REPUBLIC OF TÜRKİYE

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Information Technologies

MATLAB IMPLEMENTATION OF ARTIFICIAL NEURAL NETWORKS ALGORITHMS FOR POWER CONSUMPTION PREDICTION

Mohammad Mounam Agool AGOOL

Master's Thesis

Supervisor

Asst. prof. Dr. Abdullahi Abdu Ibrahim

Istanbul, 2022

MATLAB IMPLEMENTATION OF ARTIFICIAL NEURAL NETWORKS ALGORITHMS FOR POWER CONSUMPTION PREDICTION

Mohammad Mounam Agool AGOOL

IT Department

Master's Thesis

ALTINBAŞ UNIVERSITY

2022

The thesis/dissertation titled MATLAB İMPLEMENTATİON OF ARTİFİCİAL NEURAL NETWORKS ALGORİTHMS FOR POWER CONSUMPTİON PREDİCTİON prepared by MOHAMMAD MOUNAM AGOOL AGOOL and submitted on 13/12/2022 has been **accepted unanimously** for the degree of Master of Science in Information Technologies.

Asst. Prof. Dr. Abdullahi Abdu IBRAHIM
Supervisor

Thesis Defense Committee Members:

Asst. Prof. Dr. Abdullahi Abdu IBRAHIM	Department of Computer Engineering, Altınbaş University	
Asst. Prof. Dr. Ayça Kurnaz TÜRKBEN	Department of Software Engineering, Altınbaş University	
Asst. Prof. Dr. Serdar KARGIN	Department of Biomedical Engineering, Arel University	

I hereby declare that this thesis meets all format and submission requirements of a Master'sthesis.

Submission date of the thesis to the Institute of Graduate Studies: __/__/___

I hereby declare that all information/data presented in this graduation project has been obtained in full accordance with academic rules and ethical conduct. I also declare all unoriginal materials and conclusions have been cited in the text and all references mentioned in the Reference List have been cited in the text, and vice versa as required by the abovementioned rules and conduct.

Mohammad Mounam AGOOL

Signature

ABSTRACT

MATLAB IMPLEMENTATION OF ARTIFICIAL NEURAL NETWORKS ALGORITHMS FOR POWER CONSUMPTION PREDICTION

AGOOL , Mohammad

M.Sc., Information Technologies, Altınbaş University,

Supervisor: Asst. prof. Dr. Abdullahi Abdu Ibrahim

Date: December/2022

Pages: 70

At the same time as there is a drive to increase electrification efforts on a global scale, new technological innovations are leading to an increase in the amount of power that is consumed worldwide. As a result, we should anticipate an increase in the amount of electricity used inside the residential setting. It is becoming increasingly difficult to generate reliable projections about future power consumption as more information about the usage of energy throughout the world becomes available. Both the buyer and the seller stand to benefit from an accurate prognosis. A customer may be able to save money and reduce their carbon footprint with the assistance of a power forecast. It is essential for the effectiveness of the provider's attempts to regulate the flow of goods that they have a forecast that can be relied upon. Consequently, this type of modeling can make a contribution to the overall optimization of the supply chain in the residential electricity industry. As a result of the extensive interest in this issue, a broad variety of techniques to solving it have been investigated and assessed. In this study, we introduce the AAA-ANN and ANN-PSO algorithms for accurately predicting energy consumption. Both of these methods are based on artificial neural networks. In order to provide accurate projections of future power consumption, the algorithms underlying both approaches were developed in Matlab and then trained using data taken from the UCP database. The AAA-ANN method outperforms other approaches in terms of error curve analysis, training accuracy rate analysis,

testing accuracy rate analysis, training error rate analysis, and testing error rate analysis. After comparing AAA-ANN to ANN and ANN-PSO, it became abundantly evident that this particular method was the best approach for forecasting future power use.

Keywords: Artificial Algae Algorithm, Artificial Intelligence, Machine Learning, Neural Networks, Particle Swarm Optimization.

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

ABBREVIATIONS

PSO : Particle Swarm Optimization

ML : Machine Learning

ANN : Artificial Neural Networks

AI : Artificial Intelligence

AAA : Artificial Algae Algorithm

1 INTRODUCTION

As a method of machine learning, neural networks are gaining more and more traction because of the way they include concepts that were initially seen in organic nerve systems. A neuron is the component of a living creature that is comprised of dendrites, which are minute protrusions on the cell surface, and an axon, which is a larger protrusion on the cell surface. An axon is a long and slender projection that extends forth from the dendrites of a neuron. This is where nerve impulses are sent. On the dendrites, which are the parts of a neuron that receive connections from axons of other neurons, there are likely to be a great number of synapses. An electric impulse is sent out by the axons on occasion. This alters the permeability of the cell membrane, which, in turn, produces a little increase in voltage within the body of the neuron. The quantity of extra neurons that become active causes an increase in voltage that is directly proportional to that amount. At the origin of the axon, a dedicated facility performs continuous voltage measurements at regular intervals. As long as the voltage remains below a specific threshold, there won't be any significant changes. When this threshold is exceeded, a wave of high voltage flows from the [1] neuron toward the end of the axon, which ultimately causes an action potential to be generated. Repeating a technique identical to this one is done until all of the neurons have been connected. It is possible to imitate, albeit to a limited degree, the function of real neurons by employing artificial neuron networks. Artificial neurons, like their biological counterparts, can have one or more inputs and only one output. This is because artificial neurons are modelled after biological neurons. If a neuron receives a sufficient number of inputs that are not zero, an action potential will be formed. This output will be represented by a neuron that is not zero. Another possibility is that the neuron will continue to be dormant and will not produce any output. It is possible to create an artificial neural network by connecting a large number of neurons together.

These days, you're probably going to hear a lot of people talking about "deep learning" and "neural networks." Although these methods have previously been put to use to find solutions to a range of issues that were beyond the capacity of a computer, the study in this subject is still not yet complete.

The development of modern technical capacities is inextricably linked to the availability of reliable and affordable electrical power. Since humans rely on electricity for their survival and to maximize their productivity, the amount of electricity created and used has been rising significantly throughout the years. The ability of artificial neural networks (ANNs), especially when connected via feedback connections, to produce complex dynamics is attracting increasing interest in a wide range of control applications. The processes of search and optimization are both terms that may be used to describe the process of picking the most effective parameters for a network to employ in order to resolve a certain issue. In order to handle the issue of optimizing the parameters of artificial neural networks (ANNs) for the training of a variety of research datasets, frequent use is made of stochastic approaches such as genetic algorithms (GAs) and particle swarm optimization (PSO) [2]. Greater neural network-based training made possible by genetic algorithms and particle swarm optimization enables the robot to perform more difficult tasks than were before achievable. On the other hand, precise forecasts of future power usage are very necessary for the development of long-term planning and the implementation of energy-saving measures. To improve the performance of ANN training, this study makes use of an algorithm known as the artificial algae algorithm (AAA). The living activities of microalgae were the source of inspiration for the AAA [3]. The computer software was developed with the help of a genetic algorithm that was modeled after the processes of evolution, adaptation, and movement that occur in microalgae. In contrast to the ANN-PSO model, the recommended ANN-AAA approach has produced more encouraging outcomes across a wider variety of benchmark datasets and in the context of an actual world-scale design optimization issue. Because of the link between ANN and it, it is possible to make predictions regarding the amount of electricity that will be used. In this work, power consumption is analysed and depicted by the application of an ANN approach.

Optimization methods may help enhance the training and testing performance of an existing artificial neural network (ANN), but they cannot ensure fair and effective results.

1.1 PROBLEM STATMENT

The creation of two optimization algorithms, known as AAA and PSO, specifically designed for use in artificial neural networks is one of the primary goals of this study. Identifying the algorithms

that are employed should be done for the objectives of optimizing energy use and reporting on it. The accuracy, overfitting, and shrinkage of the ANN are three important problems that arise throughout the process of developing an ANN, and this work presents a fitness function that is based on AAA to solve these problems. The wellness capabilities attempt to better handle any concerns that may occur by taking into account all of these aspects, in addition to a range of other potential problems that may crop up while the ANN is being created, which should allow them to better deal with any potential problems. This tool may be used to identify comfort indices as well as technology related to smart cities.

1.2 BACKGROUND

The goal of swarm intelligence, a branch of AI, is to create AI systems that mimic the cooperative behavior of insects like ants and bees [4]. Computer systems that are impacted in some manner by collective intelligence are the subject of this research. Cooperative efforts by a large number of homogeneous individuals in an environment lead to collective intelligence. "Multitude Intelligence" is the term for the study of computational frameworks, which is enlivened by the "aggregate insight." Many similar agents in nature work together to form a larger whole, resulting in the emergence of aggregate intelligence. Models frequently include schools of fish, swarms of birds, and ant colonies. Instead of being centralized, this information is decentralized, self-sorting, and distributed over a domain. Such frameworks are frequently employed in nature, for example, while hunting for food or avoiding prey, or when relocating a province. It's common for the information to be kept or transferred inside the ground itself, for example, by pheromones in ants, honeybee movement, or proximity in fish and flying creatures, among other techniques. Subfields of the worldview include the following: religion and politics. Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) are two methods for analyzing ants' stigmergy and rummaging behavior. Transformational calculations are flexible systems widely utilized in the creation and improvement of places, and swarm knowledge "calculations" or "procedures" are [4].

1.3 OUTLINE

Swarm intelligence [4] is a subfield of AI that focuses on the design of smart systems that mimic the cooperative behavior of insects like ants and bees. The focus of this research is on computer

systems that, in some way, are modified as a result of the operation of collective intelligence. Collective intelligence is achieved when a large group of individuals with similar characteristics work together to solve a problem in the same setting. The study of computational frameworks is referred to as "Multitude Intelligence," and it is animated by the "aggregate insight." The creation of collective intelligence is the outcome of many comparable agents in nature working together to form a greater whole. This process takes place in nature. It is common practice to use models such as flocks of birds, colonies of ants, and schools of fish. This information is not stored in a central location; rather, it is decentralized, self-sorting, and spread over a domain. These kinds of structures are utilized rather regularly in the natural world, for example, while searching for food or avoiding being hunted, or when moving a whole province. It is usual practice for the information to be stored or transmitted inside the earth itself, for instance, through pheromones in ants, the movement of honeybees, or proximity in fish and flying species, amongst other methods. Religion and politics are two of the subfields that fall under the umbrella of the worldview. Ant Colony Optimization, also known as ACO, and Particle Swarm Optimization, often known as PSO, are two methodologies that may be used to analyse the staggery and rummaging behaviour of ants. Swarm knowledge "calculations" or "procedures" are flexible systems that are extensively used in the construction and enhancement of locations. Transformational calculations are one example of this type of system.

2 LITURETURE REVIEW

This section will explain some of the tactics and methods that are utilized the most frequently for estimating the amount of electricity that will be consumed. Additionally, research that has been done employing these technologies and has been published in the relevant academic journals is being looked into.

2.1 MACHINE LEARNING

Machine learning (ML) is a subsection of artificial intelligence (AI) that use specialized techniques, such as programming and statistics, to construct a model or algorithm for a specific purpose. Some examples of these specialized approaches are deep learning and reinforcement learning.The development of machine learning approaches was spurred on by advances in computer-based learning theory as well as pattern recognition. In the subject of data analytics, the method that is both the most popular and the most effective is to develop models and algorithms in order to make predictions about anything. By employing these models, researchers and engineers, data scientists and analysts may arrive at findings and choices that are dependable and applicable over the long term. It also helps to identify previously unknown patterns or characteristics by using the information that is already available and analysing the trends in the data. One of the most crucial things to do is to pick out the most useful characteristics for machine learning [6]. The reason for this is that the algorithms that are used in machine learning are non-interactive due to the fact that they are developed based mostly on the outcomes of the training data [7]. It does this by analysing data from the past in order to produce accurate forecasts. It is an extremely challenging challenge to devise an algorithmic prediction system that can be depended upon [8]. Techniques from the fields of mathematics and statistics are utilized in machine learning, which is the process of teaching computers to learn from data. It is possible to divide it up into the following four major categories:

a) Supervised learning.

b) Unsupervised learning.

c) Reinforcement learning.

d) Semi-supervised learning.

2.2 UNSUPERVISED LEARNING

Working with data that has not been labelled is required in unsupervised learning, which stands in contrast to supervised learning. In point of fact, labels suitable for such sorts of applications could be difficult to track down. For instance, the data could not be understandable, or the labelling might come at an unreasonable cost.

A little aside: because there are no labels available, it is difficult to define any objectives for the trained model. Because of this, it is not feasible to tell whether or not the findings are accurate. There are a number of approaches one may use to acquire a deeper comprehension of the facts [9], regardless of how challenging it could be to obtain the information.

This particular form of machine learning makes use of data that has not been tagged.

Deciphering the data is the primary purpose, and it is necessary in order to achieve the goal of comprehending the data structure of the algorithms.

Each piece of training data was submitted to the model without labels; the model seeks to infer meaning from the information and characteristics it is given in order to train and assess itself. Learning in an unstructured environment When the model is exposed to the training structure, it learns how to build a mapping from input to output that is specific to that structure.

Make a choice between two different subcategories for a dataset that is unlabelled and only includes the variables "weight" and "height" (male, female). The data is separated into two groups after being processed by cluster algorithms and graphical representations to show how the model learns from the structure of the information it is given.

Clustering and association learning are the two different types of unsupervised learning that may be distinguished from one another.

2.3 SUPERVISED LEARNING

During the training phase of supervised learning, a machine learning model is educated using data that has been labelled. Since empirical data is obtained from the researcher's personal observations

as well as the observations of others, it is typically included in the labelled dataset. It is also required to undertake data preparation in order to improve the quality of the data, fill in any gaps, or optimize it for use in training programs. Performing data preparation is a need.

Both data and a label are included in this form of machine learning, which is often referred to as "learning with a master." The training data begin with the values (X1, Y1, (X2, Y2, and (X3), respectively), which serve as a point of departure (Xn, Yn). The most significant aspect of this project is going to be designing a model that will connect the X labels with the Y labels. Learning through classification and regression are both examples of supervised learning.

It is possible to make a prediction regarding the category that a piece of information belongs to by making use of classification algorithms and data labels. It takes data as input and assigns it to the proper group or category based on that data. When using the binary system of categorization, each picture will be placed in one of two groups (like predict whether the animal is a dog or cat or the mail is spam or non-spam). When employing multi-classification, on the other hand, the picture that is fed into the system will be sorted into one of the categories that corresponds to the number of categories (like classify MNIST handwritten digit, DR levels). This encompasses technologies such as voice recognition, handwriting recognition, biometric identity, picture recognition, and others of a similar ilk. The following are the photo classification algorithms that are utilized the most frequently:

a) Support Vector Machines

b) Decision Tree

c) Feed-Forward neural network

d) Back-propagation network

e) Deep Learning

2.3.1 Support Vector Machine

Vladimir N. Vapnik developed the SVM algorithm as part of the "Summed Up Portrait Method" (SVM) for application in computerized learning and example recognition. During that time period, it was intended to be utilized for the purpose of distinctly dividing information. After four years of research and development beginning in 1962, they were first shown to the general public in 1964. Piece stunt is an approach that Vapnik and his colleagues developed in 1992 to differentiate between directly indistinguishable pieces of hard edge and soft edge information. It wasn't until 1995 that Vapnik et al. introduced delicate edges; ever since then, their use has been broadened to incorporate hard edges as well.

