high density of connectable objects
- Low operational costs

The counterpart of these advantages is the limited communication speed and the limitation of the number and volumes of exchanges, mainly to the connected object. [24]

### d) Cellular networks:

such as 2G, 3G, 4G, LTE-M, NB-IoT, and 5G now connects millions of IoT devices using embedded SIM cards. This is common in vehicle fleets, providing excellent service quality, international coverage, and strong security.

LTE-M and NB-IoT are energy-efficient 4G versions with improved indoor coverage and future compatibility with 5G. The advantage of 5G will enhance IoT capabilities further. Choosing the right network involves assessing range, latency and energy use, alongside strategic and regulatory factors. [24]

### 4.1.3 Data types

IoT applications rely on different types of data sources to gather information from various devices and sensors. These data sources can be categorized into three main types: passive data, active data, and dynamic data. Let us explore each type in more detail:

a) **Passive Data:**

Passive data refers to information collected without any direct interaction or intervention. It is typically generated by sensors, actuators, or devices that capture data passively as part of their normal operations. Passive data sources are always on and continuously collect information in the background. Passive data provides a constant stream of information that can be used to monitor and analyze various environmental factors. [25]

b) **Active Data:**

Active data sources involve interaction between the device or sensor and the user or system. It requires some form of action or input to generate data. Active data sources are triggered based on specific events or conditions, and they provide more contextual information compared to passive data sources. Active data sources allow for more precise data collection, enabling organizations to gather specific information as needed. [25]

c) **Dynamic Data:**

Dynamic data sources refer to real-time data that changes frequently based on the current conditions or context. Unlike passive and active data sources, dynamic data is highly volatile and requires immediate analysis and response. Dynamic data is often generated by sensors and devices that capture information related to motion, location, speed, or other temporal factors. Dynamic data sources power real-time analytics and enable organizations to detect anomalies, trigger alerts, and make timely decisions. [25]

*Table 1: Examples for each type of data*

| **Data source type** | **Examples** |
|---|---|
| Passive Data | Temperature sensors, humidity sensors, motion sensors, light sensors |
| Active Data | User input devices (buttons, touchscreens), voice recognition systems, RFID tags, barcode scanners. |
| Dynamic Data | GPS devices, accelerometers, gyroscopes, inventory tracking systems |

# 5. Existing Tools and Frameworks

Different tools and frameworks are used to support the development of smart energy management systems. These technologies assist in simulating energy systems, optimizing their consumption and integrating renewable sources. In below we will see the most used tools in energy management.

## 5.1 GridLAB-D

GridLAB-D is a flexible simulation environment that can be integrated with a variety of third-party data management and analysis tools [26]. A unique tool for designing and studying smart grids, it helps model the behavior of smart grids. Analyze energy demand and study the impacts of renewable energy integration. GridLAB-D is used to optimize energy distribution systems and manage distributed energy resources [27].

## 5.2 MATLAB/Simulink

Simulink is the platform for model-Based Design that supports system-level design, simulation, automatic code generation, and continuous test and verification of embedded systems [28].

With MATLAB and Simulink, we can design smart and efficient energy management systems (EMS) by implementing dynamic policies, incorporating real-time data, and increasing the level of automation in EMS operations. MATLAB and Simulink are used for your EMS development workflow, from data access and modeling to optimization and deployment [28].

## 5.3 OpenDSS (Open Distribution System Simulator)

OpenDSS is an electric power distribution system simulator (DSS) designed to support distributed energy resource (DER) grid integration and grid modernization. It enables engineers to perform complex analyses using a flexible, customizable and easy to use platform intended specifically to meet current and future distribution system challenges and provides a foundation for understanding and integrating new technologies and resources [29].

## 5.4 Pyomo

Pyomo is a Python-based open-source software package that supports a diverse set of optimization capabilities for formulating, solving, and analyzing optimization models including energy optimization. A core capability of Pyomo is modeling structured optimization applications [30].

### 5.5 TensorFlow / PyTorch

TensorFlow and PyTorch are a deep learning framework used for predictive tasks, optimization and intelligent management. These frameworks can be integrated into energy management system to forecast energy consumption in real-time.

### 5.6 IoT Platforms (Thing Speak, AWS IoT, Azure IoT)

Such IoT platforms as Thing Speak, AWS IoT, and Azure IoT offer capabilities for capturing, storing, and analyzing data in real-time from IoT sensors. These platforms are critical for smart energy management systems in smart cities as they enable the monitoring of real time energy usage data and performing analytics based on the data.

## 6. Conclusion

This chapter has explored key principles and technologies underlying smart cities and energy management. It reviewed challenges and opportunities related to urbanization, energy demand, and the increasing role of smart technologies such as IoT and AI to improve sustainability and efficiency. The dimensions of smart cities, their benefits, and integration with energy management systems from the core of this chapter for developing a clear understanding of innovative approaches for solving modern urban energy problems.

# *Chapter 02*

## *Artificial Intelligence and Deep Learning*

# Chapter 02: Artificial Intelligence and Deep Learning

## 1. Introduction

Artificial Intelligence is a part of our daily life now. Intelligent algorithms control our online behavior, help make decisions in diverse fields, and drive innovation. Machine Learning and Deep Learning are the key ideas behind this technology. In energy management, AI becomes even more valuable when combined with human expertise. It helps process large amounts of data quickly, providing real-time insights that improve efficiency. In this chapter, we will cover the basics, real-world applications, and future possibilities of these fields.

## 2. Artificial intelligence and machine learning

### 2.1 Artificial intelligence

Artificial intelligence (AI) is technology that enables computers and machines to simulate human learning, comprehension, problem solving, decision-making, creativity and autonomy [31]. AI also has an opportunity to change the way that companies manage and consume energy. AI technology that can process data quickly and efficiently, combined with technologies like blockchain and distributed energy resources, will allow both grid operators and consumers the ability to effectively manage energy in an increasingly decentralized landscape. The value of AI for energy management comes to life when technology is met with human expertise. With the specialized insight from energy market experts, insights that come from AI monitoring can be applied to corporate strategies and decision-making. With the support of digital technologies, humans can save time on large amounts of data processing, and drive faster - even instantaneous - insights that improve operations and efficiency. [32]

### 2.2 Machine learning

Machine Learning, often abbreviated as ML, is a subset of artificial_intelligence (AI) that focuses on the development of computer algorithms that improve automatically through experience and by the use of data. In simpler terms, machine learning enables computers to learn from data and make decisions or predictions without being explicitly programmed to do so [33]. ML algorithms can automatically identify the most informative features and optimize their contribution to the forecasting model. ML algorithms can provide real-time forecasts, enabling grid operators and energy managers to make informed decisions about energy production, distribution, and consumption. [34]

### 2.3 Deep Learning

Deep learning is a subset of machine learning that uses multilayered neural networks to automatically learn patterns from data. It powers many AI applications, like natural language processing, computer vision, and time series forecasting. In energy forecasting, deep learning improves predictions of consumption and production by analyzing historical data and external factors, helping optimize power grids and integrate renewable energy sources [35].

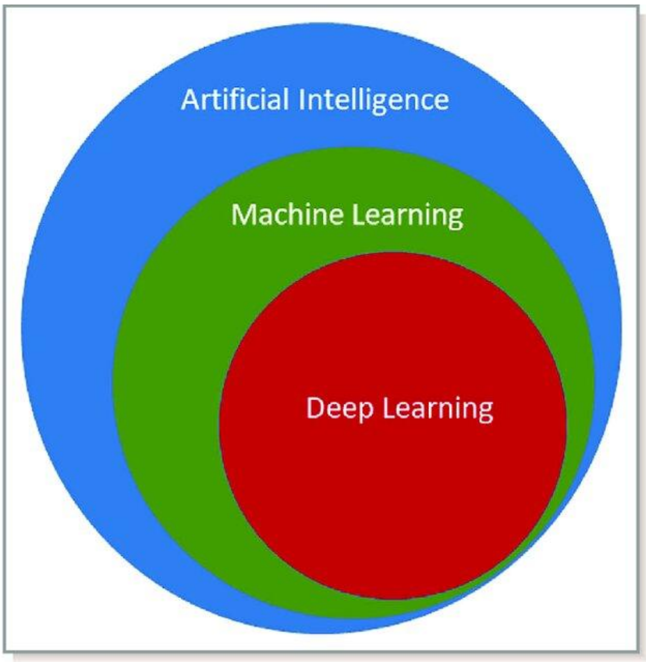



*Figure 3 : Relationship between AI, ML and DL [36]*

## 3. Types of machine learning

Machine learning (ML) techniques are generally classified into three types: supervised, unsupervised, and reinforcement learning.

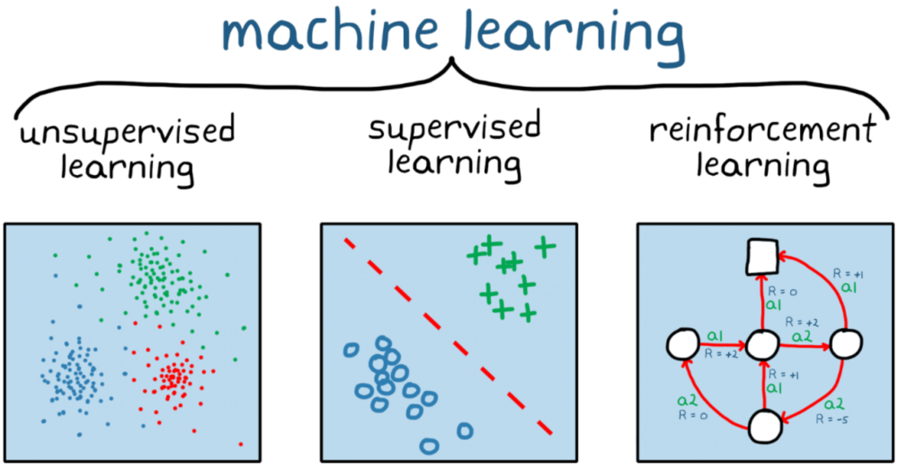


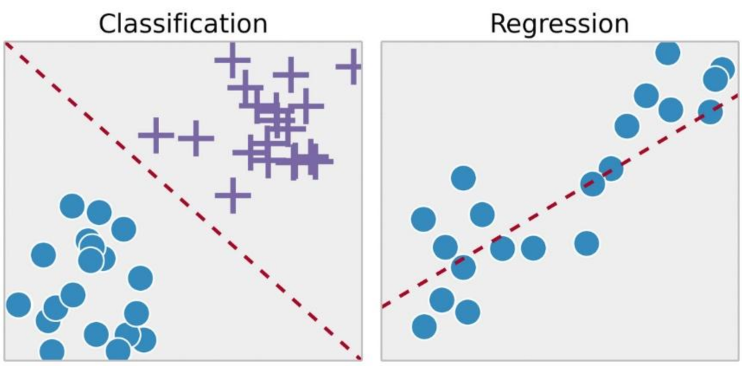

*Figure 4: the main types of machine learning [37]*

## 3.1 Supervised learning

Supervised learning is a type of machine learning in which an algorithm is trained on a labeled dataset. Each input in the dataset is associated with a desired output. The goal is to enable the algorithm to generalize to new unlabeled data by learning the relationships between inputs and outputs.

This category is divided into two main subcategories:

- **Classification:** classification is a supervised learning task where a model is trained on a labeled dataset to predict a class or category of a new observation.
- **Regression:** regression is a supervised learning task where a model is trained to predict a continuous numerical value on input features.




c

*Figure 5: supervised learning "classification vs. regression" [38]*

## 3.2 Unsupervised learning

Unsupervised learning is a branch of machine learning that deals with unlabeled data. Unlike supervised learning, where the data is labeled with a specific category or outcome, unsupervised learning algorithms are tasked with finding patterns and relationships within the data without any prior knowledge of the data's meaning. Unsupervised machine learning algorithms find hidden patterns and data without any human intervention, i.e., we don't give output to our model. The training model has only input parameter values and discovers the groups or patterns on its own. [39]

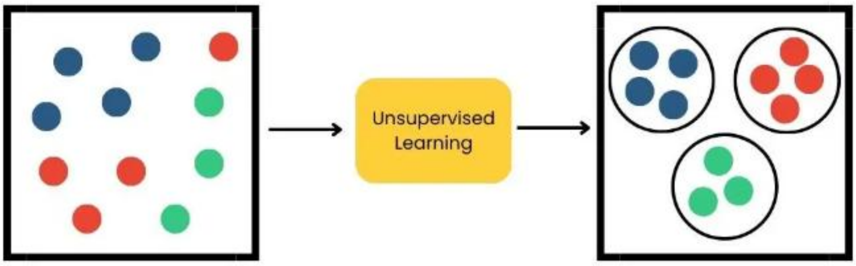



*Figure 6: unsupervised learning "clustering" [40]*

## 3.3 Reinforcement learning

Reinforcement learning (RL) is a type of machine learning process that focuses on decision making by autonomous agents. An autonomous agent is any system that can make decisions and act in response to its environment independent of direct instruction by a human user. Robots and self-driving cars are examples of autonomous agents. In reinforcement learning, an autonomous agent learns to perform a task by trial and error in the absence of any guidance from a human user.[1] It particularly addresses sequential decision-making problems in uncertain environments, and shows promise in artificial intelligence development.

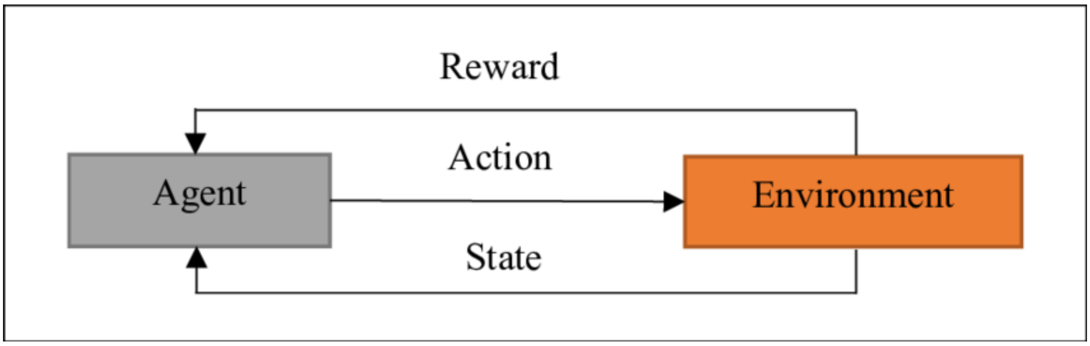



*Figure 7: Reinforcement learning [41]*

# 4. Fuzzification

Fuzzification is the process of converting crisp inputs into fuzzy sets, allowing a system to handle the inherent uncertainity and vagueness of real-world data. This is achieved by using membership functions, which define the degree of membership of each input to various categories. [42]

## 4.1 Steps of fuzzification

### 4.1.1 Input values

Fuzzification starts with real, crisp inputs (temperature, speed, height, etc.) that are measured from the environment or system. [43]

### 4.1.2 Membership functions

- A membership function (MF) defines how each input value corresponds to a fuzzy set. A fuzzy set is a collection of elements, each of which has a degree of membership between 0 and 1.

