T.C.
ISTANBUL GEDİK UNIVERSITY
INSTITUTE OF GRADUATE STUDIES

SELECTING THE IDEAL APPROACH FOR RATIONALIZATION OF ELECTRICAL ENERGY CONSUMPTION IN SMART BUILDING MANAGEMENT USING MACHINE LEARNING

MASTER'S THESIS

Ahmed Ghanim Daoud ALDAOUD

Engineering Management Department

Master of Engineering Management in English

JUNE 2023

T.C.
ISTANBUL GEDİK UNIVERSITY
INSTITUTE OF GRADUATE STUDIES

SELECTING THE IDEAL APPROACH FOR RATIONALIZATION OF ELECTRICAL ENERGY CONSUMPTION IN SMART BUILDING MANAGEMENT USING MACHINE LEARNING

MASTER'S THESIS

Ahmed Ghanim Daoud ALDAOUD
(201281020)

Engineering Management Department

Master of Engineering Management in English

Thesis Advisor: Prof. Dr. Gözde ULUTAGAY

Second Thesis Advisor: Assist. Prof. Dr. Tuğbay Burçin GÜMÜŞ

JUNE 2023

T.C.
İSTANBUL GEDİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ MÜDÜRLÜĞÜ

Yüksek Lisans Tez Onay Belgesi

Enstitümüz, Engineering Management Department İngilizce Tezli Yüksek Lisans Programı (201281020) numaralı öğrencisi Ahmed Ghanim Daoud ALDAOUD'un "Selecting the Ideal Approach for Rationalization of Electrical Energy Consumption in Smart Building Management Using Machine Learning" adlı tez çalışması Enstitümüz Yönetim Kurulunun 21/06/2023 tarihinde oluşturulan jüri tarafından *Oy Birliği* ile Yüksek Lisans tezi olarak *Kabul* edilmiştir.

Öğretim Üyesi Adı Soyadı

Tez SavunmaTarihi: 21.06.2023

1) Tez Danışmanı: Prof. Dr. Gözde ULUTAGAY

2) Jüri Üyesi: Dr. Öğr. Üyesi Bestem ESİ

3) Jüri Üyesi: Dr. Öğr. Üyesi Fatma Günseli ÇIKLAÇANDIR

DECLARATION

I, Ahmed Ghanim Daoud ALDAOUD, hereby certify that this thesis entitled "Selecting the Ideal Approach for Rationalization of Electrical Energy Consumption in Smart Building Management Using Machine Learning" is my original thesis for the award of Master's Degree in Engineering Management at the Faculty of Engineering Management. I further certify that this thesis or any part thereof has not been submitted and presented for any other degree or research thesis at any other university or institution". (21/06/2023)

Ahmed Ghanim Daoud ALDAOUD

DEDICATION

I would really be presenting my appreciation to every one supported me all through study and searching period of this voyage.

I to present my thank to my advisors
Assist. Prof. Dr. Gözde ULUTAGAY & Assist. Prof. Dr. Tuğbay Burçin GÜMÜŞ
As well as to my university, "Istanbul Gedik University".

And to the dearest people my mother and my father
And My uncle Consulting Engineer Qutaiba M.Saeed AL-SABBAGH
(May God have mercy on them and make them among the people of Paradise).

To my dear wonderful wife and children, may God protect them and make them the happiest of people for their support and prayers throughout my research.
To my brothers who sacrificed and labored for me.

Ahmed Ghanim Daoud ALDAOUD

PREFACE

In the name of Allah, the Merciful Praise be to God, Lord of the Worlds, and blessings and peace be upon our Prophet Muhammad, the best of the messengers, and upon his family the good, pure, and his companions.

"It is my pleasure, as I finish my letter, to extend my sincere thanks and gratitude to who supervised my letter for directing me and for his valuable remarks that had a great impact in producing the message in this way, and may God grant them success to what he loves and pleases I also extend my thanks, appreciation, and gratitude to Prof. Dr. Gözde ULUTAGAY & Assist. Prof. Dr. Tuğbay Burçin GÜMÜŞ for what they presented".

"Atta was the reason for my reaching this stage, and I extend my thanks and gratitude to the professors on the committee the discussion of what they provided enriches and strengthens the position of the research".

"I also extend my thanks and gratitude to my colleagues at the Master's level for their valuable cooperation in completing my study trip".

JUNE 2023 Ahmed Ghanim Daoud ALDAOUD

TABLE OF CONTENT

Page

ABBREVATIONS

ADAM : Adaptive Moment Algorithm
AI : Artificial Intelligence
AR : Auto-Regressive
ARIMA : Auto Regressive Integrated Moving Average
BPTT : Back Propagation Through Time
BRNN : Bidirectional Recurrent Neural Networks
DL : Deep learning
DN : Deep Networks
IoT : Internet of Things
LSTM : Long Short-Term Memory
MA : Moving Average
MAE : Mean Absolute Error
MAPE : Mean Absolute Percentage Error
ME : Mean Error
Minmax : Error in Min-Max
ML : Machine Learning
MPE : Mean Percentage Error
MSE : Mean Squared Error
NADAM : Nesterov Accelerated Adaptive Moment
NN : Neural Networks
RMSE : Root Mean Square Error
RNN : Recurrent Neural Networks
RPA : Random Prediction Algorithm
SARIMAX : Seasonal Auto Regressive Integrated Moving Average with eXogenous components
WSNs : Wireless Sensor Networks

LIST OF TABLES

Page

LIST OF FIGURES

Pages

SELECTING THE IDEAL APPROACH FOR RATIONALIZATION OF ELECTRICAL ENERGY CONSUMPTION IN SMART BUILDING MANAGEMENT USING MACHINE LEARNING

ABSTRACT

The infrastructure development of smart cities has gained a lot of attention in recent years, where energy efficiency was the major problem that researchers aim to improve. Anomaly detection for energy consumption was one of these issues, and it was a critical element to take into account while operating effective energy-saving systems, lowering the overall energy use and carbon emissions. Consequently, suggesting a potent method based on the Internet of Things (IoT) can have more relevance for recognizing aberrant usage in buildings and offering this information to customers and governments so that it can be handled in a correct way to lower the payments. As a result, this thesis proposes an optimization method to detect anomalies by using the LSTM algorithm and investigates three optimization algorithms of ADAM, AadMax, and Nadam. Two statistical modeling's are used for time series forecasting of ARIMA and SARIMAX. Anomaly detection results indicate that using LSTM with Nadar gives the best results, where MSE and RMSE achieved 0.15348 and 0.02356 respectively. Also, using ARIMA model gives the best overall results with AIC and MSE values of 0.13859 and 300.94365 respectively. This confirms the reliability and adaptability of the proposed model in optimizing anomaly detection.

Keywords: *Time Series Forecasting, ARIMA, LSTM, SARIMAX, ADAM, NADAM.*

AKILLI BİNA YÖNETİMİNDE ELEKTRİK ENERJİSİ TÜKETİMİNİN RASYONALİZASYONU İÇİN İDEAL YAKLAŞIMIN SEÇİLMESİ, MAKİNE ÖĞRENİMİ KULLANIMI

ÖZET

Akıllı şehirlerin alt yapı gelişimi, araştırmacıların iyileştirmeyi amaçladıkları başlıca sorunun enerji verimliliği olduğu son yıllarda büyük ilgi görmüştür. Bu sorunlar arasında, enerji tüketimine yönelik anormallik tespiti, verimli enerji tasarrufu sistemlerini kullanırken, genel enerji kullanımını ve karbon emisyonlarını azaltırken dikkate alınması gereken çok önemli bir faktördü. Bu nedenle, Nesnelerin İnternetine (IoT) dayalı güçlü bir teknik önermek, binalardaki anormal tüketimi tespit etmek ve bu bilgileri ödemeleri azaltmak için uygun bir şekilde işlenmek üzere tüketicilere ve yönetimlere sunmak için daha büyük önem taşıyabilir. Sonuç olarak, bu tez LSTM algoritmasını kullanarak anomalileri tespit etmek için bir optimizasyon yöntemi önermekte ve ADAM, AadMax ve Nadam olmak üzere üç optimizasyon algoritmasını incelemektedir. ARIMA ve SARIMAX zaman serisi tahminleri için iki istatistiksel modelleme kullanılmıştır. Anomali tespit sonuçları, Nadar ile LSTM kullanımının en iyi sonuçları verdiğini, MSE ve RMSE'nin sırasıyla 0,15348 ve 0,02356'ya ulaştığını göstermektedir. Ayrıca, ARIMA modelinin kullanılması, sırasıyla 0.13859 ve 300.94365 AIC ve MSE değerleriyle en iyi genel sonuçları vermektedir. Bu, anomali tespitini optimize etmede önerilen modelin güvenilirliğini ve uyarlanabilirliğini doğrulamaktadır.

Anahtar Kelimeler: *Zaman Serisi Tahmini, ARIMA, LSTM, SARIMAX, ADAM, NADAM.*

1. INTRODUCTION

The significant concern is about decreasing greenhouse gas discharges and their consequences on global warming. A predominant part of the world's power, almost 85%, is produced by non-renewable sources such as nuclear power plants, coal, and oil. [1]. As the usage of renewable and environmentally friendly energy sources becomes more crucial, it is equally crucial to minimize energy waste and develop strategies to make the most of this precious resource. [2]. The challenge of energy storage persists due to several constraints, such as inadequate storage capacity and overproduction [3,4]. In essence, energy efficiency may improve the way buildings utilize energy, reducing the negative consequences on society, the economy, and the global environment [5]. As a result, balancing energy output to meet actual demand is the main objective of creating a sustainable smart city. [3].

Recently, most governments throughout the globe have initiated numerous research efforts aimed at creating unique techniques to increase energy efficiency [6]. Smart meters are being widely used to collect detailed consumption patterns as a result of the Internet of Things' (IoT) quick development and the widespread application of AI algorithms for data analysis. This enables real-time usage footprint monitoring and the identification of anomalies in a variety of electrical devices. [7,8]. IoT may be used to track and manage operations, energy use, equipment, and energy sources. In addition, the IoT system is utilized to assure system stability or constant performance, to minimize the consumption of energy as much as possible, and to keep a system from being overloaded [9]. As a result, IoT helps the system be protected from predicted losses including system outages, accidents, and damaged equipment[5]. For example, essential IoT capabilities include adjusting device settings, turning off or adjusting lights, adjusting various house settings, and optimizing energy utilization. Furthermore, most problems are caused by a variety of factors, such as damaged equipment, outdated appliances, or malfunctioning system components, all these aspects can be handled by IoT [10].

Many factors influence energy consumption in smart homes, including device use, weather, and consumption value. However, although these variables may not have a direct impact on the consumption of energy, they may have a direct correlation with it. As a result, the characteristics that are highly connected with energy usage must first be determined. To locate and manage erroneous data, the data must be pre-processed, such as formatted, outlier detection, and aggregated, before it is utilized to create the forecast. To identify the mathematical model that describes this system and provide projections based on such parameters, a sizable amount of historical consumption data and parameters are needed [11]. Based on past data, statistical and Machine Learning (ML) approaches are utilized to understand probable relationships between the many aspects driving consumption and draw inferences about future consumption [12]. The anomalous detection can be classified into two types based on the used approach of supervised and unsupervised, which will handle the labeled and unlabeled datasets respectively, where each approach handles it from a different prospect As a result, the lack of annotated data sets, The creation of power abnormality detection systems is a difficult endeavor since it requires precise definitions of normal and abnormal energy use..

To contribute to the improvement of anomaly detection, this thesis proposes a framework for controlling electrical consumption and detection anomaly based on using the LSTM algorithm and two aspects of statistical modeling of ARIMA and SARIMAX with three optimization algorithms of ADAM, AdaMax, and Nadam. The optimization will be based on using a dedicated value of dropout value to reduce the impact of overfitting. Additionally, the dedicated learning rate will be adjusted to handle the optimization of the proposed system. The rest of the thesis is organized as follows: related work mentioned in the second section, the methodology related to the utilized NN method, statistical modelling, and the optimization method in section 3 will be clarified and will highlight the proposed system In Section 4, the dataset that was employed will be presented. Subsequently, Section 5 will feature an extensive discussion of the results obtained. Finally, in Section 6, a conclusive summary of the study will be provided.

1.1 Background

Smart buildings integrate communication systems, information technology, comfort, and "security together according to the user's needs and environmental requirements, as they provide useful services such as thermal comfort, air quality, security, sanitation, and heating, which makes them economic and less energy consumption.

Machine Learning (ML) is a discipline within the field of artificial intelligence that is utilized in this integration, is one of the IoT systems' many advantages in smart city applications, such as managing and lowering energy use, which allows governments to save enormous sums of money, where the computer becomes able to learn on its own through previous experiences, so it becomes able to predict and make the appropriate decision Also, it depends on training to accomplish detection of current abnormalities in order to optimize the maintenance of power and control systems, which is a crucial component of the smart city. In this study, we propose to control the energy consumption of a smart city, where energy efficiency has become a big problem so that the energy produced is close to the real demand for energy through the energy consumption of the devices depending on the dataset and using machine learning (ML) algorithms to detect the anomalies of consuming of power and optimization the maintenance of power of the smart city.

Upon completing the dataset processing, including the elimination of null values, feature selection, normalization, and partitioning into distinct subsets, namely the training set and test set, the subsequent step is to execute the training model for the purpose of detecting anomalies, followed by an assessment of the trained model's performance."

The study showed, "after comparing the group of algorithms used, that (ARIMA) algorithm gave the best results that were using a smart city dataset in Python language in Colab.

The size, quality, and sampling rate of the data set affects the performance of machine learning techniques. In order to enhance machine learning model performance, a small data set must be reviewed.

With the help of automatically controlled, internet-connected systems, IoT home automation allows for the switching of domestic usage. The process may entail configuring advanced heating and lighting systems, as well as alarm and home

security controls, which are interlinked through a central hub and managed from a distance via a smartphone application. The power of artificial intelligence is immense. On a technology level, it imitates human capacities for learning and skill development.

Recent advances in artificial intelligence have been made in cloud communication, learning human behavioral patterns, and user preference-based automation of smart home devices.

Future Internet of Things (IoT) symmetric usage between the most significant sources of creative data will be created by machine learning systems.

According to this viewpoint, network systems have the capacity to contact several sources of experimental symmetric data via a wide range of network policies, analyze the data information, gather information, and make informed decisions based on the dataset after it is removed.

This knowledge is restricted to supervised and unsupervised machine learning (ML) techniques, which are used as the basis for the examination of IoT smart data.

This study includes machine learning techniques, highlighting the advantages and limitations of each algorithm, and it also covers current research trends and suggests areas for future research.

Figure 1.1: Smart Home IoT basic Overview

Fig.1.1 represents the smart home IoT overview basics. The smart home is one application of ambient intelligence (AmI) that helps people with their daily tasks and has grown in popularity over the past ten years. The smart home can assist the occupant in many ways, including keeping an eye out for potential dangers, spotting any abnormalities, adjusting the home for environmental conditions, and inducing behavioral change. This frequently calls for the smart home to recognize the resident's behaviors. In this research, we present an approach that can precisely recognize the behaviors of the inhabitant. Segmenting the sensor flow and identifying behaviors are both included in this. We use sensor data from actual smart homes to illustrate our approach.

1.2 Research Aims

Enhancing the energy system and optimizing energy utilization in intelligent edifices involve mitigating the occurrence of anomalies subsequent to detecting events, infrequent constituents, or observations that are dubious owing to their substantial disparity from the prevailing data. These anomalies signify issues such as structural defects or errors in textual representation, and they manifest as Outliers, Novelties, Noise, Deviations, and Exceptions.

1.3 Research Benefit

Creating a state of integration between machine learning and Internet of Things systems with its applications in smart cities has many benefits, including controlling the energy system and reducing consumption, and thus will save governments, companies and citizens huge sums of money. With there is a severe energy crisis now and the population's average age is growing, the advantages come from adopting cost-effective wireless.

1.4 Purpose of the Study

The purpose of the study is to find the best model to reduce the percentage of anomalies in a dataset to obtain the highest accuracy in the results in order to rationalize energy consumption in smart buildings.

1.5 IoT vs Machine Learning

Machine learning and deep learning require enormous volumes of data to operate, and this data are being obtained through the continual deployment of billions of new sensors for the Internet of Things. Better AI is produced by IoT.

1.6 Benefits of IoT-Based Home Automation

- Accessibility.
- More Suppleness.
- Greater Safety
- Comprehensive Control.
- Energy Effectiveness.
- Enhanced Functionality.
- Recovered Home Organization.

1.7 Machine Learning

The objective is to enable the machine to operate autonomously and independently, without human intervention, by facilitating its self-understanding. This involves leveraging specialized coding techniques to enable the machine to perform intricate mathematical computations rapidly and consistently on vast datasets, thereby allowing it to proactively address potential issues. Additionally, these programs possess the capacity to learn, evolve, and adapt based on new data inputs.

Machine learning has seen an unprecedented rise in popularity after its integration with the Internet of Things technology. It is considered the best technology used to achieve the highest levels of efficiency through this integration to reach the highest efficiency in all service and production operations.

1.8 Thesis Outlines

An overview of the thesis structure and organization is provided in Figure 1.4

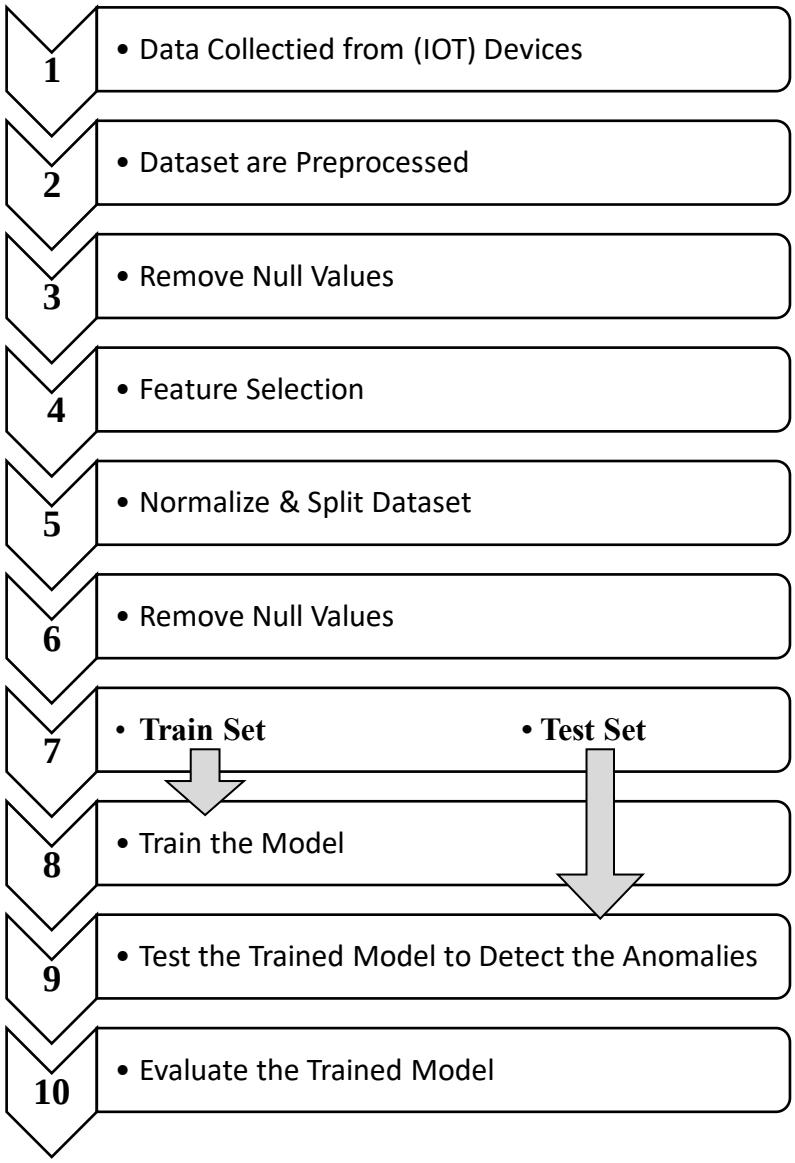


Figure 1.2: Thesis Outlines

2. LITERATURE REVIEW

2.1 The General Framework of ML Solution

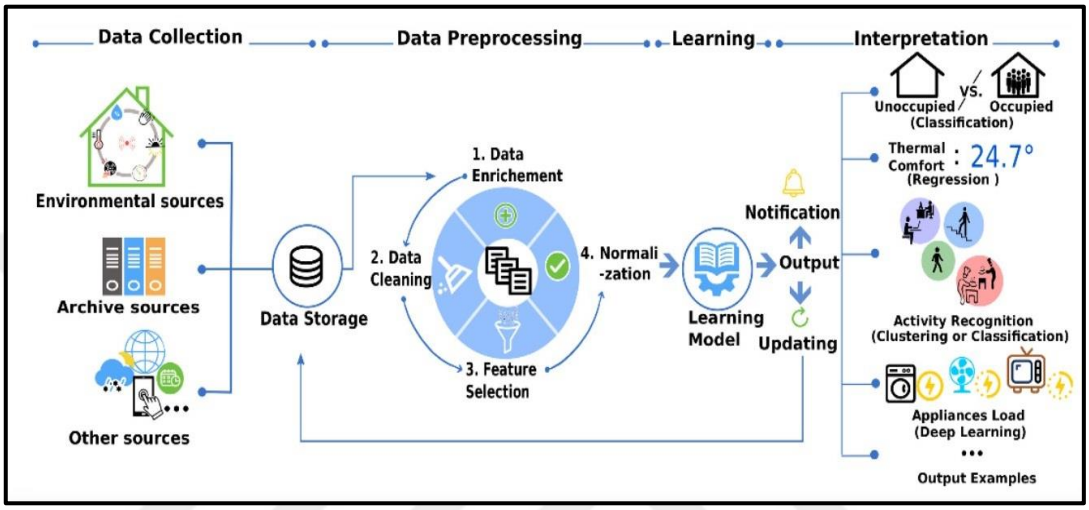


Figure 2.1: General ML Solution Framework

The general framework of ML solution as shown in Fig 2.1

1. Data Collection:

i. Environmental Sources.

ii. Archive Sources.

iii. Other Sources.