Hard edge SVM can only be used successfully if the data are completely free of errors and are distinct from one another (commotion or exceptions). In the event that errors are made, this indicates that either the edge is too thin or the hard edge SVM is not functioning very well. Vapnik suggested using delicate edge SVM as a solution to this problem. His idea was to fix it by incorporating slack variables into the model. Since support vector machines (SVMs) are effective in regression as well as classification, their use has been growing in frequency:

For the purpose of categorization, a method known as Support Vector Machine (SVM) is utilized.

In many different situations, regression methods like as SVR are utilized.

On the other hand, due to its versatility, it is frequently utilized in activities involving categorization.

2.3.2 Kernel function

A kernel function is what is utilized in order to convert a non-linear space into a linear space. With the use of this method, it is feasible to cut down on the amount of time and effort spent on computing by employing a linear model to sort non-linearly distinct data. In order to carry out this conversion, we will need to make use of the inner product of the new vectors.

Using a linear model, non-linear datasets may be categorized if a binary classification with x inputs and a single feature (non-linear separated data) is transformed to a new representation and then returned to the original form [14].

$$\boldsymbol{g(x) = w^t \emptyset(x) + b} \tag{2.1}$$

$$\boldsymbol{g(x) = \sum_{i \in sv} a_i \emptyset(x_i)^t \emptyset(x) + b} \tag{2.2}$$

$$\boldsymbol{K(x_a, x_b) = \emptyset(x_a)\, \emptyset(x_b)} \tag{2.3}$$

as: K is kernel function. , ∅ Is mapping from X to an (inner product) feature space.

The SVM technique is being utilized to learn certain features (x) as opposed to merely the input attributes x in order to provide decision boundary outcomes that are more complicated, non-linear, and non-deterministic. Before beginning the optimization process again, the previously used formulae need to have x changed to (x), so that they can be revaluated. One of the most fascinating aspects of the technique is the possibility that the kernel would be easy to compute while simultaneously corresponding to a mapping in a space with a large dimension. As a direct result of this, it would be feasible to train an SVM in this space without the requirement of manually calculating the inner product.

2.3.3 Kernel Types

a) A popular technique in the realm of image processing is the polynomial kernel. Incorporating a polynomial kernel function into models by means of support vector machines (SVMs) and similar methods, it is feasible to learn non-linear models. This function represents the similarity of vectors (training samples) in a feature space over polynomials of the original variables.

b) The Gaussian kernel is an option to consider if there is no previous knowledge on the data that may be accessed.

c) When selecting an acceptable feature subset using the GRBF-based technique of feature selection that has been presented, there are two factors that need to be taken into

consideration: The GRBF value is found to be rather high in samples that belong to the same class, but the value is found to be quite low in samples that belong to different classes [15].

d) Laplace One of the most common kernel functions, which is referred to as the RBF Kernel, has the potential to be used in a variety of kernelized learning approaches, including reinforcement learning. It finds widespread use not just in support vector machine classification but also in a great number of other fields.

e) As a direct consequence of this, the hyperbolic tangent function is frequently used.

f) The widespread use of neural networks, which traditionally make use of the sigmoid activation function, is the key factor contributing to the popularity of sigmoid kernels. Sigmoid kernels have a diminishing effect on the significance of excessive correlation. Similar to correlation coefficients, their range is limited, which draws attention to the fact that they move in the same general direction. As a consequence of this, c alters the relative significance of the angle that exists between each input.

g) First-kernel Bessel function: According to the findings of experiments, applying the Bessel kernel function of the first kind to SVR kernel functions is one way to enhance both the predictability and generalization capabilities of these kernel functions.

h) A structure that is based on multi-dimensional kernels may be formed into an Anova radial basis by using the tensor product of one-dimensional kernels as the basis for the formation of the structure.

i) This method may be quite helpful if you are working with a significant quantity of sparse data, since it can help you better organize the data. It is used rather frequently in the process of classifying texts.

2.4 SEMI-SUPERVISED LEARNING

To create a model that is capable of making accurate predictions based on a huge amount of unlabelled data, a hybrid strategy that combines supervised learning with unsupervised learning is utilized. A limited amount of label data is used at the beginning of the supervised learning process

to train the model with the intention of creating the model. This is done in order to achieve a certain goal. When predicting on unlabeled data, the same model is applied via unsupervised learning, whereas when predicting on labelled data, the same model is applied via semi-supervised learning. Data without labels can be utilized for unsupervised learning.

This method is used in many contexts, including web mining, text mining, and video mining. An instance when there is a mountain of unlabelled data but only a tiny amount of data that have been labelled.

It is advantageous for semi-supervised learning to have a smaller quantity of label data in comparison to the number of unlabelled data. This is due to the fact that acquiring label data is both time-consuming and difficult. Because of this, if you implement this technique, you will be able to include a massive quantity of unlabelled data, which will result in an increase in the accuracy of the model.

The term "semi-supervised learning" refers to a learning paradigm that focuses on the study of how computers and natural systems such as people learn in the presence of both labelled and unlabelled input. One example of such a natural system is the human brain. Learning has traditionally been studied either in the unsupervised paradigm (such as clustering and outlier detection), in which all of the data is unlabelled, or in the supervised paradigm (such as classification and regression), in which all of the data is labelled. Both of these paradigms have their advantages and disadvantages. Both unsupervised learning and learning under supervision are investigated in this study. One of the primary goals of semi-supervised learning is to develop algorithms that can make use of the combination of labelled and unlabelled data, as well as to get a better understanding of how combining different types of input might affect the learning behaviour of a system. Because it may be used to supplement supervised learning tasks in situations where labelled data is either hard to come by or costly to obtain, it is particularly prevalent in the fields of machine learning and data mining. When labelled data are unavailable or too expensive, supervised learning tasks might benefit from the use of semi-supervised learning as an alternative. Though it was developed as a quantitative approach to research human category learning, semi-supervised learning has the potential to be applied in settings where the great

majority of the input is visibly unlabelled. This is because semi-supervised learning is a form of unsupervised learning. Some of the most well-known semi-supervised learning models include self-training, mixture models, co-training and Multiview learning, graph-based techniques, and semi-supervised support vector machines. Certain fundamental assumptions must be satisfied for semi-supervised learning to be effective [13].

2.5 REINFORCEMENT LEARNING

It is referred to as "reinforcement learning" when a system is only able to learn as a result of getting reinforcements. These factors might either contribute to the achievement of the system's purpose or work against it. In the context of this discussion, the term "rewards" refers to something favourable, and "punishments" refers to something unfavourable.

Imagine a model who is engaged in the activity of playing video games. After a significant amount of points have been accumulated, the system is eligible to receive rewards. On the other hand, a penalty will be placed on the model if it ends up coming in last place in the competition. As a consequence of this, the model is in a position to determine which actions, in terms of strategy, were successful.

In order to develop both a short-term and a long-term plan, it is necessary to bring together and combine the underlying principles of each of the movements. This indicates that the model will figure out how to play the game and obtain the greatest number of rewards that are available.

The model continues to evolve with each new action and reward combination, and each new action and reward combination contributes to the model's development in some way.

It is a method of machine learning known as reinforcement learning, and it is based on interactions between agents and their environments or opponents. [10] In these exchanges, it is as if the robot is navigating a labyrinth or a chess player is facing off against an opponent. Another way to phrase it is that it is like the robot is playing against an opponent. a set of rules or regulations that the agent is required to follow at all times. The agent has an effect on the environment and is provided with feedback on their behaviour as a result of their participation in activities. In most cases, the environment reacts to player actions in some way, either badly or positively, depending on the

circumstances of the game. The ultimate purpose of the algorithms used in reinforcement learning is to locate an agent approach that maximizes the achievement of a certain goal. There are a number of routes that one might use to accomplish this objective. A decrease in the amount of robot steps required to complete the journey through the labyrinth is beneficial for the robot that is currently within the maze. One alternate strategy is to utilize any and all available resources in an effort to improve one's chances of winning.

When the robot (agent) follows the correct path that leads to our objective, the primary objective is to enhance the behaviour that is considered to result in a reward and reduce the behaviour that is assessed as being likely to result in a punishment (increase the behaviour). To put it another way, if the agent was successful in achieving the reward for the step before it, they will continue down the same path; but, if the reward was unsuccessful, they will select an alternative path. RL method Markov Decision As illustrated in Figure 2.1, processes may be utilized to describe the majority of the events that occur in the actual world (MDPs).

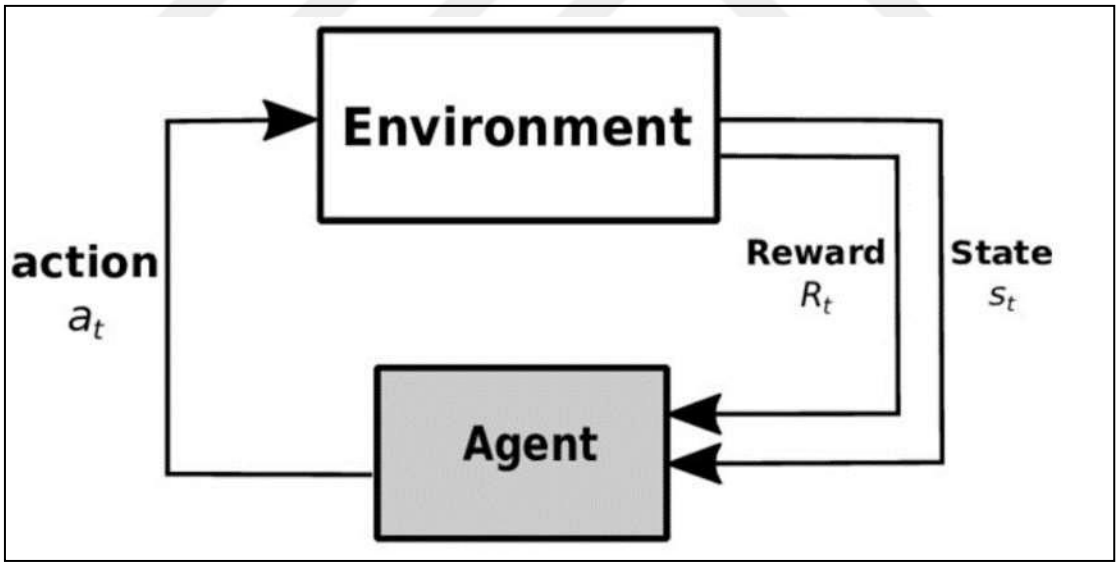

Figure 2-1: Reinforcement Learning

Types of Reinforcement Learning (RL):

a. Positive Reinforcement.

b. Negative Reinforcement.

Knowing Reinforcement Learning will help you comprehend Machine Learning in its entirety because it is a subset of ML. A few of RL's most crucial elements are as:

a. Reward (R): denote to the feedback signal that indicated how well the step t

b. Action (A): is the function of reward (R) and state (S).

c. State (S): that describes the environment.

d. Policy (P): the transfer from the environment to action is (P)

e. Value Function (V): a measurement tool to indicate how good the step is.

f. Model (M): is the demonstration of the agent's environment.

In the book written by Mark Lee [12], it is said that policies determine how learning agents should behave at a specific point in time. Policies may be thought of as a mapping between the circumstances that are believed to exist in the environment and the actions that are recommended to take place when these conditions are present. In the field of psychology, a stimulus-response rule or linkage is meant to describe the relationship between the two. There is a diverse selection of insurances to choose from. In certain circumstances, the policy may consist of something as straightforward as a function or lookup table. On the other hand, it may include computations that are more involved, such as a search algorithm. Agents that learn through reinforcement are equipped with policies that are effective enough to influence behavior on their own. When it comes to regulations in general, there is always the risk of randomness.

When working with an issue involving reinforcement learning, the reward function is what ultimately decides what the end aim will be. Maps each observable condition (or state-action pair) of the environment to a single reward, which represents the intrinsic attractiveness of a certain scenario. A key goal of a reinforcement learning agent is to maximize the overall reward it receives while it is being trained. This is done with the intention of improving the agent's performance. The reward function decides which kinds of occurrences are seen as positive for the agent and which kinds are regarded as unpleasant for the agent. It is not outside the realm of possibility for a biological system's rewards to be connected to both pleasant and negative feelings at the same

time. They are the fundamental aspects of the situation that the agent is confronted with and the ones that are most readily apparent. As a consequence of this, it is highly improbable that the agent will be able to change the reward function. In other words, it may function as a launching pad for a change in the course that the policy will take. When a policy-selected action is followed by a low return, for example, there are a number of ways in which policies may be adjusted to achieve the goal of optimizing rewards. One example of this is the situation It is not uncommon for reward functions to undergo random fluctuations.

In the context of a value function, what counts is what is good in the long run, rather than what is good in the near term in the context of a reward function. The vast majority of states are deserving of the whole benefit that a resident might anticipate receiving from the state at issue throughout the course of his or her lifetime. Values, not incentives, are what drive the immediate, intrinsic attraction of states; nonetheless, values are what define a state's attractiveness over the long run from an environmental perspective. The values of environmental states can provide insight on the desirability of such states over the long run. A state is considered to have a high value if it normally provides a rather small immediate reward, but it is frequently followed by ones that provide significant advantages. On the other hand, it's possible that the opposite is true. It is beneficial to conceive about rewards and values as equivalent to pleasure and pain, respectively, in terms of how pleased or sad one feels about the current condition of one's surroundings. This may be done when contrasting the two concepts of rewards and values. In this kind of expression, value functions unequivocally formalize a notion that is fundamental and widely recognized.

It is impossible to overestimate the significance of incentives; values, such as the expectation of gaining something in exchange for something else, take a second seat. Values can't exist without rewards, and the only reason to evaluate values is to increase one's chances of getting greater rewards for their efforts. It is an important part of one's basic beliefs to take into consideration when one is making decisions and giving them thought. Evaluation decisions serve as a compass for the process of activity selection. Instead of focusing on winning the most money, we ought to work at attaining the most valuable states that are even remotely possible, since this will bring us the greatest overall benefit in the long term. The resulting quantity known as value is the most crucial factor to take into account while making decisions and formulating plans. It's a terrible

tragedy that values are harder to pin down than incentives are since they're so important. Because the majority of incentives are provided by the environment directly, it is necessary to evaluate and reevaluate an agent's values over the entirety of the agent's life cycle. When it comes to the majority of algorithms for reinforcement learning, the component that is considered to be the most essential is an efficient way for reliably estimating values. According to the opinions of a great number of industry professionals, reinforcement learning has made significant progress over the course of the last several decades; yet, the essential function of value estimate is the most important item that we've discovered.