  For example, a temperature input of 35°C might belong to the fuzzy sets "cold", "warm", and "hot" with varying degrees of membership (0.2 for cold, 0.5 for warm, and 0.8 for hot) [43]

### 4.1.3 Fuzzy sets

- Fuzzy sets represent categories or linguistic variables. These sets could represent concepts like "low", "medium", "high", or more specific terms like "very hot", etc.
- The fuzzification process calculates the degree to which each crisp input belongs to each fuzzy set. [43]

## 4.2 Membership functions types

- **Triangular MF:** Characterized by a triangle shape, with the peak representing the highest degree of membership.
- **Trapezoidal MF:** Similar to triangular but with a flat top.
- **Gaussian MF:** A bell-shaped curve, which is ideal for smooth transitions [43]

# 5. Time-Series Analysis

Time Series Analysis (TSA) refers to the study of how a response variable changes over time, with time being the independent variable. TSA is used to predict or forecast a target variable based on patterns over time. TSA involves a sequence of observations over the same periods of time, which may be years and months, days, hours, minutes, or seconds. Some of the well-known applications of real-world applications are weather forecasting, stock market prediction, signal processing, and control systems. Unlike spatial or other types of analysis, TSA focuses on ordered time-dependent data. Various models, such as AR, MA, ARMA, and ARIMA, are commonly used for forecasting in time series analysis. [44]

## 5.1 Data Types of Time Series

Time series data can be divided into two categories: stationary and non-stationary, each with distinct characteristics and implications.

**Stationary:** A data set is called stationary if it meets certain key conditions. It must not show trends, seasonality, cycles or any irregular patterns, also:

- The **mean** value of them should be completely constant in the data during the analysis.
- The **variance** should be constant with respect to the time-frame
- **Covariance** measures the relationship between two variables.

**Non-stationary:** a dataset is classified as non-stationary if any of these properties (mean, variance, or covariance) change over time. [44]

## 5.2 Benefits of time series algorithms in Energy Management

Time series algorithms play a very important role in energy consumption optimization, grid stability, and better decision-making. Here are the key benefits:

- **Accurate Energy Demand Forecasting:** Helps predict short-term and long-term energy consumption, thereby enabling utilities and smart grids to optimize energy distribution and reduce waste.
- **Improved Load Balancing:** Time series models anticipate peak demand periods, thus enabling better load distribution and avoiding blackouts or grid overloads.
- **Anomaly and Fault Detection:** It detects anomalies in energy usage; hence, any equipment failure, energy theft, or inefficiency is detected in real time.

- **Renewable Energy Integration:** It predicts the generation of solar and wind energy based on historical weather patterns, hence enhancing grid stability and better utilization of renewable energy.
- **Optimized Energy Storage Management:** It predicts energy supply-demand fluctuations, hence optimizing battery storage and reducing reliance on fossil fuels.
- **Cost Efficiency and Reduction:** Helps businesses alter their usage pattern to dynamically priced models, hence helping them in reducing their cost of electricity [45].

## 5.3 Deep learning and time series

Deep Learning has been gaining more popularity for Time Series Analysis and Forecasting in recent years, with success in addressing problems where classical Machine Learning based methods struggled did not deliver. In this subsection, we will explore various Deep Learning techniques for time series forecasting.

### 5.3.1 Artificial Neural Networks (ANN)

Artificial neural networks (ANNs) are created to replicate how the human brain processes data in computer systems. Neurons within interconnected units collaborate to identify patterns, acquire knowledge from data, and generate predictions. Artificial neural networks (ANNs) are commonly employed in activities such as identifying images, processing language, and making decisions. Like human brains, artificial neural networks are made up of neurons that are connected like brain cells. These neurons process and receive information from nearby neurons before sending it to other neurons [46]. A neural network consists of three layers. The first layer is the input layer. It contains the input neurons that send information to the hidden layer. The hidden layer performs the computations on input data and transfers the output to the output layer. It includes weight, activation function, and cost function. The connection between neurons is known as weight, which is the numerical values. The weight between neurons determines the learning ability of the neural network. During the learning of artificial neural networks, weight between the neuron changes. [47]

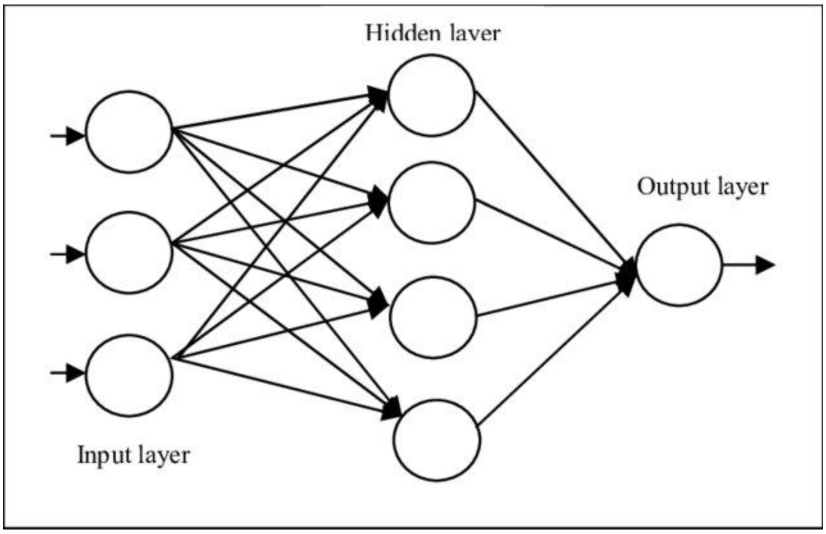



*Figure 8: ANNs Architecture*

### 5.3.2 Multilayer Perceptron (MLP)

A multi-layer perceptron is a class of feedforward artificial neural networks (ANNs) that consists of three or more layers composed of neurons. The neurons usually use nonlinear activation functions that enable the network to learn in complex patterns in data. The MLP plays a significant role in machine learning; it can model any nonlinear relationship, making them highly effective for tasks like classification, regression, and pattern recognition.

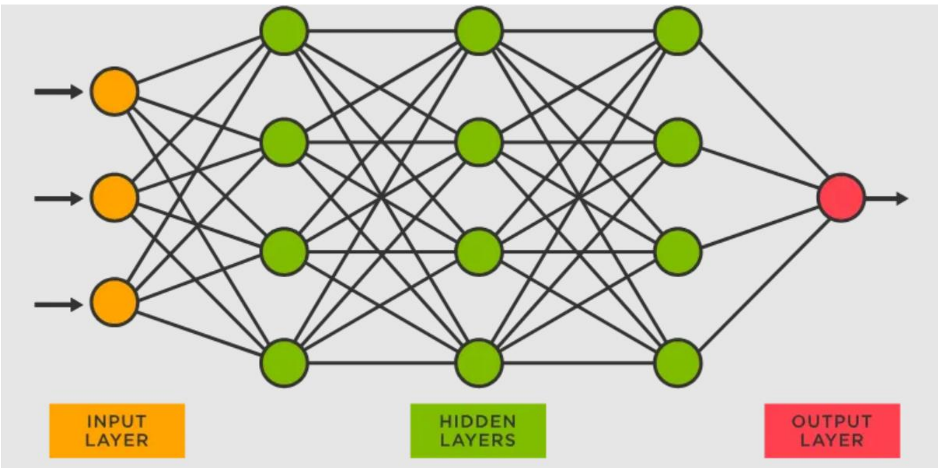



*Figure 9: MLPs architecture [48]*

### 5.3.3 Recurrent Neural Networks (RNN)

A recurrent neural network (RNN) is a deep learning model that is trained to process and convert a sequential data input into a specific sequential data output. Sequential data is data, such as words, sentences, or time series data, in which the sequential components are interconnected based on complex semantic and syntax rules. An RNN is a software system composed of many interconnected components that mimic the way humans perform sequential data conversions, such as translating text from one language to another. [49] Recurrent Neural Network (RNN) is a type of Neural Network where the output from the previous step is fed as input to the current step. In traditional neural networks, all the inputs and outputs are independent of each other. [50]

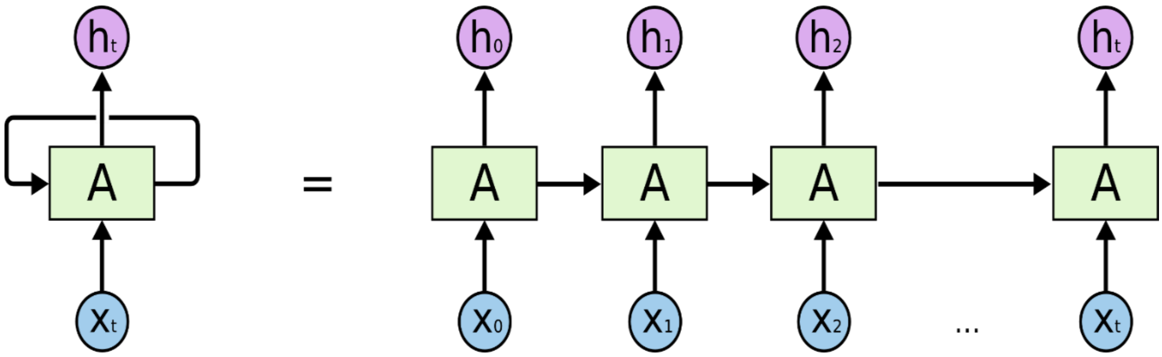



*Figure 10: RNNs Architecture [51]*

### 5.3.4 Long Short-Term Memory (LSTM)

LSTM (Long Short-Term Memory) is a recurrent neural network (RNN) architecture widely used in Deep Learning. It excels at capturing long-term dependencies, making it ideal for sequence prediction tasks. [52]

A traditional RNN has a single hidden state that is passed through time, which can make it difficult for the network to learn long-term dependencies. **LSTMs model** addresses this problem by introducing a memory cell, which is a container that can hold information for an extended period. LSTM architectures are capable of learning long-term dependencies in sequential data, which makes them well suited for tasks such as language translation, speech recognition, and time series forecasting. [53]

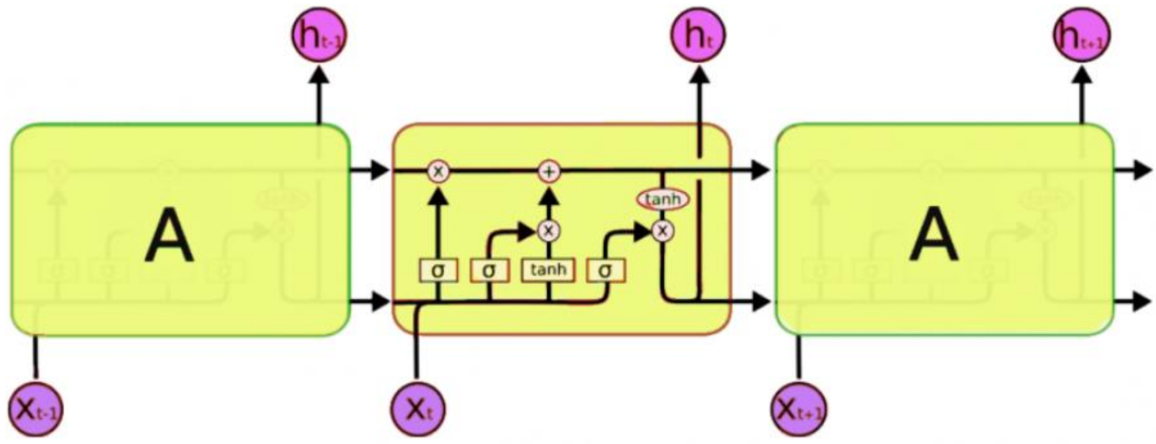



*Figure 11: LSTMs Architecture [54]*

### *5.3.5* GRU (Gated Recurrent Unit)

The Gated Recurrent Unit (GRU) is a recurrent neural network (RNN) architecture that resembles to LSTM. Like LSTM, GRU is made to simulate sequential data by permitting selective remembering or forgetting of information over time, GRU has a simpler architecture than LSTM with fewer parameters, which can make it easier to train and more computationally efficient [55].

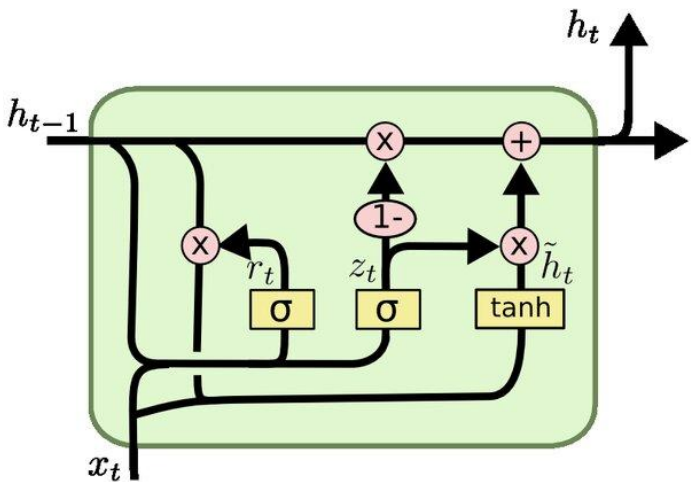



*Figure 12: GRUs Architecture [56]*

# 6. Ensemble learning Methods:

Ensemble methods are a class of machine learning techniques that combine several models with the aim of improving predictive accuracy, robustness, and generalized performance. It makes a prediction based on the aggregate contributions from many weak learners rather than depending on any single model. This learning is especially effective in handling high complexities of datasets for which over fitting reduction allows for improved classification and regression tasks [57].

## 6.1 Types of Ensemble Methods

### 6.1.1 Bagging

(Bootstrap Aggregating) is an ensemble learning technique designed to improve the accuracy and stability of machine learning algorithms. It involves the following steps:

- **Data Sampling**: Creating multiple subsets of the training dataset using bootstrap sampling (random sampling with replacement).
- **Model Training**: raining a separate model on each subset of the data.
- **Aggregation**: Combining the predictions from all individual models (averaged for regression or majority voting for classification) to produce the final output.

- **Key Benefits**:

  - **Reduces Variance**: By averaging multiple predictions, bagging reduces the variance of the model and helps prevent overfitting.
  - **Improves Accuracy**: Combining multiple models usually leads to better performance than individual models.