2. Data Storage:

3. Data preprocessing:

iv. Data Enrichment.

v. Data Cleaning.

vi. Feature Selection.

vii. Normalization.

4. Learning:

viii. Learning Model.

ix. *Notification.*

x. *Output.*

xi. *Updating.*

5. Interpretation:

xii. *Unoccupied Occupied (Classification).*

xiii. *Thermal Comfort (Regression).*

xiv. *Activity Recognition (Clustering or Classification).*

xv. *Appliances Load (Deep Learning).*

xvi. *Output Example*.

2.2 The Main Components of ML

"Learning, it is one of the disciplines of artificial intelligence focused with making the computer capable of learning on its own from any experience or past experiences, allowing it to forecast and make suitable decisions more quickly, its main components are as shown in Fig 2.2."

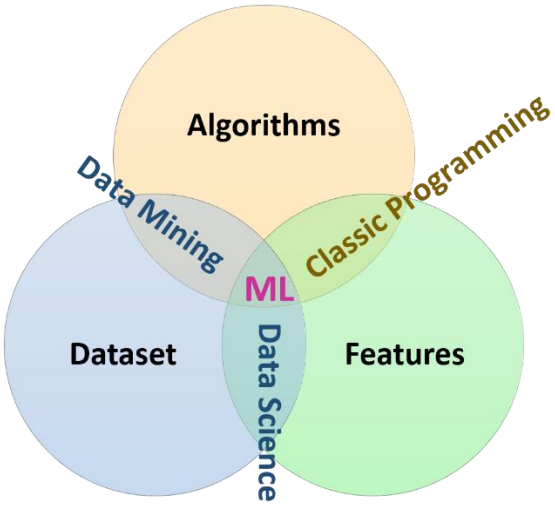


Figure 2.2: The Main Components of ML

- Dataset: The more diverse the data, the better result, they are two types Manual method & Automatic method:

Data collected manually has significantly fewer errors than its automated counterpart but takes longer to collect and is generally more expensive.

The automated method is cheaper because all we will do is collect everything we can find and hope that the quality of this data is acceptable, as seen in Table 2.1

Table 2.1: Dataset Types

Manual method	Automatic method
1- Fewer errors.	1- More errors.
2- It takes longer to assemble.	2- Less time to assemble.
3- More expensive generally.	3- Less expensive overall.

The Smart Home Dataset with Weather Information used in our study is of the automated method.

- Features: These are known by the names of parameters or variables, which are the name of columns.

-Algorithms: is the easiest and most obvious part. Any problem can be solved in different ways. However, the method you choose will affect the accuracy, performance, and size of the final model. (If the data is bad, not even the best algorithm out there will help you.)

- Model: it is a summary of what the data has learned, and it can use a ready–made model and pass the data to it, or improve an existing model.

2.3 Types and Methods of Machine Learning

There are three main categories in the classification of machine learning depending on the type and nature of the signal and feedback provided as shown in Fig 2.3.

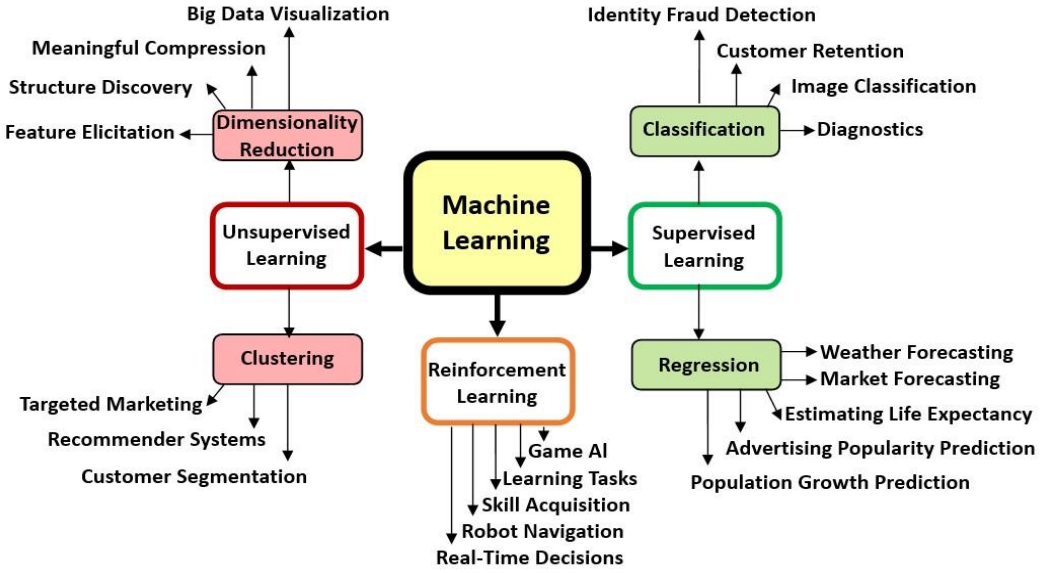


Figure 2.3: Types and Methods of Machine Learning

2.3.1 Supervised learning

- Features simple data and clear features, it is the focus of our study depending on the type of dataset used in the study from Regression type.

- An advanced predictive model based on input and output data.

An algorithm can learn by using predictive outputs that are associated with new inputs.

- The optimization function of the algorithm helps it correctly identify the outputs of inputs that were not part of the training data.

- The accuracy of the output or its predictions improves over time as it learns to do the task- Supervised learning algorithms include both classification and regression.

A. Regression

- Regression algorithms are used when the output has any numerical value within a range.

-The idea is that there are data related to each other, and what is required is to find new data values.

- Ex: There is input data (House space, Number of rooms, Address)

and output data (House price).

Giving the algorithm the inputs and outputs to make an appropriate equation through which the output can be predicted (Price of the house).

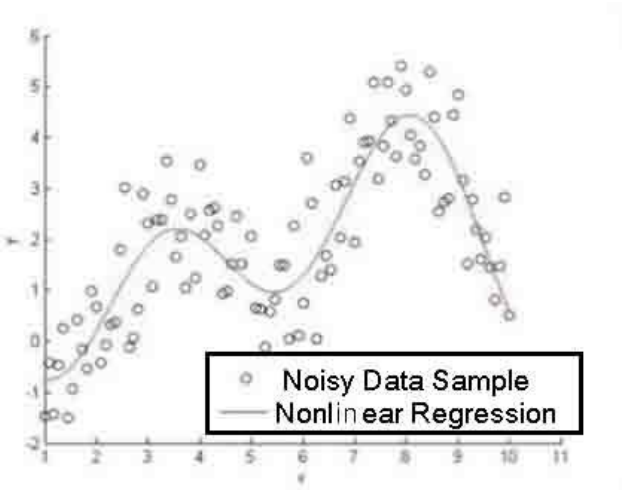


Figure 2.4: Regression

B. Classification

- Classification algorithms are used when the output is limited to a finite set of values.

- called (Classification) or (Logistic Regression).

- Divide a group of items into a number of sections.

- It is based on similar characteristics.

- Most models are of only two types.

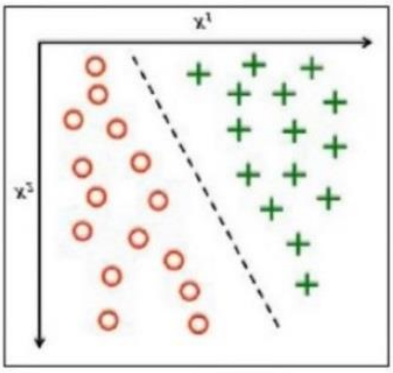


Fig. 2.5 Classification

2.3.2 Unsupervised learning

It is croup and interpret data based only on input data, and is characterized by complex data and unclear features, and it is two types:

A. Clustering

There is data without information about it, which is required to be divided into similar groups.

B. Association

Associations are used to link objects together because they are a typed relationship between two objects that must be defined in each object's class declaration in order to allow associations to occur.

A model is used to analyze data for common patterns or occurrences in a database, it defines recurring conditional associations, which are themselves rules of the association.

2.3.3 Reinforcement learning

Does not have data, but there is an environment that can be interacted with. when a computer program interacts with the dynamic environment to achieve a specific goal as the program is equipped with reward-like feedback to maximize it.

More than one type may be used in the same machine learning system to suit the development of the work method, which made deep learning (DL) the dominant and continuous one in this field [10].

2.4 Reasons Why Machine Learning Programs Fail To Achieve Expected Results

1. Lack of data.
2. Data privacy issues.
3. Difficulty accessing data.
4. Data bias.
5. Poor selection of the type of algorithm to perform the task.
6. Lack of resources and a problem with tools and error evaluation

An example of this is the death of a citizen after he collided with a self-driving vehicle while it was moving.

Some attempts to use machine learning in many areas may fail if it is used and harnessed for the purpose of investment only. An example of this is in the field of health care. After accurate detection and diagnosis of the disease, the patient is given many analyzes and additional examinations to raise the percentage of profits only.

2.5 Algorithm

"A procedure used to solve a problem or perform a mathematical operation. Algorithms act as an exact list of instructions that carry out the specified actions step by step whether in hardware or software-based procedures. Algorithms usually start with initial inputs and specific instructions that describe a computational operation and when executed, the results of that operation appear as output and are designed to accomplish different tasks, uses Algorithms are used in all different fields of

information technology, mathematics, statistics, and computer science, and in many different sectors of life (service, health, education, sports, etc.)"

Algorithms employ unprocessed data in conjunction with a prescribed sequence of instructions. "Inputs comprise fundamental information necessary for decision-making and may be expressed numerically or verbally. The input data undergoes computational processing, which may entail mathematical operations and discernment procedures. The algorithm culminates in its output, which frequently encompasses supplementary data.

Natural languages, programming languages, pseudocode, flowcharts, and control tables are all ways to represent algorithms. Natural language phrases are more difficult to find since they are more ambiguous. Programming languages are often used to express computer-implemented algorithms".

2.6 Types of Algorithms with Different Tasks

i. *Search Engine Algorithm*:

It searches the associated database for relevant web pages after taking the search strings of keywords and parameters as input and then returns the results.

ii. *Encryption Algorithm*:

Data transformation according to specific procedures to protect it. It uses the symmetric key algorithm and the data cannot be decrypted by anyone who does not have the decryption key.

iii. *Greedy Algorithm*:

Optimization problems are solved by finding the optimal solution locally, with the hope that the optimal solution will be global.

iv. *The Recursive Algorithm*:

Calls itself with a smaller value each time the recursive function is called and iteratively until the problem is resolved.

v. *Backtracking Algorithm*:

You find the solution to a problem in a step-by-step fashion, one piece at a time.

vi. Divide and conquer Algorithm:

It was divided into two parts. The first breaks down the problem into smaller subproblems. The second one works on the sub-problems and brings them together to come up with a solution.

vii. Dynamic Programming Algorithm:

It solves problems after dividing them into sub-problems. Then store the results to apply to the corresponding problems in the future [12].

2.7 Related Work

Establishing which aspects are emphasized as being unique from the others is essential when analyzing actual power usage data. Such variables are considered anomalies, and gathering all such variables in a data-driven manner is necessary to discover abnormal power consumption. In this situation, the patterns of production of anomalous power consumption may be anticipated based on previously acquired abnormal data. Many innovative algorithms have been developed by many researchers and industries to forecast energy usage anomalies in smart buildings [13,14]. For instance, a methodology proposed to detect anomalies is based on two stages consumption prediction and detection of an anomaly. The prediction is done by using Autoregressive Integrated Moving Average (ARIMA), while detection is done with two signal rules. The prediction accuracy obtained was between (89.1%-96.5%) for the case of 8 weeks of the data window. In contrast, using such an old method fails to satisfy recent IoT-based detection methods [15]. Additionally, authors in [16] proposed the method of Days Exceeding Threshold (DET)-TOA for energy consumption and temperature based on deviation difference between the simulated and measured consumption of energy. Analysis of results gives the best overall performance as compared to traditional DET due to different consecutive ways. However, the robustness of the proposed method needs to be further analyzed with different cases. A multivariate time series-based method was proposed in [17] by using single class RNN, where it has been using the Spatio Temporal-Long Short Term memory (TL-LSTM) and the trust gates indicate higher performance with an accuracy of 95% and 100% for the tested dataset. A method for feature learning and detection of anomalies by using a hybrid of the Deep Learning (DL) method of

LSTM and Multilayer Perceptron (MLP). Results obtained from the proposed method give higher accuracy as compared with other methods of historical and non-historical power data [18]. A supervised learning-based method was proposed for the detection and classification of abnormal consumption of energy by using Deep Neural Network (DNN) along with a scattering of micro-moment classes. Results of accuracy and F1 score give values of 99.58% and 97.85% respectively. However, the method needs further improvement to satisfy different outdoor conditions [19]. An electrical consumption-based method was presented in cloud computing for both centralized and decentralized. However, this method needs to be investigated to quantify with efficient detection and improve reliability [20]. Furthermore, two models for identifying anomalous consumption were proposed in [21], where the first model uses single class Support Vector Machine (SVM) without the need for annotated data. While the second model used supervised detection of micro-moment with K-Nearest Neighbors (KNN) for the issue of classifying the consumption fingerprint. Three datasets were involved in the investigation, and selecting k=5, gives high accuracy and F1 score for the proposed model. However, the accuracy of model performance has been down to 93.43 for one dataset which fails to classify data and needs to be further analyzed. Recently, a framework for controlling energy consumption is based on smart houses using ML. The data is gathered from IoT devices for time series analysis. Three models of Vector Autoregressive (VAR), prophet, and Light GBM are included in the framework investigation. The last two models give the optimal performance in detection and future prediction. The proposed method exhibits a limitation in relation to the dimensions of the dataset and the sampling rate employed. Therefore, alternative approaches should be considered in order to enhance the performance of the machine learning model [22].

2.8 General Background

By highlighting the advantages and disadvantages of the various houses' equipment's heterogeneous technologies, the use of wireless technologies in smart homes is examined, which poses issues for the monitoring of the building and its occupants. "One of the crucial challenges confronting this application involves the monitoring of representative elderly individuals and the utilization of unobtrusive wireless devices to detect their activity while also monitoring the relevant areas." In these

cases, the user's self-evaluation of his activities can improve the services' capabilities [23].

Wireless architectures have been used as examples of fully autonomous environments and as flexible, transparent technologies. "The properties of wireless signals may be used to evaluate presence and position tracking, discover patterns of resident movements and behaviors, and lower the complexity and expense of the system while preserving respect for the cutting-edge solutions on which modern architecture is founded ".

In smart homes and WSNs (wireless sensor networks), "Testing has shown that a more useful architectural instrument for offering services for detecting movement is the widely-used wireless technologies. Consider decentralized smart energy metering and senior support as examples of smart features for the home. The adoption of wireless smart home systems is driven by these two criteria. After all, there is currently a significant energy problem, and the population's average age is increasing. Adoption of low-cost wireless networks with designs that prioritize researching user behavior for the elderly results in advantages"[24].

In order to track and locate the user's behavior, special approaches for daily activity detection were researched, tested, and integrated into contemporary wireless architecture. Wireless non-intrusive solutions have been adopted, utilizing electromagnetic detection and pervasive mobile devices like cellphones. Numerous effective electromagnetic field technologies can guarantee a smart home's minimal complexity and the user's acceptance of it [25].

The pursuit of artificial intelligence led to the development of machine learning. The academic field of artificial intelligence is described as the study of how computers learn by ingesting and processing data. The researchers tested a number of approaches; however, they mostly employed the neural networks technique (Neural Networks/NN).

While classic artificial intelligence techniques were employed prior to 1992, machine learning only really started to take off after that year. "In contrast to traditional artificial intelligence systems that use symbols, machine learning employs procedures based on statistics and probability. Conventional machine learning algorithms, such as decision trees, have a straightforward algorithmic framework.

With traditional machine learning, the engineer must feed the machine with a large amount of data and the essential information before maybe manually altering the algorithm to allow the machine to work and provide the desired results".

Although making errors, deep machine learning, on the other hand, allows the computer to self-learn via repetition and feedback. Thus, until we obtain the desired outputs, the neuron weights and inputs are continuously adjusted. "Deep learning is a cutting-edge methodology that aims to enable machines to learn and perform tasks that are typically associated with human intelligence, such as image and speech recognition, natural language processing, and decision-making" [26]. This approach involves training complex algorithms across multiple layers of artificial neural networks, which are designed to mimic the structure and function of the human brain. Through this process, machines can gradually improve their performance by refining their models and adjusting their parameters to better fit the data they are given.

One of the key features of deep learning is its ability to perform unsupervised learning, which means that it can automatically discover patterns and structures in large datasets without being explicitly told what to look for. This is particularly useful for applications such as anomaly detection and data clustering, where the goal is to identify hidden patterns or anomalies in the data.

Another advantage of deep learning is its flexibility and scalability. The same deep learning model can be used for a wide range of applications and can be trained on massive datasets, making it suitable for handling complex and diverse data types. Furthermore, the modular structure of deep learning models allows for easy integration of new components and customization to meet specific requirements.

Overall, deep learning represents a powerful and rapidly evolving field that holds immense potential for a wide range of industries, including healthcare, finance, transportation, and many others. By leveraging the power of deep learning, we can unlock new insights, improve decision-making, and create innovative solutions to some of the most pressing challenges of our time.

Several academics have argued in recent years that machine learning is still a kind of artificial intelligence. Nonetheless, a sizable number of experts contend that artificial intelligence and machine learning are two entirely distinct disciplines [27].

Applications for deep learning are mostly used for voice and image recognition, brain circuit rebuilding, and predicting how DNA changes would affect a disease's genetic expression. " The domain of deep learning, which is a subset of artificial intelligence (AI), has been demonstrated to hold considerable potential for a variety of applications. One such area where deep learning shows promise is in its ability to enhance the understanding of natural language" [28]. By leveraging its ability to analyze vast amounts of data, deep learning can provide insights into the subtleties of language usage, including context, semantics, and syntax.

Furthermore, deep learning has the potential to be applied to the evaluation of emotions. By using neural networks, deep learning algorithms can be trained to identify patterns in facial expressions, speech patterns, and physiological responses, allowing for the recognition and interpretation of emotional states.

Another promising application of deep learning is in the field of language translation. With its ability to learn from large volumes of data, deep learning can be used to develop real-time translation systems that can quickly and accurately translate various languages. Such systems can be of great benefit in today's interconnected world, facilitating communication and understanding between people of different cultures and languages.

Overall, the potential uses of deep learning are vast and varied, and as the technology continues to develop and evolve, it is likely that even more innovative and exciting applications will emerge.

The application of deep networks (DNs) to unsupervised learning has proven to be an area of immense potential and significance in recent times. DNs have been widely used in various domains, including audio, imagery, and text, among others, with remarkable success" [29]. These networks have shown great potential in their ability to learn from unlabeled data, thereby enabling the extraction of useful features from the input data.

