Artificial Neural Network in Character Recognition

P.Anitha, Umadevi Pongiya, DrAravind,R.Jothi,S.Arthi

**Department of computer Applications,Dhanalakshmi Srinivasan College of Arts and Science for Women,,Perambalur , 621 212, Tamilnadu, , India., umadeviasokan@gmail.com*

Email: arthi5625@gmail.com*(Arthi S) Corresponding author: Arthi S*

ABSTRACT

Character recognition is critical in a wide range of professions. A technology known as Artificial Neural Network is used to implement this idea. When a space is encountered, the characters are combined and processed as a single entity. A thesaurus is used to match the words to thesaurus strings, and valid words are considered. A whole thought will be formed from so many of these words. Back Propagation method is used in an artificial neural network. There must be inputs and a goal output in order for this to work. When the mean squared error (MSE) of the artificial neural network is attained, the notion of training the network using input and target outputs is halted. The final weights have been recorded in a file at this stage. Testing the ANA is a step in the procedure. When the final weights are applied to a pattern, the ANN processes it. A character is indicated by the ANN's output.

Keywords: Back Propagation, Hypothesis

1.INTRODUCTION

Mathematical difficulties faced by humans before the advent of the computer, or more accurately (and this difference is highly essential), people were too sluggish in solving, existed before the era of the computer. Computers have made it possible for these arduous and time-consuming chores to be completed in a timely and efficient manner. Initially, computers were used to tackle complex physical issues, such as computing equations and providing a user-friendly interface, such as word processors. It is possible, however, to execute many ordinary activities that are easy for people to accomplish (without even a conscious effort) but difficult for a computer to define in order to simply solve [1]-[5]. Among them are:

As an example, signal processing (Character recognition, pattern recognition, voice recognition, image processing etc.) Compression Reconstruction of information (e.g. classification where part of the data is missing) Simplification of data. A neural network is an abstraction of the human brain's workings, thus it was only a matter of time until scientists, engineers, and mathematicians attempted to create one for computers.

In this, I will attempt to explain the basic principles of neural networks, and illustrate how they are similar to biological brain networks. It is only after this that you will be given a well-known neural network structure known as a multi-layer perception, which will be trained using the Feedforward Back-Propagation Network (BPN) method to learn character recognition [6]-[10].

1.1 BPN Training algorithm

Since the BP network excels in character categorization, this explanation assumes that this is the case. You may also utilise back-propagation to solve many other issues, such as compressing and predicting. If the results of your data presentation aren't what you expected, what will you do? The obvious solution is to change the weights of some of the connections. There's a good chance that the first output value will be a long way off from what you want.

We're working to make the network run more smoothly. What weights must be changed, and by how much, in order to reach this goal? To put it another way, how do you determine which connection is responsible for the most inaccuracy in the output? If we want to reduce mistakes at the output, we must utilise an algorithm that efficiently adjusts distinct connection weights. The term "optimization" is used to describe this kind of challenge in engineering. However, the neural network is a more general system and needs a more complicated algorithm to change the various network parameters.

The back-propagation technique has had a major impact on neural network popularity. Back-primary propagation's merits are its simplicity and good speed, while it may be improved in a number of ways, including some instances and tweaks. In the case of character recognition, back-propagation is ideal.