**Example: Random Forest**, which averages multiple decision trees for improved accuracy [58].

### 6.1.2 Boosting

Boosting is another ensemble learning technique that focuses on creating a strong model by combining several weak models. It involves the following steps:

- **Sequential Training**: Training models sequentially, each one trying to correct the errors made by the previous models.
- **Weight Adjustment**: Each instance in the training set is weighted. Initially, all instances have equal weights. After each model is trained, the weights of

misclassified instances are increased so that the next model focuses more on difficult cases.

- **Model Combination**: Combining the predictions from all models to produce the final output, typically by weighted voting or weighted averaging.

- **Key Benefits**:
  - **Reduces Bias**: By focusing on hard-to-classify instances, boosting reduces bias and improves the overall model accuracy.
  - **Produces Strong Predictors**: Combining weak learners leads to a strong **predictive model**.

**Examples: Gradient Boosting, XGBoost, AdaBoost, LightGBM.**

### 6.1.3 Stacking

(Stacked Generalization) is an ensemble learning technique that aims to combine multiple models to improve predictive performance. It involves the following steps:

- **Base Models**: Training multiple models (level-0 models) on the same dataset.
- **Meta-Model**: Training a new model (level-1 or meta-model) to combine the predictions of the base models. Using the predictions of the base models as input features for the meta-model.

- **Key Benefits**:
  - **Leverages Model Diversity**: By combining different types of models, stacking can capture a wide range of patterns in the data.
  - **Improves Performance**: The meta-model learns how to best combine the predictions from the base models, often leading to improved performance over individual models.

### 6.1.4 Voting & Averaging

- **Voting (for classification):** To combine several models, it selects the majority vote in hard voting or averages the probabilities in soft voting.
- **Averaging (For regression)**: calculates the mean or weighted average of several models' predictions.

## 6.2 Ensemble learning techniques:

### 6.2.1 Random Forest

**Random Forest** algorithm is a powerful tree learning technique in Machine Learning to make predictions and then we do vote of all the tress to make prediction. They are widely used for classification and regression task.

- It is a type of classifier that uses many decision trees to make predictions.
- It takes different random parts of the dataset to train each tree and then it combines the results by averaging them. This approach helps improve the accuracy of predictions. Random Forest is based on ensemble learning [59].

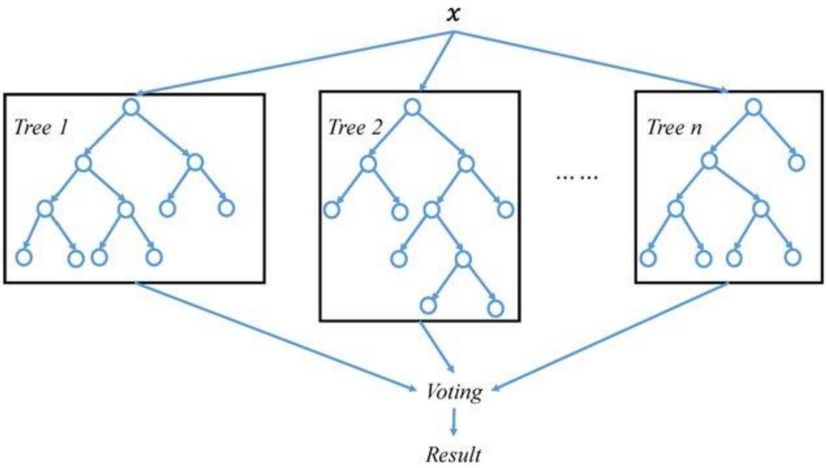



*Figure 13: A general architecture of Random Forest*

### 6.2.2 EXtreme Gradient Boosting

**Gradient Boosting** is an effective boosting algorithm that merges multiple weak learners to create strong learners, where each new model focuses on reducing the loss function, like mean squared error or cross-entropy, of the preceding model through gradient descent. In every iteration, the algorithm calculates the gradient of loss function related to the current ensemble's predictions and subsequently trains a new weak model to reduce this gradient. The forecasts from the new model are subsequently incorporated into the ensemble, and this procedure is repeated until a stopping criterion is satisfied [60].

**EXtreme Gradient Boosting (XGBoost)** is an optimized implementation of Gradient Boosting that enhances its efficiency, speed, and performance. It incorporates advanced regularization technique such as L1 (Lasso) L2 (Ridge) regularization to prevent over fitting and enhance generalization. Unlike traditional Gradient Boosting, in XGBoost, a highly optimized parallel tree boosting system is used, making it much faster and more scalable for large datasets. It also has built-in handling of missing values, sparsity-aware algorithms, and tunable hyperparameters: learning rate, maximum tree depth, and the number of boosting rounds. These enhancements make XGBoost one of the most powerful and widely used machine learning algorithms for structured data, with top-of-the-line-performance in both classification and regression tasks [61].

### 6.3 Benefits of Ensemble Methods in Energy Management

The ensemble methods are very effective in enhancing energy forecasting, anomaly detection, and optimization. Key benefits include the following:

- **Improved Energy Demand Forecasting:** The integration of different models, such as Random Forest, XGBoost, and LSTM, offers higher accuracy in both short-term and long-term energy consumption.
- **Better Handling of Seasonal Variations:** Ensemble techniques capture complex energy consumption patterns, including seasonal and weather-related fluctuations.
- **Improved Grid Stability and Load Balancing:** Better energy predictions support utilities in optimally distributing the generated power, avoiding blackout incidents while maintaining load balance.
- **Optimized Integration of Renewable Energy**: Ensemble models improve the forecast of solar and wind power generation, hence help in better grid management and storage solutions.
- **Less Overfitting and Bias:** The ensemble methods of both Bagging most famously implemented through Random Forest-and Boosting-XGBoost, LightGBM among others-help to make the models generalize better on unseen energy data.

## 7. Importance of AI in Energy Management

Artificial Intelligence (AI) is transforming energy management by improving efficiency, reducing costs, and enabling sustainable energy practices. AI technologies are pivotal in optimizing energy consumption, integrating renewable energy sources, and enhancing grid management. By leveraging AI-driven analytics, predictive modeling, and real-time monitoring, organizations can make informed decisions that drive energy efficiency and sustainability. Here's how AI works in energy management:

- ➢ Optimizing Energy Consumption

- Enhancing Renewable Energy Integration
- Improving Energy Efficiency in Industrial Processes
- Facilitating Data-Driven Decision Making
- Supporting Sustainable Energy Practices

# 8. Existing works on Energy Management

This section explores different works on energy management, focusing on prediction and optimization methods to enhance efficiency, sustainability and decision-making in energy systems.

In **Razak Olu-Ajayi [55],** the study focuses on predicting the annual energy consumption in buildings using machine-learning algorithms to improve decision-making. The authors utilized data from 300 buildings around the UK. This data includes internal variables (building characteristics) and external factors (weather conditions). Three models were developed and evaluated in this approach: Support Vector Machine (SVM), Decision Trees (DT) and Artificial Neural Networks (ANN). After the preparation (preprocessing and normalizing) of the data, the models were trained and tested using a split of 70% for training and 30% for testing. Model performance was evaluated by $R^2$, MAE, RMSE and MSE metrics. Results show that the ANN model performed best, with $R^2$ value of 0.66 and MAE of 2.20, outperforming the SVM model, which had an $R^2$ value of 0.59 and MAE of 2.40, while DT had the poorest results, with an $R^2$ value of 0.27 and MAE of 3.17. Although the training time for the SVM model was shorter, the ANN model showed superior performance in handling big data and providing more accurate predictions of annual energy consumption. It also underlines how ANN-based models may enable an early-stage design adjustment to avoid energy inefficiencies in the buildings [62].

In **S. Sehovac** [63], the study concentrates on the application of deep learning techniques, advanced like S2S, to forecast the energy consumption of a building. The research overcomes the traditional feed forward neural networks limitations in time-series forecasting by deploying RNNs, including GRU and LSTM models. These models are modified to perform short, medium and long-term energy usage predictions based on five-minute incremental data gathered through smart meters.

Then, the models are being trained on data for a year and three months in a single building with eleven features related to temperature, humidity, energy usage, etc. the GRU and LSTM based S2S architecture was compared with the classic RNN and the DNN. In the performance evaluation, MAE and MAPE are considered. The GRU S2S model gives the best accuracy among all models in this research across all prediction lengths. For example, in the long-term scenario, GRU S2S had a 14.05% lower MAPE than that of the LSTM model. These findings describe the effectiveness and efficiency of the GRU model for long-term time-series forecasting due to the simplified structure resulting in a faster convergence.

This work highlights the potential of Sequence-to-Sequence models in enhancing energy load forecasting and reducing energy inefficiencies in commercial buildings. It also opens up further research avenues for optimizing these architectures for longer-term predictions and applying them to diverse datasets [63].

In **Roberto Morcillo-Jimenez** [64]**,** The study investigates the application of deep learning methodologies in predicting energy consumption in office buildings, which is one of the critical needs for energy efficiency. Using data from the ICPE office building in Bucharest, Romania, this research investigates memory-based and feed forward neural network architectures, along with time-series analysis. The paper aims to demonstrate the comparison in the performance of different models for short-term energy consumption forecasting over a 12-hour horizon. The different models that were reviewed are Seq2Seq, RNN, MLP, CNN, and XGBoost.

This data has a span of three winter months and underwent exhaustive preprocessing-noisy and missing data. The best features extracted included outside temperature, occupancy, and heating consumption of the buildings.

Performance highlights include:

- **The Seq2Seq models** yielded the best accuracy during January, with a Normalized Mean Absolute Error (NMAE) of 0.21 in Zone D1 and 0.20 for Zone D5/2.
- **RNNs** supplied robust predictions for March, achieving an NMAE of 0.29 in both zones.
- **CNNs** performed best in February, thanks to their ability in capturing local patterns when missing values reached a peak.

- **XGBoost and MLPs** showed relatively poor performance, as they cannot capture temporal dependencies.

This work confirms that memory-based architectures, like S2S and RNN, outperform others in energy consumption forecasting tasks, mainly when proper preprocessing is applied to the data. On the other hand, it points to the potential of CNNs in scenarios where stable trends can be found with lacking data. These results come in support of using state-of-the-art deep learning models to further improve energy management is non-residential buildings. Future work will investigate transformer-based architectures and larger datasets for better scalability and prediction accuracy. [64]

## 8.1 Comparison of the previous works

This table provides a concise comparison of the three works, summarizing their objectives, datasets, methodologies, and key conclusions.

*Table 2: Comparison of Studies on Energy Consumption Prediction and Forecasting Methods*

| Paper | Title | Objective | Dataset | Models used | Performance Metrics |
|---|---|---|---|---|---|
| **01** | Building energy consumption prediction using ML | Predict annual energy consumption in buildings using ML models | 300 buildings in the UK (internal variables + weather conditions) | SVM, Decision Trees (DT), Artificial Neural Networks (ANN) | ANN performed best ($R^2$ = 0.66, MAE = 2.20), while DT had the lowest performance ($R^2$ = 0.27, MAE = 3.17). ANN models help in early-stage design adjustments for energy efficiency. |
| **02** | Forecasting Building Energy Consumption with Deep Learning (Seq2Seq Approach) | Improve time-series forecasting of energy consumption using deep learning | One building, 1 year and 3 months of smart meter data (11 features) | GRU S2S, LSTM S2S, RNN, DNN | GRU S2S outperformed all models, with 14.05% lower MAPE than LSTM for long-term forecasting. S2S models improve energy load |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | forecasting and efficiency in commercial buildings. |
| **03** | Deep learning for energy consumption prediction in office buildings | Compare deep learning models for short-term energy forecasting (12-hour horizon) | ICPE office building (Bucharest, Romania), 3 winter months (preprocessed data) | Seq2Seq, RNN, MLP, CNN, XGBoost | Seq2Seq performed best in January (NMAE = 0.21), RNN in March (NMAE = 0.29), CNN in February (handling missing data). XGBoost and MLP struggled with temporal dependencies. Memory-based models are superior for energy forecasting. |

# 9. Conclusion

This chapter covered the fundamentals of Artificial intelligence (AI), Machine learning (ML), and Deep learning (DL), and their applications in energy management. We explored how AI and ML improve energy forecasting. Deep learning models, such as LSTM and GRU, are particularly useful for analyzing time-series data, helping predict energy demand and optimize usage. Additionally, fuzzification techniques help handle uncertainty in data, enhancing the accuracy of AI systems. Ensemble learning methods, which combine multiple models for improved predictions, further enhance the effectiveness of energy management systems. In conclusion, these techniques are transforming energy management, making systems smarter, more efficient, and more sustainable.

# *Chapter 03*

## *Proposed approach*

# Chapter 03: Proposed approach

## 1. Introduction

In this chapter, we propose an approach for building energy consumption analysis using the Building Data Genome Project 2 data set [65]. The goal is to forecast energy usage by applying advanced deep learning techniques. We outline the principal steps of our method, ranging from data preparation to results analysis.

## 2. Description of the dataset (Building Data Genome Project 2)

The Building Data Genome Project 2 (BDG2) is an open dataset consisting of 1,636 non-residential buildings' energy consumption data. It includes 3,053 energy meters' readings, taken at an hourly frequency for two years (2016 and 2017), which total approximately 53.6 million readings. The data was collected at 19 sites across North America and Europe, with multiple meters per building collecting electricity, heating and cooling water, steam, solar, and water and irrigation data. Part of this data was used in the ASHRAE Great Energy Predictor III (GEPIII) competition that was hosted on Kaggle in 2019.

This dataset is being made available for many applications like energy forecasting, anomaly detection, energy analysis, and building type classification. This dataset also has corresponding weather data, enabling investigation of the influence of climatic conditions on buildings' energy use. [66]

*Table 3: Dataset Overview*

| | |
|---|---|
| **Dataset name** | Building Data Genome Project 2 (BDG2) |
| **Dataset Type** | Multi-variable, time series |
| **Release year** | 2020 |
| **Total instances** | 53.6 million measurements |
| **Number of features** | 83 |
| **Number of buildings** | 1636 |
| **Total Meters** | 3053 |
| **Collection period** | 2016-2017 |
| **Measurement Frequency** | Hourly |

| | |
|---|---|
| **Meter types** | Electricity, hot water, chilled water, steam, gas, irrigation, solar energy |

## 2.1 Metadata of the dataset

The Building Data Genome Project 2 (BDG2) dataset includes metadata that describes 1,636 buildings and 3,053 energy meters. This metadata provides essential information such as building identifiers, primary use types (e.g., office, education), subcategories, industry classifications, time zones, floor areas, and years of construction. Additionally, it details the types of energy meters (like electricity, heating, cooling), their measurement units, and data recording frequencies. This comprehensive metadata supports effective analysis and comparison of energy usage across diverse buildings, facilitating research in energy efficiency and machine learning applications.