While DNs have been extensively utilized in single-task settings, certain DNs have demonstrated the ability to effectively collaborate on multiple tasks and acquire diverse features. For instance, DNs have been employed to process real-time video and audio in a video format, showcasing their remarkable ability to learn from complex and dynamic data.

In such a scenario, the application of deep networks is particularly relevant, as it enables the networks to efficiently process large amounts of data in real time. By employing deep networks in the analysis of video and audio data, we can effectively extract meaningful information from the input data in real time, thereby enabling us to make decisions and take actions based on the extracted insights.

Thus, it is evident that deep networks have immense potential in a wide range of applications, including unsupervised learning. The ability of these networks to learn from unlabeled data and to collaborate on multiple tasks makes them valuable assets for the analysis of complex and dynamic data, thereby enabling us to extract valuable insights from the data in real-time.

utilizing the Deep Learning methodology, the computer is now able to mimic the way the human brain learns things to a large extent. Both humans and machines now learn from their experiences and observations. Concepts can now be learned by machines without having to be entered manually by humans. The machine learns new, more complicated concepts by building on and combining those it already knows through repetitions, continuous feedback, and simultaneous classification of concepts.

Prioritizing well-known notions on many levels is the fundamental task to be completed in order to achieve this. There are numerous uses for machine learning, including facial recognition, search engines, advertising, and picture recovery [30].

2.9 Survey of Previous Studies

Posted by [31] Several" M.L. models were used to assess the time-series data produced by smart meters through thorough tests and trials. By giving the IoT devices in a smart home a spam score, different IoT device contribution levels were identified with the use of ensemble machine-learning techniques. The findings demonstrate that the devices' spam city scores contribute to improving the prerequisites for an IoT device to perform successfully in a smart home."

The weather information and the data from the smart home may be efficiently used to determine the security of IoT devices and to assign a spam score.

Writer in [32] Due to the inadequacy of ARIMA's pattern identification and the issue with lag determination, the forecasting power of traditional approaches is insufficient

to estimate future values. Due to the utilization of several processing threads to do multiple jobs concurrently, a parallel GA is faster than a sequential approach. Using the top performers across many populations, the parallel GA delivers superior results. The SARIMA approach has a huge number of factors, which makes its prediction power less effective. In light of this, genetic algorithms offer an appreciable method for these parameters' estimation in the ARIMA model. In order to improve the accuracy of temperature prediction, the parallel GA-SARIMA forecasting model is applied.

Comparing the estimated values with the actual values serves as a test step to demonstrate forecasting accuracy. The model's testing procedure demonstrates if the suggested model makes accurate predictions or has any overfitting or underfitting issues. Because of its acceptable accuracy, the chosen model is utilized to forecast future values. "A different method for estimating the parameters (p, d, q) x (P, D, Q) that enhance the performance of the SARIMA model to forecast parameters, including temperature data, with more precision is the GA based on the SARIMA model. Author in [33] Vertical plant wall systems, embedded with sensors and actuators, have become a promising application for indoor climate control, explore the possibility of applying machine learning based anomaly detection methods to vertical plant wall systems so as to enhance the automation and improve the intelligence to realize predictive maintenance of the indoor climate. It also serves as an illustration of how research on the construction environment may fully take advantage of advances in machine learning and the Internet of Things to speed up the creation of solutions. By using the models on the CO and temperature datasets gathered from real indoor environments, the study benchmarked the performances of two categories of approaches, i.e., prediction-based and pattern recognition-based detection methods. The findings indicate that while the LSTM-ED ensemble model outperforms the others in the identification of point anomalies, the autoencoder neural network model excels in the detection of contextual anomalies. Both of them perform better than baseline models that are offered as default tools by the Azure cloud platform." Posted by [34] Traditional statistical models are frequently employed for identifying abnormalities in streaming data produced by IoT-based sensors, notably because of their transparency." Despite certain limitations inherent in both types of models, deep learning-based anomaly detection techniques have

made significant progress. However, no single technique can adequately address every use case. To overcome this challenge, the Fuse AD approach combines statistical and deep learning-based models for time-series anomaly detection, thereby leveraging the strengths of both models. Unlike previous model fusion techniques, Fuse AD adopts a novel residual learning framework that enables the network to learn when to favoura specific model's forecast. This approach enables the system to capitalize on the advantages of both models. Results from experiments on the Yahoo Web scope dataset unequivocally demonstrate the superiority of Fuse AD over cutting-edge anomaly detection techniques. Ablation testing and comparative analysis show that the use of Fuse AD significantly improves performance when compared to the use of standalone statistical or deep learning models."

Author in [35] "As compared to the DET-Date approach, the DET-Toa method has a better ability to identify irregular building energy usage. To detect persisting small increases or decreases in normal building energy consumption, a temperature-based method known as the Days Exceeding Threshold-Toa (DET-Toa) method identifies an abnormal energy consumption fault when the deviation between measured and simulated consumption is greater than one standard deviation of the residuals in the baseline period and persists. In the 20 simulation test instances, it correctly recognized 19 synthetic control changes". During simulated experiments with two campus buildings, the method's capacity to detect faults is assessed.

3. ALGORITHMS METHODOLOGY

This section will outline the associated theoretical concepts and methodologies that comprise the suggested model.

3.1 Time Series Forecasting

"Demand forecasting is regularly performed using several kinds of econometric models. The collection of observations known as a time-series, which may be identified by its stationarity, was made progressively through time (mean, variance, and autocorrelation function) [36,37], is one of the basic approaches of quantitative forecasting. To attain the least probability accuracy based on the current and anticipated values, Forecasts produced at t are required at some future time t + l, or at lead time l, which varies depending on the issue and property [36]. In addition, time series forecasting is a data science technique that is frequently used in manufacturing, supply chain management, inventory planning, and business and finance. Many prediction problems have a time component, thus extrapolating time series data or predicting time series is necessary. Another important use of ML is time series forecasting, which may be seen as a supervised learning issue. Together with other ML methods, it may be utilized with regression, NN, SVM, and random forests (RF). Using models based on prior data to foresee future observations is the process of forecasting [38].

Conventional time-series models have been used for forecasting and are useful, such as moving averages, exponential smoothing, and candy regression. higher level of sophistication models, like ARIMA in numerous forms, have also been utilized. These statistical models frequently use historical data, but the outcome depends on the selection of appropriate parameters as a prerequisite. Due to improved Artificial Neural Networks (ANN), algorithm modeling, have demonstrated incredible success in creating forecasting models for several sectors. Different trends or seasonal patterns may be captured by forecasting models. As a consequence, to produce a workable time series forecasting model based on given parameters, the connected parts explore seasonality, stationarity, and autocorrelations. But still, the magnitude outliers that cause problems and have an impact on the fitted model's state can also

have an impact on predicting models [37,38]. This thesis will use the seasonal SARIMAX and ARIMA statistical models with exogenous variables".

3.2 ARIMA

An analytical time series statistical model is called ARIMA. The ARIMA model has two parts: the Moving Average (MA) and Autoregression (AR) models [39, 40]. "The AR model takes use of the dependent relationship between an observation and its delay. The time series are made stationary by using raw observation differencing, which is referred to as the integrated term. Equation (1) demonstrates the calculation of the AR model for the order of P." The association between an observation and a residual error from a moving average applied to delayed data is used by the MA model. According to the moving average, the variable's future value will be decided by the average of k prior values [41]. Moveable average model may be calculated using equation (2) [22]

$$y_t = c + \Phi 1 y_{t-1} + \Phi 2 y_{t-2} + \ldots\ldots + \Phi P y_{t-P} + \varepsilon_t \quad \textbf{(3.1)}$$

$$\widehat{y_t} = \sum_{n=1}^{k} y_{t-n} \quad \textbf{(3.2)}$$

Where $\Phi$is the parameters, ε_t is the white noise, $\widehat{y_t}$ represent the predicted value, y_{t-n} is the variable value, and t represents the time. Using ARIMA provides some benefits, which are represented by being suitable for short-term forecasting, only requiring historical data, and providing models for nonstationary data.

3.2.1 ARIMA models

- ARIMA models, short for "Auto Regressive Integrated Moving Average," are statistical techniques used to model time series data. They employ a set of parameters to capture the relationships between prior values, including any delays and lagged errors in prediction, to produce a forecast for future values of the time series.
- With ARIMA models, we can normalize any Time Series that do not reflect seasonal patterns and are not random noise.

"There are three terms describing An ARIMA model:

P, Q, and D where,

P = the command of the AR term

Q = the command of the MA term

D = the number of changes essential to type the time series stationary"

In cases where a time series exhibits cyclical patterns, it is necessary to incorporate cyclical periods, resulting in a Seasonal ARIMA or SARIMA model.

The ARIMA model is constructed by first establishing a stationary time series. The "AR" or auto-regressive component of the model utilizes linear regression with lagged predictors. It is important to note that linear regression models perform best when working with independent and uncorrelated data. To achieve stationarity, the most commonly used method involves differencing the time series by subtracting past values from the current value. Multiple subtractions may be necessary depending on the complexity of the series. The parameter "D" represents the minimum number of differences required to achieve stationarity, and becomes zero if the time series is already stationary.

Moving on to the parameter's "P" and "Q", "P" refers to the number of lagged predictors used in the auto-regressive component, while "Q" represents the number of lagged forecast errors used in the moving average component of the ARIMA model.

AR and MA models are in detail

- Moving average (MA) and auto-regressive (AR) models.
- The true mathematical formulation for the AR and MA models.

A common model that depends simply on its individual lags is an auto-regressive model known as a pure AR model. As a result, we may likewise conclude that the "lags of Yt" serve this role.

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \ldots + \beta_p Y_{t-p} + \varepsilon_t$$

"where, $Y_{t-1} \rightarrow$ is the lag1 of the series.

$\beta_1 \rightarrow$ The model calculates the term of capture as 1, which is the coefficient of lag1.

A Pure MA (Moving Average alone) model is one where $\boldsymbol{Y_t}$ depends exclusively on the lagged predicted errors."

$$Y_t = \alpha + \varepsilon_1 + \Phi_1 \varepsilon_{t-1} + \Phi_2 \varepsilon_{t-2} + \ldots + \Phi_q \varepsilon_{t-q}$$

The matching lags errors of the AR models are the error relations everywhere. Since the following equations have been given, the errors t and t-1 are the errors:

$$Y_t = \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \ldots + \beta_0 Y_0 + \varepsilon_t$$

$$Y_{t-1} = \beta_2 Y_{t-2} + \beta_3 Y_{t-3} + \ldots + \beta_0 Y_0 + \varepsilon_{t-1}$$

As a result, we used Moving Average (MA) and Auto-Regressive (AR) models, respectively.

3.2.2 The Equation of an ARIMA Model

An ARIMA model is one in which the time sequence was at least once made inactive during instruction and in which the moving average (MA) and auto-regressive (AR) factors were combined. Following that, we arrived at the following calculation:

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \ldots + \beta_p Y_{t-p} \varepsilon_t + \Phi_1 \varepsilon_{t-1} + \Phi_2 \varepsilon_{t-2} + \ldots + \Phi_q \varepsilon_{t-q}$$

3.2.3 ARIMA Model in words

Mathematically we can represent the formula

$$\Delta y_t = c + \Phi_1 \Delta y_{t-1} + \Phi_1 \varepsilon_{t-1} + \varepsilon_t \quad \textbf{(3.3)}$$

"Where:

- C is an intercept of the ARMA model.
- Δ is the first difference operator.
- Y is the time lags

Three factors must be taken into account in the ARIMA model.

Values that, while implementing it, we must provide in our parameters. As a result, we may depict it as (P, D, Q).

P= lags in the autoregressive model.

D= differencing /integration order.

Q= moving average lags."

"Forecasted Yt = Constant + Linear Combination of Lagged Predicted Errors + Lagged Y Lags (up to P Lags) (up to Q lags)

Let's start by locating the letter "D" in the ARIMA model.

The primary aim of the ARIMA model is to achieve temporal steadiness of the time series, which is accomplished by employing the differencing operator "D".

ARIMA, which stands for Autoregressive Integrated Moving Average, is a widely used statistical technique in the field of time series analysis. This approach is particularly useful for modeling and forecasting the behavior of data that varies over time, such as stock prices, weather patterns, and economic indicators. By combining three key components - autoregression, differencing, and moving averages - ARIMA models can capture complex patterns and trends in time series data, making them valuable tools for prediction and decision-making in a wide range of fields. Whether you are analyzing market trends, predicting consumer behavior, or tracking changes in the climate, ARIMA offers a powerful framework for understanding the past and predicting the future. Now, let us realize the suitable differencing order.

-To attain a stable series with a discernible mean and an ACF plateau that reaches zero in a relatively short period across all regions, it is advisable to employ the minimal differencing order.

- If the autocorrelations are positive for a number of lags, the series requires further differencing (often 10 or more)."

- In contrast, the series is likely over-differenced if lag 1 auto correlation is fairly negative.

- When choosing between two differencing orders is difficult, the order with the smallest difference in the differenced series must be chosen.

3.2.4 Assessing linear models

Three primary criteria are employed for computing linear models. These are:

- The Mean Absolute Error (MAE) can be expressed in a more formal and professional manner as follows:

The Mean Absolute Error (MAE) is a statistical measure that quantifies the average magnitude of the errors between a set of predictions and their corresponding actual values. It is calculated by taking the absolute difference between each prediction and its corresponding actual value, then averaging those differences.

The MAE is commonly used in various fields, such as finance, economics, and engineering, to assess the accuracy of predictive models and forecasting techniques.

MAE: The easiest to identify. Indicates average error

$$\text{MAE} - \text{ Mean Absolute Error value} = \frac{1}{n}\sum_{i=1}^{n}|y_t - \hat{y}|^2 \quad \textbf{(3.4)}$$

- The Mean Squared Error (MSE) is a statistical metric used to assess the quality of a prediction or an estimator by calculating the average of the squared differences between the actual and predicted values. It is a common measure of the accuracy and precision of a model, and is widely used in various fields such as finance, engineering, and machine learning.

MSE: Similar to MAE, but with overblown noise and punished for higher errors.

$$\text{MSE} - \text{ Mean Squared Error value} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

It is tougher to understand than MAE as it's not in base units; however, it is usually more common.

- The Root Mean Squared Error (RMSE) is a statistical metric used to measure the difference between predicted and observed values in a regression analysis. It is calculated by taking the square root of the mean of the squared differences between the predicted and observed values. The RMSE is a widely used measure of model accuracy and is often used in fields such as economics, engineering, and science.

RMSE: Greatest general metric, similar to MSE, though, because it is in base units, the result is squarely rooted to make it more interpretable.

$$\text{RMSE} - \text{Root Mean Squared Error value} = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

It is optional that RMSE be used as the primary metric to understand your model.

3.2.5 Time series forecast accuracy metrics

The following metrics are frequently used to assess forecast accuracy:

ME – Mean Error = Sum of all error values / Number of records

$$\text{MAPE} - \text{Mean Absolute Percentage Error} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i} * 100\right|$$

$$\text{MPE} - \text{Mean Percentage Error} = \frac{100\%}{n}\sum_{t=1}^{n}\frac{a_t - f_t}{a_t}$$

$$\text{MSE} - \text{Mean Squared Error value} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$\text{RMSE} - \text{Root Mean Squared Error} = \sqrt{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y - \hat{y})^2$$

$$\text{MAE} - \text{Mean Absolute Error} = \frac{1}{n}\sum_{i=1}^{n}|y_t - \hat{y}|^2$$

ACF1 - Lag 1 autocorrelation of error: $y_t = \beta_0 + \beta_1 y_{t-1} + \varepsilon_t$

corr - correlation between the actual and the forecast $= \frac{\text{Cov}(x,y)}{S_x S_y}$

minmax - Error in Min-Max: $x' = \frac{x - \min(x)}{\max(x) - \min(x)}$

When comparing predictions from two different series, the MAPE, Correlation, and Min-Max Error are typically used.

3.2.6 Advantages of using ARIMA model

i. The main benefit of ARIMA forecasting is that it needs data on the period series in query only.

ii. First, this feature is beneficial if one is forecasting a huge quantity of time series. Second, this escapes a problematic that occurs occasionally with multivariate models.

iii. To generalize the forecast, all that is needed is the time series' prior data.

iv. Meets expectations for short-term forecasts.

v. Non-stationary time series are modeled.

3.2.7 Disadvantages of using ARIMA model

- Minor performance for long term forecasts. Cannot be used for cyclical time series.
- "Turning points are difficult to predict.
- The model's choice of (p, d, q) order is very subjective.
- In terms of computing, it is expensive.
- Long-term forecast performance is lower.
- Difficult to explain in comparison to exponential smoothing."

3.2.8 Concluding ARIMA

By simulating the correlations in the data, the ARIMA approach is a statistical tool for assessing and developing forecasting models that properly depict a time series.

To "generalize the forecast and enhance prediction accuracy while keeping the model's parsimony, ARIMA models only require past data from a time series."

Although being economical, using ARIMA models has a number of potential disadvantages. This is one of the most important because p and q parameter identification involve subjectivity.

While autocorrelation and partial autocorrelations are used, the model creator will choose which p and q to utilize based on their skill and experience.

Moreover, "compared to straightforward exponential smoothing and the Holt-Winters approach, ARIMA models are more complicated and have less explanatory power. And lastly, like other forecasting approaches, ARIMA models struggle to make long-term estimates and spot turning points since they are backward-looking." The cost of computing them might potentially be high.

So, when time series data are used alone, Short-term forecasting with ARIMA models is straightforward and accurate; however, finding the right combination of parameters for each application may need some skill and experimenting.

3.2.9 ARIMA model benefits and drawbacks

i. "Because it is linear, it is valuable and commonly used in the field for testing, data interpretation, and setting baseline forecasting scores.

ii. They can perform considerably better if correctly calibrated with delayed values (p, d, and q). Because of its simplicity and explainability, the algorithm is a popular choice among analysts and data scientists."

iii. Working with ARIMA at scale is not without its advantages and disadvantages, though. Let's talk about both of those.

3.2.10 Advantages of ARIMA

i. Simple to comprehend and apply: The simplicity and interpretability of the model are the one thing that your coworkers and colleagues would value.

ii. Presentations to stakeholders will go better if you concentrate on these whiles preserving the standard of the output.

iii. There are just a few variables Due to the fewer hyper parameters, maintaining the configuration file will be straightforward if the model is used.

3.2.11 Drawbacks of ARIMA

i. "***Exponential time complexity:*** If p and q are high, there are more coefficients to fit as p and q increase, which multiplies the time complexity."

ii. Due to their difficulty in implementation, this algorithm has Data Scientists looking towards Prophet and other techniques.

iii. However, it also relies on how complex the dataset is.

iv. **"*Data complexity:*** It's likely that your facts are too complicated and that p and q don't have the best feasible answers. The failure of ARIMA is quite unlikely, but if it does happen, you might have to go elsewhere."

v. Algorithm needs a lot of information to work, especially if the information is seasonal. Three years of historical demand, for instance, is probably not enough to produce a realistic prediction for products with short life cycles.

3.2.12 Real-world use-cases of ARIMA

ARIMA, one of the most extensively used econometrics models, "is used to forecast stock prices, demand, and even the spread of dangerous diseases. For studying the stock market or retail sales, it is usually better to utilize ARIMA or a related statistical model rather than complex deep algorithms like RNNs since the underlying mechanisms are unclear", too difficult, or not totally understood.

However, there are some circumstances when using ARIMA can produce acceptable results.

The selected articles that use ARIMA are provided below:

- An ARIMA Model Application to Predict COVID-19 Epidemic Dynamics in India.
- This research employed the ARIMA methodology to forecast the COVID-19 case count in India.
- The drawback of using ARIMA in this situation is that it only uses past data to predict the future. But unlike past values, COVID-19 fluctuates through time and depends on a range of additional behavioral features that ARIMA is unable to capture.
- A Case Study of Seoul, South Korea Using a Time Series ARIMA Model to Forecast Daily and Monthly Average Global Solar Radiation.
- The utilization of time series data is another example of how the ARIMA model is used in disease management, demonstrating its broad application. A few real-world use examples for ARIMA are mentioned in the research papers. For instance, during the SARS pandemic, A Singapore hospital precisely predicted the amount of beds they would require in 3 days.

- Forecasting of demand using the ARIMA model: This use case focuses on utilizing ARIMA to estimate and forecast demand in a food firm.

3.2.13 Conclusion of ARIMA

However, there are a few things to bear in mind while you work on them in your actual use case:

- The temporal complexity of training can rise exponentially when p, q is raised. So, it is advised to ascertain their values first, then investigate them.
- They have a tendency to over fit. So, be sure to configure the hyper settings accurately and do validation before transitioning to production.