2.BACK PROPAGATION

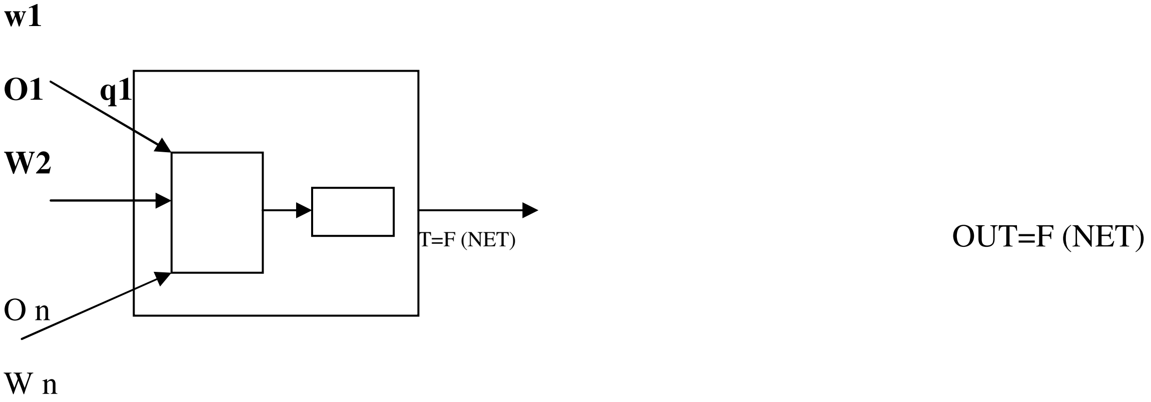
A theoretically valid approach for training multi-layer artificial neural networks did not exist. Because of the limitations of a single-layer network, the whole field was transformed into a virtual ellipse.

Multiplayer artificial neural networks may be trained via back propagation.

powerful if not very applicable mathematics are at play here. .

2.1THE BACKPROPAGATION TRAINING ALGORITHM:

NETWORK CONFIGURATION:



OUT=F (NET)

Figure 1 illustrates the neuron that serves as the basic building block for the Back propagation network.

This layer receives a set of inputs, either externally or from a preceding layer. Input data is weighted before being multiplied and the results are added together. The word "NET" refers to this summing of items. The signal is generated when the activation function F is applied after the calculation NET IS SHOWN IN FIGURE 1.

NET=O1W1+O1W1+……….+OnWn

OUT=F (NET)

Tactivation function used to propagate backwards

A "squashing function" or "logistic" is a term for this kind of function. When the ranges of Net are condensed by the sigmoid, Out can only be found between zero and one. Only if no-linearity is added can we say that multi-layer networks have more representational power than single-layer networks.

2.1 Multi-layer Network

A backpropagation-ready network with many layers. The initial set of points serves just as distribution points, and they do not execute point summing. " The weights' outputs receive the point signal.

2.2 Static backpropagation

A supervised neural network may be trained via backpropagation, which is the most common method. If you want an immediate mapping of the input to the output, you employ a technique called "static backpropagation." Classification challenges like optical character recognition may be solved with these networks (OCR).

At the core of all backpropagation algorithms is the chain rule for ordered partial derivatives, which is used to assess the sensitivity of a cost function to the internal states and weights of a network. Backpropagation also refers to the backward transmission of error across each internal node in the network, which is then used to build weight gradients for that node. Activations may also be transmitted forward while errors are propagated backward

.

3. HMM Methods in Character Recognition

Automated Character Recognition (ACR) systems of the future are likely to be based on software designs that generate word hypotheses from an auditory signal, rather than hardware. Statistical approaches are often used in these designs to implement the most popular algorithms. Additionally, a library of publications detailing many systems and their history and mathematical basis may be accessed. Every 10 to 30 msec, a vector of auditory characteristics is calculated. Section contains further information about this component. Different vectors and their effect on recognition performance are described in. Word models are represented as acoustic vectors, and the likelihood of a word occurring is computed using these vectors. observing a sequence y_1^T of vectors when a word sequence W is pronounced. The ASR system uses a search procedure based on the rule: given a sequence, a word sequence is created

to an individual who has the highest a-posteriori likelihood of winning (MAP). In contrast to Acoustic Models (AM), Language Models (LM) are used to calculate (LM). One of the first things to keep in mind while searching through huge databases is that it takes two steps to do it. Word lattices of the best sequences are generated using basic models to determine approximation probabilities in real time. With fewer hypotheses, more accurate likelihoods are compared in the second stage. A single word sequence hypothesis may be generated by certain systems in a single step. Dictation-related searches return a possible word sequence. The work of comprehending requires a procedure that may take more than two phases to accomplish. A pair of stochastic processes is what is meant by a hidden Markov model.. The model may be described using the following parameters:

$$\begin{array}{lcll} A & \equiv & \{a_{i,j} | i,j \in \mathcal{X}\} & \text{transition probabilities} \\ B & \equiv & \{b_{i,j} | i,j \in \mathcal{X}\} & \text{output distributions} \\ \Pi & \equiv & \{\pi_i | i \in \mathcal{X}\} & \text{initial probabilities} \end{array}$$