*Table 4: Number of buildings and meters per site ID*

| Site | Buildings | Meters |
|---|---|---|
| Panther | 136 | 299 |
| Robin | 52 | 67 |
| Fox | 137 | 306 |
| Rat | 305 | 305 |
| Bear | 92 | 92 |
| Lamb | 147 | 265 |
| Eagle | 106 | 298 |
| Moose | 15 | 43 |
| Gator | 74 | 74 |
| Bull | 124 | 308 |
| Bobcat | 36 | 116 |
| Crow | 5 | 15 |
| Wolve | 36 | 66 |
| Hog | 163 | 336 |
| Peacock | 47 | 106 |
| Cockatoo | 124 | 282 |
| Shrew | 9 | 13 |
| Swan | 21 | 55 |
| Mouse | 7 | 7 |

# 3. Hardware and Software

## 3.1 Execution Environment

Our experiment requires suitable hardware to operate properly, we ensure, before we implement our approach that we possess the necessary hardware specifications.

Below are the specifications of the PC used for this work:

- Device Name: DESKTOP-IF401A1
- Processor: Intel(R) Core(TM) i5-8365U CPU @ 1.60GHz, 1.90GHz
- Installed RAM: 16.0 GB

## 3.2 Tools used

### 3.2.1 Anaconda

Anaconda is an open-source Python and R programming languages, which are used for data science, machine learning, and artificial intelligence applications. It uses the Conda package manager that analyzes the environment before installing packages to prevent compatibility problems with existing frameworks. [67]

### 3.2.2 Jupyter

Jupyter Notebook is an open-source web application that allows users to create and share documents containing live code, equations, visualizations, and narrative text. It is widely used by data scientists, researchers, and educators for interactive computing and data analysis. [68]

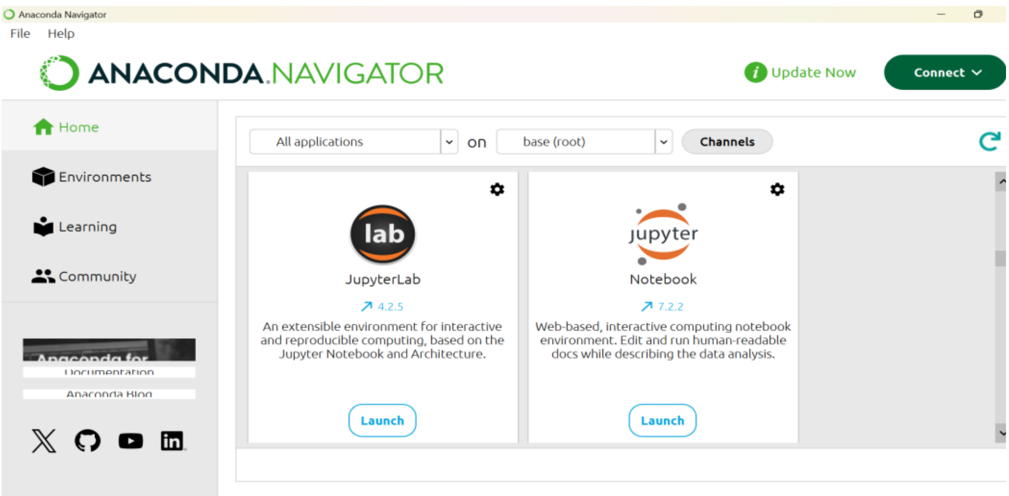



*Figure 14: Anaconda's interface*

## 3.3 Languages used

### 3.3.1 Python

Python is a programming language that is interpreted, object-oriented, and considered to be high-level too. The latest version, Python 3.13, is used for web development, machine learning applications, and new software industry technologies. One of Python's biggest strengths is its vast collection of standard libraries, including **NumPy, Pandas, Scikit-learn, TensorFlow, and Keras**.

## 3.4 Libraries used

*Table 5: Libraries used*

| Library | Description |
|---|---|
| TensorFlow | Open-source library for machine learning. |
| Keras | High-level interface for TensorFlow. |
| Scikit-learn | Machine learning library with efficient implementations of ML algorithms. |
| Pandas | Library for data manipulation and analysis. |
| NumPy | Library for numerical computation in python. |
| Matplotlib | Library for creating static, animated, and interactive visualizations in Python. |
| Seaborn | Library based on Matplotlib for statistical data visualization. |

# 4. Our proposed approach

## 4.1 Experiment 01: Apply times series on Genome dataset

In this experiment, we apply two deep learning models (LSTM and GRU) for energy consumption prediction using the Genome dataset. Three types of consumption are considered: Gas, Water, and Electricity. The data is prepared by merging the metadata and meter datasets (electricity, gas, and water) to create a complete dataset. Preprocessing steps include handling missing values, scaling features, and creating sequences for time series modeling. The dataset is then split into training (80%) and testing (20%) sets. Finally, the predictions are evaluated using standard metrics: MAE, RMSE, and MAPE. The overall process is shown in Figure 15.

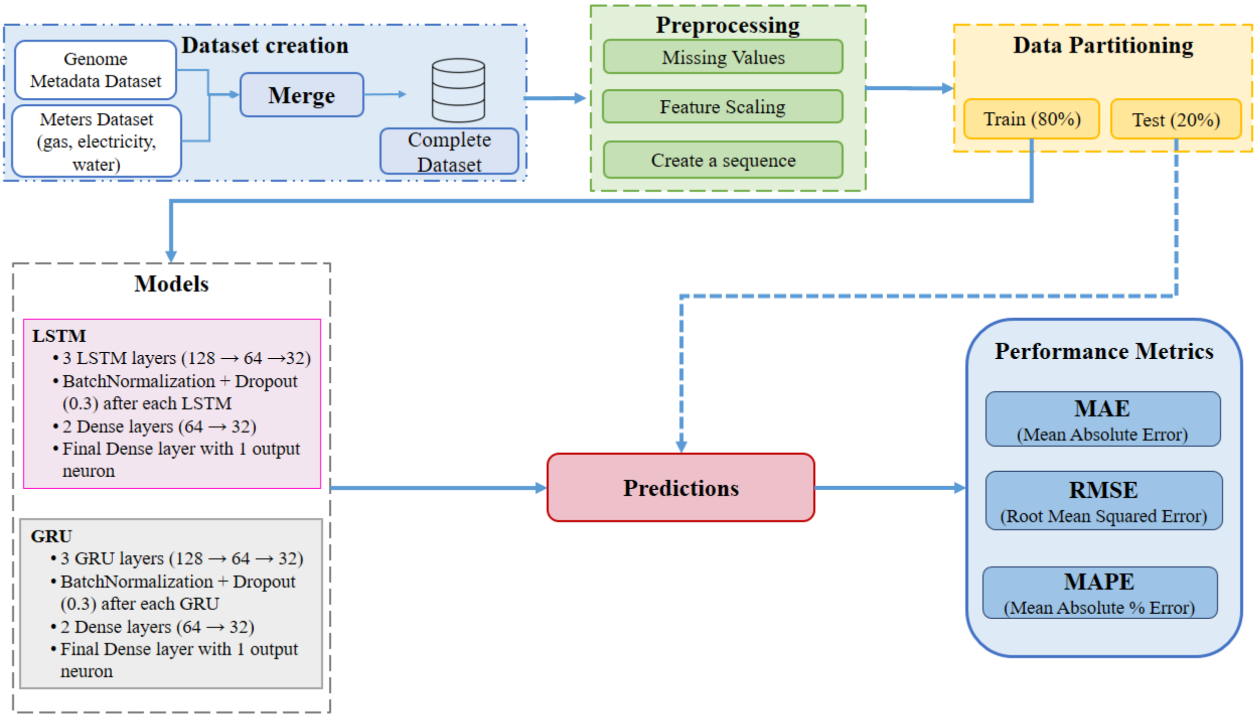



*Figure 15: Proposed model on Genome times series*

#### 4.1.1 LSTM (long-short term memory)

The following table 7 presents the architecture of the LSTM model used in our experiments, along with its performance results for predicting electricity, gas, and water consumption. The model consists of multiple LSTM layers followed by dense layers, with regularization techniques such as Batch Normalization and Dropout applied to improve generalization. The results are reported using three standard evaluation metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE), separately for each type of energy consumption.

*Table 6: LSTM Model Architecture and Performance Results by Energy Type*

| **LSTM Architecture** | **Electricity results** | **Gas results** | **Water results** |
|---|---|---|---|
| • Three LSTM layers (128 → 64 →32).<br>• Batch Normalization + Dropout (0.3) after each LSTM. | **MAE :** 300602.74<br><br>**RMSE:** 354400.51 | **MAE :** 566883.78<br><br>**RMSE:** 744246.88 | **MAE :** 506889.94<br><br>**RMSE :** 597035.39 |

| | | | |
|---|---|---|---|
| • Two Dense layers (64 → 32).<br>• Final Dense layer with one output neuron. | **MAPE**: 6.23% | **MAPE:** 23.13% | **MAPE:** 38.93% |

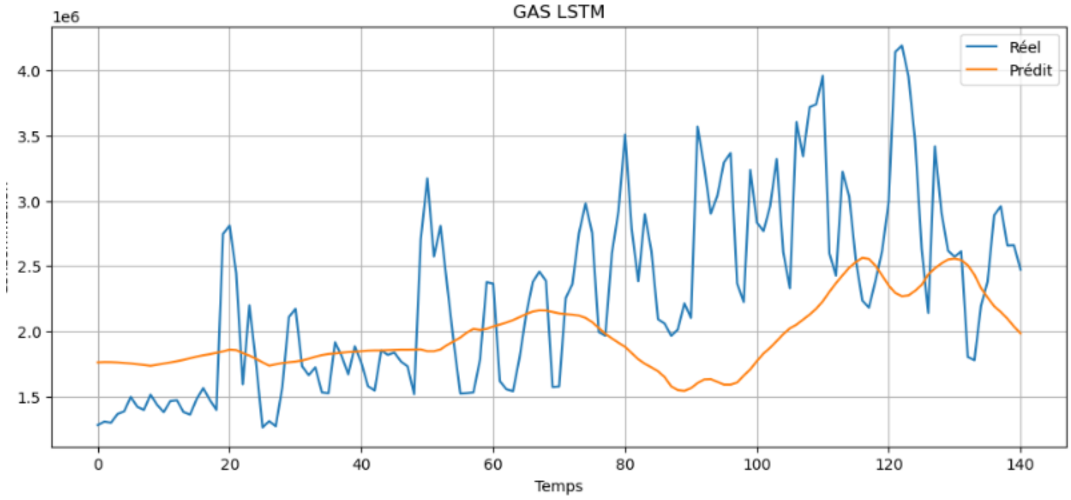



*Figure 16: Actual vs Predicted Gas Consumption – LSTM*

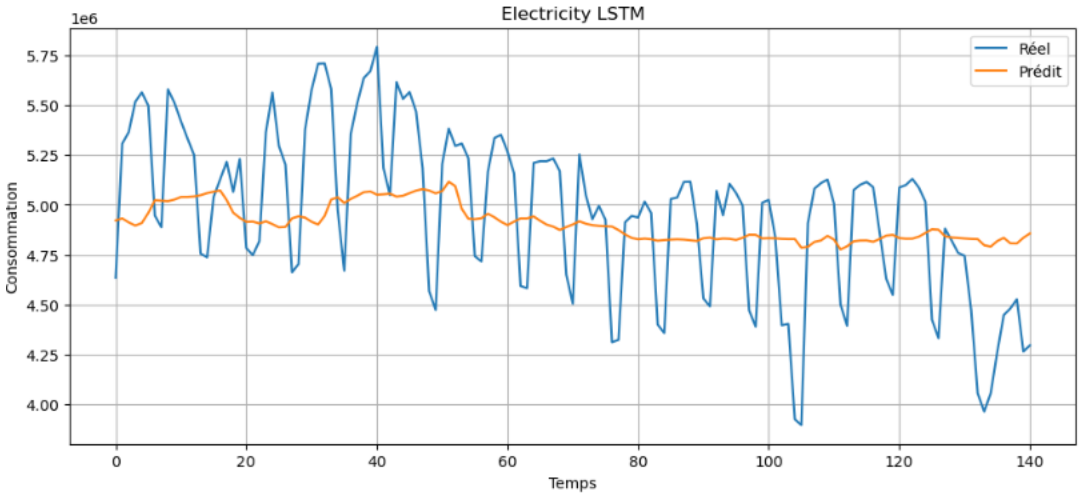



*Figure 17: Actual vs Predicted Electricity Consumption – LSTM*

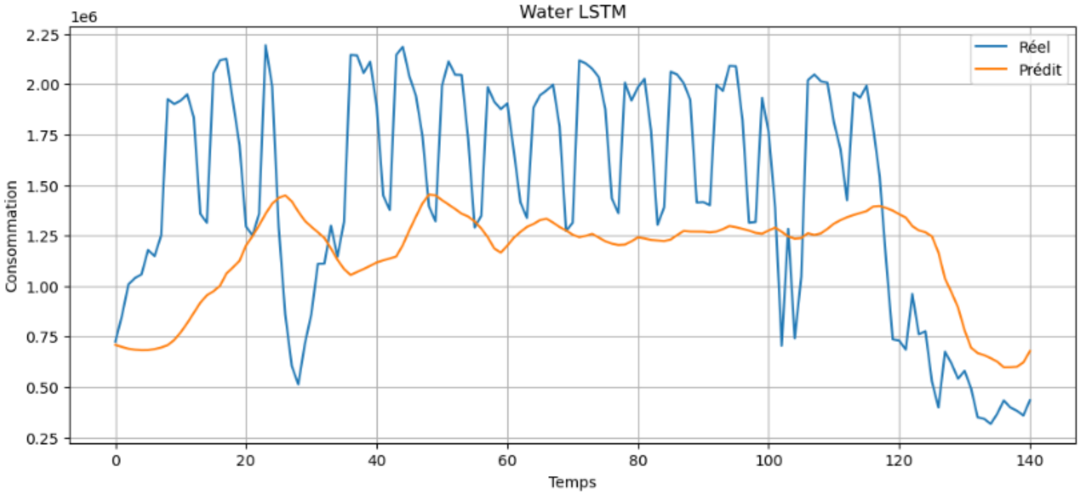



*Figure 18: Actual vs Predicted water Consumption – LSTM*

### 4.1.2 GRU (Gated Reccurent Unit)

The following table 8 presents the architecture of the GRU model used in our experiments, along with its performance results for predicting electricity, gas, and water consumption. Similar to the LSTM model, the GRU network is composed of several recurrent layers followed by dense layers. Batch Normalization and Dropout are applied after each GRU layer to reduce over fitting and enhance model stability. The model's performance is evaluated using three metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE), for each type of energy consumption.