3.3 SARIMAX

It may be thought of as an improved version of the ARIMA model. "ARIMA includes an autoregressive integrated moving average, whereas SARIMAX adds seasonal impacts and exogenous factors to the autoregressive and moving average components. As a result, like SARIMA and Auto ARIMA, SARIMAX is a seasonal equivalent model. SARIMAX may be expressed mathematically as an equation (3)" [42].

$$\varphi_p(B)\Phi_p(B^s)\nabla_S^D y_t = \beta_k x_{k,\grave{t}+\theta_q(B)\emptyset_Q(B^s)\varepsilon_t} \quad \textbf{(3.5)}$$

Where p, d, q and P, D, Q represent the polynomial order with non-negative integers, $x_{k,\grave{t}}$ represents the vector with the k^{th} input variable, β_k is the coefficient value of the k^{th} value of the exogenous input variable.

- SARIMAX "is an efficient variant of the ARIMA model (Seasonal Auto-Regressive Integrated Moving Average with eXogenous components)".
- SARIMAX is a seasonal equivalent model that is comparable to Auto ARIMA and SARIMA.
- With outside influences, it might also contract. This aspect of the vehicle varies from other models.
- The data can be measured as time-series data when it is indexed in a way that the data concepts are the size of variations occurring over time.

- For instance, a product's sales unit for a specific "day, week, month, or year", or a change in temperature over time.
- Thus, this is one of the key areas of data science where we anticipate future value by using historical data from time series.
- Many models exist that help us predict the future and the morals we will need to follow in order to attain our goals while meeting the demands of the situation.
- We only have two options if the records are not stationary: make the data stationary or fulfill the SARIMAX model.
- Alternative seasonal equivalent models can account for external impacts while maintaining the seasonal design. This aspect of the model is unique compared to other models.
- For instance, the temperature has cyclical impacts in a time series, being short in the winter and great in the summer.
- The temperature in winter is still higher as a result of external factors like moisture, and there is a potential that it will be colder owing to rain.
- If these problems don't exhibit cyclical or seasonal fulfill, we can't predict their exact value. Other copies are incapable of handling this handling of data.

We must fulfill two different types of orders based on SARIMAX model parameters.

- "This order is referred to as a seasonal order, and we are obliged to supply four numbers. The first is analogous to the ARIMAX model (p, d, and q), while the second is to illustrate the seasonality's impact.
- (Seasonal AR specification, Seasonal Integration order, Seasonal MA, Seasonal periodicity) (Seasonal AR specification, Seasonal Integration order, Seasonal MA, Seasonal periodicity) This is how the model can be represented."

mathematically. $\Phi_p(L)\Phi_p(L^s)\Delta^d\ \Delta_s^D y_t = A(t) + \theta_q(L)\theta Q(L^s)\varepsilon_t$ **(3.6)**

Where:

"$\boldsymbol{\Phi_p(L)}$ is the non-seasonal autoregressive lag polynomial.

$\boldsymbol{\Phi_p(L^s)}$ is the seasonal autoregressive lag polynomial.

$\boldsymbol{\Delta^d\ \Delta_s^D y_t}$ is the time series, differenced $\boldsymbol{d}$ times, and seasonally differenced $\boldsymbol{D}$ times.

$\boldsymbol{A(t)}$ is the trend polynomial (including the intercept).

$\boldsymbol{\theta_q(L)}$ is the non-seasonal moving average lag polynomial.

$\boldsymbol{\Theta Q(L^s)}$ is the seasonal moving average lag polynomial."

3.4 LSTM

The Long Short-Term Memory (LSTM) is a second-order RNN design that excels in storing and retrieving successive short-term memories across multiple time steps. The original LSTM training technique offers crucial spatial and temporal locality qualities that other training approaches lack, at the expense of restricting its application to a narrow range of network designs [43].

The Long Short-Term Memory (LSTM) architecture is known for its ability to acquire and depict intricate and robust representations of long-term dependencies present in input sequential data, which it achieves through a gating mechanism that regulates an internal memory cell. This attribute renders LSTM highly suitable for feature learning over a sequence of temporal data, thereby positioning it as an extension of the Recurrent Neural Network (RNN) model. [17].

i. Time series forecasting can be done using Long Short-Term Memory networks, or LSTMs.

ii. Each particular sort of time series forecasting problem can be solved with a different variety of LSTM models.

"Recurrent neural networks of the Long-Short-Term Memory (LSTM) type may learn order dependency in sequence prediction tasks. The current phase's input is taken from the results of the previous RNN step. Created by Hochreiter & Schmid Huber, the LSTM. It addressed the problem of long-term reliance of RNNs, this occurs when the RNN may predict words depending on the present input but not words stored in long-term memory. When the gap length rises, RNN performance

declines. The LSTM has the ability to retain data indefinitely by default. It is used for forecasting, categorization, and analyzing time-series data.

Unlike traditional feed-forward neural networks, LSTM has connections for feedback.

It can manage large data streams in addition to single data points (like photos) (such as speech or video)."

Two applications of LSTM are unsegmented, linked handwriting recognition and speech recognition.

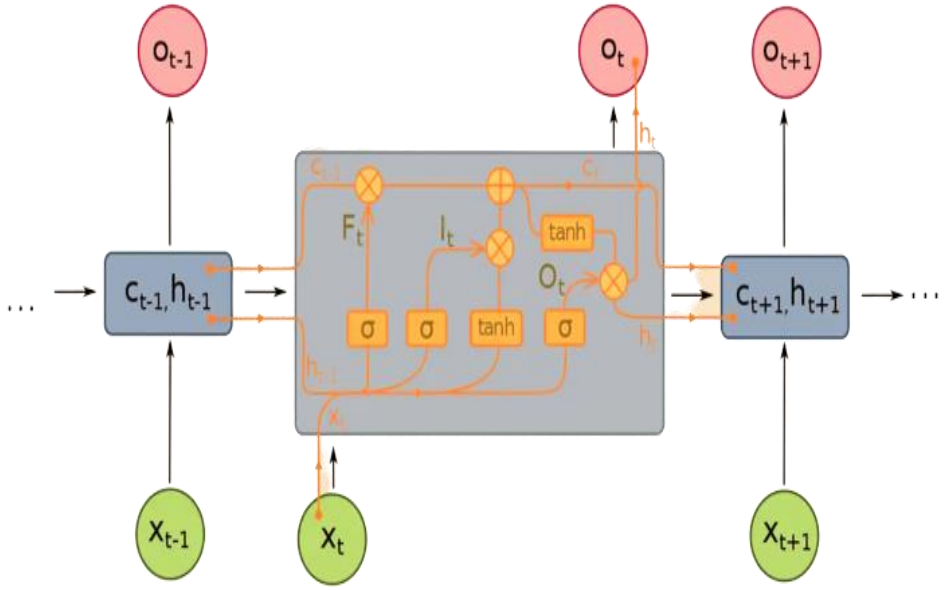


Figure 3.1: LSTM

3.4.1 LSTM structure

"The LSTM is made up of four neural networks and numerous memory cells that are linked together in a chain. An LSTM unit is made up of a cell, an input gate, an output gate, and a forget gate. Three gates govern the flow of information into and out of the cell, and the cell may store data for any period of time.

The LSTM (Long Short-Term Memory) algorithm is capable of categorizing, analyzing, and making predictions on time series data with an indefinite duration.

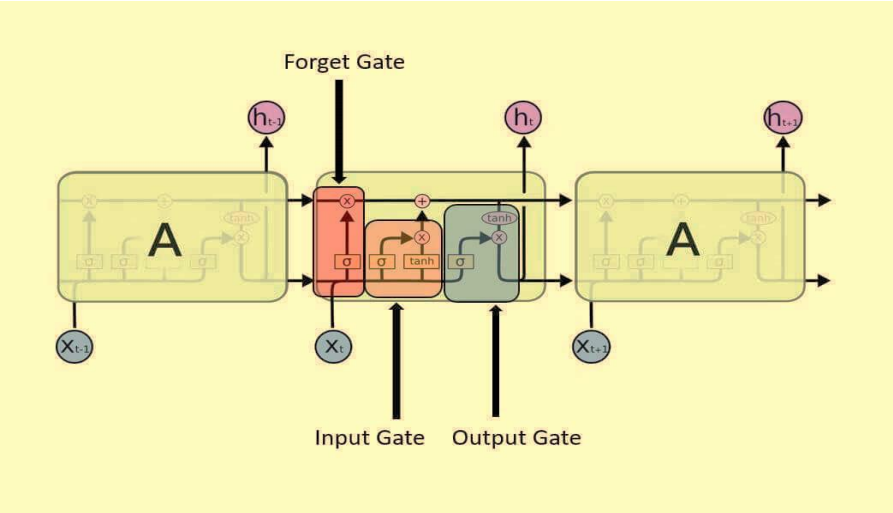


Figure 3.2: Structure of LSTM

The gates control memory, whereas the cells store information. Three entrances are present:

"Input Gate:"

It determines "which input values should be used to update the memory. The sigmoid function decides whether 0 or 1 data should be sent through. The tanh function also provides extra weight to the given data by ranking its relevance on a scale of -1 to 1."

$$\mathbf{i_t} = \sigma\,(W_i \mathbf{.}\, [h_t - 1, x_t] + b_i) \qquad \textbf{(3.7)}$$

$$\mathbf{C_t} = \tanh\,(W_C \mathbf{.}\, [h_t - 1, x_t] + b_c) \qquad \textbf{(3.8)}$$

"Forget Gate:"

"It identifies the data that has to be removed from the block. A sigmoid function determines the outcome.

It examines the prior state (ht-1), the content input (Xt), and each number in the cell state Ct-1 to produce a number between 0 (omit this), and 1. (keep this)."

$$\text{"}\mathbf{f_t} = \sigma\,(W_f \mathbf{.}\, [h_t - 1, x_t] + b_f)\text{"} \qquad \textbf{(3.9)}$$

Output Gate:

The resultant output is determined by the input as well as the memory held within the block. The sigmoid function is employed to determine the suitability of transmitting binary data, either 0 or 1. On the other hand, the tanh function plays a crucial role in

determining which integers can pass through the binary gate, using a Gaussian output to weigh input values and ascertain their relevance on a scale ranging from -1 to 1.

$$\mathbf{O_t} = \sigma\,(W_o \mathbf{.}\, [h_t - 1,\, x_t] + b_o) \qquad \textbf{(3.10)}$$

$$\mathbf{h_t} = O_t * \tanh\,(C_t) \qquad \textbf{(3.11)}$$

The recurrent neural network employs "long short-term memory blocks to provide context for how the software analyzes inputs and creates outputs. Because the program uses a framework based on short-term memory processes to generate longer-term memory, the unit is referred to as a long short-term memory block. Natural language processing makes extensive use of these systems."

3.4.1.1 LSTM networks

Recurrent- neural- networks are built from a set of repeating modules. Traditional RNNs frequently feature a basic structure for this repeating module, such as a single tanh layer.

Recurrent time steps are ones in which the previous time step's output is used as the input for the following time step.

In addition to considering the current input, the model also considers what it already knows about the inputs that came before. The repeating module in a standard RNN has " only one layer:

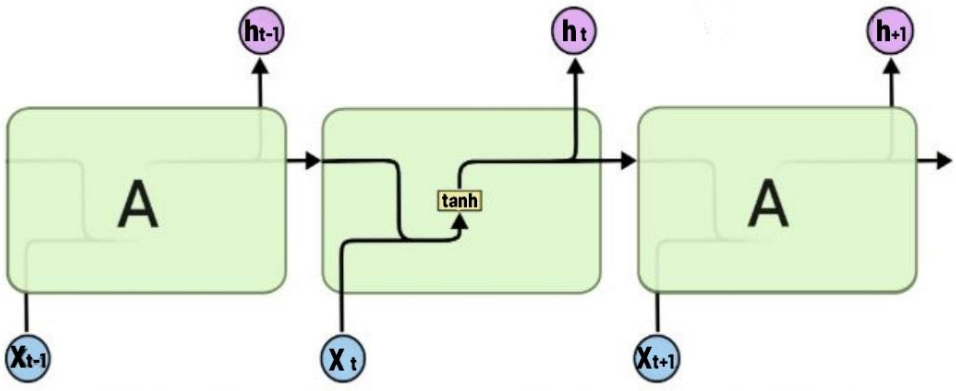


Figure 3.3: Typical RNN

The LSTM repeating module is made up of four linked layers:

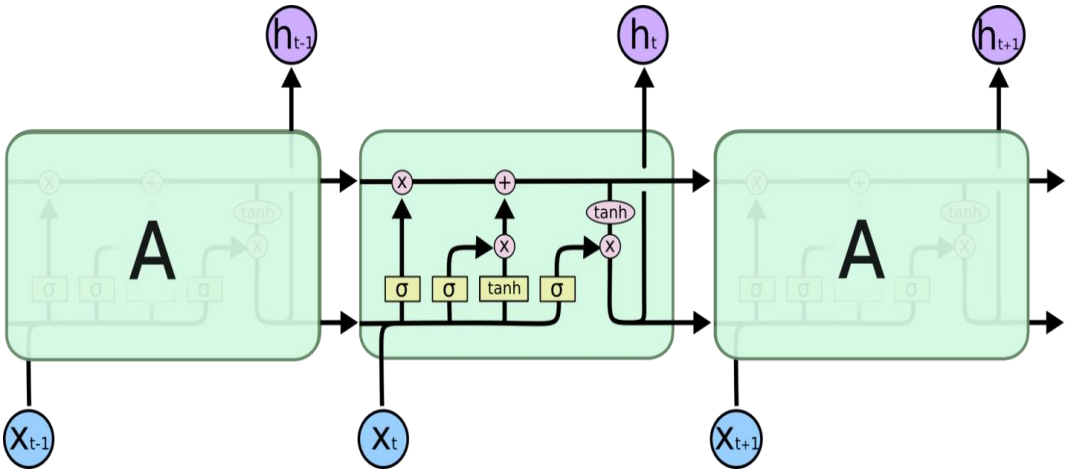


Figure 3.4: Overview on LSTM

- The cell state, which is important to LSTMs, is indicated by the horizontal line at the top of the diagram.
- The condition of the cell is somewhat comparable to that of a conveyor belt. Just a few brief linear exchanges take place while the signal moves through the whole chain. Data may just travel through it unmodified, which is a very easy process.
- The LSTM will change the cell state via removing or adding information. The cell state is methodically regulated by structures called gates.
- Information may be regulated as it passes through gates. They consist of a point-wise multiplication step and a layer of sigmoid neural networks.
- "The sigmoid layer generates values from 0 to 1 that represent the amount of each component that should be permitted to pass. When a value is zero, "nothing" should be permitted to pass, and when a value is one, "everything" should be let to pass."
- An LSTM contains three of these gates to safeguard and control the cell state.

3.4.1.2 Cycle of LSTM

Four steps make up the LSTM cycle:

- Information from a previous time step is used to identify what should be forgotten using the forget gate.

- New data is requested for updating cell state by use of tanh and input gate.
- Data from the two gates mentioned above are required to update the state of the cell.
- Information is usefully provided by the output gate and squashing procedure.

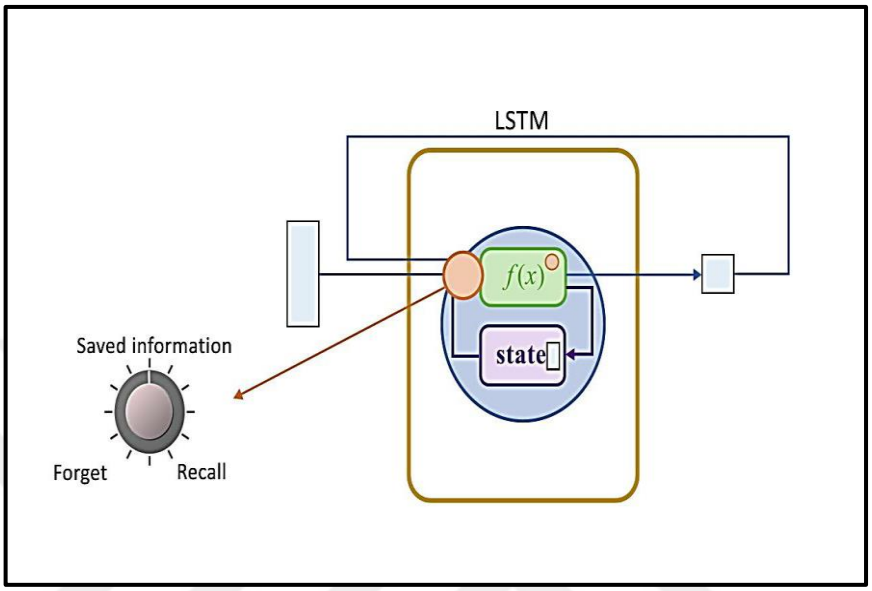


Figure 3.5: Cycle of LSTM

"An LSTM cell's output is sent to a dense layer.

After the dense layer, the output stage receives the SoftMax activation function."

3.4.1.3 Bidirectional LSTM

Each "training sequence is shown forward and backward to two independent recurrent nets that are coupled to the same output layer via bidirectional recurrent neural networks (BRNN).In other words, the BRNN is fully aware of every point in a given sequence that comes before and after it."

Furthermore, because the" internet can utilize as much or as little of this context as it wants, there is no need to provide a (task-dependent) time frame or desired delay size."

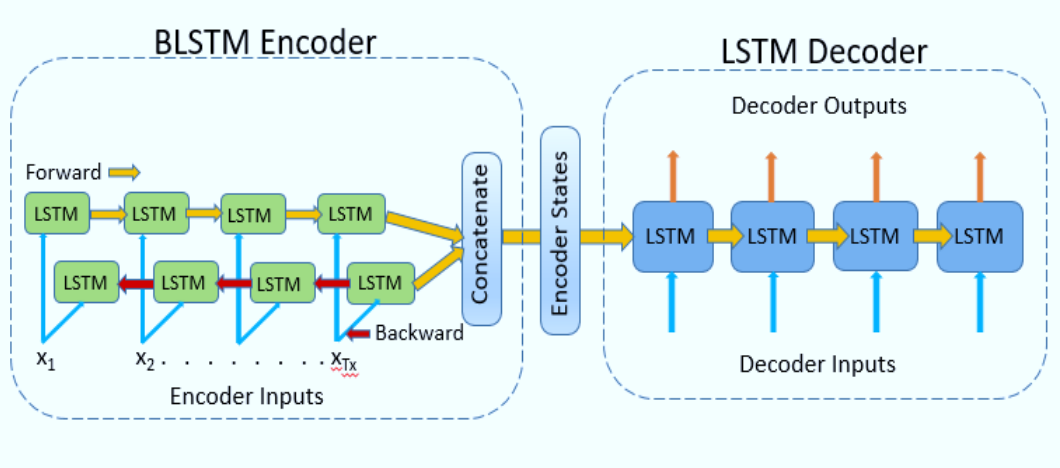


Figure 3.6: Bidirectional LSTM

The limitation of just being able to use the prior contexts is a drawback of conventional RNNs.

"Bidirectional RNNs (BRNNs) do this by processing data in both directions utilizing two hidden layers that feed-forward to the same output layer.

A bidirectional-LSTM When BRNN and LSTM are combined, a system that could also access long-range information in both input directions is developed."

3.4.2 LSTM models - multivariate

- Data with many comments for each time step are referred to as multivariate time series data.
- With multivariate time series data, we may need the following two main models:
- Input Series - Multiple
- Parallel Series - Multiple
- Let's examine each one separately.

3.4.3 LSTM advantage

i. "I. LSTMs provide us with a wide variety of parameters including learning rates and input and output biases. As a result, no precise modifications are required.

ii. II. LSTMs have the benefit of making updating each weight to (1) easier, which is analogous to BPTT (Back Propagation Through Time).

iii. The Long Short-Term Memory (LSTM) model, which is a type of recurrent neural network, has the ability to acquire knowledge of the sequential relationship among components.

iv. LSTMs carry the potential of being able to learn the context required to make predictions in time series forecasting issues, as opposed to having this context pre-specified and fixed."

3.4.4 Limitation of LSTM

i. "Memory is a significant drawback of LSTMs. Or, to be more precise, how memory may be misused.

ii. An LSTM model may be forced to recall a single observation over a very large number of input time steps.

iii. This is a bad use of LSTMs, and expecting an LSTM model to recall many observations will result in failure."