2.6 FEED FORWARD NEURAL NETWORK

In 1943, Warren Mcculloch and Walter Pitts [16], the men who established threshold logic and gave a neural network model, made their first suggestion for a model of the neural network found in the human brain. To better comprehend the brain's structure, researchers are simulating the activity of the brain's external cortex, which consists of billions of neurons engaged in a distributed, parallel communication process to boost learning's speed and robustness. With the hope of better comprehending how the brain functions, this is being carried out. We employ weights that require human touch in the form of manual adjustment in order to function properly.

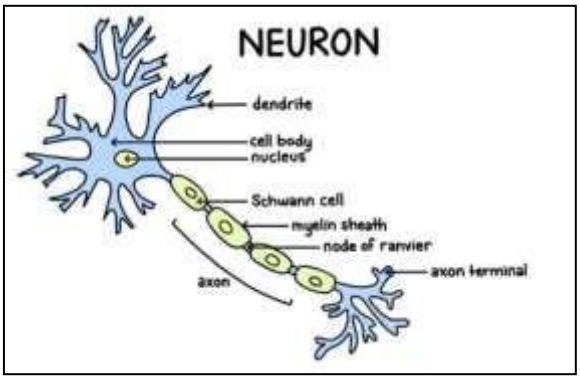

Figure 2-2: Single Neuron of Human Brain

As can be seen in Figure 2.2, each neuron in the human brain is responsible for one of three separate tasks. These functions include input acts that are facilitated by synapses (receive signals from another neuron) It is vital to have knowledge of the chemicals that are present in the cell in order

to determine whether or not the signals that have been gathered are stimulating or inhibiting (soma). The axon is responsible for transmitting the final signal to another cell.

As a consequence of this, the output of one neuron acts as the input to another neuron in the chain, which results in a non-linear activation of each neuron (turning it on or off). The synaptic transmission rate in the human brain is around 100 bits per second, and there are approximately 100 trillion connections that exist between the neurons in the human brain.

In 1949, Donald Hebb came up with the name "Hebbian Learning" to refer to the very first collection of instructional techniques that had been conceived of. This was done in order to define what had been developed. [19] The concept known as "relationship learning" relates to the idea that the significance of a link might be altered based on the assessments of the neurons with which it interacts:

$$\Delta w_{ij} = \alpha a_i a_j \tag{2.4}$$

Rosenblat's 1959 introduction of the single-layer perceptron approach for solving linear classification problems is seen in Figure 2.3. (binary image classification)[20].

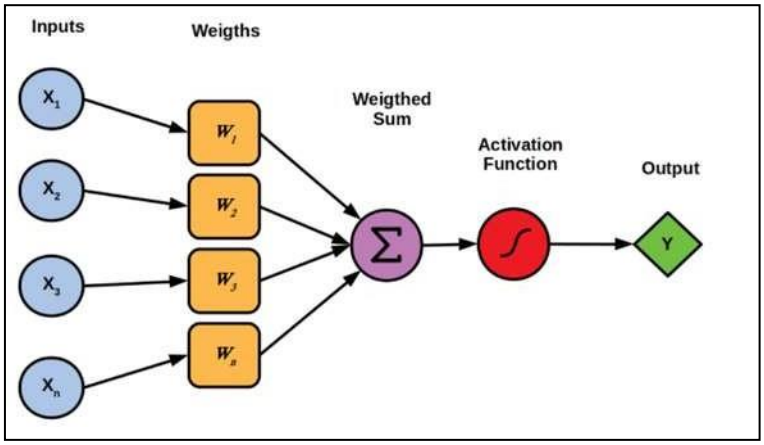

Figure 2-3: Perceptron Architecture

Marvin Minsky made the announcement that the perceptron approach only has a limited capacity when it comes to the non-linear classification (XOR) problem. As a result of his assumption concerning the problem of non-linear classification, a great number of individuals, including himself, were skeptical regarding neuron networks [22]. In 1974, PAUL J. WERBOS made a

significant contribution to the field of artificial intelligence by developing the Backpropagation method using a three-layer perceptron network. This marked the beginning of the renaissance of the neural network.

In the 1980s, there was a surge in interest in the Backpropagation algorithm, which led Rumelhart et al. (1986) to develop an MLP-based Backpropagation algorithm. This algorithm, which can handle non-linearity issues, is made up of three primary components, and it was named after the multilayer perceptron (input layer, hidden layers, output layers), Neural networks have made it possible to find solutions to a broad variety of problems that, in the past, required significant effort to solve but that can now be solved.

A non-recurrent range that incorporates information sources and yields as well as veiled layers, feedforward neural networks are a type of neural network that uses the feedforward computation mode. The output of one layer's worth of processing components is used as the input for the next layer's worth of processing components, and so on, until the output of the last layer's worth of processing components is used to calculate the final yield. This process is repeated until the desired result is reached. This continues until all of the layers have been processed and the yield has been determined. In order to determine a neuron's yield in the yield layer, it is sometimes necessary to perform a limit move task on that neuron.

2.7 BACKPROPAGATION NEURAL NETWORK

There are fully connected nodes in a Backpropagation Neural Network (Figure 2.4). Communication between the nodes is simplified by stacking them in layers. Data collection is the responsibility of the input layers. The output layers are the ones that are responsible for producing the finished result. Internal connections, also known as hidden connections, are what are in charge of bringing the various components of the system together.

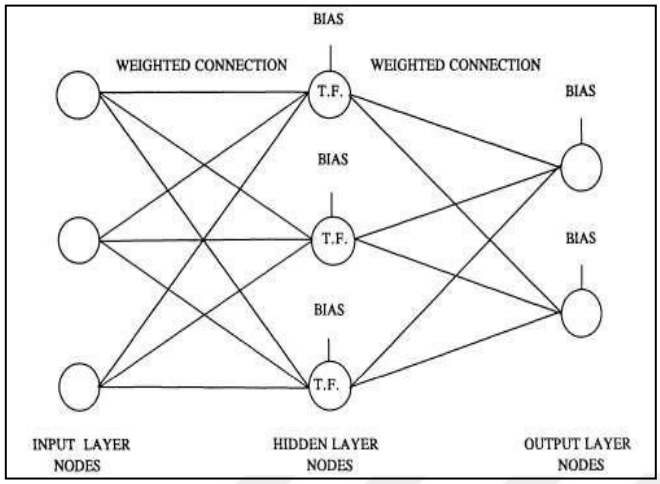

Figure 2-4: Backpropagation Neural network

As part of building a neural network, we take a look at backpropagation overview (Bp), a technique utilized by gradient descent to calculate the gradient and hence minimize the cost function via derivative computing. Supervision learning involves information of the intended output for each neuron, and it is performed as part of the expansion of a neural network , This building is comprised of three completely different types of strata at various levels (input layer, hidden layer, output layer). As a consequence of this, output values are computed by making use of the weights of the currently active nodes, and the backpropagation phase makes use of a differential activation function to compute error values for the output node values. The neuron weights in various areas of the network are recalculated during the update phase depending on the error values that were previously determined. The iterative method shown in Figure 2.5 is applied to each pattern in the training database in order to get the desired results [33].

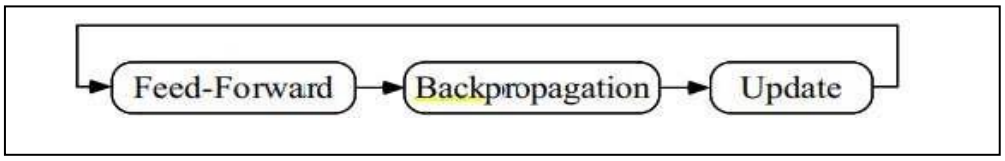

Figure 2-5: The stages of Backpropagation training algorithms

2.8 DEEP LEARNING

Deep learning is another name for machine learning, which is a subset of artificial intelligence that enables computers to mimic human behaviour by utilizing algorithms that have been trained on large amounts of data [34]. Machine learning is a subset of artificial intelligence, but it is also known as "deep learning."

The physical make-up of the human brain may be thought of as a prototype for the process of deep learning. An artificial neural network, often known as an ANN, is a type of structure that engages in deep learning and has the ability to automatically extract features from input data. Deep learning is a technique that was first proposed in the late 20th century. In order to develop a model utilizing deep learning, which was first presented, a substantial number of data is necessary [35].

Because of this, In the past, figuring out how to apply machine learning to extract representative features for pattern recognition required either domain expertise or human engineering, both of which were time-consuming, expensive endeavours. In contrast, a multi-layered deep learning model can fast convert raw data into the representation necessary for pattern detection without the intervention of a human expert. This is possible because of the model's ability to learn from its own mistakes. Since nonlinear processes may be pre-programmed, they can be utilized in the construction of the layers that comprise a deep learning architecture. The more layers that are added, the more complicated and abstract the final representations of the world that we see around us become. This is one of the ways in which a deep learning architecture might pick up on difficult functions thanks to the skills it possesses for deep learning. Deep learning algorithms have made it possible to do a wide variety of previously impossible tasks over the course of human history. Some examples of these tasks include object categorization [36], [37], machine translation [38], [39], and speech recognition [40], [41]. Because there is so much data available in the medical field, deep learning has shown to be quite useful in the healthcare and medical businesses [42]. The application of deep learning in fields such as healthcare and medicine may sometimes provide very positive results. Deep learning has proven effective in part because there is an abundance of high-quality training data, as well as the availability of vast amounts of high-quality training data. This has allowed deep learning to tackle some of the most difficult societal concerns of our day.

In the world as it is right now, there are both 1) vast amounts of data that are available to the public and can be used to train complex deep learning models and 2) adequate computational resources to support these models. For complex deep learning models to be trained on massive amounts of data, enormous amounts of computational power are required. In recent years, it has been essential to satisfy these criteria that powerful computing resources be made available, specifically advancements in GPU performance and the development of methodologies to employ GPU for computation; 3) the availability of deep learning frameworks is also an extremely important factor. Academics from a diverse range of fields are increasingly turning to open-source websites for a variety of reasons. The availability of deep learning implementations like as GoogLeNet [43], ResNet [44], and DenseNet [45] has resulted in a significant reduction in the amount of time required to apply deep learning to applications that are used in the real world.

When it comes to the architecture of deep learning, the standard for neural networks in general consists of an input layer (which contains raw data), a hidden layer (which processes and mixes input data), and an output layer (which contains output data). The use of deep learning becomes more realistic with each new layer of a neural network.

3 SIMILAR WORK AND METHODOLOGY

As we move forward with our own research, we will highlight the benefits of the work done by others in the field of power consumption prediction by utilizing artificial neural networks algorithms and an approach based on artificial intelligence. In the following section, we will analyse the previous work that was done using these approaches.

3.1 PREDICTION OF BUILDING ENERGY CONSUMPTION USING ANN AND PRINCIPAL COMPONENT ANALYSIS

The principal component analysis (PCA) is applied in [57] with the goal of simplifying the model and isolating the inputs that are most important. The research makes use of two different sets of historical data, one of which was gathered by the author from the Energy Prediction Shootout competition, and the other from a university building in East China. Both of these sets of data are updated hourly, and the study makes use of both sets. In addition, for the purpose of comparison, an ANN model and a hybrid Genetic Algorithm-ANN (GA-ANN) model for forecasting were built over the course of this research. The findings of the study indicate that both the iPSO-ANN and the GA-ANN models, when compared to the conventional ANN models, produce results that are more precise. The iPSO-ANN model saves more time throughout the development process than the GA-ANN technique does, making it the most efficient option. The suggested prediction model might be seen as an alternate method for use in situations such as the online forecast of the amount of electricity a building uses, as well as in energy audits.

While gradient-based neural network training approaches perform exceptionally well in terms of local optimization, they are severely lacking when it comes to locating the globally optimal solution. However, the iPSO algorithm is certain that it can search for global optimums and converge quickly. In this study, a hybrid iPSO-ANN model is given for energy prediction in buildings, allowing users to reap the benefits of both methods. The proposed model employs a three-layer feed forward network trained using back-propagation for model initialization. The iPSO technique is then used to optimize the network parameters, taking into consideration the weight and threshold values of the whole network when deciding where each particle should be

placed. Figure 3.1 depicts a possible flowchart for a thorough procedure using the hybrid iPSO-ANN approach.

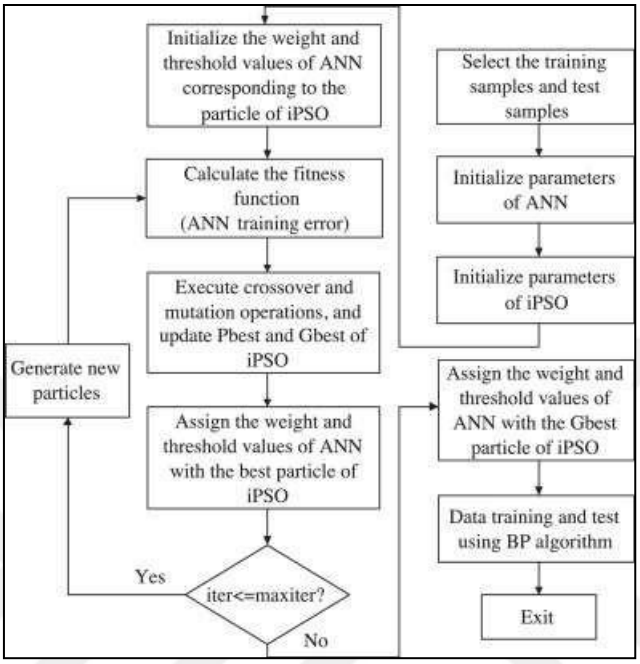

Figure 3-1: A Flow Diagram for a Hybrid iPSO-ANN Prediction Model

The entire WBE electrical power consumption of the building is estimated with the assistance of the hybrid iPSO-ANN model and data set A. Tansig neurons make up the network's one and only hidden layer and are connected to one another in the form of a web. During this experiment, the hybrid iPSO-ANN prediction is tested a total of twelve times to see how well it performs. Figure **3.2** presents the outcomes of the contest in descending order of performance, beginning with the best. For the purpose of evaluating the hybrid iPSO-ANN model's capacity for prediction, two more models are developed and compared with it.

Over the past several years, artificial neural networks (ANNs) have seen widespread application as a tool for estimating the amount of energy a structure would require. In the following, we are going to take a more in-depth look at a three-layer BP neural network that has a regular topology. The very same datasets are utilized in both the training and testing phases of the ANN model. There were a total of 12 guesses, and the results of the best 5 predictions are displayed in Figure 3.3. In addition, the median and the best possible outcomes are shown.