*Table 7: GRU Model Architecture and Performance Results by Energy Type*

| **GRU Architecture** | **Electricity results** | **Gas results** | **Water results** |
|---|---|---|---|
| • 3 GRU layers (128 → 64 → 32)<br>• BatchNormalization + Dropout (0.3) after each GRU<br>• 2 Dense layers (64 → 32)<br>• Final Dense layer with 1 output neuron | **MAE :** 326753.95<br><br>**RMSE :** 382337.35<br><br>**MAPE**: 6.69% | **MAE :** 5474832.27<br><br>**RMSE :** 594111.77<br><br>**MAPE:** 19.63% | **MAE :** 395926.26<br><br>**RMSE :** 453872.15<br><br>**MAPE:** 39.04% |

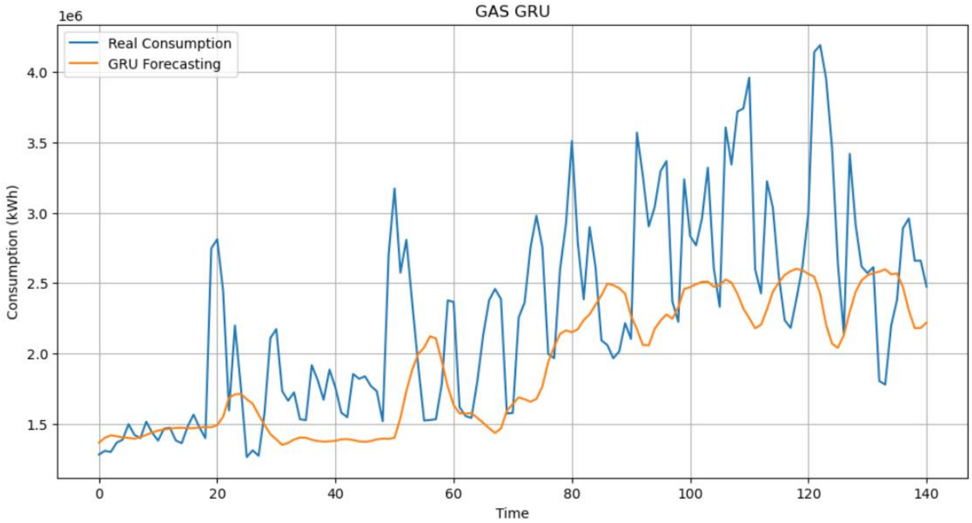



*Figure 19: Actual vs Predicted Gas Consumption – GRU*

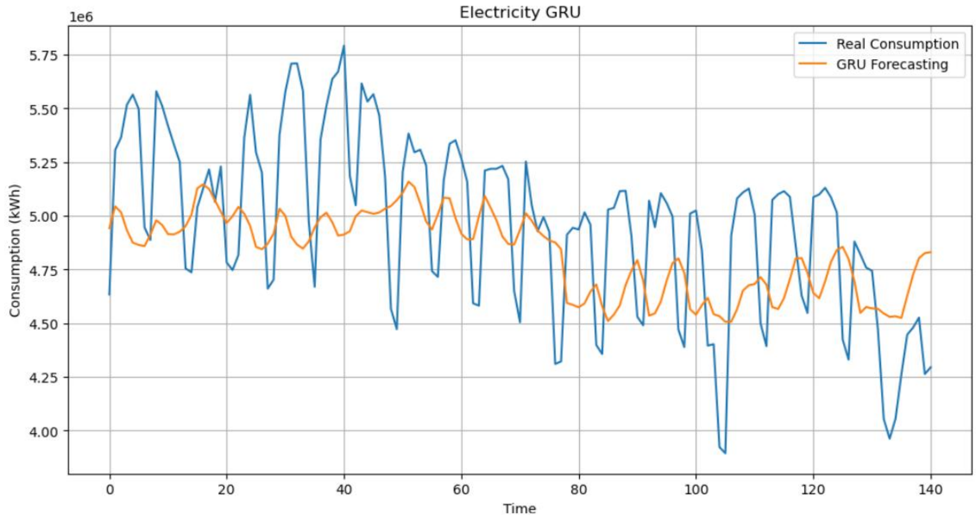



*Figure 20: Actual vs Predicted Electricity Consumption – GRU*

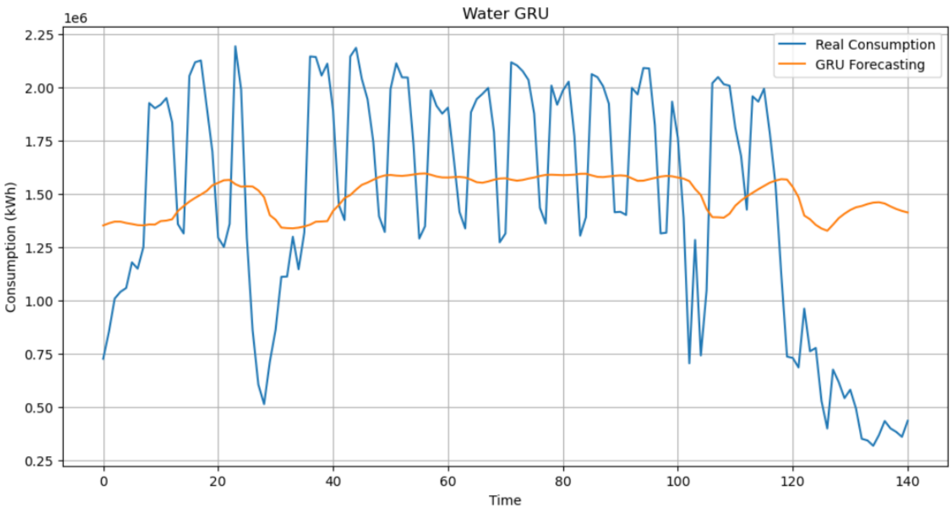



*Figure 21: Actual vs Predicted Water Consumption – GRU*

## 4.2 Experiment 02: Classification using ensemble learning

We begin by preparing the raw data through the merging of the metadata and meter datasets (electricity, gas and water) to form a complete dataset. This data passes through several preprocessing steps, including fuzzification, which transforms numerical values into fuzzy membership values using improved fuzzy logic functions. Additional preprocessing includes label encoding, handling missing values, and feature scaling. After this step, the dataset is divided into a training set (80%) and testing set (20%). Three different models are trained on the data: Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Multi-Layer Perceptron (MLP). Each model consists of multiple layers with Batch Normalization and Dropout applied to prevent over fitting. The predictions from the three models are them combined using an ensemble learning technique based on stacking. Finally, the performance of the final prediction is evaluated using standard metrics such as accuracy, precision, recall, and F1-score. Our overall approach is illustrated in figure 22.

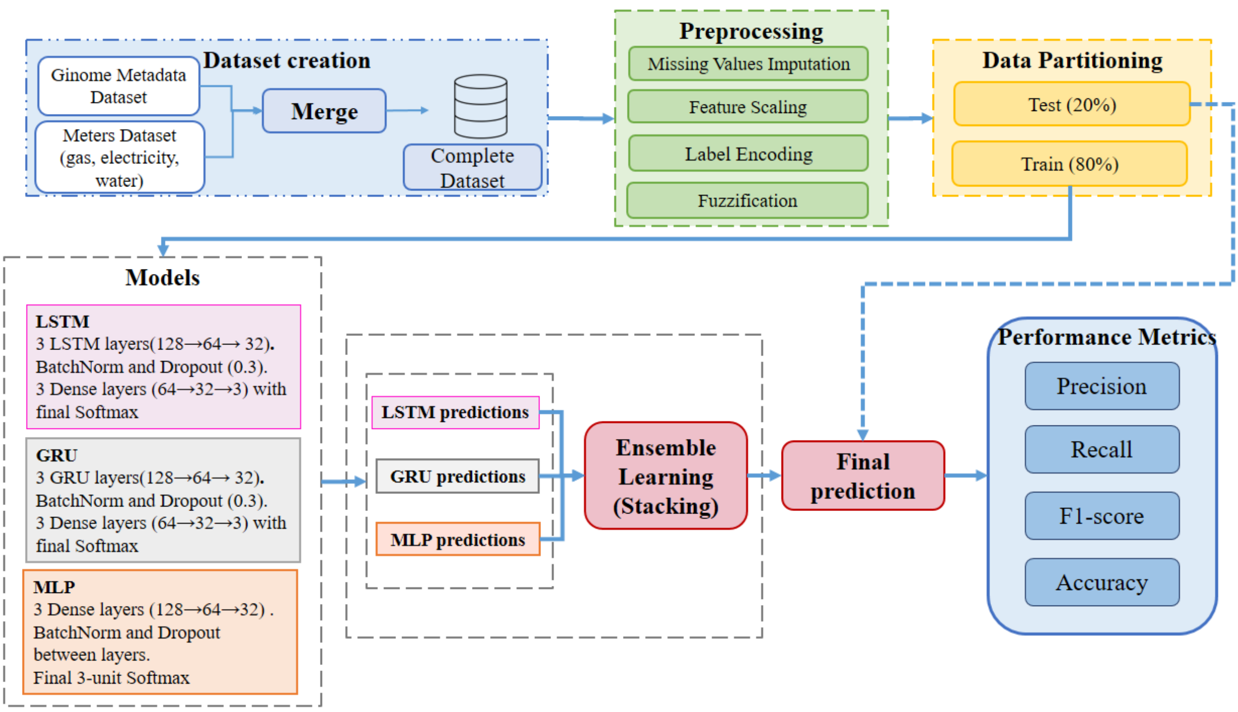



*Figure 22: Proposed model on Genome classification*

#### 4.2.1 Step 01: Dataset Creation

The first step consists of assembling the dataset by merging two main sources: the metadata dataset, which contains descriptive information about the buildings (such as site ID, primary use, etc.), and the meters dataset, which includes time-series data from different meters (electricity, gas, and water). The merging of these two datasets results a complete dataset that combines contextual and consumption data for each building.

- **Merging the meters into one dataset**

**Syntax:**

```
path_raw = "C:/Users/DELL/Documents/MEMOIRE/code/dataset/meters/N_cleaned/"
files = glob(path_raw + "*.csv")
```

```
['C:/Users/DELL/Documents/MEMOIRE/code/dataset/meters/N_cleaned\\electricity.csv',
 'C:/Users/DELL/Documents/MEMOIRE/code/dataset/meters/N_cleaned\\gas.csv',
 'C:/Users/DELL/Documents/MEMOIRE/code/dataset/meters/N_cleaned\\water.csv']
```

```
complete_data = pd.concat(dfs, axis=0, ignore_index=True)
complete_data = pd.DataFrame(columns=["timestamp", "building_id", "meter_reading", "meter"])
```

```
complete_data
```

| | index | building_id | meter | date | meter_reading |
|---|---|---|---|---|---|
| **0** | 0 | Bear_assembly_Angel | electricity | 2016-01-01 | 12808.16 |
| **1** | 1 | Bear_assembly_Angel | electricity | 2016-01-02 | 9251.00 |
| **2** | 2 | Bear_assembly_Angel | electricity | 2016-01-03 | 14071.65 |
| **3** | 3 | Bear_assembly_Angel | electricity | 2016-01-04 | 12860.38 |
| **4** | 4 | Bear_assembly_Angel | electricity | 2016-01-05 | 12212.86 |
| **...** | ... | ... | ... | ... | ... |
| **1389626** | 1389626 | Wolf_science_Alfreda | electricity | 2017-12-27 | 1781.39 |
| **1389627** | 1389627 | Wolf_science_Alfreda | electricity | 2017-12-28 | 1778.28 |
| **1389628** | 1389628 | Wolf_science_Alfreda | electricity | 2017-12-29 | 1769.79 |
| **1389629** | 1389629 | Wolf_science_Alfreda | electricity | 2017-12-30 | 1755.74 |
| **1389630** | 1389630 | Wolf_science_Alfreda | electricity | 2017-12-31 | 1736.93 |

1389631 rows × 5 columns

- **Cleaning the metadata dataset**

The metadata dataset was cleaned by deleting unnecessary columns that are not useful for the analysis. Other steps like removing duplicates and filling the missing values. These steps help to keep only the important information.

**Syntax:**

```
path = "C:/Users/DELL/Documents/MEMOIRE/code/dataset/metadata/"
```

```
metadata = pd.read_csv(path + "metadata.csv")
```

```
# Liste des colonnes à supprimer
colonnes_a_supprimer = ["building_id_kaggle", "site_id_kaggle", "sqft", "sub_primaryspaceusage", "leed_level", "solar",
                        "irrigation","hotwater", "rating",  "chilledwater", "steam","site_eui","source_eui"]

# Supprimer ces colonnes
metadata_cleaned = metadata .drop(columns=colonnes_a_supprimer)

# Sauvegarder le fichier nettoyé
metadata_cleaned.to_csv( "C:/Users/DELL/Documents/MEMOIRE/code/dataset/metadata/metadata_cleaned.csv", index=False)
```

- **Creating the complete dataset**

**Syntax:**

```
meters = pd.read_csv(path_proc + "allmeters_daily.csv")
meters
```

```
path1 = "C:/Users/DELL/Documents/MEMOIRE/code/dataset/metadata/"
metadata_cleaned = pd.read_csv(path1+"metadata_cleaned.csv")
```

```
dev = meters.merge(metadata_cleaned, on="building_id", how = "left")
```

### 4.2.2 Step 02: Data Preprocessing

Data preprocessing is the step where data is cleaned and transformed into a suitable format for machine learning. It includes handling missing values, converting data types, and scaling features to improve model accuracy and performance.

- **Handling Missing Values**

To complete the dataset the missing values are handled using many methods.

**Syntax:**

```python
df_apply[['electricity', 'water', 'gas']] = df_apply[['electricity', 'water', 'gas']].fillna("No")
df_apply.loc[:, ['industry', 'subindustry', 'heatingtype']] = df_apply[['industry',
                                        'subindustry', 'heatingtype', 'occupants']].fillna("Unknown")
df_apply.loc[:, ['yearbuilt', 'date_opened', 'numberoffloors', 'occupants','eui']] = df_apply[['yearbuilt',
                                        'date_opened', 'numberoffloors', 'eui']].fillna(method='ffill')
```

- **Fuzzification**

Fuzzification is used to transform continuous data (electricity, gas, and water) into linguistic categories such as **low**, **medium** and **high**. For each type of consumption, fuzzy membership functions are defined based on value ranges in the dataset. These functions assign a degree of membership to each category, allowing for better handling of uncertainty and variability in the data. The value is then classified into the category with the highest membership degree, helping improve model learning and interpretation.