3.4.5 Applications of LSTM

- Robotic mechanism
- Prediction of time series
- recognition Speech
- Rhythm learning
- Composition of Music
- Learning Grammar
- Recognition of Handwriting
- Recognition of Human action
- Sign language translation
- Protein homology recognition
- Calculating subcellular localization of proteins
- Time series anomaly gratitude
- Several forecast tasks in the area of business process organization
- Prediction in medical care paths
- Semantic analyzing
- Thing co-segmentation

- Airport passenger organization
- Short-term traffic forecast
- Drug Strategy
- Market Forecast"

3.4.6 Conclusion of LSTM

i. "LSTM is a deep learning architecture based on an artificial recurrent neural network (RNN).

ii. LSTMs provide a viable solution for scenarios requiring sequences and time series.

iii. Its difficulties in training models—even basic ones—requires a significant amount of time and system resources, which is one of its drawbacks. Nevertheless, this is only a hardware limitation.

iv. A limitation of traditional RNNs is that they can only use earlier contexts.

v. Bidirectional Recurrent Neural Networks (BRNNs) achieve this feat by effectively processing data in both forward and backward directions."

3.5 Optimization Algorithms

Optimization is a field of mathematics that seeks to mathematically formulate, analyze, and resolve analytical or numerical challenges that involve minimizing or maximizing a function over a specific dataset. This thesis will elaborate on the optimization algorithms examined for the proposed model.

3.5.1 Adaptive moment algorithm (ADAM)

The Adam algorithm amalgamates the RMSProp and Momentum techniques and incorporates the computation of adaptive learning rates for each parameter. While AdaDelta and RMSprop employ an exponentially decaying average of the preceding gradient squares vt, Momentum follows suit, but Adam does not. [17]. The Adam algorithm equations are as follows [14]:

$$m \leftarrow \beta 1_m - (1 - \beta 1)\nabla_\theta J(\theta) \quad \textbf{(3.12)}$$

$$vt \leftarrow \beta 2_{vt} + (1 - \beta 1)\nabla_\theta J(\theta) \otimes \nabla_\theta J(\theta) \quad \textbf{(3.13)}$$

$$m \leftarrow m \otimes (1 - \beta_1^t) \quad \textbf{(3.14)}$$

$$vt = \frac{vt}{(1-\beta_1^t)} \quad \textbf{(3.15)}$$

$$\theta = \theta + \frac{\eta m}{\sqrt{vt+U}} \quad \textbf{(3.16)}$$

The structure of the ADAM algorithm is expressed as seen below.

***Input**: η, decay rate β1 and β2, small constant a (about 10^{-7}), initial θ.*

Initialize gradient accumulation variable $r \leftarrow 0$

$m_0 \leftarrow 0$; $v_0 \leftarrow 0$; $t \leftarrow 0$. (Initialize 1st moment, 2nd moment and time step)

***While** θ_t not converged do:*

$t \leftarrow t + 1$

Calculate gradients for step t: $g_t \leftarrow \nabla_\theta J_t(\theta_{t-1})$

Update biased 1st moment estimate: $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1).g_t$

Update biased 2nd raw moment estimate: $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1- \beta_2)\, g^2{}_t$

Calculate bias-corrected 1st moment estimate: $\hat{m}_t \leftarrow m_t / (1- \beta^t{}_1)$

Calculate bias-corrected 2nd moment estimate: $\hat{v}_t \leftarrow v_t / (1- \beta^t{}_2)$

Update parameters: $\theta t \leftarrow \theta_{t-1} - \eta.\hat{m}_t / (\sqrt{\hat{v}t} + \alpha)$ ADAM structure [14]

3.5.2 AdaMax algorithm

The Adam algorithm is extended by the AdaMax method using infinite norms. The factor vt in the update rule of the Adam algorithm adjusts the gradient inversely proportionate to the l_2 norm of previous gradients v_{t-1}, and the current gradient $|g_t|^2$[14]. the equations that illustrate the AdaMax procedure can be seen below:

$$vt \leftarrow \beta_2{}_{vt-1} + (1 - \beta 2)|g_t|^2 \quad \textbf{(3.17)}$$

$$vt \leftarrow \beta_2^p{}_{vt-1} + \left(1 - \beta_2^p\right)|g_t|^p \quad \textbf{(3.18)}$$

$$ut \leftarrow \beta_2^\infty{}_{vt-1} + (1 - \beta_2^\infty)|g_t|^\infty \quad \textbf{(3.19)}$$

$$ut \leftarrow \max(\beta_2 . v_{t-1}, |g_t|) \quad \textbf{(3.20)}$$

The structure of the AdaMax algorithm is expressed as seen in below.

***Input**: η, decay rate β1 and β2, small constant a (about 10^{-7}), initial θ.*

$m_0 \leftarrow 0$; $v_0 \leftarrow 0$; $t \leftarrow 0$. (Initialize 1st moment, 2nd moment and time step)

While *θ_t not converged do:*

$t \leftarrow t + 1$

Calculate gradients for step t: $g_t \leftarrow \nabla_\theta J_t(\theta_{t-1})$

Update biased 1st moment estimate: $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1).g_t$

Update exponentially weighted infinity norm: $u_t \leftarrow \max(\beta_2. u_{t-1}, |g_t|)$

Update parameters: $\theta_t \leftarrow \theta_{t-1} (\eta / (1- \beta^t_1)). m_t / u_t$

AdaMax Structure [14]

3.5.3 Nadam algorithm

Adam and Nesterov-accelerated algorithms are combined in the Nesterov-accelerated Adaptive Moment (Nadam) algorithm [18]. This method tries to gradually raise and lower the decay factor. For clarity, a sequence of parameters β_1 , β_1 ,..., β_t corresponding to steps 1, 2,..., t are performed. The momentum step in step t+1 is applied once by updating step t instead of t+1 as represented in the equations follows [14]:

$$g_t \leftarrow \nabla_{\theta_t - 1} j_t(\theta_{t-1}) \quad \textbf{(3.21)}$$

$$m_t \leftarrow \beta_t m_{t-1} + \eta g_t \quad \textbf{(3.22)}$$

$$\theta_t \leftarrow \theta_{t-1} - (\beta_{t+1} m_t + \eta g_t) \quad \textbf{(3.23)}$$

The steps related to the gradient and momentum depend on the current gradient as expressed in the following equations [14].

$$\theta_t \leftarrow \theta_{t-1} - \eta \left(\frac{\beta_t m_{t-1}}{1 - \prod_{i=1}^{t} \beta_t} + \frac{(1-\beta_t) g_t}{1 - \prod_{i=1}^{t} \beta_t} \right) \quad \textbf{(3.24)}$$

$$\theta_t \leftarrow \theta_{t-1} - \eta \left(\frac{\beta_{t+1} m_t}{1 - \prod_{i=1}^{t+1} \beta_t} + \frac{(1-\beta_t) g_t}{1 - \prod_{i=1}^{t} \beta_t} \right) \quad \textbf{(3.25)}$$

The Nadam algorithm's groundwork is discussed further below.

*"**Input**: learning rates $\eta_1, \eta_2, ..., \eta_t$; decay rates $\beta_1, \beta_2, ..., \beta_t$; small constant ε; initial θ;*

hyperparameter v. "

$m_0 \leftarrow 0$; no $\leftarrow 0$. (Initialize 1st and 2nd moments)

While θ_t, not converged do:

Calculate gradients for step t: $g_t \leftarrow \nabla_{\theta t-1} J_t(\theta_{t-1})$

Update biased 1st moment estimate: $m_t \leftarrow \beta_t m_{t-1} + (1-\beta_t) g_t$

Update biased 2nd moment estimate: $n_t \leftarrow v n_{t-1} + (1-v) g^2_t$

$$\hat{m} \leftarrow \left[\beta_{t+1} m_t / [1 - \prod_{i=1}^{t+1} \beta_i]\right] + \left[(1 - \beta_t) g_t / [1 - \prod_{i=1}^{t} \beta_i]\right] \quad \textbf{(3.26)}$$

$$\hat{n} \leftarrow v n_t / (1 - v^t)$$

Update parameters: $\theta_t \leftarrow \theta_{t-1} - \frac{\eta t}{\sqrt{\hat{n} t + \acute{U}}} \hat{m}_t$

Nadam structure [14]

3.6 IoT

The Internet of Things (IoT) links common objects with devices, sensors, software, and networks. Things: Sensor-equipped objects may gather and transfer data via the network and collaborate. A refrigerator's sensor collects outside temperature data and adjusts its temperature to match. In healthcare, energy, and environmental research, the IoT is crucial. IoT sensors on home appliances measure energy and temperature in real-time. The IoT platform monitors and controls business and residential energy usage, devices, and sources. IoT optimization relies on extensive analysis that most businesses and individuals lack [22].

3.7 Anomaly Detection

Is the process of analyzing data to find data points that don't fit with the pattern of standard data. "The experimental results show that it is possible to utilize energy consumption data to identify change points in the usage trend of energy consumption. By tracking changes in consumption patterns, it has been objectively demonstrated and reported that energy usage increases in months with high consumption and decreases in months where demand is anticipated to be lower." The moving average is the simplest technique to find anomalies in time-series data, and points that depart from the moving average are termed anomalies [22].

4. FRAMEWORK AND DATASET

4.1 Proposed Framework

The proposed framework of this thesis for anomaly detection and smart home consumption-based ML will present in this section. The steps of the proposed framework can be seen in Fig. 4.1

First, the data is collected from a dataset available on the Kaggle website and known as (the smart home dataset with weather information), which will include information about the energy consumption related to different appliances that were gathered from sensors and IoT based devices. "

The dataset appliance consumption selected from the utilized dataset with its related values can be seen in Table 4.1 In addition, the different environmental data were obtained from the requested dataset and selected to be included for the performance evaluation of the proposed framework and as listed in Table 4.2 Thereby, achieving a higher impact of anomaly detection and reducing overall consumption. After that, the data is handled by preprocessing eliminating steps to maintain the required column to guarantee the right process of ML. Then data is modified with extraction and feature selection.

The next step is to design the proposed framework based on utilizing the LSTM algorithm with 50 layers and using the Rectified Linear Activation Function (ReLU), which outputs the positive input directly, else output zero. It's the default activation function for many neural networks since it's simpler to train and performs better. Additionally, two statistical modeling of ARIMA and SARIMAX for effective time series forecasting with different orders are included.

In LSTM, the major enhancement that has been considered to improve the model performance, which is the Dropout and set to the value of 0.001. Dropout regularizes LSTM units by probabilistically excluding input and recurrent connections from activation and weight changes during training. Overfitting decreases and model performance improves.

Moreover, three optimization method has been included for performance evaluation, which is Adam, AdaMax, and Nadam with a selected learning rate of 0.001 as a tuning parameter to determine the size of step-taking in optimization algorithms.

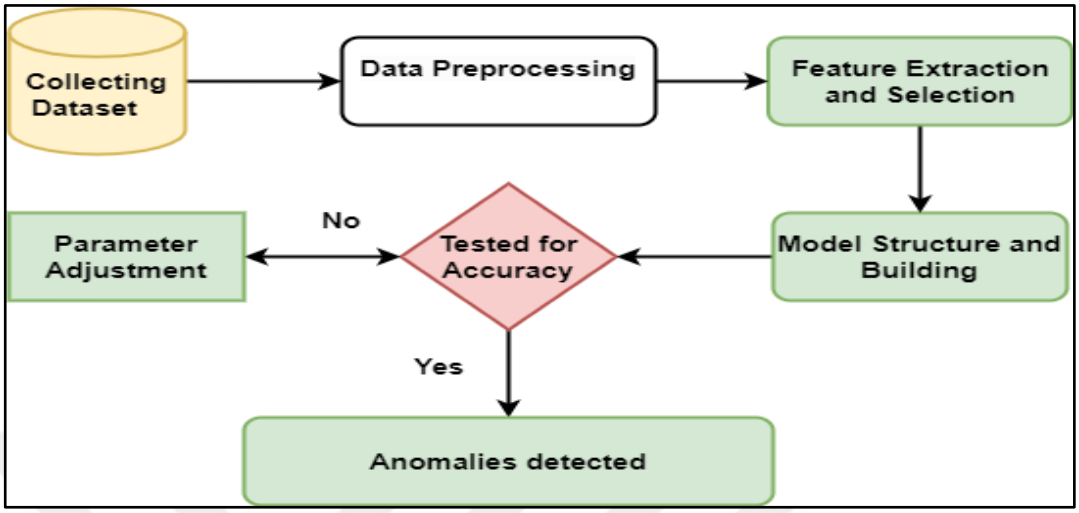


Figure 4.1: Flowchart Steps of the Proposed Framework

Table 4.1: Sample Dataset Collected for the Proposed Framework

time	House overall	Dishwasher	Home office	Fridge	Garage door	Barn	Well	Microwave	Living room	Furnace	Kitchen	Solar
2016-01-01 05:00:00	0.932833	0.000033	0.442633	0.12415	0.013083	0.03135	0.001017	0.004067	0.001517	0.020700	0.000417	0.003483
2016-01-01 05:01:00	0.934333	0.000000	0.444067	0.12400	0.013117	0.03150	0.001017	0.004067	0.001650	0.020717	0.000417	0.003467

Table 4.2: The Utilized Environmental Data from the Selected Dataset

temperature	humidity	visibility	apparentTemperature	pressure	windSpeed	cloudCover	windBearing	precipIntensity	dewPoint	precipProbability	month	day	weekday	hour	minute
36.14	0.62	10.0	29.26	1016.91	9.18	0.75	282.0	0.0	24.4	0.0	1	1	Friday	5	0
36.14	0.62	10.0	29.26	1016.91	9.18	0.75	282.0	0.0	24.4	0.0	1	1	Friday	5	1

4.2 Smart Home Dataset with Weather Information

"The dataset for this Case Study is named Smart Home Dataset with Weather Information, and it was obtained from Kaggle. The dataset contains 32 columns and

over 500,000 measurements with a time span of one minute of the energy consumption (in kW) by smart home appliances and the meteorological conditions in that location at that time.

To generate more accurate findings, 25 columns were used after 7 redundant and unneeded columns were removed" [44].

4.2.1 Column analysis

"From January 1st, 2016 (5 a.m.) until December 16th, 2016, this dataset captured the energy use of each room/appliance every minute (3:29 am). We divided the Smart Home Dataset into subsets in order to work with it: Weather and energy data."

4.2.1.1 Energy data

Contains all information on how much energy is used by various rooms and equipment, such as the kitchen, furnace, home office, etc. All energy is rated in KW.

- The energy use of each room in the house is shown in KW in the following columns: living room, kitchen, and home office.
- Fridge, Barn, Dishwasher, Microwave, Furnace, Garage Door, Well - These are the energy meter readings for the specific household appliances utilized on a given day.
- Generator, Home overall - These columns show the total amount of energy used by all the appliances together with the total amount of energy the generator produced on a given day."

4.2.1.2 Weather data

- Contains all information about the weather, such as the temperature, humidity, precipitation, dew point, visibility, etc.
- Temperature, Humidity, and Apparent Temperature - These represent the location's actual daytime temperature and humidity readings.
- Visibility, Dew Point, Wind Bearing, Wind Speed, and Pressure - They include the wind speed and other climatic parameters at that location at various scales" [44]. The steps of the Dividing a Smart Home dataset into Energy data and Weather can be seen in Fig. 4.2

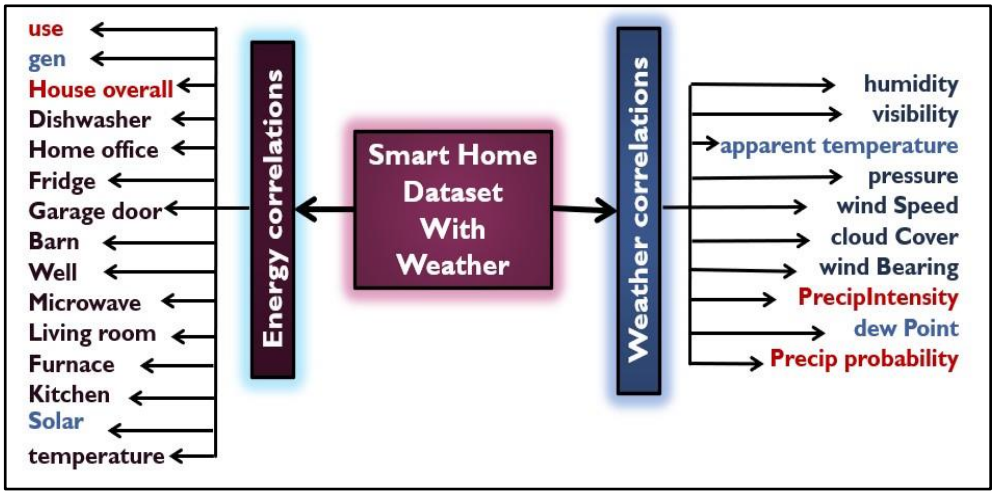


Figure 4.2: Dividing a Smart Home dataset into Energy data and Weather

5. RESULTS AND DISCUSSIONS

This section will demonstrate the results of the proposed framework for anomaly detection-based IoT energy consumption devices.

5.1 Results-Based Anomaly Detection

For the results of utilizing the LSTM algorithm with the three mentioned previously optimization methods for anomaly detection. The smart home dataset studied uses testing and training stages which indicate simple overfitting occurs at the beginning of the curves, while when moving towards the curves gets more stability Green indicates model training, red color is model testing, and data is blue color, as seen in Fig. 5.1, this may confirm the reliability of the proposed framework for testing and training stages. To quantify the achieved stability, the training loss is obtained as seen in Fig. 5.2, which gives a measure of how well a deep learning model matches training data.

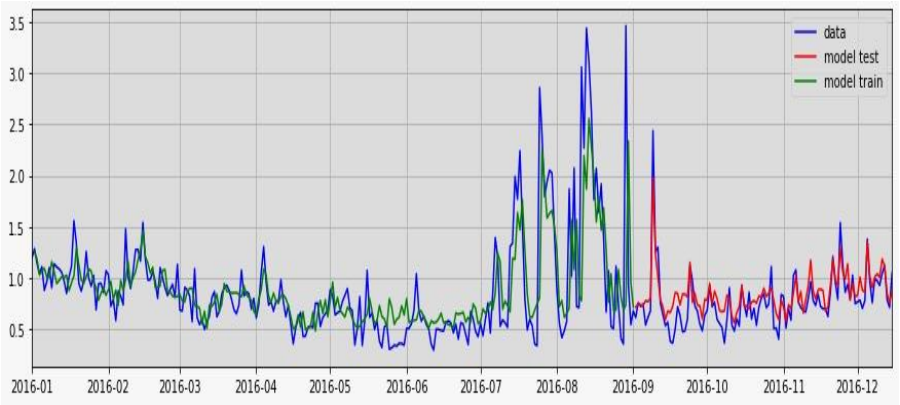


Figure 5.1: Proposed Framework Data Testing and Training

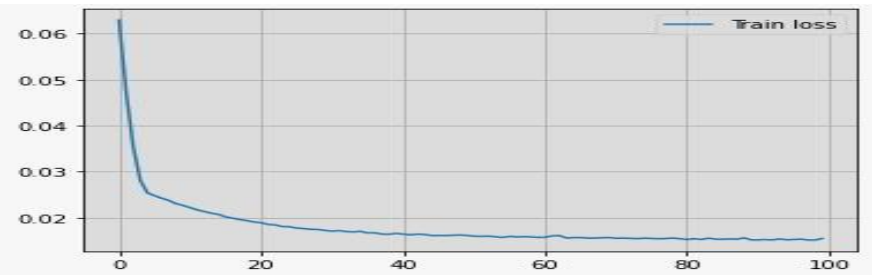


Figure 5.2: Train Loss for the Proposed Model

In addition, using the moving average for data trends can be defined as the simplest way to detect anomalies in time series forecasting by using the proposed framework. As a result, the (LSTM with Nadar) gives the best, where MSE and RMSE achieved 0.02576 and 0.16049 overall results for achieving the best results concerning the studied parameters as listed in Table 5.1

The green color indicates the Rolling mean trend, the orange color is the upper and the lower bound, and the blue color the actual values.

but regarding the Anomaly The red dots, which peak in August month, gradually decreases, and then fade away, as the extent of its effect is from the end of the middle of July to the beginning of the middle of September, as shown in Figure 5.3.