The research that has been done up to this point demonstrates that the two hybrid ANN models frequently attain higher levels of accuracy than the conventional ANN method. Figure 3.4 displays the accuracy ratings given to a wide variety of various prediction models.

Index	1	2	3	4	5	Average	Best
CV	0.0259	0.0254	0.0266	0.0267	0.0262	0.0262	0.0254
MAPE	0.0163	0.0162	0.0169	0.0166	0.0171	0.0166	0.0162
Time(s)	20.7	23.2	25.3	23.7	25.3	23.4	20.7

Figure 3-2: Predicted building electricity loads using iPSO-ANN

Index	1	2	3	4	5	Average	Best
CV	0.0352	0.0325	0.0340	0.0333	0.0339	0.0339	0.0325
MAPE	0.0222	0.0211	0.0238	0.0213	0.0224	0.0222	0.0211
Time(s)	11.6	12.2	10.4	11.9	7.3	10.7	7.3

Figure 3-3: Building power consumption forecasted via artificial neural network

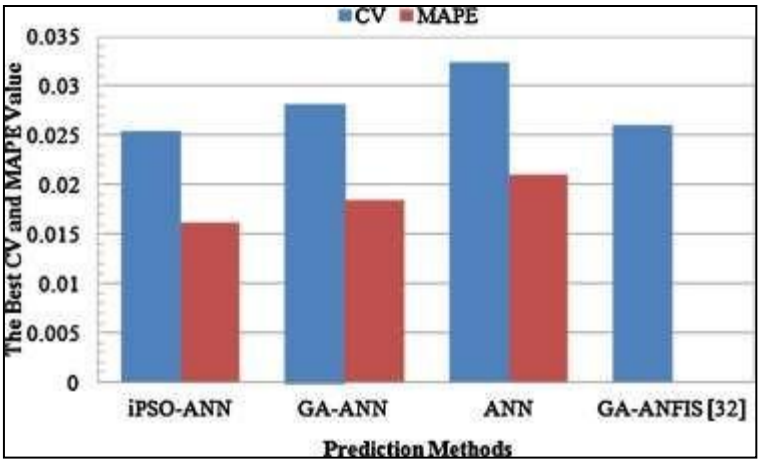

Figure 3-4: Statistical analysis of prediction accuracy using a variety of models

3.2 INNOVATIVE ENERGY-SAVING EVOLUTIONARY ALGORITHMS IN NEURAL NETWORKS FOR LONG- AND SHORT-TERM MEMORY

The proposed method for estimating the amount of energy a building would need in the future [58] is organized into three separate layers: data collection and storage, data pre-processing, and data analysis. Together, the genetic algorithm (GA) and the long-short term memory (LSTM) neural network model that make up the data analytics layer produce an accurate and reliable energy prediction model. In order to simulate the time series of energy consumption data, we make use of a recurrent neural network that is built on top of the LSVM method. A genetic algorithm (GA) is used to determine the optimal design for LSTM neural networks so that both the accuracy and resilience of predictions may be increased. The LSTM architecture is determined by hyper-parameters such as the number of LSTM layers, the number of neurons included inside each LSTM layer, the dropping rate of each LSTM layer, and the pace at which the network is able to acquire new information. Concurrently, an investigation of the implications of the historically significant weather profile and the temporal horizon of past information is being carried out. The suggested method for forecasting the amount of energy used by a building is currently being put to the test in two actual schools, and the results are being compared to forecasts. In terms of accuracy and resilience, the outcomes of the experiments show that the proposed adaptive LSTM neural network outperforms both the state-of-the-art feedforward neural network as well as the LSTM-based prediction models. The performance of LSTM networks with hyper-parameters determined by grid search, Bayesian optimization, and PSO is superior to that of LSTM networks with hyper-parameters selected by any of the other available methods. This kind of accurate prediction of energy consumption has the potential to be used in a variety of contexts, including the day-to-day management of building energy, the decision-making of facility managers, the construction of building information models, the operation of zero-energy buildings, the mitigation of climate change, and the circular economy.

The system that is being suggested to anticipate energy consumption is composed of three subsystems, which are referred to as data collection and storage, data preparation, and data analytics. The organizational structure of the suggested system for predicting energy use is depicted in figure 3.5.

The optimal GA parameters to utilize may be determined by using relevant prediction performance, such as convergence performance and final MAE value. The effectiveness of the LSTM neural network is evaluated both with and without the weather profile as an input dataset, as well as with and without a predetermined time horizon. Both of these scenarios are considered. Evaluation of the adaptive LSTM neural network's prediction performance with various reference parameters using the optimal design that has been selected is what determines the network's level of efficacy.

As can be seen in Figure 3.6, the performance of the GA's ability to converge is evaluated utilizing two distinct structures in order to simulate varied selection probabilities, crossover probabilities, and mutation probabilities.

We present two reference prediction models for comparison in order to highlight the usefulness of GA in identifying the most appropriate hyper-parameters for the LSTM network. In other words, the ideal hyper-parameters for the LSTM network are determined with the assistance of Bayesian optimization and PSO, and the results of this determination are reported. The use of the grid search approach has the potential to ensure that the most effective combination of hyper-parameters is located. However, if grid search were used for the hyper-parameter selection, 2,948,760 distinct LSTM architectures would be trained, and each design would be trained and evaluated five times (see Figure 3.7). This requires far too much processing time than is reasonable. As a consequence of this, it is not deemed a reference model, and as a result, it is disregarded. Figure 3.7 provides a concise summary of the assessment of the performance of the forecast. In comparison to the PSO and Bayesian optimization methods, which need 60 and 90 iterations, respectively, the GA optimization method could attain convergence after only 40 iterations. Even though the number of LSTM layers identified via Bayesian optimization and PSO is the same as that determined by the proposed GA, the number of neurons contained inside each LSTM layer, the rate at which neurons drop out of each layer, and the learning rate are all unique. Because of this, the GA-LSTM prediction model that is proposed yields results that are somewhat better than the results achieved using Bayesian optimization and PSO, which both reach comparable levels of performance. This indicates that the R2 value of the LSTM network that is determined by Bayesian optimization and PSO results in a decrease of 0.42–2.50 percentage points, an increase of 0.36–14.7 percentage

points in RMSE, and an increase of 2.47–31.4% in MAE, depending on the choices that are made regarding the parameters. This is due to the fact that it is a distinct possibility that the optimisation function for LSTM hyper-parameters does not adhere to a Gaussian distribution. This is a foundation upon which Bayesian optimisation is based. Furthermore, when the optimization issue is continuous, PSO performs better than GA, and GA performs better when the optimization problem is discrete. Because there is a predetermined search space, it is much easier to determine what the correct value should be for the LSTM hyper-parameter. Therefore, the suggested GA-determined LSTM is superior to both Bayesian optimization and PSO in terms of performance.

With the help of a system that is driven by an adaptive LSTM neural network, it is possible to make predictions regarding the amount of energy that a building consumes that are both accurate and reliable. It has been suggested that the proposed technique for energy forecasting should incorporate three significant advancements that are enhancements made over currently available deep learning models. Several examples of these are as follows:

a) The efficiency of the LSTM neural network can be attributed to the fact that its neurons are connected to one another in a recurrent fashion. As a result, it could reveal every aspect of the temporal connection present in the data set.

b) Because GA is so effective at discrete optimization, it was used to choose the values for the LSTM network's hyper-parameters. The number of neurons in each LSTM layer, the number of LSTM layers, the dropping rate of each LSTM layer, and the learning rate of weighting matrices and bias are some of the optimal hyper-parameters of the LSTM network. Other optimal hyper-parameters include the number of layers in the LSTM network. Because the hyper-parameters of the LSTM network are customized to each specific energy dataset, it is possible for the network to be quickly adapted to new kinds of data.

The building energy forecasting system that has been proposed is currently undergoing testing in two educational establishments as part of a pilot program. The system is currently being trained on one year's worth of weather and energy data from the past, and it will be tested on the profile from the subsequent year.

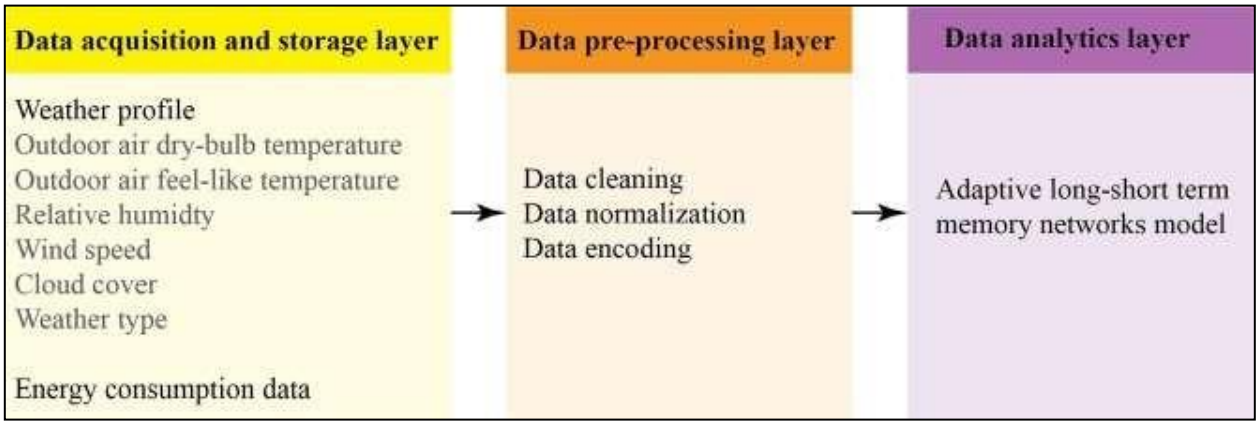

Figure 3-5: framework for energy forecasting system proposal

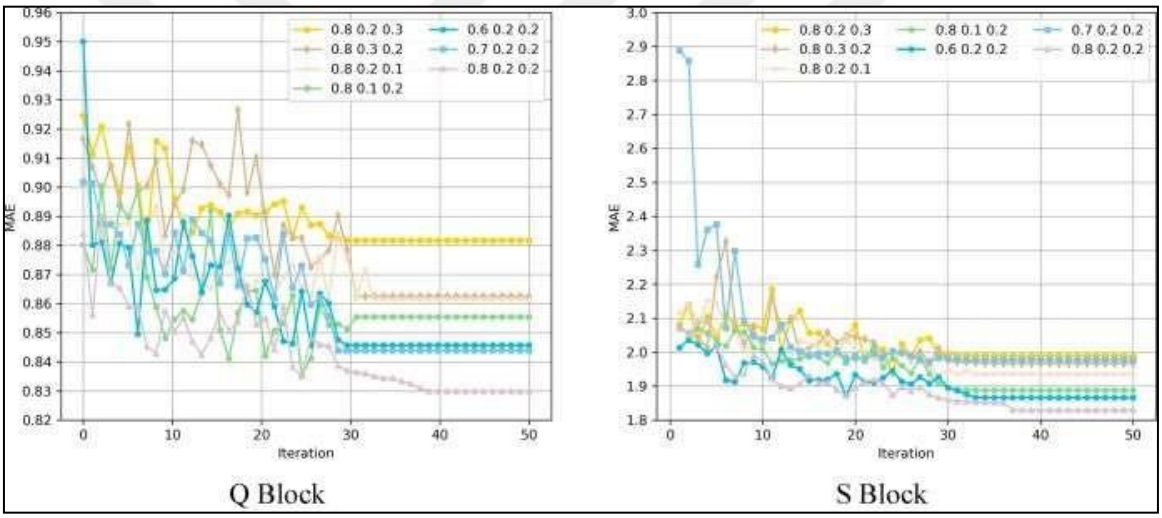

Figure 3-6: Variations in GA parameters and their effects on MAE convergence

Building	Change of LSTM architecture	Number of LSTM layers	Number of neurons in each layer	Dropping rate	Learning rate	R^2 (%)		RMSE (kW)		M
						Train	Test	Train	Test	T
Q block	Optimal	2	[60, 60]	[0.3, 0.5]	0.01	92.3	86.7	1.83	2.75	0
	BO	2	[40, 50]	[0.3, 0.7]	0.03	90.0	86.3	2.10	2.78	0
						↓2.50	↓ 0.46	↑14.7	↑ 1.09	↑
	PSO	2	[70, 70]	[0.4, 0.5]	0.01	91.4	86.4	1.94	2.76	0
						↓0.98	↓ 0.35	↑6.01	↑ 0.36	↑
S block	Optimal	2	[70, 40]	[0.2, 0.2]	0.01	95.6	86.6	3.24	4.80	1
	BO	2	[60, 40]	[0.1, 0.1]	0.001	94.9	85.1	3.52	4.92	1
						↓0.73	↓ 1.73	↑8.64	↑ 2.50	↑
	PSO	2	[60, 30]	[0.2, 0.5]	0.01	95.2	85.8	3.32	4.87	1
						↓0.42	↓ 2.47	↑2.47	↑ 1.46	↑

Figure 3-7: In Bayesian optimization, PSO and GA are employed to hasten the convergence of the mean absolute error (MAE)

3.3 ARTIFICIAL NEURAL NETWORK MODELS FOR ENERGY CONSUMPTION FORECASTING

Panagiotis Karampelas and colleagues [59] developed a number of different artificial neural network models with the intention of predicting the use of energy in the foreseeable future. In order to develop each model with the highest possible generalization potential, a variety of architectures, learning algorithms, and transfer functions were utilized. This was done to guarantee that the models were as similar as possible. Throughout the training, validation, and testing processes, data taken from actual life situations served as input and output. Several different models of neural networks were built, and then they were compared to one another to see which one performed the best. The very last thing that needed to be done was to apply the chosen ANN model to the problem of forecasting Greece's future energy use.

The end objective is to build a neural network in a laboratory that is capable of predicting Greece's need for energy. It was decided that the output of the neural network would be the final energy consumption, and the selection of the four criteria was based on the likelihood that each of those factors would have a significant impact on the final energy consumption. The Hellenic National Meteorological Service [60] provided the data for the annual ambient temperature T, the Public Power Corporation S.A. [61] provided the data for the installed power capacity C and the annual per resident electricity consumption R, and EUROSTAT [62] provided the data for the gross domestic product G, which resulted in the lira [63] being the outcome variable.

In the course of this investigation, we developed and analysed a variety of FF ANN models. Because of their better generalization capabilities, they were chosen from a huge pool of potential combinations of two learning algorithms, three transfer functions, and a number of distinct structures. It was discovered that the learning techniques Gradient Descent and Levenberg-Marquardt were being utilized. The number of neurons in each hidden layer ranged from two to forty, while the number of hidden layers used ranged from one to five. The transfer functions used were the hyperbolic tangent sigmoid, the logarithmic sigmoid, and the hard-limit function (Table 3.1).