**Syntax:**

```python
# Improved Fuzzy Membership Functions
def low_membership(x, a, b):
    return max(0, min((b - x) / (b - a), 1)) if a <= x <= b else 0

def medium_membership(x, a, b, c):
    if a <= x < b:
        return (x - a) / (b - a)
    elif b <= x <= c:
        return (c - x) / (c - b)
    return 0

def high_membership(x, a, b):
    return max(0, min((x - a) / (b - a), 1)) if x >= a else 0
```

In this case, these ranges are for electricity only.

```python
# Define fuzzy ranges based on dataset distribution
low_range = (100, 3500)
medium_range = (3400, 4000, 9000)
high_range = (8000, 30000)
```

- **Label Encoding**

Label encoding is used to convert categorical features into numerical values so that they can be used by machine learning algorithms. We used **LabelEncoder** from **Scikit-learn** to change the labels into numeric values. This makes it easier for the machine learning models to understand and work with the data.

**Syntax:**

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df_apply11.building_id = le.fit_transform(df_apply11.building_id)
df_apply11.site_id = le.fit_transform(df_apply11.site_id)
df_apply11.primaryspaceusage = le.fit_transform(df_apply11.primaryspaceusage)

df_apply11.timezone = le.fit_transform(df_apply11.timezone)
df_apply11.electricity = le.fit_transform(df_apply11.electricity)
df_apply11.meter = le.fit_transform(df_apply11.meter)

df_apply11.water = le.fit_transform(df_apply11.water)

df_apply11.gas = le.fit_transform(df_apply11.gas)
df_apply11.industry = le.fit_transform(df_apply11.industry)
df_apply11.subindustry = le.fit_transform(df_apply11.subindustry)
df_apply11.heatingtype = le.fit_transform(df_apply11.heatingtype)

df_apply11.date_opened = le.fit_transform(df_apply11.date_opened)
```

- **Feature Scaling**

Feature scaling is a crucial step in data preprocessing that aims to make variables comparable by bringing them onto same scale, which facilitates analysis and model training. In our work, we relied **on StandardScaler normalization a**s a key part of the preprocessing pipeline. This method transforms the data so that it has a mean of 0 and a standard deviation of 1.

The standardized score for a given sample x is calculated using the formula:

z = (x-u)/s. where u is the mean and s is the standard deviation of the feature.

### 4.2.3 Step 03: Data splitting

Data splitting is commonly used in machine learning to evaluate and validate the models. In this step, we divide the dataset into two parts: 80% of the data for training the model, while the remaining 20% is used to test its performance. We used the **train_test_split** function from the scikit-learn library.

### 4.2.4 Step 04: Model Training

Model training is the step where machine learning models learn from the training data. The goal is to help the models understand patterns and relationships between the features and the target values. In our case, we trained the models LSTM, GRU, and MLP to predict energy consumption levels.

*Table 8: Trained models*

| Model | Architecture | Regularization | Output Layer |
|---|---|---|---|
| **LSTM** | 3 LSTM layers (128 → 64 → 32) | Batch Normalization & Dropout (0.3) after each LSTM layer | 3 Dense layers (64 → 32 → 3) with Softmax |
| **GRU** | 3 GRU layers (128 → 64 → 32) | Batch Normalization & Dropout (0.3) after each GRU layer | 3 Dense layers (64 → 32 → 3) with Softmax |
| **MLP** | 3 Dense layers (128 → 64 → 32) with ReLU | Batch Normalization & Dropout between layers | Final Dense layer with 3 units & Softmax |

#### 4.2.5 Step 05: Ensemble Learning (Stacking)

The predictions from the LSTM, GRU, and MLP models are combined using **stacking**, an ensemble learning technique. In this approach, the outputs of the base models are used as inputs to a **meta-model**, which in our case is **XGBoost**. The XGBoost model learns how to best combine the base predictions to improve the overall accuracy and robustness of the final output.

```
X_meta = np.column_stack((pred_LSTM, pred_GRU, pred_MLP))
```

```
Classification Report:
              precision    recall  f1-score   support

           0       0.72      0.78      0.74      6094
           1       0.94      0.96      0.95     41319
           2       0.70      0.60      0.64      8173

    accuracy                           0.89     55586
   macro avg       0.79      0.78      0.78     55586
weighted avg       0.88      0.89      0.88     55586