Table 5.1: Results of the Proposed Framework

	Optimizer	**MSE**	**RMSE**	**MAE**	**MAPE**	**R^2**
LSTM	Adam	0.02576	0.16049	0.129	0.197	0.655
	Adamax	0.02416	0.15545	0.122	0.181	0.677
	Nadam	**0.02356**	**0.15348**	**0.122**	**0.187**	**0.685**

Figure 5.3: Anomaly Detection with Moving Average

For the results of utilizing the statistical models that were included with the proposed model for performance evaluation. The results using ARIMA in order (1,1,2) gives the best results with AIC and MSE values of 300.944 and 0.13859 as compared to the utilized SARIMAX model. These results were based on the evaluation of the AIC and MSE for different orders as seen in Table 5.2

The green color indicates the Model, the orange color the upper and the lower bound, and the blue color the Data House overall.

but regarding the Anomaly The red dots, which peak in August month, gradually decreases, and then fade away, as the extent of its effect is from the end of the middle of July to the beginning of the middle of September, as shown in Figure 5.4.

Table 5.2: Results of the Proposed Statistical Models

Algorithm	Order (p, d, q)	AIC	MSE
ARIMA	**(1,1,2)**	**300.944**	**0.13859**
ARIMA	(2,1,1)	300.98	0.13861
SARIMAX	(2,1,1)	305.247	0.13536
SARIMAX	(1,1,2)	307.032	0.13615

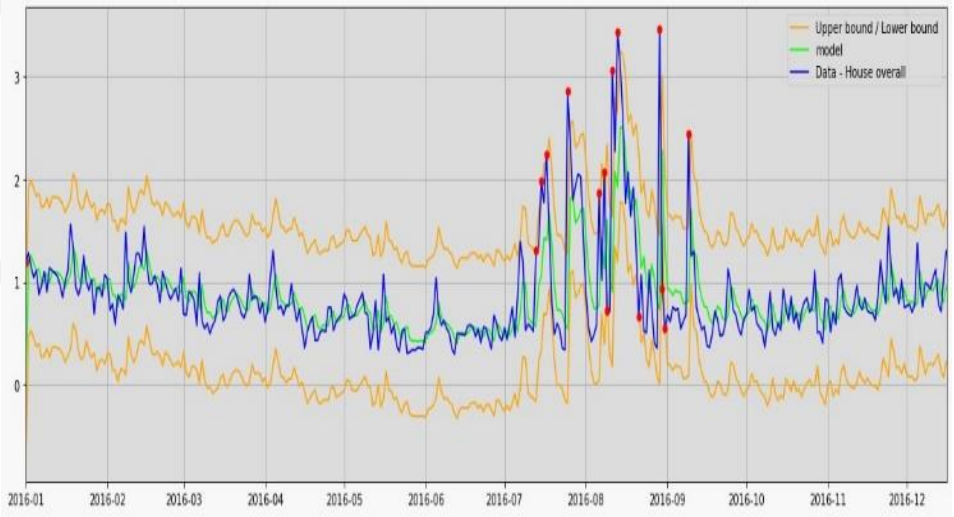


Figure 5.4: ARIMA Statistical Model with Moving Average

The results of correlation analysis for devices information in the Energy data can be seen in Matrix values in Fig. 5.5, it was found that there is a highest correlation between the (use) and (house overall), as well as between (generation) with (solar).

While the correlation between (Furnace) with (use & house overall) of about (0.31).

In addition, the results of correlation analysis for Weather information can be seen in Fig. 5.6, it was found that there is the highest correlation between (Apparent temperature) and (dew Point), also (Precip Probability) with (Precip Intensity).

While the is a correlation between (Precip Probability) and (cloud cover) of about (0.48).

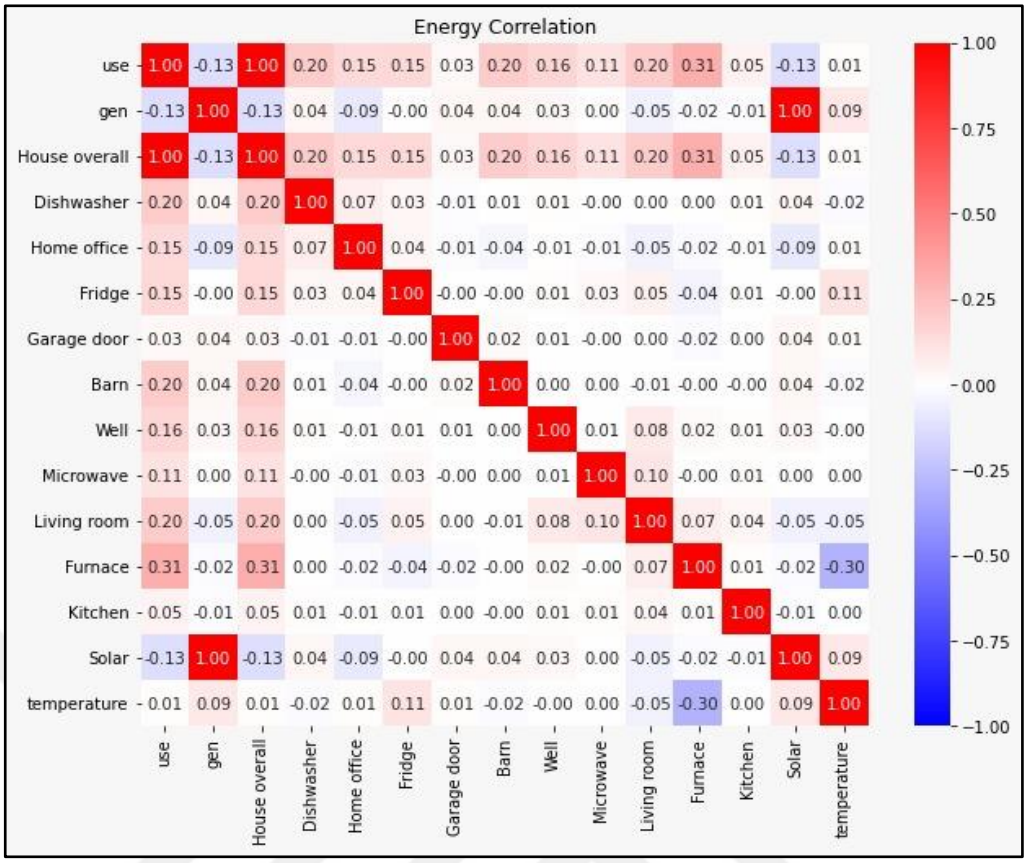


Figure 5.5: Correlation Analysis of Devices

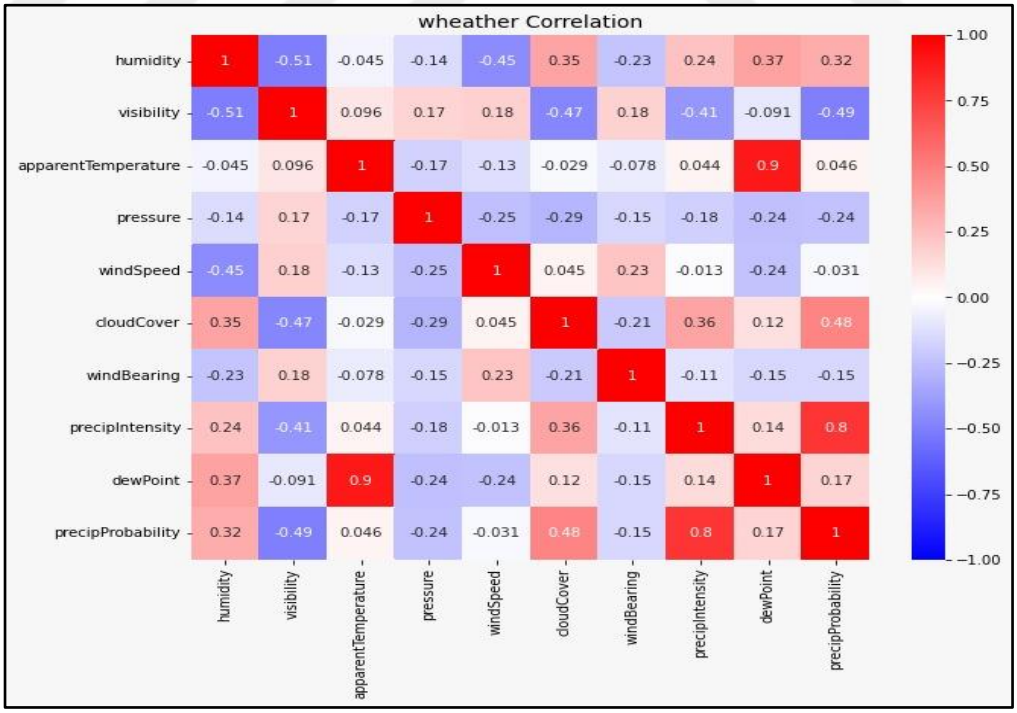


Figure 5.6: Correlation Information of Weather

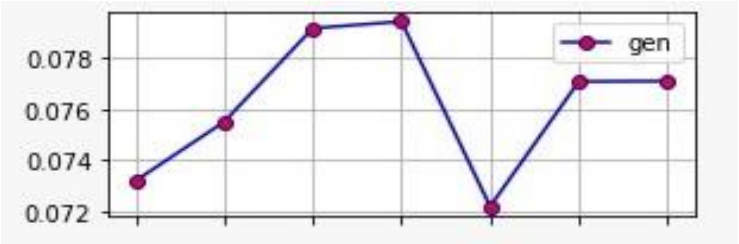
5.2 Results-Based Data Visualization

The visualization of the utilized data can be represented based on the different parameters within the used dataset. For instance, energy generation and consumption will be selected to be analyzed for three aspects weekdays, hours, and months.

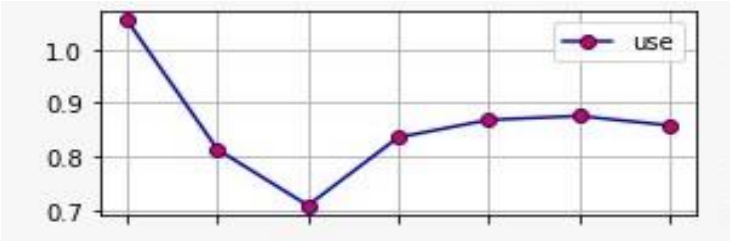
The visualization of the utilized data can be represented based on the different parameters within the used dataset. For instance, energy generation and consumption will be selected to be analyzed for three aspects weekdays, hours, and months.

5.2.1 Visualization of the weekdays

- For the visualization, which represents the time series over the weekdays as expressed in Fig. 5.7 it can be seen that the height peak of generation was between (Wed and Thu). Meanwhile, the generation was lowest on (Fri).
- While it can be seen that the height peak of consumption was (Mon). Meanwhile, the consumption was lowest on (Wed), with constant consumption during the remaining weekdays in the middle.
- It was clearly shown that (Mon) was the highest consumption and the lowest generation, and this is a problem, while the best day was (Wed), which was characterized by the highest generation and the lowest consumption.



(a)



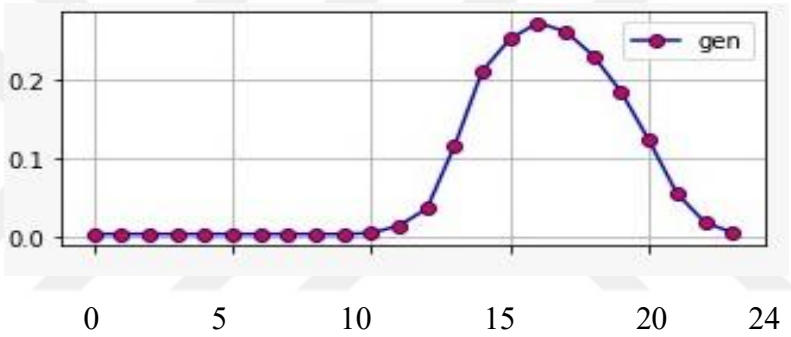
(b)

Figure 5.7: Time Series by Weekday for Total Energy (a) Gen. & (b) Consumption

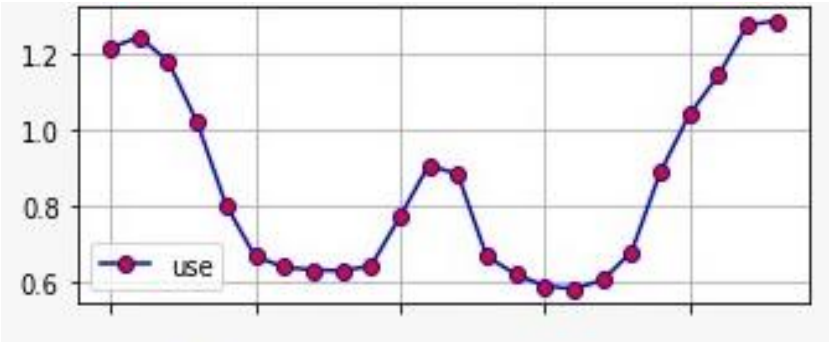
5.2.2 Visualization of the hours

- For the visualization of hours as expressed in Fig. 5.8, the generation was lowest during the first half of the day in (the morning) and it is stable, and the highest generation was during the second half of the day in (the evening) and it is gradual.
- It can be seen that the height peak of consumption was between (22 to 24). Meanwhile, the consumption was lowest from (05 to 09) & (13 to 17).
- The problem is the height peak of consumption and lowest for the generation at (midnight). as well as the height peak of generation and lowest for the consumption at an hour (17).

Also, hour visualization could give a better understanding of the energy behavior.



(a)



(b)

Figure 5.8: Time Series by Hour for Total Energy (a) Generation and (b) Consumption

5.2.3 Visualization of the year months

For the last visualization as expressed in Fig. 5.9, which represent the time series over the year months it can be seen that the height peak of consumption was between July and august. Meanwhile, the generation peak gradually raised from February to September.

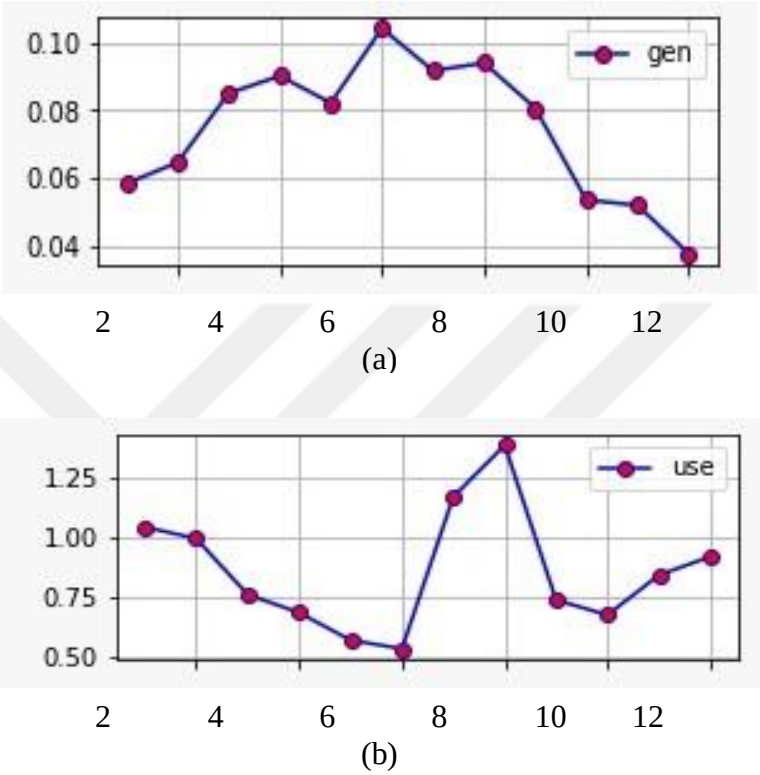


Figure 5.9: Time Series by Months for Total Energy (a) Generation and (b) Consumption

6. CONCLUSION AND RECOMMENDTIONS

6.1 Conclusions

This thesis introduces an optimization framework designed to detect anomalies in energy consumption among sensors and devices utilized in smart homes that are based on Internet of Things (IoT) technology. The proposed model designed and implemented based on using the powerful LSTM algorithm for time series forecasting approaches and utilize two statistical models of ARIMA and SARIMAX to achieve the optimal model. Additionally, three optimization algorithms have been selected for performance evaluation. Results obtained indicate that using LSTM with Nadam optimization gives the best overall results based on the studied parameter, where MSE and RMSE achieved 0.15348 and 0.02356 respectively. Also, using ARIMA model gives the best overall results with AIC and MSE values of 0.13859 and 300.94365 respectively. This confirms the reliability and adaptability of the proposed model in optimizing anomaly detection and reducing the consumption cost of electricity.

6.2 Why ARIMA is best

- It is "often used in demand forecasting, such as when predicting upcoming demand in the food production industry.
- This is because the model provides managers with solid guidance for making supply chain-related decisions.
- You can use ARIMA models to predict the price of your stocks based on past prices."

6.3 The Future Artificial Intelligence (AI) in Smart Homes

Analysts estimate that there will be more than 300 million "smart" residential houses in the world by 2023.

With the "growth of the market for intelligent homes, new security risks are predicted to materialize. Remotely connected devices are more frequently the target of cyber-attacks.

Protecting connected devices from security threats and vulnerabilities is essential to win consumers' trust and increase the sale of smart home equipment".

For example, in 2016, the Mirai IoT botnet succeeded in controlling nearly 600,000 smart home devices globally, including air quality sensors, routers, and security cameras. Services for websites like Twitter and Netflix were suspended as a result of this significant online traffic redirection.

However, businesses are seeking to integrate emotions and artificial intelligence. Clio, a robot from LG, is cheerful, whereas Aibo, Sony's next-generation robotic dog, is unique in personality and feeling.

With the help of EmoShape's Emotion Processing, personal assistants and avatars can convey 12 various emotions, including, among others, happiness, displeasure, pain, and contentment.

"The Emotion Processing unit can modify the facial expression and body language of a robot or an avatar on a desktop screen."

Google created multilingual capabilities to let the Google Assistant speak and understand numerous languages at once.

This made it easier for the Assistant to understand family members who were bilingual.

As a result of advancements in speech recognition technology, the Assistant can now understand two languages at once.

Google developed multilingual features so that the Google Assistant could simultaneously understand and speak multiple languages.

As a result, the Assistant could comprehend family members who spoke different languages better. Thanks to breakthroughs in speech recognition technology, the Assistant is now capable of understanding two languages simultaneously.

REFERENCES

[1] **Himeur, Yassine, (2020).** Robust event-based non-intrusive appliance recognition using multi-scale wavelet packet tree and ensemble bagging tree. Applied Energy, 267: 114877.

[2] **Sohani, Ali, et al**. (2022).Using machine learning in photovoltaics to create smarter and cleaner energy generation systems: A comprehensive review. Journal of Cleaner Production, 132701.

[3] **Liu, Yu, et al.** (2016). Non-intrusive energy use monitoring for a group of electrical appliances. IEEE Transactions on Smart Grid, 9.4: 3801-3810.

[4] **Zanella, A., et al.** (2014). Internet of things for smart cities. IEEE Internet Things J 1 (1): 22–32.

[5] **Mosavi, Amir; Bahmani, Abdullah.** (2019). Energy consumption prediction using machine learning; a review.

[6] **Alsalemi, Abdullah, et al.** (2020). Achieving domestic energy efficiency using micro-moments and intelligent recommendations. IEEE Access, 8: 15047-15055.

[7] **Sardianos, Christos, et al.** (2020). A model for predicting room occupancy based on motion sensor data. In: 2020 IEEE international conference on informatics, IoT, and enabling technologies (ICIoT). IEEE, p. 394-399.

[8] **Sardianos, Christos, et al.** (2020). Rehab-c: Recommendations for energy habits change. Future Generation Computer Systems, 112: 394-407.

[9] **Zhao, Guilin; Xing, Liudong.** (2020). Reliability analysis of IoT systems with competitions from cascading probabilistic function dependence. Reliability Engineering & System Safety, 198: 106812.

[10] **Zekić-Sušac, Marijana; Mitrović, Saša; Has, Adela.** (2021). Machine learning based system for managing energy efficiency of public sector as an approach towards smart cities. International journal of information management, 58: 102074.

[11] **Essiet, Ima O.; Sun, Yanxia; Wang, Zenghui.** (2019). Optimized energy consumption model for smart home using improved differential evolution algorithm. Energy, 172: 354-365.

[12] **Hartman, Wesley Tyler, et al.** (2018). Energy monitoring and control using Internet of Things (IoT) system. In: 2018 Systems and Information Engineering Design Symposium (SIEDS). IEEE,. p. 13-18.

[13] **Luo, Yuansheng; LI, Wenjia; Qiu, Shi.** (2019). Anomaly detection based latency-aware energy consumption optimization for IoT data-flow services. Sensors, 20.1: 122.