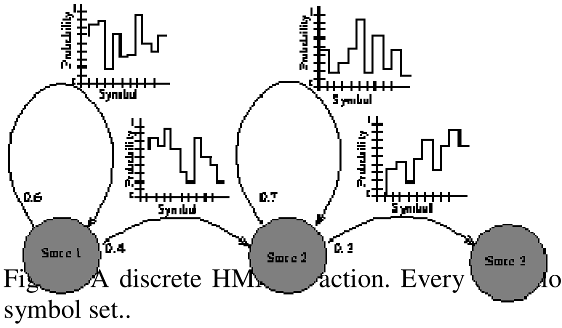
with the following definitions:

$$\begin{array}{rcl} a_{i,j} & \equiv & p(X_t = j | X_{t-1} = i) \\ b_{i,j}(y) & \equiv & p(Y_t = y | X_{t-1} = i, X_t = j) \\ \pi_i & \equiv & p(X_0 = i) \end{array}$$

Types of Hidden Markov Models

According on the distribution functions in the B matrix, HMMs may be categorised. Discrete HMMs define distributions on finite spaces. A finite alphabet of N symbols is used to represent observations as vectors of symbols. It's important to keep in mind that this definition presupposes that each component is autonomous.

A discrete HMM with one-dimensional observations is seen in the figure. Model transitions are linked to distributions.



Fi A discrete HM action. Every on has a transition probability and an output distribution on the symbol set..

Probability densities on continuous observation spaces may also be used to define distributions. In order to keep the number of statistical parameters to a tolerable level, rigorous limits must be put on the functional form of the distributions is shown in figure 2. The most common strategy is to use mixes of base densities g of a family G with a simple parametric form to define model transitions. The mean vector and the covariance matrix may be used to parameterize the underlying densities, which are typically either Gaussian or Laplacian. A continuous HMM is an HMM that has a wide range of distributions like this. A high number of base densities must be employed in each combination in order to describe complicated distributions in this manner. In order to estimate the distribution parameters, a large training dataset may be required. Shared distributions across various models may be alleviated if the provided corpus is not big enough Semi-continuous HMMs, for example, use a common set of base densities for all mixes. The sole distinguishing feature of various combinations is their weight.

Semi-continuous modelling is sometimes generalised by considering the input vector y to be made up of many components, each of which is connected with a unique set of base distributions. The components are considered to be statistically independent, therefore the distributions associated with model transitions are the product of the density functions of the individual components. '

Calculating mixture densities may be made quicker by using vector quantization (VQ) on the mixes' gaussians rather than using continuous models for probability computation.

3.1 WORD AND UNIT MODELS

Phoneme networks are the most common way to represent words. In a word network, each node represents a unique way to say a word.

If the same phoneme is spoken in various settings, the acoustic distribution of observations might vary. Different situations may be represented by the same phoneme by using allophone models. For a particular phoneme, the number of allophones to examine may depend on several aspects, such as the amount of training data available to infer model parameters.

Polyphones are a novel approach to the problem. For every word in which a phoneme is used, an allophone should be considered. Another method is to choose allophones based on the grouping of relevant circumstances. With the use of Classification and Regression Trees, this decision may be done automatically (CART). As each node in the CART is linked to a specific query, the root of the tree is a phoneme. Nose consonants are referred to as, "Is the preceding phoneme nasal?" Links to more questions are provided with each potential response (YES or NO) in the form of nodes. Various methods exist for automatically assigning questions to CART nodes from a manually specified pool of questions. Allophone symbols may be used to designate the tree's leaves. Examples of how this notion has been used and references to a formalism for training and utilising CARTs are provided in papers by.