Accuracy: 0.8851
Log Loss: 0.2674
```

*Table 9: Performance Metrics of Meta Model (XGBoost)*

| Algorithm | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| Meta model (XGboost) | 89% | 79% | 78% | 78% |

The ROC curve analysis demonstrates exceptional performance of the Meta Model (likely an ensemble) across three classes, with outstanding AUC scores of 0.9714 (Class 0), 0.9813 (Class 1), and 0.9413 (Class 2). All values exceed 0.94, indicating near-perfect discriminative ability, with Class 1 showing particularly remarkable separation (AUC ≈ 0.98). While Class 2's performance is slightly lower, it remains excellent (AUC > 0.94), suggesting the model maintains strong predictive power across all categories. These results position the Meta Model as highly reliable for classification tasks, with minimal confusion between classes. The consistently high AUC values across all three classes underscore the model's robustness and suitability for precision-demanding applications.

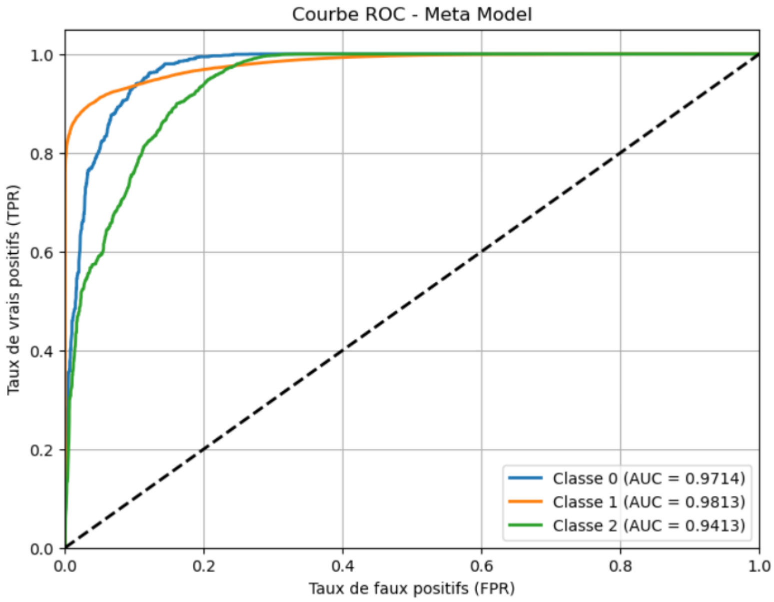



*Figure 23: ROC curve for META model*

The Meta Model stands out due to several key advantages. Its ensemble approach, likely combining multiple models through techniques like voting or stacking, significantly reduces errors and enhances overall accuracy. Unlike the MLP model, which struggles with minority classes, or the GRU model, which shows slight performance drops for Class 2, the Meta Model demonstrates remarkable robustness across all classes. This consistent high performance makes it particularly suitable for high-stakes applications where misclassification could have serious consequences, ensuring reliable and precise results even in demanding scenarios.

### 4.2.6 Results and discussions

- **LSTM**

This figure shows the training and validation accuracy over 20 epochs. Both accuracies steadily increase, with validation accuracy slightly higher than training accuracy throughout, indicating good generalization. The model reaches over 89% validation accuracy without signs of overfitting, suggesting effective learning and regularization.

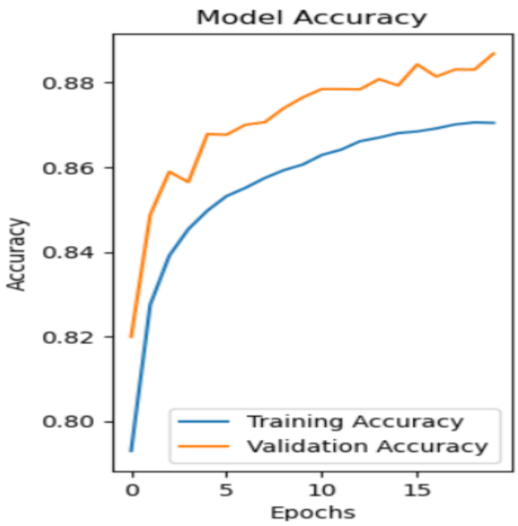



*Figure 24: Accuracy of the LSTM model*

This figure represents a **confusion matrix** for a classification model predicting three classes: **Low**, **Medium**, and **High**.

- The model performs **best on the "Medium" class**, which has the highest number of correct predictions.
- **High class has notable confusion with Medium**, with 12,448 instances misclassified.
- **Overall**, the model demonstrates strong performance but struggles a bit with distinguishing between **High** and **Medium** categories.

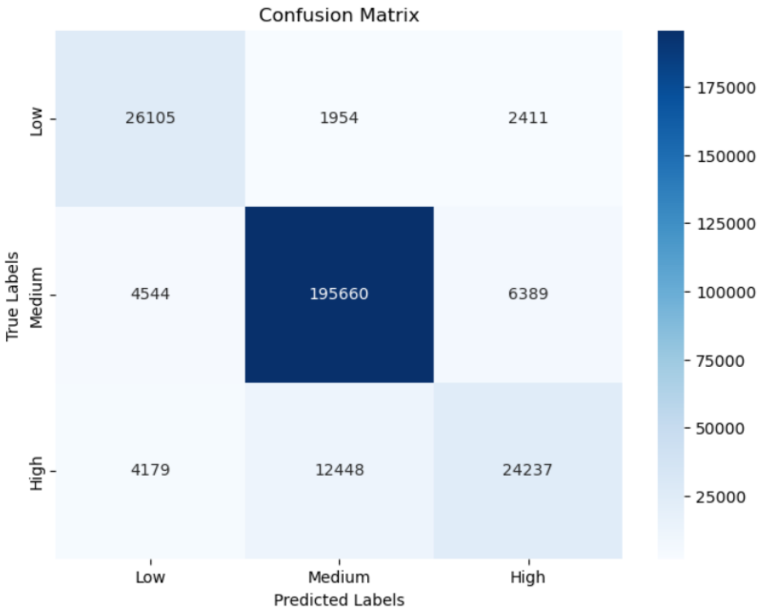



*Figure 25: Confusion matrix for the LSTM model*

*Table 10: Performance of evaluation metrics for the improved model (LSTM)*

| **Algorithm** | **Accuracy** | **Precision** | **Recall** | **F1-Score** |
|---|---|---|---|---|
| LSTM | 89% | 80% | 80% | 80% |

This figure displays a **ROC (Receiver Operating Characteristic) curve** for an LSTM model, used to evaluate its classification performance across three classes: **Classe 0**, **Classe 1**, and **Classe 2**.

- The model shows **strong discriminatory power** for all three classes, especially for **Classe 0**, with the curve very close to the top-left corner.
- Even the lowest-performing class (Classe 2) still achieves a high AUC, indicating the model reliably distinguishes it from others.
- Overall, the LSTM model demonstrates **high accuracy and strong classification capability** across all classes.

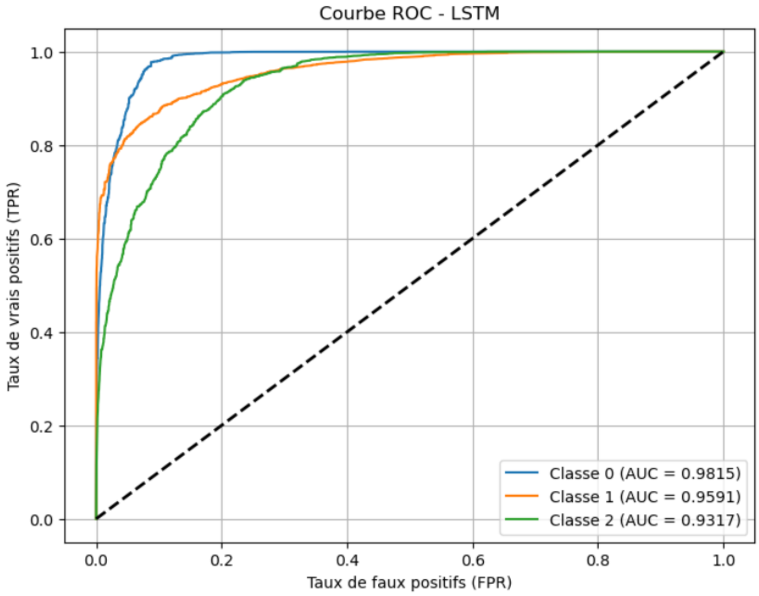



*Figure 26: ROC curve for LSTM model*

- **GRU**

The figure titled "Model Accuracy" illustrates the training and validation accuracy of a machine learning model over 20 epochs. The x-axis shows epochs (0 to 15) and the y-axis shows accuracy (0.80 to 0.87). The training accuracy (likely in blue) starts at 0.81 and increases steadily to 0.87, while the validation accuracy (likely in orange) starts at about 0.805, increases rapidly at first, and reaches a plateau around 0.845 to 0.85. Both curves level off, indicating effective training without significant overfitting, although the slight gap between them suggests slight overfitting. The model achieves a training accuracy of about 87% and a validation accuracy of about 85%, demonstrating good generalization. To address possible overfitting, techniques such as regularization or early stopping could be applied, and the plateau implies that subsequent epochs may not yield significant improvements.

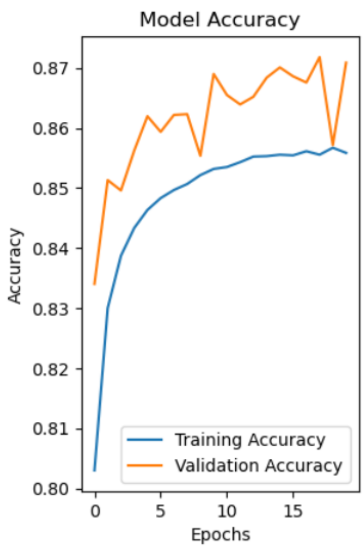



*Figure 27: Accuracy of the GRU model*

The image shows a **confusion matrix** for a model classifying data into three categories: **Low**, **Medium**, and **High**. The matrix reveals a strong performance for the "Medium" class (197,726 correct predictions) but lower accuracy for "Low" (25,095 correct) and "High" (18,489 correct). The model struggles most with distinguishing "High" from "Medium," possibly due to class imbalance or overlapping features. The overall performance suggests the need for further refinement to enhance accuracy, particularly for minority classes.

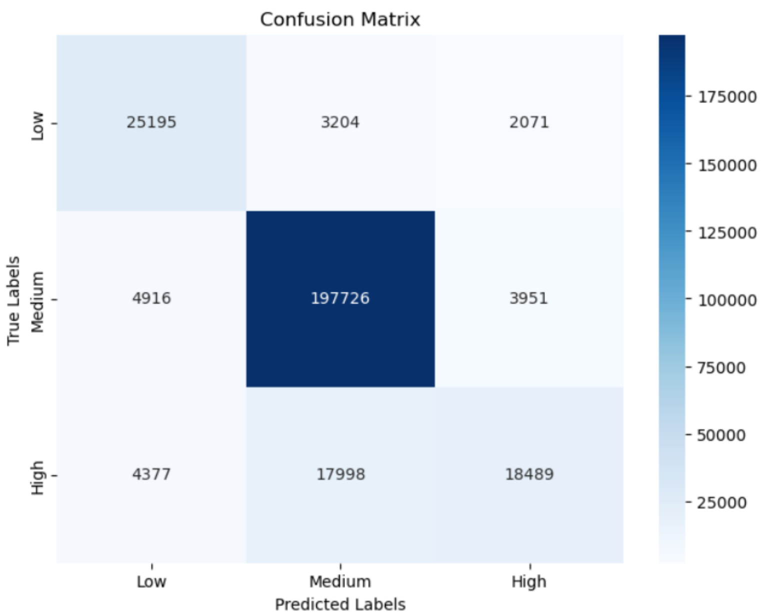



*Figure 28: Confusion matrix for the GRU model*

*Table 11: Performance of evaluation metrics for the improved model (GRU).*

| Algorithm | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| GRU | 87% | 80% | 75% | 76% |

The image displays a ROC curve analysis for a GRU model's performance across three classes (0, 1, and 2), plotting the True Positive Rate (TPR) against the False Positive Rate (FPR). The model demonstrates excellent performance for Class 0 (AUC = 0.9776), near-perfect classification, followed by strong results for Class 1 (AUC = 0.9547). Class 2 shows good but comparatively weaker discrimination (AUC = 0.9194), suggesting potential challenges such as class overlap or data imbalance. The steep curves indicate the model effectively maintains high true positive rates while keeping false positives low across all thresholds. While the model performs robustly overall, Class 2 may benefit from targeted improvements like data augmentation or reweighting to enhance its classification accuracy.

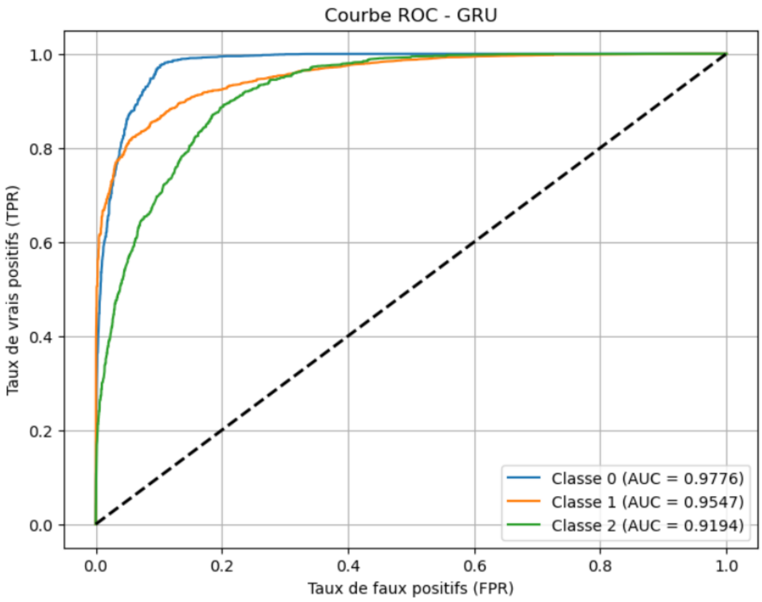



*Figure 29: ROC curve for GRU model*

- **MLP**

The graph illustrates the training and validation accuracy of a machine learning model across. Both accuracy curves show an increasing trend, indicating effective learning. The training accuracy starts around 0.81 and rises to approximately 0.85, while the validation accuracy begins slightly lower and reaches about 0.84 by the final epoch. The close alignment between training and validation accuracy suggests the model is generalizing well without significant overfitting. The steady improvement across epochs implies the model benefits from additional training iterations, though the convergence pattern suggests diminishing returns beyond 8 epochs. This performance indicates a well-balanced model with good learning progression and generalization capability.

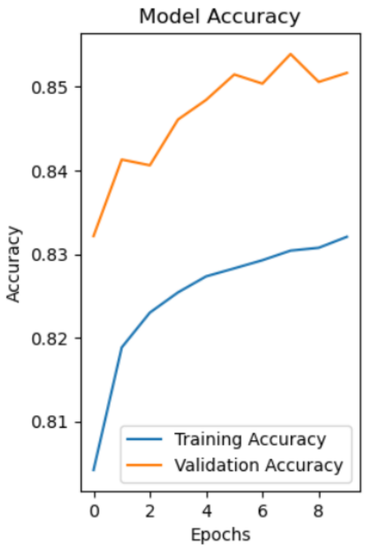



*Figure 30: Accuracy of the MLP model.*

The confusion matrix reveals significant class imbalance, with the model performing well for the majority "Medium" class (184,454 correct predictions) but struggling with minority "Low" and "High" classes. The "Low" class is frequently misclassified as "Medium" (17,046 errors), while the "High" class shows substantial confusion with "Low" (30,675 errors). The presence of implausible values (e.g., 150,000) suggests data formatting issues. To improve performance, addressing class imbalance through resampling techniques and enhancing feature discrimination for minority classes is recommended. The model requires refinement to better handle imbalanced data and improve accuracy for all classes.

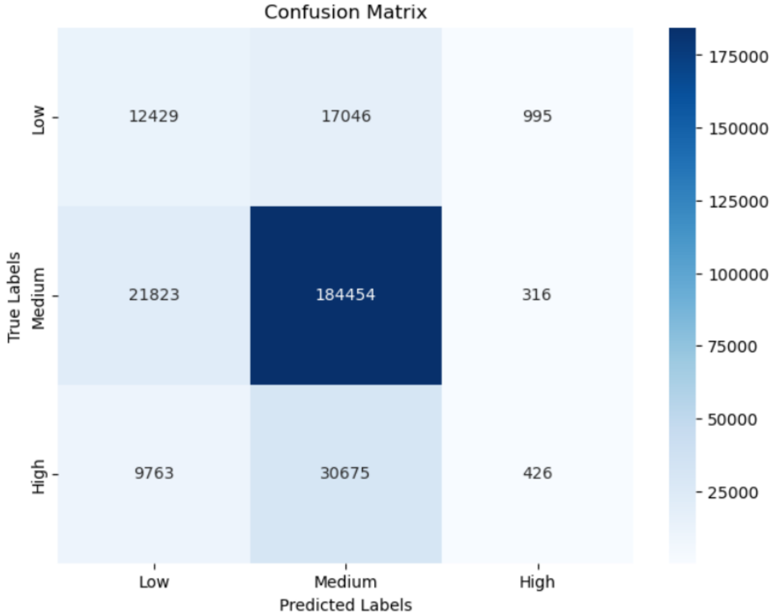



*Figure 31: Confusion matrix for the MLP model*

The ROC curve analysis for the MLP model shows moderate performance across three classes, with Class 0 achieving the highest AUC score of 0.7322, followed by Class 1 (AUC = 0.6748) and Class 2 (AUC = 0.6466). These results indicate that the model has limited discriminative power, with all classes performing below the 0.8 threshold typically considered good. The performance degrades progressively from Class 0 to Class 2, suggesting increasing difficulty in distinguishing these classes. The similar AUC values for both True Positive Rate (TPR) and False Positive Rate (FPR) across classes indicate consistent but suboptimal performance. To improve results, feature engineering, model architecture adjustments, or addressing potential class imbalance should be considered. The current performance level suggests the model may not be sufficiently reliable for critical applications without further optimization.

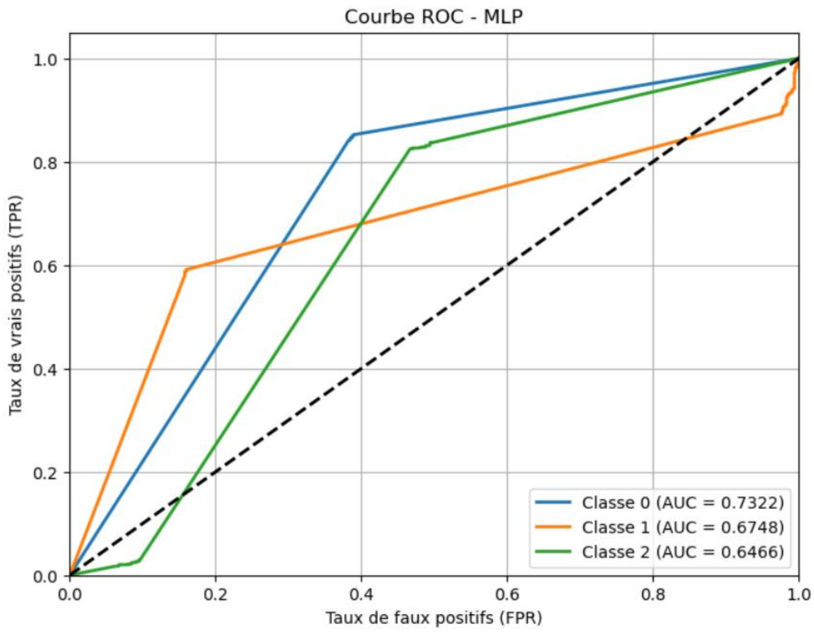



*Figure 32: ROC curve for MLP model*

*Table 12: Performance of evaluation metrics for the improved model (MLP).*

| Algorithm | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| MLP | 71% | 44% | 44% | 40% |

The trained meta-model is used to forecast energy consumption levels for electricity, gas and water.

*Table 13: Performance Accuracy of Different Meters*

| Meter | Electricity | Gas | Water |
|---|---|---|---|
| **Accuracy** | 82,66% | 99,87% | 99,12% |

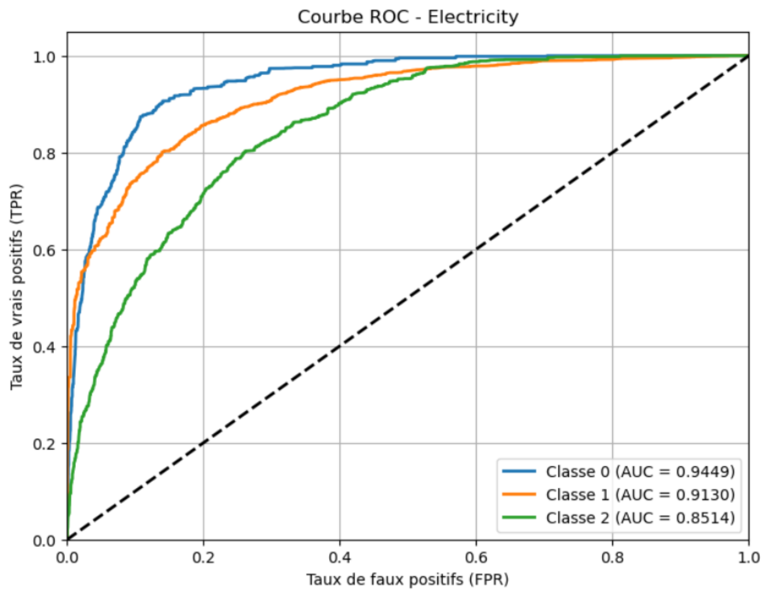



*Figure 33: ROC curve for Electricity*

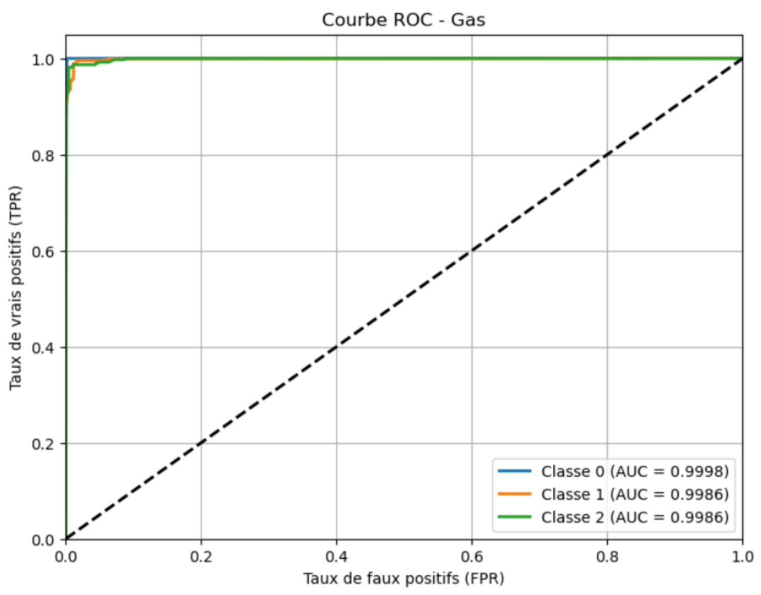



*Figure 34: ROC curve for Gas*

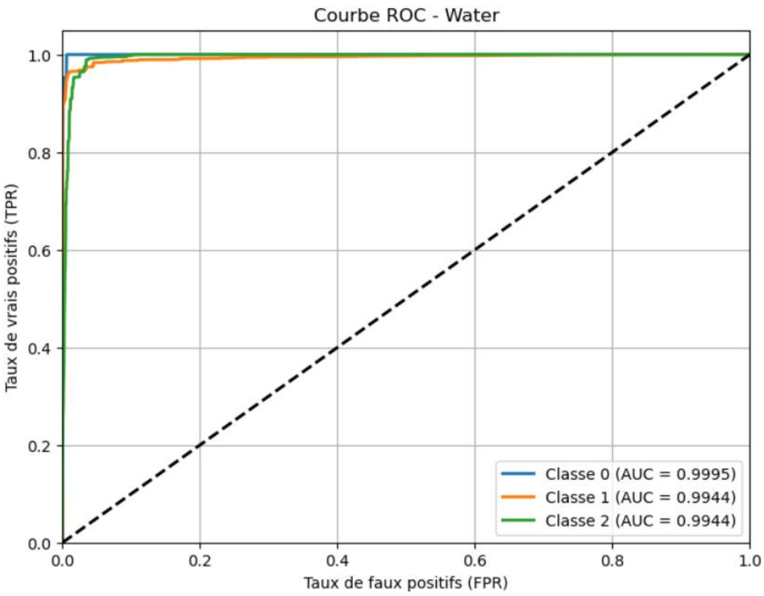



*Figure 35: ROC curve for Water*

### 4.3 Comparison of Results

**Experiment 01:**

Both LSTM and GRU models showed good performance in predicting energy consumption. LSTM performed slightly better overall, while GRU gave competitive results, especially for water consumption.

**Experiment 02:**

The ensemble learning method achieved excellent results. It was especially effective for classifying gas and water consumption, showing high accuracy and reliability across all meter types.

## 5. Conclusion

In this chapter, we presented an approach for energy forecasting based on two experiments: time series regression and classification using ensemble learning. We used deep learning models like LSTM and GRU to predicts energy consumption and identify patterns. The results proved that the approach is both effective and practical.

# General Conclusion

# General Conclusion

In this thesis, we proposed an intelligent and robust approach for energy consumption forecasting within the context of smart cities, leveraging deep learning and ensemble learning methodologies. Recognizing the pressing challenges of sustainability, urban growth, and energy efficiency, this work contributes to the ongoing efforts in optimizing resource management through advanced technological solutions.

Our approach utilized the Building Data Genome Project 2 dataset, incorporating deep learning models such as LSTM and GRU, enhanced with **fuzzification** and **ensemble learning** techniques. The integration of fuzzification allowed the system to handle uncertain and heterogeneous data more effectively, while ensemble learning improved classification accuracy and model generalization. The experimental results demonstrated high prediction accuracies across different consumption types — electricity, gas, and water — validating the strength and applicability of the proposed approach.

This work not only confirms the potential of AI in transforming urban energy management but also highlights the critical role of combining multiple advanced techniques to enhance predictive performance and system resilience. Our contribution align with global sustainability goals, supporting the design of smarter, more energy-efficient infrastructures in modern cities.