[14] **Mustapha, Aatila; Mohamed, Lachgar; Ali, Kartit.** (2021). Comparative study of optimization techniques in deep learning: Application in the ophthalmology field. In: Journal of Physics: Conference Series. IOP Publishing. p. 012002.

[15] **Chou, Jui-Sheng; Telaga, Abdi Suryadinata.** (2014). Real-time detection of anomalous power consumption. Renewable and Sustainable Energy Reviews, 33: 400-411.

[16] **Lin, Guanjing; Claridge, David E.** (2015). A temperature-based approach to detect abnormal building energy consumption. Energy and Buildings, 93: 110-118.

[17] **Liu, Jun, et al.** (2016). Spatio-temporal lstm with trust gates for 3d human action recognition. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14. Springer International Publishing, p. 816-833.

[18] **Buzau, Madalina-Mihaela, et al**. (2019). Hybrid deep neural networks for detection of non-technical losses in electricity smart meters. IEEE Transactions on Power Systems, 35.2: 1254-1263.

[19] **Himeur, Yassine, et al.** (2020). A novel approach for detecting anomalous energy consumption based on micro-moments and deep neural networks. Cognitive Computation, 12: 1381-1401.

[20] **Feng, Longji, et al.** (2020). Anomaly detection for electricity consumption in cloud computing: framework, methods, applications, and challenges. EURASIP Journal on Wireless Communications and Networking, 1: 1-12.

[21] **Himeur, Yassine, et al.** (2021). Smart power consumption abnormality detection in buildings using micromoments and improved K-nearest neighbors. International Journal of Intelligent Systems, 36.6: 2865-2894.

[22] **Malki, Amer; Atlam, El-Sayed; Gad, Ibrahim.** (2022). Machine learning approach of detecting anomalies and forecasting time-series of IoT devices. Alexandria Engineering Journal, 61.11: 8973-8986.

[23] **Xiao, Guangyi, et al.** (2014). User interoperability with heterogeneous IoT devices through transformation. IEEE Transactions on Industrial Informatics, 10.2: 1486-1496.

[24] **Chettri, Lalit; Bera, Rabindranath.** (2019). A comprehensive survey on Internet of Things (IoT) toward 5G wireless systems. IEEE Internet of Things Journal, 7.1: 16-32.

[25] **Viani, Federico, et al**. (2013). Wireless architectures for heterogeneous sensing in smart home applications: Concepts and real implementation. Proceedings of the IEEE, 101.11: 2381-2396.

[26] **Vardakis, George, et al.** (2022). Smart Home: Deep Learning as a Method for Machine Learning in Recognition of Face, Silhouette and Human Activity in the Service of a Safe Home. Electronics, 11.10: 1622.

[27] **Chollet, Francois**. (2021). Deep learning with Python. Simon and Schuster.

[28] **Liu, Li, et al.** (2020). Deep learning for generic object detection: A survey. International journal of computer vision, 128: 261-318.

[29] **Ngiam, Jiquan, et al.** (2011). Multimodal deep learning. In: Proceedings of the 28th international conference on machine learning (ICML-11). p. 689-696.

[30] **Vardakis, George, et al.** (2022). Smart Home: Deep Learning as a Method for Machine Learning in Recognition of Face, Silhouette and Human Activity in the Service of a Safe Home. Electronics, 11.10: 1622.

[31] **Zainab, Ameema; S. Refaat, Shady; Bouhali, Othmane**. (2020). Ensemble-based spam detection in smart home IoT devices time series data using machine learning techniques. Information, 11.7: 344.

[32] **Farsi, Mohammed, et al.** (2021). Parallel genetic algorithms for optimizing the SARIMA model for better forecasting of the NCDC weather data. Alexandria Engineering Journal, 60.1: 1299-1316.

[33] **Liu, Yu, et al.** (2020). Anomaly detection based on machine learning in IoT-based vertical plant wall for indoor climate control. Building and Environment, 183: 107212.

[34] **Munir, Mohsin, et al**. (2019). FuseAD: unsupervised anomaly detection in streaming sensors data by fusing statistical and deep learning models. Sensors, 19.11: 2451.

[35] **Rahman, Saifur.** (2017). Energy efficiency and the role of IoT in a smart city connected community. In: 2017 20th International Conference of Computer and Information Technology (ICCIT). IEEE, p. 1-1.

[36] **Box, George EP; Jenkins, Gwilym M**. (1976). Time series analysis: Forecasting and control San Francisco. Calif: Holden-Day.

[37] **Hyndman, Rob J.; Athanasopoulos, George.** (2018). Forecasting: principles and practice. OTexts.

[38] **Ampountolas, Apostolos**. (2021). Modeling and forecasting daily hotel demand: A comparison based on sarimax, neural networks, and garch models. Forecasting, 3.3: 580-595.

[39] **Yang, Lintao; Yang, Honggeng.** (2019). A Combined ARIMA-PPR Model for Short-Term Load Forecasting. In: 2019 IEEE Innovative Smart Grid Technologies-Asia (ISGT Asia). IEEE,. p. 3363-3367.

[40] **Farsi, Mohammed, et al.** (2021). Parallel genetic algorithms for optimizing the SARIMA model for better forecasting of the NCDC weather data. Alexandria Engineering Journal, 60.1: 1299-1316.

[41] **Wu, Dongmei; Zhang, Li; Lin, Lihua**. (2018). Based on the moving average and target motion information for detection of weak small target. In: 2018 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS). IEEE, p. 641-644.

[42] **Vagropoulos, Stylianos I., et al.** (2016). Comparison of SARIMAX, SARIMA, modified SARIMA and ANN-based models for short-term PV generation forecasting. In: 2016 IEEE international energy conference (ENERGYCON). IEEE, p. 1-6.

[43] **Monner, Derek; Reggia, James A.** (2012). A generalized LSTM-like training algorithm for second-order recurrent neural networks. Neural Networks, 25: 70-83.

[44] **Smart Home Dataset with weather Information.** (n.d.). Www.kaggle.com. https://www.kaggle.com/datasets/taranvee/smart-home-dataset-with-weather-information

APPENDIX

Python Software Code

Appendix-A: Import Dataset & Preprocessing

```
#import dataset

# connect with drive
from google.colab import drive
drive.mount('/content/drive')

# copy the data from drive to colab content
import shutil
shutil.copy("/content/drive/MyDrive/Datasets/HomeC.csv", "/content/")
data = pd.read_csv("/content/HomeC.csv",low_memory=False)
#delete last row (NaNs)
data = data[:-1]
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 503910 entries, 0 to 503909
Data columns (total 32 columns):
 #   Column                 Non-Null Count     Dtype
---  ------                 --------------     -----
 0   time                   503910 non-null  object
 1   use [kW]               503910 non-null  float64
 2   gen [kW]               503910 non-null  float64
 3   House overall [kW]     503910 non-null  float64
 4   Dishwasher [kW]                503910 non-null  float64
 5   Furnace 1 [kW]                 503910 non-null  float64
 6   Furnace 2 [kW]                 503910 non-null  float64
 7   Home office [kW]               503910 non-null  float64
 8   Fridge [kW]            503910 non-null  float64
 9   Wine cellar [kW]               503910 non-null  float64
 10  Garage door [kW]       503910 non-null  float64
 11  Kitchen 12 [kW]                503910 non-null  float64
 12  Kitchen 14 [kW]                503910 non-null  float64
 13  Kitchen 38 [kW]                503910 non-null  float64
 14  Barn [kW]              503910 non-null  float64
 15  Well [kW]              503910 non-null  float64
 16  Microwave [kW]                 503910 non-null  float64
 17  Living room [kW]       503910 non-null  float64
 18  Solar [kW]             503910 non-null  float64
 19  temperature            503910 non-null  float64
```

```
 20  icon                 503910 non-null  object
 21  humidity             503910 non-null  float64
 22  visibility           503910 non-null  float64
 23  summary              503910 non-null  object
 24  apparentTemperature  503910 non-null  float64
 25  pressure             503910 non-null  float64
 26  windSpeed            503910 non-null  float64
 27  cloudCover           503910 non-null  object
 28  windBearing          503910 non-null  float64
 29  precipIntensity              503910 non-null  float64
 30  dewPoint             503910 non-null  float64
 31  precipProbability    503910 non-null  float64
dtypes: float64(28), object(4)
memory usage: 123.0+ MB
```

---

1. Preprocessing

```
#Convert Unix timestamp to datetime, use sample frequency of minutes and make it dataframe index
data['time'] = pd.to_datetime(data['time'], unit='s')
data['time'] = pd.DatetimeIndex(pd.date_range('2016-01-01 05:00', periods=len(data),  freq='min'))
data = data.set_index('time')
data.head(2)

#Delete '[kW]' in columns name, sum similar consumtions and delete 'summary' column
data.columns = [i.replace(' [kW]', '') for i in data.columns]
data['Furnace'] = data[['Furnace 1',]].sum(axis=1)
data['Kitchen'] = data[['Kitchen 12']].sum(axis=1)
#We could also use the mean
data.drop(['Furnace 2','Kitchen 14','Kitchen 38','Wine cellar'], axis=1, inplace=True)

#Replace invalid values in column 'cloudCover' with backfill method
data['cloudCover'].replace(['cloudCover'], method='bfill', inplace=True)
data['cloudCover'] = data['cloudCover'].astype('float')

#Reorder columns
data = data[['use', 'gen', 'House overall', 'Dishwasher', 'Home office', 'Fridge', 'Garage door', 'Barn', 'Well', 'Microwave', 'Living room', 'Furnace', 'Kitchen', 'Solar', 'temperature', 'humidity', 'visibility',
'apparentTemperature', 'pressure', 'windSpeed', 'cloudCover', 'windBearing', 'precipIntensity', 'dewPoint', 'precipProbability']]
data.head(2)
```

Energy correlations

```
#Checking (Only Energia)
fig = plt.subplots(figsize=(10, 8))
```

```
sns.heatmap(data[data.columns[0:15].tolist()].corr(), annot=True, fmt='.2f', vmin=-
1.0, vmax=1.0, center=0, cmap="bwr")
plt.title('Energy Correlation', fontsize=12);

#Drop the duplicate ones
data.drop(['use', 'gen'], axis=1, inplace=True)
```

Weather correlations

```
#wheather Correlation
fig = plt.subplots(figsize=(10, 8))
sns.heatmap(data[data.columns[13:].tolist()].corr(), annot=True, vmin=-
1.0, vmax=1.0, center=0, cmap="bwr")
plt.title('wheather Correlation', fontsize=14);
```

Groupings

```
#Define new columns from datetime
data['month'] = data.index.month
data['day'] = data.index.day
data['weekday'] = data.index.day_name()
data['hour'] = data.index.hour
data['minute'] = data.index.minute
data.head(2)
```

Month

```
##Averge consuption per month
mean_month = data.groupby('month').agg({i:'mean' for i in data.columns[:-
5].tolist()})
mean_month[mean_month.columns[0:13].tolist()].plot(subplots=True, layout=(1, 3),
 figsize=(15, 10), grid=True, rot=45, xlabel=None, marker='o', mfc = 'r', color=('blue'
));
```

Weekday

```
#Averge consuption per day of the week
days = [ 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
mean_weekday = data.groupby('weekday').agg({i:'mean' for i in data.columns[:-
5].tolist()}).reindex(days)

mean_weekday[mean_weekday.columns[0:13].tolist()].plot(subplots=True, layout=(
-1, 3), figsize=(15, 10),
grid=True, rot=45, xlabel=None, marker='o', mfc = 'r', color=('blue'));
# ax = plt.gca()
# ax.set_facecolor("orange")
```

Hour

```
#Averge consuption per hour
mean_hour = data.groupby('hour').agg({i:'mean' for i in data.columns[:-5].tolist()})
```

```
mean_hour[mean_hour.columns[0:13].tolist()].plot(subplots=True, layout=(-
1, 3), figsize=(18, 12),
grid=True, rot=45, xlabel=None, marker='o', mfc = 'r', color=('blue'));
```

Baseline Model

Moving Average

Prendiamo la media mobile come modello "baseline" di riferimento:

$$\hat{y}_t = \frac{1}{n}\sum_{i=1}^{n} y_{t-i}$$

per avere un errore RMSE da migliorare

```
#Data resampling by day
data_daily = data['House overall'].resample('d').mean()
#Define la moving average
baseline = data_daily.rolling(window=10).mean()
#Plot
plt.figure(figsize=(15,3))
plt.plot(data_daily, c='blue',label='Data - House overall')
plt.plot(baseline, c='red', label='Rolling mean')
plt.legend(fontsize=12)
plt.ylabel('kW')
plt.margins(x=0)
plt.grid()
```

Train, Validation, Test split attempt

```
n_steps = 30
X, Y = sampling(data_daily.tolist(), n_steps)
X = X.reshape((X.shape[0], X.shape[1], 1))
size = int(len(data_daily)*0.7)
size2 = int(((len(data_daily)-size)/2)+size)
X_train, Y_train = X[:size], Y[:size]
X_val, Y_val = X[size:size2], Y[size:size2]
X_test, Y_test = X[size2:], Y[size2:]
print('Training size:', size)
print('Training + Validation size:', size2)
plt.plot(data_daily[:size])
plt.grid(True)
plt.plot(data_daily[size:size2], color=("red"))
plt.plot(data_daily[size2:], color=("green"))
```

Appendix-B: LSTM Models

LSTM & ADAM

```
#Reduce size of dataframe with only the columns we are interested in
tf.keras.backend.clear_session()

data_daily = data[['House overall', 'Furnace', 'Living room', 'Barn', 'temperature', 'hu
midity', 'apparentTemperature', 'pressure', 'cloudCover', 'precipIntensity',
 'dewPoint', 'precipProbability']]
#Rescale
data_daily = data_daily.resample('D').mean()
#Normalize the features
scaler = MinMaxScaler(feature_range=(0, 1))
data_daily[data_daily.columns[1:]] = scaler.fit_transform(data_daily[data_daily.colu
mns[1:]])
scaler_target = MinMaxScaler(feature_range=(0, 1))
data_daily[['House overall']] = scaler_target.fit_transform(data_daily[['House overall'
]])

size = int(len(data_daily)*0.7)
data_daily_train = data_daily[:size]
data_daily_test = data_daily[size:]
X_train, X_test = [], []
Y_train, Y_test = [], []
n_past=1
n_future=1
for i in range(n_past, len(data_daily_train)-n_future+1):
    X_train.append(data_daily_train.iloc[i-n_past:i, 0:data_daily.shape[1]])
    Y_train.append(data_daily_train.iloc[i+n_future-1:i+n_future, 0])
for i in range(n_past, len(data_daily_test)-n_future+1):
    X_test.append(data_daily_test.iloc[i-n_past:i, 0:data_daily_test.shape[1]])
    Y_test.append(data_daily_test.iloc[i+n_future-1:i+n_future, 0])

X_train, Y_train = np.array(X_train), np.array(Y_train)
X_test, Y_test = np.array(X_test), np.array(Y_test)

print('X_train shape', X_train.shape)
print('X_test shape', X_test.shape)
print('Y_train shape', Y_train.shape)
print('Y_test shape', Y_test.shape)
```

X_train shape (244, 1, 12)
X_test shape (105, 1, 12)
Y_train shape (244, 1)
Y_test shape (105, 1)

```
model = Sequential()
model.add(LSTM(50, activation='relu', return_sequences = False, input_shape=(X_tr
ain.shape[1], X_train.shape[2])))
model.add(Dropout(0.001))
```

```
model.add(Dense(Y_train.shape[1]))

optimizer = keras.optimizers.Adam(learning_rate=0.001)
model.compile(optimizer=optimizer, loss='mse')
model.summary()

model_fit = model.fit(X_train, Y_train, epochs=100, verbose=0)

Train_pred = model.predict(X_train, verbose=0)
Y_pred = model.predict(X_test, verbose=0)

plt.plot(model_fit.history['loss'], label='Train loss')
#plt.plot(model_fit.history['val_loss'], label='Validation loss')
plt.rcParams.update({'axes.facecolor':'gainsboro'})
plt.grid(True)
plt.legend()
print('Train MSE minimum:', min(model_fit.history['loss']))
#print('Validation MSE minimum:', min(model_fit.history['val_loss']))
#Invert scaling
data_daily[['House overall']] = scaler_target.inverse_transform(data_daily[['House o
verall']])
Y_pred = scaler_target.inverse_transform(Y_pred)
Train_pred = scaler_target.inverse_transform(Train_pred)
plt.figure(figsize=(15,4))
plt.rcParams.update({'axes.facecolor':'gainsboro'})
plt.grid(True)
plt.plot(data_daily[['House overall']][size:-1].values)
plt.plot(Y_pred)
np.sqrt(mean_squared_error(Y_pred[:,0].tolist(), data_daily[['House overall']][size:-
1].values))
```

0.1604906630432789

```
Y_pred_series = pd.Series(Y_pred.flatten().tolist(), index=data_daily['House overall'
][size:-n_past].index)
Train_pred_series = pd.Series(Train_pred.flatten().tolist(), index=data_daily['House
overall'][n_past:size].index)
plt.figure(figsize=(15,4))
plt.rcParams.update({'axes.facecolor':'gainsboro'})
plt.plot(data_daily['House overall'][:-n_past], c='blue', label='data')
plt.plot(Y_pred_series, c='red', label='model test')
plt.plot(Train_pred_series, c='green', label='model train')
plt.legend()
plt.grid(), plt.margins(x=0);
Y_test = data_daily['House overall'][size:-n_past]

# calcolo errore
print('MSE: %.5f' % (mean_squared_error(Y_pred, Y_test)))
print('RMSE: %.5f' % np.sqrt(mean_squared_error(Y_pred, Y_test)))
MAE = mean_absolute_error(Y_test, Y_pred)
MAPE = np.mean(np.abs(Y_pred[:,0] - Y_test.values)/np.abs(Y_test.values))
```

```
#MASE = np.mean(np.abs(Y_test -
 Y_pred))/(np.abs(np.diff(X_train)).sum()/(len(X_train)-1))
print('MAE: %.3f' % MAE)
print('MAPE: %.3f' %MAPE)
print('R^2 score: %.3f' % r2_score(Y_test, Y_pred))
```

MSE: 0.02576
RMSE: 0.16049
MAE: 0.129
MAPE: 0.197
R^2 score: 0.655

```
##############################here is the anom#############################
from sklearn.metrics import r2_score, median_absolute_error, mean_absolute_error,
mean_squared_error

def plotMovingAverage(series, window, plot_intervals=False, scale=1.96, plot_anom
alies=False):

    rolling_mean = series.rolling(window=window).mean()

    plt.figure(figsize=(15,5))
    plt.rcParams.update({'axes.facecolor':'gainsboro'})
    plt.title("Moving average with window size = {}".format(window))
    plt.plot(rolling_mean, "r", label="Rolling mean trend", color = ('lime'))

    # Plot confidence intervals for smoothed values
    if plot_intervals:
        mae = mean_absolute_error(series[window:], rolling_mean[window:])
        deviation = np.std(series[window:] - rolling_mean[window:])
        lower_bond = rolling_mean - (mae + scale * deviation)
        upper_bond = rolling_mean + (mae + scale * deviation)
        plt.plot(upper_bond, "orange" , label="Upper Bond / Lower Bond")
        plt.plot(lower_bond, "orange")

        # Having the intervals, find abnormal values
        if plot_anomalies:
        anomalies = pd.DataFrame(index=series.index, columns=series.columns)
        anomalies[series<lower_bond] = series[series<lower_bond]
        anomalies[series>upper_bond] = series[series>upper_bond]
        plt.plot(anomalies, "ro", markersize=5)

    plt.plot(series[window:], 'blue', label="Actual values")
    plt.legend(loc="upper left")
    plt.grid(True), plt.margins(x=0);

data_anom = data.resample('d').mean()
cols=['House overall']
plotMovingAverage(data_anom[cols], window=20, plot_intervals=True, plot_anoma
lies=True)
```