When we were ready to train the resultant network models, we turned to the MATLAB® neural network tools [63]. The neural network models were trained and evaluated using input and output data collected over the course of the past fifteen years (1991-2005). Before determining the amount of validation error for each iteration of the training process, twenty percent of the random samples that had been collected were culled from the training set. Either the maximum number of epochs (the presentation of the set of training data to the network and the computation of new weights and biases) were reached, or the root mean square error (RMSE) between the actual output and the planned output was less than 1%. In the end, the estimated value of energy consumption was evaluated in light of both the values obtained during training and the values obtained from unfamiliar settings.

The selected ANN model will be used to calculate energy consumption forecasts for 2007 and 2008. We can examine how the 2007 and 2008 ANN energy consumption estimates relate to the

actual amount of energy that was used in the real world by looking at Table 3.2. It has been proven that the results that the suggested ANN model generates are extremely consistent with the outcomes that occur in the actual world; this demonstrates both the model's feasibility and a sufficient degree of accuracy.

The research presents an artificial intelligence system that is based on neural networks in order to make projections about electricity usage in Greece. We used actual input and output data captured during the training, validation, and testing processes. These data, which obviously have a significant impact on the amount of energy that is utilized, were captured. Forecasts have been produced with results that are quite near to those obtained from actual occurrences, and these predictions have been made several years in advance. Because accurate projections of energy consumption affect capital investment, environmental quality, revenue analysis, and market research management, in addition to maintaining supply security, the technique that has been suggested may be useful in the effective implementation of energy regulations.

Table 3.1: The Models were Designed by Ann

Structure	**Learning Algorithm**	**Transfer Function**
−1 to 5 hidden layers	−Gradient Descent	−Hyperbolic Tangent Sigmoid
−2 to 40 neurons in each hidden layer	−Levenberg-Marquardt	−Logarithmic Sigmoid
		−Hard-Limit

Table 3.2: Final Ann Model Proposal Outcomes

Year		2007	2008	2010	2015	2020
Final Energy Consumption (1000 TOE)	**ANN results**	23,463	23,341	25,307	28,478	32,293
	Real results	22,552	22,834	-	-	-

3.4 METHODOLOGY FOR PREDICTING HOUSEHOLD MICROGRID ENERGY USE BASED ON LEARNING

An integral part of the country's economic and social growth is attributable to domestic energy use. Decision making and home electricity demand forecasts can benefit from very precise forecasting, which is essential for the efficient management of power system operations. However, human behaviour, seasonal change, and other social factors also impact residential load characteristics. As a result, there will be a lot of unpredictability in the workload. The operator of the electrical grid faces a significant challenge in attempting to acquire reliable predictions of future load. For residential short-term energy consumption forecasting, the authors of [64] offer a recurrent neural network-based LSTM, GRU, Bi-LSTM, and Bi-GRU based learning technique. These are the findings of a simulation run using a 9-month dataset of energy use made up of 30-minute intervals, collected from a central-Norway home prosumer microgrid. Numerical findings on the provided load data set demonstrate that the Bi-GRU model outperforms its competitors.

Models in a learning architecture can take use of the load history data in a hierarchical fashion to improve their performance. Several layers of the model are responsible for the various degrees of learning. Higher-level elements of the load data have been used to train the model properly. Here, we go down each of the several types of learning models in further depth.

This is Ronald and David's [65] RNN algorithm, created in 1989. It is a subset of neural networks that can translate input-output data in a sequential fashion over a period of time appropriate for making predictions. When compared to a neural network, RNN performs far better. Figure 3.8 depicts the skeleton of an RNN.

The architecture includes a multiplicative gate that supplies a memory cell for storing and retrieving data for later use. This solves the problems with vanishing weight gradients. The LSTM memory element is constructed from an input gate, forget gate, and an output gate. In LSTM, the forget gate is not integrated into the network like in a regular RNN, but rather functions as a standalone memory unit.

The latest iteration of GRU's network architecture is implemented. In this design, we have two gates: the update gate and the reset gate.

In order to concurrently record past and future data from a pattern, a bidirectional technique was designed. The learning model's bidirectional mechanism consists of two distinct processes. A forward neural network (FNN) first reads the input data in a left-to-right sequence, and then a backward neural network (BNN) does the same thing in reverse.

For a particular data set, the effectiveness of a learning framework is evaluated using metrics such the mean absolute percentage error (MAPE), mean squared error (MSE), and root mean squared error (RMSE). Results of learning models applied to energy consumption data are shown in Figure 3.9. Models for each home, both as trained and as tested, are shown. From what can be seen in fFigure 3.10, the Bidirectional-GRU algorithm achieves respectable results in both training and testing, especially when compared to other models for both houses. Maximum non-linearity in the data set of energy consumption is captured by this model. Data factors, such as time, location, and season, might influence the model performance measurements. Results from learning do not take into account the effect of seasonal unpredictability. It's possible that other algorithms' performance might improve if the data sample length was increased to account for seasonal uncertainty and seasonal changes.

Blackouts, power outages, fault diagnostics, operation management, and maintenance are just a few of the issues that arise when renewable energy sources are integrated into a smart home prosumer microgrid. Throughout the decades before, scientists have attempted to develop clever techniques for addressing these issues. Additionally, the electric power system has been upgraded from its antiquated structure to one that is more easily controlled and monitored. The government and the energy company have also built sophisticated monitoring and metering systems in every

home and commercial structure. In this study, we utilize a historical data collection and a learning-based framework to predict the energy use of a community of dwellings. Five learning algorithms that fall under the umbrella of "deep learning" have been analyzed in this work. In terms of performance in a learning framework, the Bi-GRU algorithm appears to have fared well.

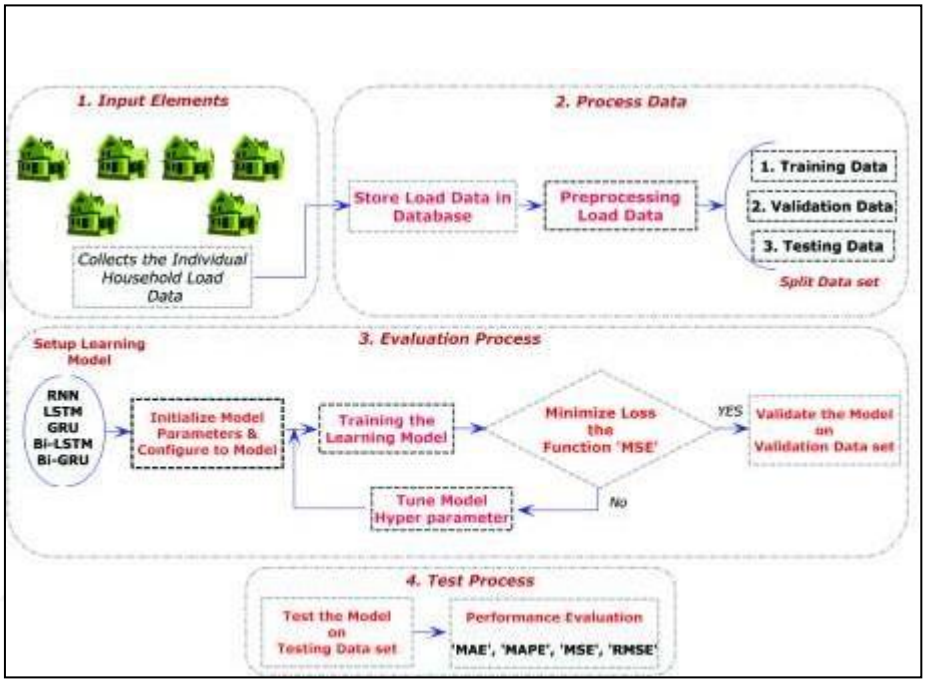

Figure 3-8: framework

Sr.No	Hyperparameters	Values
1	Neurons in input layers	12
2	Neurons in 1^{st} hidden layers	64
2	Neurons in 2^{nd} hidden layers	8
3	Neurons in output layers	1
4	Epochs	100
5	window size	12
6	Learning rate	0.001
7	Optimizer	Adam
8	Loss function	mse

Figure 3-9: Sr. No Hyperparameters of Learning Models

House No.	Learning Model	Train				Test			
		MAPE	MAE	MSE	RMSE	MAPE	MAE	MSE	RMSE
H1	RNN	0.4941	0.0979	0.0307	0.1753	0.5147	0.0814	0.0184	0.1357
	LSTM	0.4661	0.0957	0.0314	0.1772	0.4935	0.0796	0.0185	0.1361
	GRU	0.4582	0.0946	0.0297	0.1723	0.3898	0.0711	0.0169	0.1303
	Bi-LSTM	0.4698	0.0958	0.0312	0.1767	0.3713	0.0759	0.0179	0.1338
	Bi-GRU	**0.4176**	**0.0914**	**0.0295**	**0.1719**	**0.3226**	**0.0653**	**0.0168**	**0.1296**
H2	RNN	0.8763	0.1622	0.1134	0.3368	0.9511	0.1665	0.1038	0.3222
	LSTM	0.8687	0.1620	0.1131	0.3363	1.0540	0.1733	0.1020	0.3194
	GRU	0.9426	0.1644	0.1644	0.3409	1.0863	0.1779	0.1038	0.3222
	Bi-LSTM	0.9103	0.1678	0.1186	0.3444	1.0636	0.1827	0.1106	0.3325
	Bi-GRU	**0.8415**	**0.1561**	**0.1110**	**0.3332**	**0.9387**	**0.1669**	**0.1001**	**0.3165**
H3	RNN	0.3071	0.1741	0.0728	0.2698	0.3008	0.1584	0.0733	0.2708
	LSTM	0.3058	0.1743	0.0737	0.2715	0.2777	0.1527	0.0726	0.2695
	GRU	0.2926	0.1707	0.0725	0.2693	0.2646	0.1487	0.0718	0.2680
	Bi-LSTM	0.2981	0.1724	0.0729	0.2700	0.2621	0.1498	0.0726	0.2694
	Bi-GRU	**0.2893**	**0.1695**	**0.0724**	**0.2692**	**0.2482**	**0.1451**	**0.0709**	**0.2663**
H4	RNN	0.4107	0.2012	0.0957	0.3094	0.3083	0.2955	0.2313	0.4810
	LSTM	0.4114	0.2012	0.0950	0.3082	0.3089	0.2954	0.2332	0.4829
	GRU	0.4094	0.2011	0.0944	0.3073	0.3055	0.2958	0.2420	0.4919
	Bi-LSTM	0.4285	0.2048	0.0948	0.3080	0.3223	0.2946	0.2303	0.4799
	Bi-GRU	**0.4074**	**0.2001**	**0.0925**	**0.3042**	**0.2958**	**0.2926**	**0.2284**	**0.4779**
H5	RNN	0.2783	0.2365	0.1273	0.3568	0.2955	0.1737	0.0733	0.2709
	LSTM	0.2804	0.2410	0.1293	0.3596	0.2706	0.1633	0.0699	0.2644
	GRU	0.2850	0.2420	0.1295	0.3599	0.2669	0.1605	0.0700	0.2647
	Bi-LSTM	0.2715	0.2378	0.1297	0.3602	0.2575	0.1578	0.0691	0.2630
	Bi-GRU	**0.2593**	**0.2333**	**0.1231**	**0.3508**	**0.2189**	**0.1407**	**0.0672**	**0.2593**
H6	RNN	0.4341	0.1255	0.0453	0.2128	0.3872	0.1269	0.0517	0.2274
	LSTM	0.4223	0.1243	0.0456	0.2136	0.3751	0.1245	0.0492	0.2218
	GRU	0.4027	0.1221	0.0448	0.21171	0.3570	0.1233	0.0503	0.2242
	Bi-LSTM	0.3930	0.1218	0.0451	0.2125	0.3718	0.1254	0.0502	0.2241
	Bi-GRU	**0.3858**	**0.1193**	**0.0444**	**0.2107**	**0.3492**	**0.1242**	**0.0520**	**0.2281**

Figure 3-10: Brief overview of energy consumption projections

3.5 BRIEF

Numerous ANN algorithms are utilized in order to generate accurate forecasts of future energy consumption. This chapter discusses and illustrates a variety of different algorithmic approaches, including but not limited to Gradient Descent, Levenberg-Marquardt, hybrid genetic algorithm (GA), long-short term memory (LSTM) neural network model, and Hybrid iPSO-ANN. Their application in the real world and theoretical investigation are both extensively covered here, along with pertinent results and new perspectives.

3.6 DATABASE

We will utilize 2075259 measurements that were obtained at a house in Sceaux, France (which is located 7 kilometers from Paris, France) during the months of December 2006 and November 2010 in order to test and train the suggested model. Sceaux is located in France. The suggested model was tested and trained using data from a dataset known as the Individual Household Electric Power

Consumption Data Set, which was downloaded from the UCI Machine Learning repository (47 months).

The date consists of time and date attributes in addition to other pieces of information. Forecasting the use of energy in residential buildings will involve the utilization of sub-metering, global active power, and global reactive power.

3.7 ANN-PSO

When Kennedy and Eberhart were developing PSO in 1995 [64], they found that the cooperative foraging techniques of flocks of birds and schools of fish were a source of inspiration for them. A community of users is utilized in the PSO heuristic optimization approach, which is known as PSO.

PSO has been utilized in the training of a wide variety of neural networks, such as recurrent networks [65] and product unit neural networks [66]. One of the early applications of PSO was found in the training of feed-forward neural networks [66, 67].

The PSO approach looks for a state space by employing the topology of the most advantageous neighboring regions and a weight representing inertia [68] (Figure 3.11). Using this method, initial velocities are arbitrarily given to a certain number of particles in order to construct the population and conduct research into the state space. The fitness function takes into account positional information from the target space for each individual particle. This makes it possible to map the input space onto the output space. Following each repetition, the location of each particle is modified according to its velocity in accordance with the calculations that are stated in [69].

$$x_i = x_i(t) + v_i(t+1) \tag{3.1}$$

$$v_i(t+1) = a * v_i(t) + b * r_1 \left(x_{pb}(t) - x_i(t)\right) + c * r_2(x_{gb}\ (t) - x_i\ (t)\ ) \tag{3.2}$$

The moment of inertia can be described as (a). Where xib represents the best location for the individual, xgb represents the best position for the group, and r1 and r2 represent random deviations that are evenly distributed from those two positions (0,1).

Particle swarm, in contrast to other searching methods such as gradient descent, does not become stalled on local minima or peaks when looking for a solution. The random variables r1 and r2 bring about an element of uncertainty, which makes it possible to conduct a more in-depth analysis of the goal space. The initial velocity of the particle provides it with "inertia," which enables it to continue its exploration even after it has identified a stable minimum or maximum value. When the location undergoes a change, fresh fitness values are computed at the corresponding point in time. Any new personal bests that have been achieved as a direct result of the promotion to a higher rank are carefully documented and stored away. What is true for the greatest in one part of the globe is also true for the best in other parts of the world. The algorithm is complete once the requirements for terminating it have been satisfied. Limiting criteria are frequently formulated in the form of a maximum number of iterations that must be completed before the process is terminated.