State, transition and probability distributions provide the basis for an allophone model's HMM structure. As a way to facilitate the estimate of statistical parameters, certain distributions might be identical or linked. When a phoneme's allophones have the same core piece, the distributions may be connected to show that they represent the same stable (context-independent) physical manifestation of the centre part of the phoneme, pronounced with a stationary configuration of the vocal tract.

If you have a few thousand cluster distributions, you can use them to build all of your models from a single pool. Additionally, the fundamental structures that make up word models or allophone models may be combined in a variety of ways. Fenones, as they're known, were first presented by. Multiones are more complex models of the same kind, but with more advanced construction elements. Another technique is to have a set of Gaussian probability density functions for each cluster of distributions. In order to create allophone distributions, one considers mixes with the same components but with different weights.

4. LANGUAGE MODELS

The probability $p(W)$ of a sequence of words $W = w_1, \ldots, w_L$ is computed by a Language Model (LM). In general $p(W)$ can be expressed as follows:

$$p(W) = p(w_1, .., w_n) = \prod_{i=1}^{n} p(w_i | w_0, .., w_{i-1})$$

Detailed explanations of how these probabilities were derived are provided in the next section. .

4.1 GENERATION OF WORD HYPOTHESES

To generate word hypotheses, you may either create a single word sequence, a collection of the n-best word sequences, or a lattice of word hypotheses that partly overlap one another.

An auditory feature vector sequence and word models are compared in this generation. First, the calculations involved in speech recognition algorithms will be presented, concentrating on the situation of a single-word utterance and then looking at the extension to continuous speech.

Since word borders are not clearly defined in voice signals and their transformations, a search is conducted for word boundaries in hypotheses. A series of acoustic characteristics are used to compare each word model to each other. When an auditory sequence and its model are compared in a probabilistic framework, it is necessary to compute the model's probability of assigning it to the given sequence. Recognizing something is all about this. The following variables are utilised in this calculation:

$\alpha_t(y_1^T, i)$:

probability of having observed the partial sequence y_1^t and being in state i at time t

$$\alpha_t(y_1^T, i) \equiv \begin{cases} p(X_0 = i), & t = 0 \\ p(X_t = i, Y_1^t = y_1^t), & t > 0 \end{cases}$$

$\beta_t(y_1^T, i)$:

probability of observing the partial sequence y_{t+1}^T given that the model is in state i at time t

$$\beta_t(y_t^T, i) \equiv \Big\{ p\left(Y_{t+1}^T = y_{t+1}^T | X_t = i\right), \quad t < T$$

$\psi_t(y_1^T, i)$

:

probability of having observed the partial sequence y_1^t along the best path ending in state i at time t:

$$\psi_t(y_1^T, i) \equiv \begin{cases} p(X_0 = i), & t = 0 \\ \max\limits_{i_0^{t-1}} p\left(X_0^{t-1} = i_0^{t-1}, X_t = i, Y_1^t = y_1^t\right) & t > 0 \end{cases}$$

α and β can be used to compute the total emission probability $p(y_1^T | W)$ as

$$\begin{aligned} p(Y_1^T = y_1^T) &= \sum_i \alpha_T(y_1^T, i) \\ &= \sum_i \pi_i \beta_0(y_1^T, i) \end{aligned}$$

This probability may be approximated by pursuing just the route with the highest likelihood. The amount may be used to do this:

$$p^*(Y_1^T = y_1^T) = \max_i \psi_T(y_1^T, i) \qquad (13)$$

All of the aforementioned probabilities are computed using a trellis, a matrix represented in the figure below. Let's assume that the HMM in Figure represents a word and that the input signal reflects the pronunciation of a single word for the purpose of simplification.

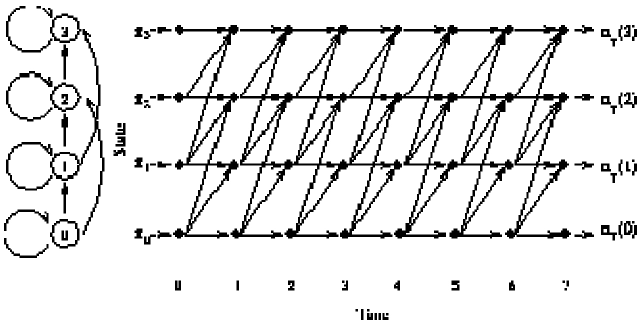


Figure.3. A state-time trellis.