LSTM & Adamax

```
#Reduce size of dataframe with only the columns we are interested in
tf.keras.backend.clear_session()
data_daily = data[['House overall', 'Furnace', 'Living room', 'Barn', 'temperature', 'hu
midity', 'apparentTemperature', 'pressure', 'cloudCover', 'precipIntensity', 'dewPoint', '
precipProbability']]
#Rescale
data_daily = data_daily.resample('D').mean()
#Normalize the features
scaler = MinMaxScaler(feature_range=(0, 1))
data_daily[data_daily.columns[1:]] = scaler.fit_transform(data_daily[data_daily.colu
mns[1:]])
scaler_target = MinMaxScaler(feature_range=(0, 1))
data_daily[['House overall']] = scaler_target.fit_transform(data_daily[['House overall'
]])

size = int(len(data_daily)*0.7)
data_daily_train = data_daily[:size]
data_daily_test = data_daily[size:]
X_train, X_test = [], []
Y_train, Y_test = [], []
n_past=1
n_future=1
for i in range(n_past, len(data_daily_train)-n_future+1):
    X_train.append(data_daily_train.iloc[i-n_past:i, 0:data_daily.shape[1]])
    Y_train.append(data_daily_train.iloc[i+n_future-1:i+n_future, 0])
for i in range(n_past, len(data_daily_test)-n_future+1):
    X_test.append(data_daily_test.iloc[i-n_past:i, 0:data_daily_test.shape[1]])
    Y_test.append(data_daily_test.iloc[i+n_future-1:i+n_future, 0])

X_train, Y_train = np.array(X_train), np.array(Y_train)
X_test, Y_test = np.array(X_test), np.array(Y_test)

print('X_train shape', X_train.shape)
print('X_test shape', X_test.shape)
print('Y_train shape', Y_train.shape)
print('Y_test shape', Y_test.shape)
```

X_train shape (244, 1, 12)
X_test shape (105, 1, 12)
Y_train shape (244, 1)
Y_test shape (105, 1

```
model = Sequential()
model.add(LSTM(50, activation='relu', return_sequences = False, input_shape=(X_tr
ain.shape[1], X_train.shape[2])))
model.add(Dropout(0.001))
model.add(Dense(Y_train.shape[1]))
```

```
optimizer = keras.optimizers.Adamax(learning_rate=0.001)
model.compile(optimizer=optimizer, loss='mse')
model.summary()

model_fit = model.fit(X_train, Y_train, epochs=100, verbose=0)

Train_pred = model.predict(X_train, verbose=0)
Y_pred = model.predict(X_test, verbose=0)

plt.plot(model_fit.history['loss'], label='Train loss')
#plt.plot(model_fit.history['val_loss'], label='Validation loss')
plt.rcParams.update({'axes.facecolor':'gainsboro'})
plt.grid(True)
plt.legend()
print('Train MSE minimum:', min(model_fit.history['loss']))
#print('Validation MSE minimum:', min(model_fit.history['val_loss']))
#Invert scaling
data_daily[['House overall']] = scaler_target.inverse_transform(data_daily[['House o
verall']])
Y_pred = scaler_target.inverse_transform(Y_pred)
Train_pred = scaler_target.inverse_transform(Train_pred)

plt.figure(figsize=(15,4))
plt.rcParams.update({'axes.facecolor':'gainsboro'})
plt.grid(True)
plt.plot(data_daily[['House overall']][size:-1].values)
plt.plot(Y_pred)
np.sqrt(mean_squared_error(Y_pred[:,0].tolist(), data_daily[['House overall']][size:-
1].values))
```

0.1554461428379365

```
Y_pred_series = pd.Series(Y_pred.flatten().tolist(), index=data_daily['House overall'
][size:-n_past].index)
Train_pred_series = pd.Series(Train_pred.flatten().tolist(), index=data_daily['House
overall'][n_past:size].index)
plt.figure(figsize=(15,4))
plt.rcParams.update({'axes.facecolor':'gainsboro'})
plt.plot(data_daily['House overall'][:-n_past], c='blue', label='data')
plt.plot(Y_pred_series, c='red', label='model test')
plt.plot(Train_pred_series, c='green', label='model train')
plt.legend()
plt.grid(), plt.margins(x=0);
Y_test = data_daily['House overall'][size:-n_past]

# calcolo errore
print('MSE: %.5f' % (mean_squared_error(Y_pred, Y_test)))
print('RMSE: %.5f' % np.sqrt(mean_squared_error(Y_pred, Y_test)))
MAE = mean_absolute_error(Y_test, Y_pred)
MAPE = np.mean(np.abs(Y_pred[:,0] - Y_test.values)/np.abs(Y_test.values))
```

```
#MASE = np.mean(np.abs(Y_test -
 Y_pred))/(np.abs(np.diff(X_train)).sum()/(len(X_train)-1))
print('MAE: %.3f' % MAE)
print('MAPE: %.3f' %MAPE)
print('R^2 score: %.3f' % r2_score(Y_test, Y_pred))
```

MSE: 0.02416
RMSE: 0.15545
MAE: 0.122
MAPE: 0.181
R^2 score: 0.677

```
from sklearn.metrics import r2_score, median_absolute_error, mean_absolute_error,
mean_squared_error

def plotMovingAverage(series, window, plot_intervals=False, scale=1.96, plot_anom
alies=False):

    rolling_mean = series.rolling(window=window).mean()

    plt.figure(figsize=(15,5))
    plt.rcParams.update({'axes.facecolor':'gainsboro'})
    plt.title("Moving average with window size = {}".format(window))
    plt.plot(rolling_mean, "r", label="Rolling mean trend", color = ('lime'))

    # Plot confidence intervals for smoothed values
    if plot_intervals:
        mae = mean_absolute_error(series[window:], rolling_mean[window:])
        deviation = np.std(series[window:] - rolling_mean[window:])
        lower_bond = rolling_mean - (mae + scale * deviation)
        upper_bond = rolling_mean + (mae + scale * deviation)
        plt.plot(upper_bond, "orange" , label="Upper Bond / Lower Bond")
        plt.plot(lower_bond, "orange")

        # Having the intervals, find abnormal values
        if plot_anomalies:
anomalies = pd.DataFrame(index=series.index, columns=series.columns)
            anomalies[series<lower_bond] = series[series<lower_bond]
            anomalies[series>upper_bond] = series[series>upper_bond]
            plt.plot(anomalies, "ro", markersize=5)

    plt.plot(series[window:], 'blue', label="Actual values")
    plt.legend(loc="upper left")
    plt.grid(True), plt.margins(x=0);

data_anom = data.resample('d').mean()
cols=['House overall']
plotMovingAverage(data_anom[cols], window=20, plot_intervals=True, plot_anoma
lies=True)
```

LSTM & Nadam

```
#Reduce size of dataframe with only the columns we are interested in
tf.keras.backend.clear_session()
data_daily = data[['House overall', 'Furnace', 'Living room', 'Barn', 'temperature', 'hu
midity',
'apparentTemperature', 'pressure', 'cloudCover', 'precipIntensity', 'dewPoint', 'precipPr
obability']]
#Rescale
data_daily = data_daily.resample('D').mean()
#Normalize the features
scaler = MinMaxScaler(feature_range=(0, 1))
data_daily[data_daily.columns[1:]] = scaler.fit_transform(data_daily[data_daily.colu
mns[1:]])
scaler_target = MinMaxScaler(feature_range=(0, 1))
data_daily[['House overall']] = scaler_target.fit_transform(data_daily[['House overall'
]])

size = int(len(data_daily)*0.7)
data_daily_train = data_daily[:size]
data_daily_test = data_daily[size:]
X_train, X_test = [], []
Y_train, Y_test = [], []
n_past=1
n_future=1
for i in range(n_past, len(data_daily_train)-n_future+1):
    X_train.append(data_daily_train.iloc[i-n_past:i, 0:data_daily.shape[1]])
    Y_train.append(data_daily_train.iloc[i+n_future-1:i+n_future, 0])
for i in range(n_past, len(data_daily_test)-n_future+1):
X_test.append(data_daily_test.iloc[in_past:i, 0:data_daily_test.shape[1]])
Y_test.append(data_daily_test.iloc[i+n_future-1:i+n_future, 0])

X_train, Y_train = np.array(X_train), np.array(Y_train)
X_test, Y_test = np.array(X_test), np.array(Y_test)

print('X_train shape', X_train.shape)
print('X_test shape', X_test.shape)
print('Y_train shape', Y_train.shape)
print('Y_test shape', Y_test.shape)
```

X_train shape (244, 1, 12)
X_test shape (105, 1, 12)
Y_train shape (244, 1)
Y_test shape (105, 1)

```
model = Sequential()
model.add(LSTM(50, activation='relu', return_sequences = False, input_shape=(X_tr
ain.shape[1], X_train.shape[2])))
model.add(Dropout(0.001))
model.add(Dense(Y_train.shape[1]))
```

```
optimizer = keras.optimizers.Nadam(learning_rate=0.001)
model.compile(optimizer=optimizer, loss='mse')

model_fit = model.fit(X_train, Y_train, epochs=100, verbose=0)

Train_pred = model.predict(X_train, verbose=0)
Y_pred = model.predict(X_test, verbose=0)

plt.plot(model_fit.history['loss'], label='Train loss')
#plt.plot(model_fit.history['val_loss'], label='Validation loss')
plt.rcParams.update({'axes.facecolor':'gainsboro'})
plt.grid(True)
plt.legend()
print('Train MSE minimum:', min(model_fit.history['loss']))
#print('Validation MSE minimum:', min(model_fit.history['val_loss']))

#Invert scaling
data_daily[['House overall']] = scaler_target.inverse_transform(data_daily[['House o
verall']])
Y_pred = scaler_target.inverse_transform(Y_pred)
Train_pred = scaler_target.inverse_transform(Train_pred)

plt.figure(figsize=(15,4))
plt.rcParams.update({'axes.facecolor':'gainsboro'})
plt.grid(True)
plt.plot(data_daily[['House overall']][size:-1].values)
plt.plot(Y_pred)
np.sqrt(mean_squared_error(Y_pred[:,0].tolist(), data_daily[['House overall']][size:-
1].values))
```

0.17133033351877708

```
Y_pred_series = pd.Series(Y_pred.flatten().tolist(), index=data_daily['House overall'
][size:-n_past].index)
Train_pred_series = pd.Series(Train_pred.flatten().tolist(), index=data_daily['House
overall'][n_past:size].index)
plt.figure(figsize=(15,4))
plt.rcParams.update({'axes.facecolor':'gainsboro'})
plt.plot(data_daily['House overall'][:-n_past], c='blue', label='data')
plt.plot(Y_pred_series, c='red', label='model test')
plt.plot(Train_pred_series, c='green', label='model train')
plt.legend()
plt.grid(), plt.margins(x=0);
Y_test = data_daily['House overall'][size:-n_past]

# calcolo errore
print('MSE: %.5f' % (mean_squared_error(Y_pred, Y_test)))
print('RMSE: %.5f' % np.sqrt(mean_squared_error(Y_pred, Y_test)))
MAE = mean_absolute_error(Y_test, Y_pred)
MAPE = np.mean(np.abs(Y_pred[:,0] - Y_test.values)/np.abs(Y_test.values))
```

```
#MASE = np.mean(np.abs(Y_test -
 Y_pred))/(np.abs(np.diff(X_train)).sum()/(len(X_train)-1))
print('MAE: %.3f' % MAE)
print('MAPE: %.3f' %MAPE)
print('R^2 score: %.3f' % r2_score(Y_test, Y_pred))
```

MSE: 0.02935
RMSE: 0.17133
MAE: 0.142
MAPE: 0.219
R^2 score: 0.607

```
from sklearn.metrics import r2_score, median_absolute_error, mean_absolute_error,
mean_squared_error

def plotMovingAverage(series, window, plot_intervals=False, scale=1.96, plot_anom
alies=False):

    rolling_mean = series.rolling(window=window).mean()

    plt.figure(figsize=(15,5))
    plt.rcParams.update({'axes.facecolor':'gainsboro'})
    plt.title("Moving average with window size = {}".format(window))
    plt.plot(rolling_mean, "r", label="Rolling mean trend", color = ('lime'))

    # Plot confidence intervals for smoothed values
    if plot_intervals:
        mae = mean_absolute_error(series[window:], rolling_mean[window:])
        deviation = np.std(series[window:] - rolling_mean[window:])
        lower_bond = rolling_mean - (mae + scale * deviation)
        upper_bond = rolling_mean + (mae + scale * deviation)
        plt.plot(upper_bond, "orange" , label="Upper Bond / Lower Bond")
        plt.plot(lower_bond, "orange")

        # Having the intervals, find abnormal values
        if plot_anomalies:
        anomalies = pd.DataFrame(index=series.index, columns=series.columns)
        anomalies[series<lower_bond] = series[series<lower_bond]
        anomalies[series>upper_bond] = series[series>upper_bond]
        plt.plot(anomalies, "ro", markersize=5)

    plt.plot(series[window:], 'blue', label="Actual values")
    plt.legend(loc="upper left")
    plt.grid(True), plt.margins(x=0);

data_anom = data.resample('d').mean()
cols=['House overall']
plotMovingAverage(data_anom[cols], window=20, plot_intervals=True, plot_anoma
lies=True)
```

Appendix-C: ARIMA

Arima v1

```
import statsmodels.api as sm
from statsmodels.tsa.arima.model import ARIMA
model = ARIMA(data_anom['House overall'], order=(2,1,1))
model_fit = model.fit()
print('AIC: ', model_fit.aic)
print('MSE: ', model_fit.mse)
```

AIC: 300.9797957439473
MSE: 0.13861045047899023

```
squared_errors = (model_fit.resid)
threshold = np.mean(squared_errors) + 1.96*np.std(squared_errors)
upper_bond = model_fit.predict(dynamic=False)+threshold
lower_bond = model_fit.predict(dynamic=False)-threshold
anomalies_v1 = data_anom['House overall'][(data_anom['House overall']<lower_bon
d)|(data_anom['House overall']>upper_bond)]

plt.figure(figsize=(15,5))
plt.rcParams.update({'axes.facecolor':'gainsboro'})
plt.plot(model_fit.predict(dynamic=False)+threshold, c='orange', label='Upper bound
 / Lower bound')
plt.plot(model_fit.predict(dynamic=False)-threshold, c='orange',)
plt.plot(model_fit.predict(dynamic=False), c='lime', label='model')
plt.plot(data_anom['House overall'], c='blue',label='Data - House overall')
plt.plot(anomalies_v1, "ro", markersize=5)
plt.legend(), plt.grid(), plt.margins(x=0);
```

anomalies_v1

ARIMA v2

```
tf.keras.backend.clear_session()

predictions = model_fit.predict(dynamic=False)

confidence = model_fit.get_prediction().conf_int(alpha=0.01)
#Errors = np.abs(data_anom['use'] - predictions)
Errors = np.sqrt(mean_squared_error(data_anom['House overall'], predictions))
Uncertainty = confidence['upper House overall'] - confidence['lower House overall']
Anomalies_v2 = data_anom['House overall'][(data_anom['House overall']> confiden
ce['upper House overall'])|
(data_anom['House overall'] < confidence['lower House overall'])]
#Anomalies = dati_anom['use'][Errors > Uncertainty]

plt.figure(figsize=(15,5))
plt.rcParams.update({'axes.facecolor':'gainsboro'})
```

```
plt.plot(data_anom['House overall'], c='blue',label='Data - House overall')
plt.plot(predictions, c='green', label='model')
plt.plot(confidence.index[1:],confidence['lower House overall'][1:], confidence['uppe
r House overall'][1:], color='orange')
plt.plot(Anomalies_v2, "ro", markersize=5)
plt.legend()
plt.grid(), plt.margins(x=0);

## ARIMA with order=(1,1,2)
import statsmodels.api as sm
from statsmodels.tsa.arima.model import ARIMA
model = ARIMA(data_anom['House overall'], order=(1,1,2))
model_fit = model.fit()
print('AIC: ', model_fit.aic)
print('MSE: ', model_fit.mse)
```

AIC: 300.94365018890267
MSE: 0.13859761091437336

```
squared_errors = (model_fit.resid)
threshold = np.mean(squared_errors) + 1.96*np.std(squared_errors)
upper_bond = model_fit.predict(dynamic=False)+threshold
lower_bond = model_fit.predict(dynamic=False)-threshold
anomalies_a2_1 = data_anom['House overall'][(data_anom['House overall']<lower_b
ond)|(data_anom['House overall']>upper_bond)]

plt.figure(figsize=(15,5))
plt.rcParams.update({'axes.facecolor':'gainsboro'})
plt.plot(model_fit.predict(dynamic=False)+threshold, c='orange', label='Upper bound
 / Lower bound')
plt.plot(model_fit.predict(dynamic=False)-threshold, c='orange',)
plt.plot(model_fit.predict(dynamic=False), c='lime', label='model')
plt.plot(data_anom['House overall'], c='blue',label='Data - House overall')
plt.plot(anomalies_a2_1, "ro", markersize=5)
plt.legend(), plt.grid(), plt.margins(x=0);

tf.keras.backend.clear_session()
predictions = model_fit.predict(dynamic=False)

confidence = model_fit.get_prediction().conf_int(alpha=0.01)
#Errors = np.abs(data_anom['use'] - predictions)
Errors = np.sqrt(mean_squared_error(data_anom['House overall'], predictions))
Uncertainty = confidence['upper House overall'] - confidence['lower House overall']
Anomalies_a2 = data_anom['House overall'][(data_anom['House overall']> confiden
ce['upper House overall'])|(data_anom['House overall'] < confidence['lower House ov
erall'])]
#Anomalies = dati_anom['use'][Errors > Uncertainty]

plt.figure(figsize=(15,5))
plt.rcParams.update({'axes.facecolor':'gainsboro'})
```

```
plt.plot(data_anom['House overall'], c='blue',label='Data - House overall')
plt.plot(predictions, c='green', label='model')
plt.plot(confidence.index[1:],confidence['lower House overall'][1:], confidence['uppe
r House overall'][1:], color='orange')
plt.plot(Anomalies_a2, "ro", markersize=5)
plt.legend()
plt.grid(), plt.margins(x=0);
```

Anomalies_a2

Appendix-D: SarimaX

```
## order=(2,1,1)
import statsmodels.api as sm
from statsmodels.tsa.statespace.sarimax import SARIMAX
model = SARIMAX(data_anom['House overall'], order=(2,1,1), seasonal_order=(5,0,
1,12))
model_fit = model.fit()
print('AIC: ', model_fit.aic)
print('MSE: ', model_fit.mse)
```

AIC: 305.24731028762744
MSE: 0.1353603219083357

```
squared_errors = (model_fit.resid)
threshold = np.mean(squared_errors) + 1.96*np.std(squared_errors)
upper_bond = model_fit.predict(dynamic=False)+threshold
lower_bond = model_fit.predict(dynamic=False)-threshold
anomalies_s1 = data_anom['House overall'][(data_anom['House overall']<lower_bon
d)|(data_anom['House overall']>upper_bond)]

plt.figure(figsize=(15,5))
plt.plot(model_fit.predict(dynamic=False)+threshold, c='orange', label='Upper bound
 / Lower bound')
plt.plot(model_fit.predict(dynamic=False)-threshold, c='orange')
plt.plot(model_fit.predict(dynamic=False), c='lime', label='model')
plt.plot(data_anom['House overall'], c='blue',label='Data - House overall')
plt.plot(anomalies_s1, "ro", markersize=5)
plt.legend(), plt.grid(), plt.margins(x=0);


## order=(1,1,2)
import statsmodels.api as sm
from statsmodels.tsa.statespace.sarimax import SARIMAX
model = SARIMAX(data_anom['House overall'], order=(1,1,2), seasonal_order=(5,0,
1,12))
model_fit = model.fit()
print('AIC: ', model_fit.aic)
print('MSE: ', model_fit.mse)
```

AIC: 307.03246025436187
MSE: 0.13615675858682566

```
squared_errors = (model_fit.resid)
threshold = np.mean(squared_errors) + 1.96*np.std(squared_errors)
upper_bond = model_fit.predict(dynamic=False)+threshold
lower_bond = model_fit.predict(dynamic=False)-threshold
anomalies_s2 = data_anom['House overall'][(data_anom['House overall']<lower_bon
d)|(data_anom['House overall']>upper_bond)]
plt.figure(figsize=(15,5))
```

```
plt.plot(model_fit.predict(dynamic=False)+threshold, c='orange', label='Upper bound
 / Lower bound')
plt.plot(model_fit.predict(dynamic=False)-threshold, c='orange')
plt.plot(model_fit.predict(dynamic=False), c='lime', label='model')
plt.plot(data_anom['House overall'], c='blue',label='Data - House overall')
plt.plot(anomalies_s2, "ro", markersize=5)
plt.legend(), plt.grid(), plt.margins(x=0);
```

RESUME

Ahmed Ghanim Daoud ALDAOUD

EDUCATION:

- Dip.E.T. Diploma in Electronics Technology from the Technical Institute in Mosul 1991-1994.
- B.Sc. Bachelor in Computer Techniques Engineering from Northern Technical University in Mosul 2001-2005.

WORK EXPERIENCE:

- Working as (Ass. Chief Engineer) of Mosul Municipality Directorate in Mosul 2005-2023.