A collection of particles may converge to local minima or maxima due to initialization or the attraction towards the local or global bests before they have arrived at their ultimate destination. This can happen even though the particles have travelled a great distance. After then, in order to continue the search, it is best to "mutate" the individuals that performed the worst by providing them with new beginning positions and velocities.

The following are the actions that make up the PSO training algorithm:

Determine the topology of the network, the parameters of the PSO, and the fitness function.

Method 2: Draft the possible weight solution using the equation x = wit, who. First, we will define wit as the weight difference between the input nodes and the hidden nodes, and we will define who as the weight differential between the hidden nodes and the output nodes.

Third, within the range that is permitted, randomly decide the position (weights) and velocity of each individual particle (weight change in a single iteration). Create a ranking system for the fitness of each particle by using the error function we developed before.

First, determine which of the particles' greatest fitness values can be relocated, and then determine which of the best fitness values can be relocated globally.

Fifth, recalculate the swarm's current velocity and position with the help of equation (3.1). (3.2).

Proceed to step 6 if the condition that should cause the process to end is not met. If this is not the case, the iteration should be stopped and the global best solution should be used to establish the ideal weight.

It is vital to determine the settings for the training algorithm's parameters before commencing the process of training the model.

In this thesis, PSO is utilized with n=10, C=1.5, and C=2.5 respectively. The weight that is caused by inertia decreases in a linear fashion, going from 0.9 to 0.4. When establishing the initial weights, a value chosen at random from the range [0, 1] is utilized. To avoid any misunderstandings, the termination criteria will be explained as follows: Either (1) the training error is less than the threshold e, or (2) the maximum number of iterations, Itermax, has been achieved. There can be only one of these two outcomes.

PSO combines random variables, inertia speed, and mutations to explore exceptional solutions and converge on the best personal position tracking. This allows it to identify the greatest global/personal position monitoring available. The PSO is shown here in the form of a flowchart in Figure 3.12.

PSO's potential applications in the field of artificial neural network education have lately attracted the attention of a number of academics (ANNs). The training and education process of an ANN that makes use of PSO has the potential to boost productivity while simultaneously cutting expenses. Using the PSO approach is one strategy to improve a neural network's performance. One way to do this is to determine the ideal weight and bias values for the neural network's training and testing data sets. For this reason, studies on ANN-PSO and the ways by which to train data from a selected database were carried out as part of a research project.

The PSO-based NN model diagram may be found depicted in Figure 3.13. Data on energy use was utilized to train neural networks, which were then put to use in forecasting energy consumption. PSO is applied during the training of neural networks, as well as throughout the process of fine-tuning and optimizing the weights and biases of these networks. This process determines the

locations of the particles and sends the results on to the next learning phase so that they may be utilized there. When a neural network that employs NN is trained using PSO, the particles that make up the population do their best to identify the appropriate weights and biases. This is because PSO utilizes NN. Iterations of the PSO algorithm are utilized in the pursuit of finding the optimum weights for neural network training in order to fine-tune local and global locations. The analyzer will indicate whether or not the optimal outputs have been calculated or whether or not more training is required.

The algorithms that are searching for a set of link weights will be trained using seventy percent of the data that is available. After the training phase is over, the remaining thirty percent of the dataset is taken and utilized as the testing set. This is done so that the results can be compared and the work can be validated.

For the purpose of evaluating the PSO algorithm, data pertaining to training accuracy, sensitivity, and specificity; testing accuracy; training error; and testing error are acquired.

In this thesis, an artificial neural network with many layers is utilized, and the results are quite encouraging. An input layer, an output layer, and a concealed layer are the kinds of things one should look for. This layer illustrates the method by which data is transported throughout the network. The number of input nodes in this layer is equal to the number of characteristics in the dataset, and it is represented by the layer below it. The number of input nodes that corresponds to each attribute in the dataset is specified explicitly. Two of the parameters that determine the layout of the output layer of an artificial neural network (ANN) are the number of output classes and the type of dataset that was used to train the ANN. Because the number of target outputs in the dataset is equal to the number of nodes in the output layer when dealing with a regression problem, this means that there are no excess nodes. For the sake of this thesis, the numbers 3, 6, and 9 were chosen for their comparability despite the fact that a hidden layer can include any number of hidden nodes.

We divide the data into two distinct groups: the samples that are used for training and the samples that are used for testing. Data used for instruction are referred to as instructional data. Training

samples are used to instruct the network, while testing examples are utilized to evaluate the degree to which the network has retained the information.

In order to train a neural network, we must first choose the weights that are ideal for the network, and then we must search over the weight space in order to find a solution that minimizes the error function. PSO generates possible solutions using a mechanism known as forward propagation and assesses these potential solutions through the use of an error function. After that, the error value is utilized as a fitness function in order to direct the particle toward a more optimal solution. After a significant amount of repetitions, the best particle on a global scale is identified and then connected to the essential trained network.

Artificial neural networks, also known as ANNs, are intelligence tools that swiftly learn patterns and anticipate the solutions of complicated issues in high dimensions by using signals from biological neural networks seen in humans and other animals. ANNs utilize these clues to model their own behaviour. Even with a noisy and complicated dataset, they are able to map inputs to outputs. MLPs, also known as multilayer perceptron's, are a sort of feed-forward artificial neural network that is both straightforward and reliable. An input layer, one or more hidden layers, and an output layer are the three layers that make up the conventional architecture of a multi-layer perceptron (MLP) network. The data is received by the input layer, which then relays it to the exposed neurons of the hidden layer.

In order to train a database that was utilized for the purposes of this thesis and was built on three distinct hidden layers, an artificial neural network particle swarm optimization was employed. The database was stored in an excel sheet (3, 6, 9). The construction of the MATLAB code that was used to train the aforementioned data was guided by the following techniques.

1. *get and read the input file data*
2. *read the file data*
3. *measure the size*
4. *specify total number of hidden layers*
5. *Create the object of feedforward network*
6. *Set up Division of Data for Training, Validation, Testing (trainRatio = 70/100, testRatio = 30/100)*
7. *configure the neural network object with input dataset features*
8. *Compute the value of maximum fitness number*
9. *Compute the PSO initization using upper and lower bounds*
10. *Start the PSO algorithm to estimate the best particle for ANN training with fitness value computation*
11. *define the objective function*
12. *pso initialization: initial population, velocity, initial pbest, initial gbest, global weight C1 and C2, pso position update, process handling boundary violations, process evaluating fitness*
13. *finding out the best particle*
14. *storing best fitness, iteration count*
15. *calculating tolerance*
16. *Final neural network model*
    a. *Compute Training Accuracy*
    b. *Compute the sensitivity and specificity by corss validation*
    c. *Perform the cross validation*
    d. *Computing the TP, FP, TN, FN, Precision and Recall Rates, Accuracy*
    e. *Calculation of MSE*
    f. *Compute testing Accuracy*

Figure 3-11: Process of the MATLAB code consturction

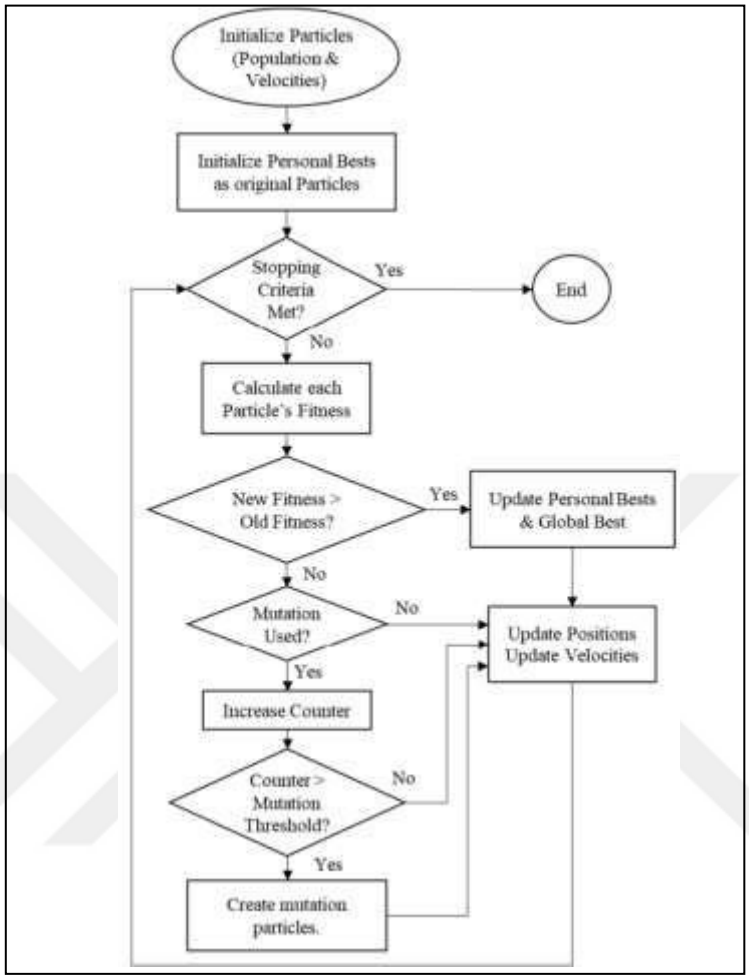

Figure 3-12: flowchart of PSO

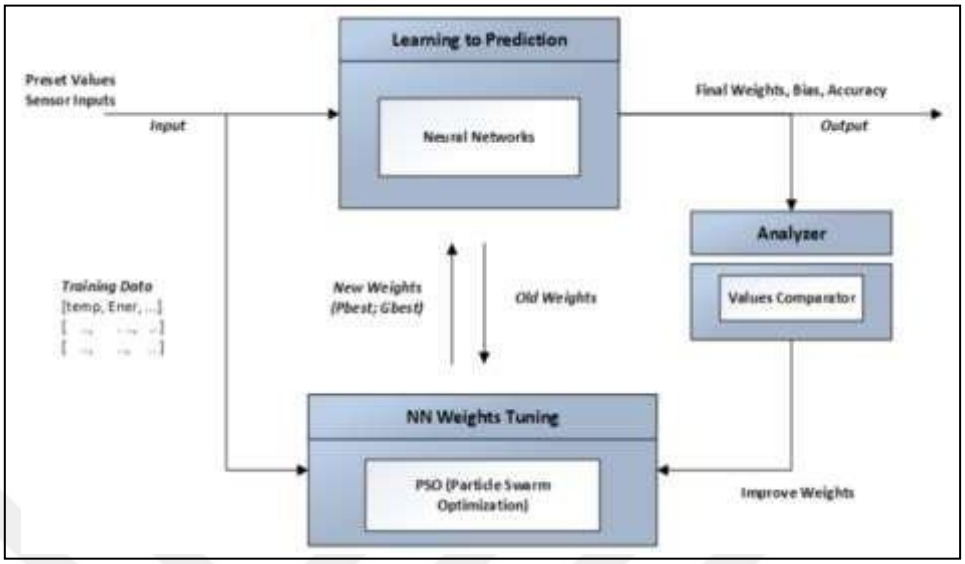

Figure 3-13: PSO-ANN Model

3.8 ARTIFICIAL ALGAE ALGORITHM

The Artificial Algae Algorithm (AAA) is a new bio-inspired metaheuristic that is inspired by the distinct physiology of microalgae [70]. The design of the algorithm is influenced by a number of factors, including the mobility of microalgae, the adaptive process, and the evolutionary process.

Heuristic approaches, as opposed to purely mathematical techniques, have the ability to search the whole solution space and produce good sub-optimal answers. This gives them the potential to be applied to issues that occur in the real world [70]. The objective is to develop a robust algorithm that consistently produces high-quality results. Similar to heuristics, metaheuristics offer a broad algorithmic framework that may be altered to solve a wide range of optimization issues. This is in contrast to heuristics, which only handle specific problems. A metaheuristic is a process that integrates one or more processes (heuristic approaches) with a higher-level strategy. It is also known as a metaheuristic procedure. Each metaheuristic strategy is an unknown method that may be substituted with any other approach. These actions may be as simple as switching a representation, or they could be as complicated as utilizing a new metaheuristic [72].

Darwin's principles of evolution or the intelligent activities of swarming can serve as a source of inspiration for successful bio-inspired metaheuristic approaches [73].

Scientists refer to a wide variety of eucaryotic, chlorophyll- and nucleus-containing organisms as "algae." These organisms are capable of producing their own food by using carbon dioxide (CO2) and water as their primary nutrients (H2O).

To migrate toward the light source and complete the process of photosynthesis, artificial algae, just like their natural counterparts, utilize helical swimming. In addition to this, they may adapt to their surroundings, cause a shift in the species that predominate, and reproduce through the process of mitosis. The algorithm may be broken down into three separate steps, which are referred to as the "Evolutionary Process," the "Adaptation," and the "Helical Movement." An algal colony is a clump of algae cells that have evolved to coexist and thrive together as a community. Every settlement offers a potential solution to the problem. A flowchart providing a high-level understanding of the AIA may be seen in Figure 3.14. Initiation of an algae colony, calculation of energy loss and adaptation parameters, and performance of a fitness evaluation using the same dataset that was used to train the PSO algorithm are all conducted (with 70% of the data being used for training and 30% being used for testing). The accuracy, sensitivity, and specificity of training and testing, in addition to the error rate of training and testing, are all components of the evaluation criterion. After that, a new map depicting the locations of the algal colonies is produced, and the procedure is repeated until either the maximum number of possible iterations is reached or the desired rate of error is attained. It was estimated that there were around 40 people in the population. A total of 10,000 people's fitness levels were determined via algorithms.

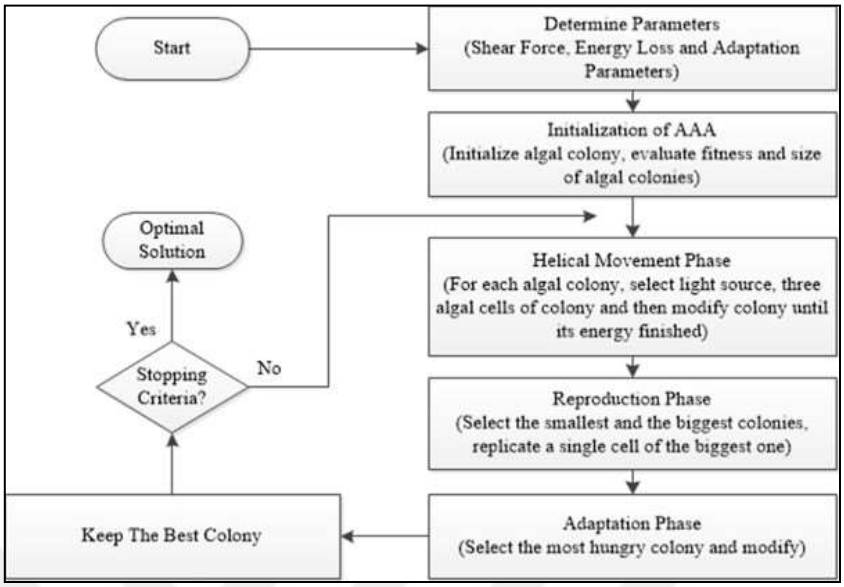

Figure 3-14: AAA Flowchart

An Excel-based database was trained using ANN-AAA in this thesis, and three distinct hidden layers were employed to do the training (3, 6, 9). Figure 3.15 shows script was used to produce the MATLAB code that was employed in the training process for the data.