Future directions could include the extension of this model to other domains of urban resource management, real-time system implementation, and the use of transformer-based architectures for further performance gains. Moreover, improving the interpretability of AI models will be essential to foster trust and understanding among stakeholders and end users.

In our research, we focus on writing and submitting a scientific paper that reflects our efforts in the field of energy management in smart cities. We also had the opportunity to participate in study days on artificial intelligence in Algeria, where we shared our work, learned from others, and strengthened our knowledge in the field. These events enriched our research experience and helped us connect with the scientific community.

In conclusion, this research marks a significant step toward the development of scalable, accurate, and intelligent energy management systems, paving the way for more sustainable and responsive smart cities.

# REFERENCES


1] I. ZIAYA , "Une approche de recommandation pour le tourisme dans un environnemnt smart city," 2022.

2] G. Alice and J. Alexandra , "IBM," 22 Novembre 2023. [Online].

3] A. P. Doru , C. O. Bogdan and D. Z. Ion , Retrofitting for Optimal Energy Performance, Romania, 2019.

4] G. Rudolf, K. Robert, F. Christian, M. P. Nataša, K. Hans and M. Evert, Smart cities - Ranking of European medium-sized cities, Vienna, 2007.

5] S. Amira, B. M. Ramdan and H. G. Aditya, "Jakarta Smart City," 10 march 2022. [Online]. Available: https://smartcity.jakarta.go.id/en/blog/membedah-smart-economy-dan-manfaatnya-buat-jakarta/. [Accessed 15 november 2024].

6] "nomadia," 28 august 2023. [Online]. Available: https://www.nomadia-group.com/en/resources/blog/advantages-and-challenges-of-smart-mobility/. [Accessed 15 november 2024].

7] Pinheiro, pedro, S. Ricardo and B. Ricardo, "Pinheiro, Pedro, Ricardo Santos, and Ricardo Barbosa. "Industry 4.0 multi-agent system based knowledge representation through blockchain.," Springer International Publishing, 2019.

8] M. Catriona , . C. Gavin, . C. Jonathan, M. Jeremy, J. k. PEDERSON, R. . K. THAARUP and A. LIEBE, "Mapping Smart Cities in the EU," © European Union, 2014.

9] P. Laurent, M. Erica, F. Laurent and C. Daniela, "Business Innovation Observatory," European Union, Luxembourg, 2014.

10] K. Majda , . C. Ivan, K. . D. Nives and S. Katarina, "On the move: indicators of urban mobility – smart city concept “City of Varaždin”," Koprivnica, 2023.

11] "SMART CITY," [Online]. Available: https://www.smartcity.co.nz/blog/challenges-of-smart-cities/. [Accessed 30 Octobre 2024].

12] "explorama," 10 Avril 2019. [Online]. Available: https://explorama.app/exemples-de-smart-cities/.

13] S. Morteza and P. M. Mohsen, "International Power System Conference (PSC)," Iran, 2013.

14] "Connaissance Des energies," 17 juin 2024. [Online]. Available: https://www.connaissancedesenergies.org/fiche-pedagogique/reseau-intelligent-smart-grid. [Accessed 31 10 2024].

15] C. Qi, W. Wei, W. Fangyu, D. Saparna, I. R. W. Member, I. B. Z. Member and H. Xin, "A Survey on an Emerging Area: Deep Learning for Smart City Data," IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE, 2019.

16] B. Jacob, "builtin," 21 february 2023. [Online]. Available: https://builtin.com/articles/iot-sensors. [Accessed 31 october 2024].

17] "Building Intelligent Networks: Feed in Tariffs and Smart Grids," 2024.

18] D. Linh Chu, "The Advantages and Disadvantages of IoT in Business," 2023.

19] "Oracle," [Online]. Available: https://www.oracle.com/internet-of-things/. [Accessed 31 October 2024].

20] S. Jessica, "coursera," 2 octobre 2024. [Online]. Available: https://www.coursera.org/articles/internet-of-things. [Accessed 31 octobre 2024].

21] "SAP," [Online]. Available: https://www.sap.com/suisse/products/artificial-intelligence/what-is-iot.html. [Accessed 31 octobre 2024].

22] K. Masroor, *IoT Sensors for Energy Management: A Diverse Variety*, 2024.

23] "fogwing," [Online]. Available: https://www.fogwing.io/types-of-iot-networks/. [Accessed 01 novembre 2024].

24] "orange," [Online]. Available: https://iotjourney.orange.com/en/explore/solutions-iot/iot-networks. [Accessed 02 november 2024].

25] "types," 10 february 2024. [Online]. Available: https://www.types.blog/types-of-data-in-iot/. [Accessed 02 november 2024].

26] "gridlabd," february 2023. [Online]. Available: https://www.gridlabd.org/. [Accessed 09 november 2024].

27] C. David P, "GridLAB-D Smart Grid Simulation," Washington.

28] "mathworks," [Online]. Available: https://www.mathworks.com/solutions/electrification/building-energy-management.html. [Accessed 11 november 2024].

29] "EPRI," [Online]. Available: https://www.epri.com/pages/sa/opendss#:~:text=What%20is%20OpenDSS%3F,grid%20integration%20and%20grid%20modernization.. [Accessed 09 november 2024].

30] "Pyomo," [Online]. Available: https://www.pyomo.org/about#:~:text=Pyomo%20is%20a%20Python%2Dbased,is%20modeling%20structured%20optimization%20applications.. [Accessed 09 november 2024].

31] S. Cole and K. Eda , "IBM," 16 August 2024. [Online]. Available: https://www.ibm.com/topics/artificial-intelligence. [Accessed 02 November 2024].

32] "Artificial Intelligence and Energy Management," [Online]. Available: https://perspectives.se.com/visualize-data/artificial-intelligence-and-energy-management. [Accessed 02 November 2024].

33] C. Matt, "datacamp," 19 july 2023. [Online]. Available: https://www.datacamp.com/blog/what-is-machine-learning. [Accessed 02 november 2024].

34] R. S. Arvind, K. R. Seshu, B. Mohit, B. K. Chetan and Z. Ievgen, "Machine learning-based energy," Scientific Reports, 2024.

35] H. Jim, "IBM," 17 June 2024. [Online]. Available: https://www.ibm.com/think/topics/deep-learning. [Accessed 18 February 2025].

36] K. . S. Rahul, M. Wan-Tai , Au-Yeung, P. Jagmeet and H. Kevin, "State-of-the-Art Machine Learning Techniques Aiming to Improve Patient Outcomes Pertaining to the Cardiovascular System," *Journal of the American Heart Association,* no. 2, p. 15, 2022.

37] P. Junjie, C. J. Elizabeth, D. Pierre and C. Coziana, "Machine Learning Techniques for Personalised Medicine Approaches in Immune-Mediated Chronic Inflammatory Diseases: Applications and Challenges," *Frontiers in Pharmacology,* p. 18, 2021.

38] P. Shivam, "Linkedin," 09 March 2018. [Online]. Available: https://www.linkedin.com/pulse/machine-learning-classification-day-1-shivam-panchal/. [Accessed 27 February 2025].

39] "geeksforgeeks," 15 January 2025. [Online]. Available: https://www.geeksforgeeks.org/unsupervised-learning/. [Accessed 25 February 2025].

40] M. Rohit, "Bombay softwares," 10 September 2023. [Online]. Available: https://www.bombaysoftwares.com/blog/introduction-to-unsupervised-learning. [Accessed 10 February 2025].

41] S. H. Tan, K. I. Mohamed and F. P. Mohamed, "Researchgate," February 2021. [Online]. Available: https://www.researchgate.net/figure/Block-diagram-of-reinforcement-learning_fig1_356715295. [Accessed 27 february 2025].

42] "library.fiveable," 30 July 2024. [Online]. Available: https://library.fiveable.me/neural-networks-and-fuzzy-systems. [Accessed 07 February 2025].

43] M. Adnan , "Medium," 11 February 2025. [Online]. Available: https://medium.com/%40adnan.mazraeh1993/fuzzification-and-defuzzification-6fbfe059047e. [Accessed 11 April 2025].

44] . P. Shanthababu, "analyticsvidhya," 04 february 2025. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-to-time-series-analysis/#Data_Types_of_Time_Series. [Accessed 18 february 2025].

45] "geeksforgeeks," 13 August 2024. [Online]. Available: https://www.geeksforgeeks.org/time-series-analysis-and-forecasting/. [Accessed 01 February 2025].

46] Gourav, "Analytics Vidhya," 17 october 2024. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/09/introduction-to-artificial-neural-networks/.

47] G. Shubham, "knoldus," 28 December 2021. [Online]. Available: https://blog.knoldus.com/architecture-of-artificial-neural-network/.

48] N. H. Parsan, "Medium," 11 October 2024. [Online]. Available: https://medium.com/@prasanNH/artificial-neural-networks-anns-and-their-role-in-machine-learning-e19a214e46ec. [Accessed 28 January 2025].

49] "Amazon," [Online]. Available: https://aws.amazon.com/what-is/recurrent-neural-network/. [Accessed 03 november 2024].

50] "geeksforgeeks," 23 july 2024. [Online]. Available: https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/. [Accessed 03 november 2024].

51] "TEX," january 2022. [Online]. Available: https://tex.stackexchange.com/questions/494139/how-do-i-draw-a-simple-recurrent-neural-network-with-goodfellows-style. [Accessed 28 january 2025].

52] "Analytics Vidhya," 01 october 2024. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/#h-what-is-lstm.

53] "geeksforgeeks," 10 june 2024. [Online]. Available: https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/.

54] Pranjal, "Analytics Vidhya," 20 December 2024. [Online]. Available: https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/. [Accessed 28 january 2025].

55] Anishnama, "Medium," 4 may 2023. [Online]. Available: https://medium.com/@anishnama20/understanding-gated-recurrent-unit-gru-in-deep-learning-2e54923f3e2. [Accessed 23 November 2024].

56] J. Mohammed and M. Antonio, "ResearchGate," 17 March 2019. [Online]. Available: https://www.researchgate.net/publication/331848495_A_Deep_Learning-Based_Approach_for_Multi-Label_Emotion_Classification_in_Tweets. [Accessed 28 january 2025].

57] M. Jacob and K. Eda, "IBM," 18 March 2024. [Online]. Available: https://www.ibm.com/think/topics/ensemble-learning. [Accessed 02 February 2025].

58] K. Mbali, "Analytics Vidhya," 03 January 2025. [Online]. Available: https://www.analyticsvidhya.com/blog/2023/01/ensemble-learning-methods-bagging-boosting-and-stacking/. [Accessed 02 February 2025].

59] "geeksforgeeks," 16 january 2025. [Online]. Available: https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/. [Accessed 24 january 2025].

60] "geeksforgeeks," 31 March 2023. [Online]. Available: https://www.geeksforgeeks.org/ml-gradient-boosting/. [Accessed 01 February 2025].

[61] "geeksforgeeks," 06 December 2023. [Online]. Available: https://www.geeksforgeeks.org/ml-xgboost-extreme-gradient-boosting/. [Accessed 01 February 2025].

[62] O.-A. Razak and A. Hafiz, "Building energy consumption prediction using deep learning," Big Data Technologies and Innovation Laboratory, University of Hertfordshire, United Kingdom.

[63] S. Sehovac, C. Nesen and K. Grolinger, "Forecasting Building Energy Consumption with Deep Learning: A Sequence to sequence Approach," Western University, London, Canada, 2019.

[64] M. J. Roberto, M. Jesus, G. R. Juan, V. Amparo and B. Martin, "Deep learning for prediction of energy consumption: an applied use," 2024.

[65] «Kaggle,» October 2020. [En ligne]. Available: https://www.kaggle.com/datasets/claytonmiller/buildingdatagenomeproject2?resource=download.

[66] M. Clayton, K. Anjukan , P. Bianca, A. Pandarasamy and N. Zoltan, "The Building Data Genome Project2, energy meter data from the ASHRAE Great Energy Predictor III," Scientific data, 2020.

[67] C. Austin, "datacamp," 2024 December 2024. [Online]. Available: https://www.datacamp.com/blog/anaconda-vs-python-key-differences. [Accessed 25 February 2025].

[68] "DataScientist," 30 March 2021. [Online]. Available: https://datascientest.com/jupyter-notebook-tout-savoir. [Accessed 25 February 2025].

[69]

[70] Furizal, . F. Aldi Bastiatul, M. Hari , . M. Alfian, A. F. Asno and S. Iswanto, "Long Short-Term Memory vs Gated Recurrent Unit," Indonesia, 2024.

71] P. Ishaani , S. Sandipan, K. Raghvendra and T. David, "A machine-learning ensemble model for predicting energy consumption in smart homes," 2022.

72] S. V. Chang , K. S. Gan , K. O. Chin and A. Patricia, "ResearchGate," November 2013. [Online]. Available: https://www.researchgate.net/publication/261206036_A_review_of_stock_market_prediction_with_Artificial_neural_network_ANN. [Accessed 06 February 2025].