1. *get and read the input file data*
2. *read the file data*
3. *measure the size*
4. *specify total number of hidden layers*
5. *Create the object of feedforward network*
6. *Set up Division of Data for Training, Validation, Testing (trainRatio = 70/100, testRatio = 30/100)*
7. *configure the neural network object with input dataset features*
8. *Compute the value of maximum fitness number*
9. *Initialize the training algorithm (AAA) parameters*
10. *Perform the training using AAA algorithm*
11. *compute the error values for AAA algorihm*
12. *Compute Training Accuracy*
13. *Compute the sensitivity and specificity by corss validation*
14. *Computing the TP, FP, TN, FN, Precision and Recall Rates and Accuracy*
15. *Calculation of MSE*
16. *Compute Testing Accuracy*

Figure 3-15: Script used for producing the MATLAB code

4 RESULTS

In this chapter, we report the outcomes of the ANN-PSO and ANN-AAA approaches that were investigated throughout the course of this thesis. We did so by making use of the dataset from the UCI Machine learning repository, which includes time, date, global active power, global reactive power, global intensity, and voltage as predictors of energy consumption. It is anticipated that the updated version of ANN would produce better results. It is possible that choosing between ANN-PSO and ANN-AAA will prove to be difficult. The subsequent sections of this chapter will give further findings to the reader.

The quality of the data that was utilized for the prediction is shown in figure 4.1.

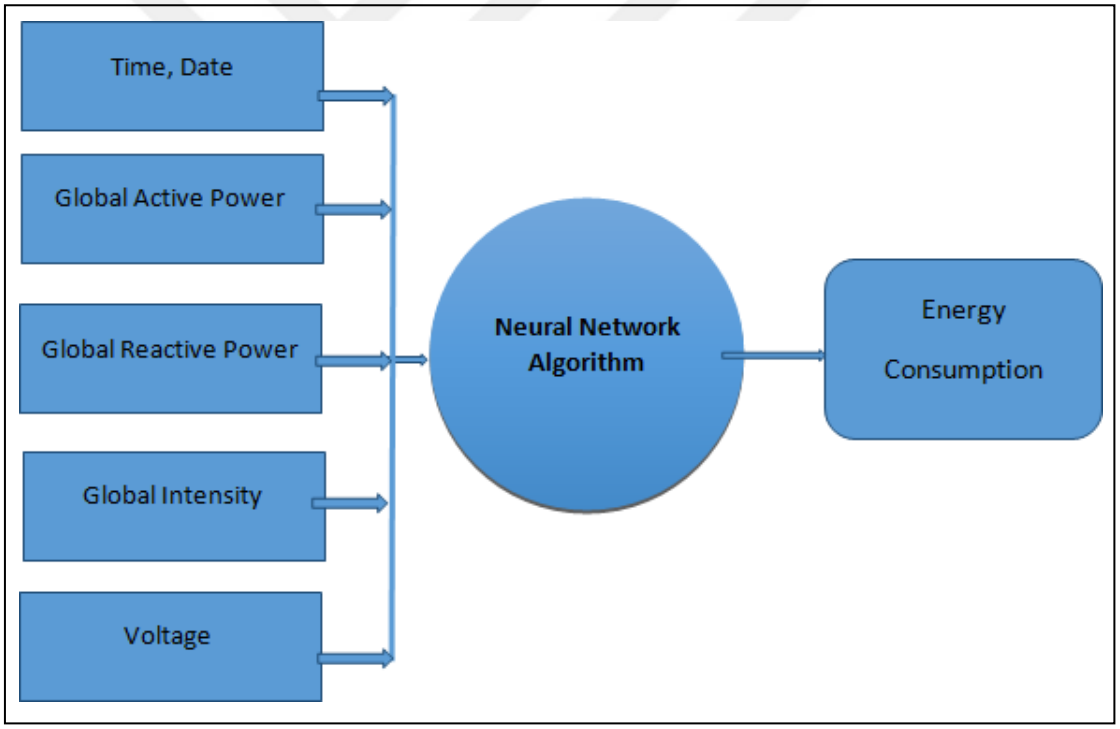

Figure 4-1: Energy usage forecasting

4.1 EVALUATION OF ENERGY CONSUMPTION FORECASTING RESULTS

4.1.1 Error Curve Prediction

70% of the data was utilized for training, while the remaining 30% was used for testing with ANN-PSO and ANN-AAA, respectively. To achieve the same level of accuracy, these two algorithms made use of various numbers of hidden layers (3, 6, and 9), but the overall effect was the same.

The ANN-PSO and ANN-AAA models were both initially trained on the same dataset with the same number of hidden layers (three in each case). Figure 4.2 illustrates the outcomes of the error curve when there are three hidden layers to consider.

The error curve for ANN's prediction of power consumption decreases from 0.24 to 0.175 after 200 minutes of forecasting, while the error curves for ANN-PSO and ANN-AAA decrease even further, to 0.1 and 0.09, respectively, after the same amount of time.

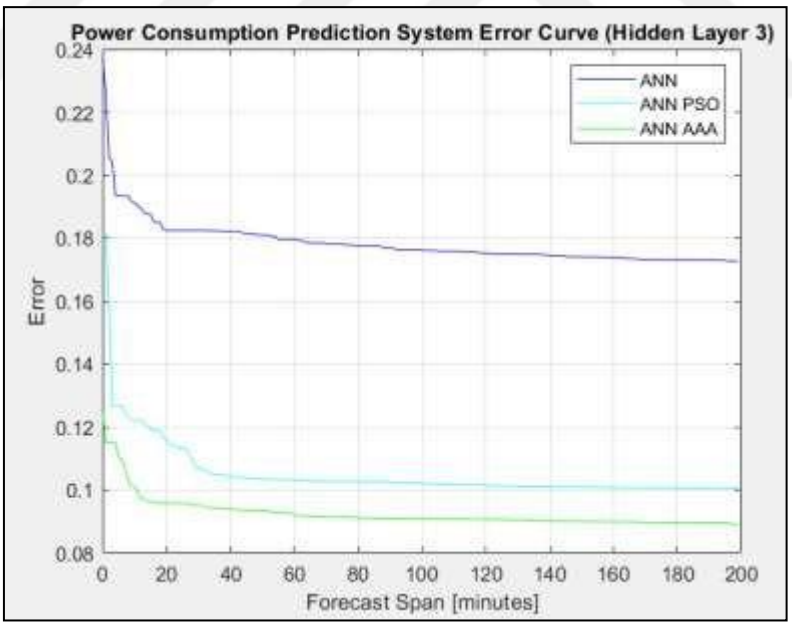

Figure 4-2: Power Consumption Prediction System Error Curve (HL3)

Figure 4.3 depicts the error curves for power consumption, ANN, ANN-PSO, and ANN-AAA, when the algorithms are initialized with 6 hidden layers. The error curve for the ANN indicates

the discrimination of the error at 0.28 to 0.175 throughout the span, while the error curves for ANN-PSO and ANN-AAA show a similar drop from 0.1 to 0.09.

A total of 9 hidden layers were used to begin the algorithms, and as can be seen in Figure 4.4, the ANN, ANN-PSO, and ANN-AAA all saw improvement during the course of the experiment, with the ANN dropping from 0.22 to 0.15, ANN-PSO to 0.1, and ANN-AAA to 0.08.

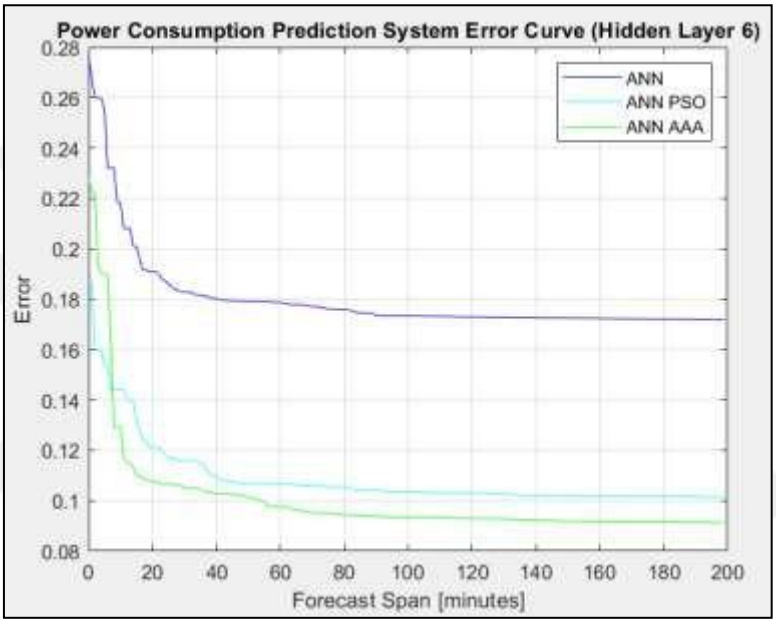

Figure 4-3: Power Consumption Prediction System Error Curve (HL6)

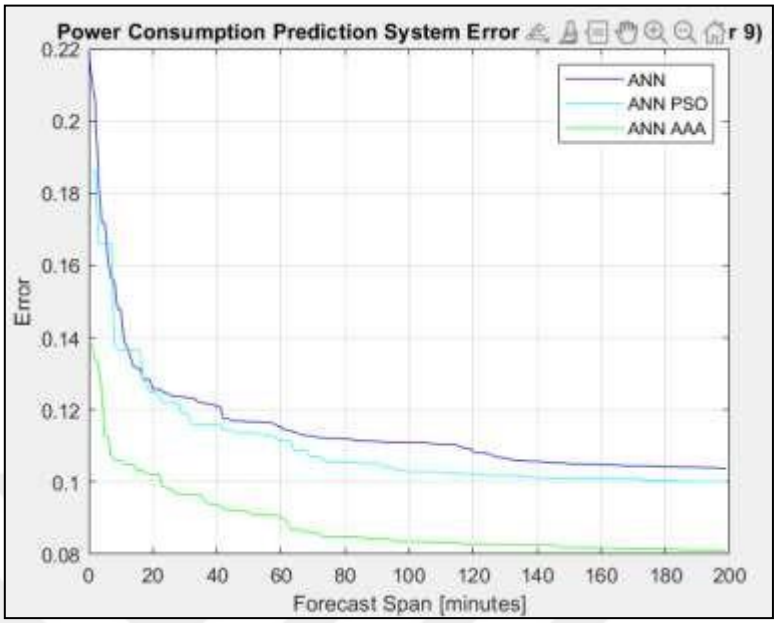

Figure 4-4: Power Consumption Prediction System Error Curve (HL9)

According to the findings, the ANN-AAA approach beat the ANN-PSO technique in terms of error performance and mean squared error performance, as well as other critical performance characteristics during testing and training. It is impossible to dispute the superiority of the ANN-AAA technique, as demonstrated by the divergence in the error graphs. The findings that were computed and reported above were determined for five different values of the "hidden layer number." The relative training error is shown in Figures 4.2, 4.3, and 4.4 for the 3, 6, and 9 hidden layer configurations, respectively. The findings demonstrate that overall mistake rates decreased, despite the fact that training circumstances were varied. These findings make it abundantly evident that there is a closing gap in the error rates that may be produced with a variety of numbers of hidden layers. As was to be predicted, the suggested technique shows an improvement in error performance in direct proportion to the decreasing number of hidden layers.

4.1.2 Accuracy Rate Performance

Figure 4.5 illustrates the findings of an investigation into the training accuracy rate for power usage across all of the different techniques that were utilized. It is clear that out of all of the specified numbers of hidden layers, ANN-AAA has the highest training accuracy rate performance. When contrasted with the outcomes of classic ANN, the ones produced by ANN-PSO appear to have a lot of potential.

The following table provides information on the accuracy rate of training for each approach with 3, 6, and 9 hidden layers. Accuracy levels of 96 or above are attained by each of the recommended hidden layers in AAA-ANN. When all of the hidden layers in ANN-PSO are used, the algorithm achieves an accuracy of around 95.

The results of the tests, together with their respective accuracy, are shown in Figure 4.6.

Table 4.1 presents the results of testing the algorithms with 3, 6, and 9 hidden layers to see how accurate they are.

Figure 4.3 depicts training accuracy rate experiments that were conducted for the purpose of power consumption forecasting. According to Table 4.2, the ANN training model, the ANN-PSO training model, and the ANN-AAA training model each include 3, 6, and 9 hidden layers, respectively. According to the evidence at hand, ANN performs significantly better than both ANN-PSO and ANN-AAA.

Figure 4-5: Training Accuracy Rate Performance

Figure 4-6: Testing Accuracy Rate Performance

Table 4.1: Training Accuracy Rate Performance

Hidden layers	ANN	ANN-PSO	AAA-ANN
3	93.2151	95.447	96.7744
6	93.1971	95.409	96.6937
9	93.84	95.39	96.81

Table 4.2: Testing Accuracy Rate Performance

Hidden layers	ANN	ANN-PSO	AAA-ANN
3	93.21	95.44	96.77
6	93.19	95.409	96.69
9	93.84	95.39	96.81

4.2 ANALYSIS

By utilizing AAA-ANN, improvements may be made to the error Curve, the training accuracy rate, and the testing accuracy rate. Based on the results of our experiments and examinations, it is abundantly evident that the AAA-ANN method is the most efficient technique for forecasting future power use.

5 CONCLUSION

In the course of this research, a variety of machine learning techniques have been applied in order to derive forecasts for future energy consumption.

An introduction and a few experiments are offered in the first chapter to set the stage for this research, the objective of which is to suggest two optimization algorithms that may be used to improve the performance of artificial neural networks. These algorithms are referred to as AAA and PSO. It is important to identify the algorithms that are utilized to monitor and report on energy usage, as well as optimize the use of energy. In this work, we suggest a fitness function that is based on AAA in order to handle the difficulties of accuracy, overfitting, and ANN size reduction, all of which are essential during the process of creating an ANN.

In the second chapter, we will discuss the structure of the neural network and carry out an extensive literature review that will cover all of the different types of machine learning algorithms.

In chapter three, we made an effort to lay the framework for a new method by reviewing and summarizing prior debates and simulations of related wok techniques.

In Chapter 4, a UCP database of energy consumption prediction is used as a training dataset to explain the approach of the study. In MATLAB, AAA and ANN algorithms are created using the dataset's parameters, and the chapter concludes with a discussion of future work.

Seventy percent of the dataset was utilized so that we could train on it and make comparisons for the purposes of this chapter. When compared to other methods, the AAA-ANN technique yields a flatter error Curve, a larger proportion of correct answers during training, and a more trustworthy prediction during testing. In addition, the percentage of correct responses during training is significantly higher. According to the findings of this inquiry, AAA-ANN performs much better than both ANN and ANN-PSO when it comes to forecasting future energy use.

REFERENCES

[1] E. Journal, O. S. Eluyode, and D. T. Akomolafe, "Scholars Research Library Comparative study of biological and artificial neural networks," of Applied Engineering and Scientific Research, vol. 2, no. 1, pp. 36–46, 2013.

[2] K. Premalatha and A. M. Natarajan, "Hybrid PSO and GA for Global Maximization" , Int. J. Open Problems Compt. Math, vol. 2, no. 4, 2009.

[3] M. A. Tawhid and V. Savsani, "A novel multi-objective optimization algorithm based on artificial algae for multi-objective engineering design problems" , Applied Intelligence, vol. 48, pp. 3762–3781, 2018.

[4] C. Blum and X. Li, "Swarm Intelligence in Optimization" , Swarm Intelligence, pp. 43–85, Sep. 2008, doi: 10

[5] X. da Zhang, "A matrix algebra approach to artificial intelligence," A Matrix Algebra Approach to Artificial Intelligence, pp. 1–805, Jan. 2020.

[6] S. Angra and S. Ahuja, "Machine learning and its applications: A review," Proceedings of the 2017 International Conference On Big Data Analytics and Computational Intelligence, ICBDACI 2017, pp. 57–60, Oct. 2017.

[7] F. Min, Q. Hu, and W. Zhu, "Feature selection with test cost constraint," International Journal of Approximate Reasoning, vol. 55, no. 1, pp. 167–179, Jan. 2014.

[8]R. E. Schapire, "The Boosting Approach to Machine Learning: An Overview," pp. 149–171, 2003.

[9] K.R. Dalal, "Analysing the Role of Supervised and Unsupervised Machine Learning in IoT," Proceedings of the International Conference on Electronics and Sustainable Communication Systems, ICESC 2020, pp. 75–79, Jul. 2020.

[10] K. Kwon, S. Member, H. Zhu, and S. Member, “Reinforcement Learning Based Optimal Battery Control Under Cycle-based Degradation Cost,” IEEE TRANSACTIONS ON SMART GRID (SUMBITTED), 2018.

[11] R. Amiri, H. Mehrpouyan, L. Fridman, R. K. Mallik, A. Nallanathan, and D. Matolak, “A Machine Learning Approach for Power Allocation in HetNets Considering QoS,” IEEE International Conference on Communications, vol. 2018-May, Jul. 2018.

[12] Mark Lee, Elements of Reinforcement Learning. 2005. Accessed: Oct. 10, 2021. [Online]. Available: http://incompleteideas.net/book/first/ebook/node9.html

[13] X. Goldberg, “Introduction to semi-supervised learning,” Synthesis Lectures on Artificial Intelligence and Machine Learning, vol. 6, pp. 1–116, Jun. 2009.

[14] A. Patle and D. S. Chouhan, “SVM kernel functions for classification,” 2013 International Conference on Advances in Technology and Engineering, ICATE 2013.

[15] Z. Liu, M. J. Zuo, and H. Xu, “A Gaussian radial basis function based feature selection algorithm,” IEEE International Conference on Computational Intelligence for Measurement Systems and Applications Proceedings, pp. 53–56, 2011.

[16] W. S. Mcculloch and W. Pitts, “A Logical Calculus Of The Ideas Immanent In Nervous Activity” , Bulletin of Mothemnticnl Biology, vol. 52, no. 2, pp. 99–115, 1990.

[17] J. Moini and P. Piran, “Histophysiology,” Functional and Clinical Neuroanatomy, pp. 1–49, 2020.

[18] S. Kiranyaz, M. Zabihi, A. B. Rad, T. Ince, R. Hamila, and M. Gabbouj, “Real-time phonocardiogram anomaly detection by adaptive 1D Convolutional Neural Networks,” Neurocomputing, vol. 411, pp. 291–301, Oct. 2020.

[19] E. Kuriscak, P. Marsalek, J. Stroffek, and P. G. Toth, “Biological context of Hebb learning in artificial neural networks, a review,” Neurocomputing, vol. 152, pp. 27–35, Mar. 2015.

[20] B. Widrow and M. A. Lehr, “30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation,” Proceedings of the IEEE, vol. 78, no. 9, pp. 1415–1442, 1990.

[21] L. E. Givon and A. A. Lazar, “Neurokernel: An Open Source Platform for Emulating the Fruit Fly Brain,” PLOS ONE, vol. 11, no. 1, p. e0146581, Jan. 2016.

[22] M. Minsky and S. Papert, Perceptrons; an introduction to computational geometry. MIT Press, 1969.

[23] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” Nature, vol. 323, no. 6088, pp. 533–536, 1986.

[24] T. Szandała, “Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks,” vol. 903, Oct. 2020.

[25] P. Li, J. Qian, and T. Wang, “Automatic Instrument Recognition in Polyphonic Music Using Convolutional Neural Networks,” Nov. 2015, Accessed: Dec. 07, 2021.

[26] T. Jónás, “Sigmoid functions in reliability based management,” Periodica Polytechnica Social and Management Sciences, vol. 15, no. 2, pp. 67–72, 2007.

[27] A. C. Marreiros, J. Daunizeau, S. J. Kiebel, and K. J. Friston, “Population dynamics: Variance and the sigmoid activation function,” 2008, doi: 10.1016/j.neuroimage.2008.04.239.

[28] M. Chandra, “Hardware Implementation of Hyperbolic Tangent Function using Catmull-Rom Spline Interpolation,” Jul. 2020, Accessed: Dec. 07, 2021.

[29] A. M. Fred Agarap, “Deep Learning using Rectified Linear Units (ReLU),” Mar. 2018, Accessed: Dec. 07, 2021.

[30] D. A. Clevert, T. Unterthiner, and S. Hochreiter, “Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs),” 4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings, Nov. 2015.

[31] C. Enyinna Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, “Activation Functions: Comparison of Trends in Practice and Research for Deep Learning”.

[32] E. RJ, “Introduction to backpropagation neural network computation.,” Pharmaceutical Research, vol. 10, no. 2, pp. 165–170, Feb. 1993.

[33] P. and S. Z. Roop, “Run-Time Circuit Reconfiguration”, Ph.D. thesis , 2004.

[34] N. Farhana Hordri, S. Yuhaniz, S. Mariyam Shamsuddin, and S. Sophiayati Yuhaniz, “Deep Learning and Its Applications: A Review Predicting Financial Frauds using Deep Learning Approaches View project recognition View project Deep Learning and Its Applications: A Review,” , 2016, Accessed: Oct. 12, 2021.

[35] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, “A survey of deep neural network architectures and their applications,” Neurocomputing, vol. 234, pp. 11–26, Apr. 2017.

[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” Adv Neural Inf Process Syst, pp. 1097–1105, 2012.

[37] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller, and W. Burgard, “Multimodal deep learning for robust RGB-D object recognition,” IEEE International Conference on Intelligent Robots and Systems, vol. 2015-December, pp. 681–687, Dec. 2015.

[38] Y. Wu et al., “Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation,” Computer Science, 2016.

[39] J. Zhou, Y. Cao, X. Wang, P. Li, and W. Xu, “Deep Recurrent Models with Fast-Forward Connections for Neural Machine Translation,” Trans Assoc Comput Linguist, vol. 4, pp. 371–383, Dec. 2016, doi: 10.1162/TACL_A_00105.

[40] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, “Attention-Based Models for Speech Recognition,” Advances in Neural Information Processing Systems, vol. 28, 2015.

[41] D. Amodei et al., “Deep Speech 2: End-to-End Speech Recognition in English and Mandarin,” 33rd International Conference on Machine Learning, ICML 2016, vol. 1, pp. 312–321, Dec. 2015, Accessed: Oct. 12, 2021.

[42] A. Esteva et al., “A guide to deep learning in healthcare,” Nature Medicine, vol. 25, no. 1, pp. 24–29, Jan. 2019.

[43] C. Szegedy et al., “Going deeper with convolutions,” Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 07-12-June-2015, pp. 1–9, Oct. 2015.

[44] R. Rahadian and S. Suyanto, “Deep Residual Neural Network for Age Classification with Face Image,” 2019 2nd International Seminar on Research of Information Technology and Intelligent Systems, ISRITI 2019, pp. 21–24, Dec. 2019.

[45] Y. Que and H. J. Lee, “Densely connected convolutional networks for multi-exposure fusion,” Proceedings - 2018 International Conference on Computational Science and Computational Intelligence, CSCI 2018, pp. 417–420, Dec. 2018.

[46] M. M. , N. M. and V. V. T. M. Merzenich, "Neuroplasticity: introduction.’, Progress in brain research. 2013.

[47] G. Hinton, “A Practical Guide to Training Restricted Boltzmann Machines,”, 2010, Accessed: Oct. 12, 2021. [Online]. Available: http://learning.cs.toronto.edu

[48] O. Fink, E. Zio, and U. Weidmann, “Fuzzy Classification With Restricted Boltzman Machines and Echo-State Networks for Predicting Potential Railway Door System Failures,” IEEE Transactions on Reliability, vol. 64, no. 3, pp. 861–868, Sep. 2015.

[49] N. Lu, T. Li, X. Ren, and H. Miao, “A Deep Learning Scheme for Motor Imagery Classification based on Restricted Boltzmann Machines,” IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 25, no. 6, pp. 566–576, Jun. 2017.

[50] Y. Hua, J. Guo, and H. Zhao, “Deep Belief Networks and deep learning,” Proceedings of 2015 International Conference on Intelligent Computing and Internet of Things, ICIT 2015, pp. 1–4, May 2015.

[51] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, “Greedy Layer-Wise Training of Deep Networks”.

[52] X. Luo and S. Xu, “Forest mapping from hyperspectral image using deep belief network,” Proceedings - 2019 15th International Conference on Mobile Ad-Hoc and Sensor Networks, MSN 2019, pp. 395–398, Dec. 2019.

[53] D. D. Pham, G. Dovletov, S. Warwas, S. Landgraeber, M. Jager, and J. Pauli, “Deep learning with anatomical priors: Imitating enhanced autoencoders in latent space for improved pelvic bone segmentation in MRI,” Proceedings - International Symposium on Biomedical Imaging, vol. 2, pp. 1166–1169, 2019.

[54] F. B. M. Suah, “Preparation and characterization of a novel Co(II) optode based on polymer inclusion membrane,” Analytical Chemistry Research, vol. 12, pp. 40–46, Jun. 2017.

[55] K. Simonyan and A. Zisserman, “VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION,” 2015, Accessed: Oct. 12, 2021.

[56] F. Ayberk, U. H. Ozer, E. Nurba, and E. Onat, “Direction Finding Using Convolutional Neural Networks and Convolutional Recurrent Neural Networks,” 2020 28th Signal Processing and Communications Applications Conference, SIU 2020 - Proceedings, Oct. 2020.

[57] K. Li, C. Hu, G. Liu, and W. Xue, “Building’s electricity consumption prediction using optimized artificial neural networks and principal component analysis,” Energy and Buildings, vol. 108, pp. 106–113, Dec. 2015.

[58] X. J. Luo and L. O. Oyedele, “Forecasting building energy consumption: Adaptive long-short term memory neural networks driven by genetic algorithm,” Advanced Engineering Informatics, vol. 50, p. 101357, Oct. 2021.

[59] P. Karampelas, V. Vita, C. Pavlatos, V. Mladenov, and L. Ekonomou, “Design of artificial neural network models for the prediction of the Hellenic energy consumption,” 10th Symposium on Neural Network Applications in Electrical Engineering, NEUREL-2010 - Proceedings, pp. 41–44, 2010.

[60] “Greece Weather Forecast,HNMS, Hellenic National Meteorological Service.” http://www.hnms.gr/emy/en/forecast/xartis_prognosis_kairou_elladas (accessed Oct. 16, 2021).

[61] “PPC S.A.” https://www.dei.gr/en (accessed Oct. 16, 2021).

[62] https://ec.europa.eu/info/departments/eurostat-european-statistics_en (accessed Oct. 16, 2021), “Eurostat - European statistics | European Commission.”

[63] X. J. Luo and L. O. Oyedele, “Forecasting building energy consumption: Adaptive long-short term memory neural networks driven by genetic algorithm,” Advanced Engineering Informatics, vol. 50, p. 101357, Oct. 2021, doi: 10.1016/J.AEI.2021.101357.

[64] V. K. Saini, R. Singh, D. K. Mahto, R. Kumar and A. Mathur, "Learning Approach for Energy Consumption Forecasting in Residential Microgrid," 2022 IEEE Kansas Power and Energy Conference (KPEC), 2022, pp. 1-6, doi: 10.1109/KPEC54747.2022.9814744.

[65] L. Medsker and L. C. Jain, Recurrent neural networks: design and applications., CRC press, 1999.

[66] A. Ismail and A. P. Engelbrecht, “Global optimization algorithms for training product unit neural networks,” Proceedings of the International Joint Conference on Neural Networks, vol. 1, pp. 132–137, 2000, doi: 10.1109/IJCNN.2000.857826.

[67] J. Kennedy, “Particle swarm: Social adaptation of knowledge,” Proceedings of the IEEE Conference on Evolutionary Computation, ICEC, pp. 303–308, 1997, doi: 10.1109/ICEC.1997.592326.

[68] J. Wen and B. Cao, “A modified particle swarm optimizer based on cloud model,” IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM, pp. 1238–1241, 2008, doi: 10.1109/AIM.2008.4601839.

[69] H. Kuwahara and J. Theodoras, “Guest editorial,” IEEE Communications Magazine, vol. 44, no. 5, p. 30, May 2006, doi: 10.1109/MCOM.2006.1637942.

[70] E. Kaya, S. A. Uymaz, and B. Kocer, “Boosting galactic swarm optimization with ABC,” International Journal of Machine Learning and Cybernetics, vol. 10, no. 9, pp. 2401–2419, Sep. 2019, doi: 10.1007/S13042-018-0878-6.

[71] “Xin-She Yang (DPhil) - Google Akademik.” https://scholar.google.com.tr/citations?user=fA6aTlAAAAAJ&hl=tr (accessed Nov. 15, 2021).

[72] L. Díez, “Clever Algorithms Nature-Inspired Programming Recipes.” Accessed: Nov. 15, 2021.https://www.academia.edu/32823084/Clever_Algorithms_Nature_Inspired_Programming_Recipes

[73] S. S. S. Binitha S, “A Survey of Bio inspired Optimization Algorithms 138,” International Journal of Soft Computing and Engineering (IJSCE), vol. 2, no. 2, pp. 137–151, May 2012.