It is important to note that each trellis column has a value for one of the newly added probabilities, and each interval between two columns corresponds to one input frame. The trellis' arrows reflect model transitions that may take the model from the beginning of time to its conclusion. To update the scores of nodes in a column at each time frame, recursion formulae are used, including the values of a neighbouring column, transition probabilities of the models, and corresponding values of output distributions for each of the columns. It begins with the leftmost column's values, which are initialised using (), and finishes with the rightmost column's values, which are calculated using () or (). (). Coefficients are computed in a right-to-left fashion is shown in figure 3.

The Viterbi method, used to compute coefficients, may be thought of as an application of dynamic programming to build a graph with weighted arcs with a maximum probability route. In order to compute it, the recursion formula follows:

Because of the aforementioned formula, it is feasible, at the conclusion of a series of input frames to extract the states that were visited by the optimal route and execute a time-alignment of input frames with models' states.

They are all time-complex because M is the number of possible transitions and T is the duration of the input. Assuming that the transition probability matrix is sparse, M can only be as large as S, although in practise it is frequently considerably less. In reality, it is typical practise in voice recognition to set strict limits on the permitted state sequences, such as j=i, j=i+2, as in the model shown in Figure 1. All conceivable word-segmentation possibilities and the a-priori probability that the LM assigns to word sequences are taken into consideration in the recognition process.

With basic LMs based on bigram or trigram probabilities, good results may be achieved. Let's have a look at a bigram language model as an example. A finite state automata may easily include this model, as illustrated in Figure, where the dashed arcs represent to transitions between words with the LM's probability is shown in figure 4.

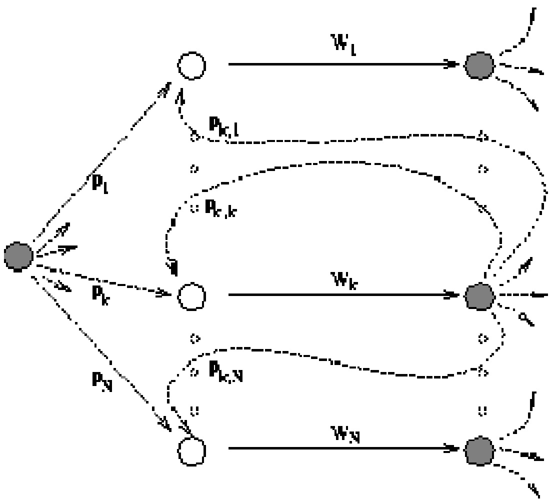


Figure.4.: Bigram LM represented as a weighted word graph. $p_{h,k}$stands for $p(W_k|W_h)$, $p_h$stands for $p(W_h)$. The leftmost node is the starting node, rightmost ones are finals.

Substituting word-labeled arcs for their respective high-dimensional models (HMMs) results in an automaton that may be used to search for the most likely route given an observation series. Transitions that have no related output distribution are shown with dashed lines. This

The Viterbi algorithm has to be altered in some way. Backtracking is maintained to a minimum throughout the Viterbi algorithm so that a route in terms of word labels may be reconstructed. The search space widens as the language expands. However, the unequal distribution of odds across alternative pathways might be beneficial. There are a lot of states that have an accumulated probability that is considerably lower than the highest one, which means that it is very improbable that a route via one of these states would be the best path at the conclusion of the utterance if there are many states. Beam search is a method for reducing the amount of computation required. states with a cumulative score lower than the best one minus a set threshold are ignored. Using this method, the computational burden associated with expanding faulty nodes is minimised. Because of its lack of sophistication, pruning has the undesired attribute of being inadmissible, which may result in the loss of the best possible route. Beam threshold modification may increase search performance by an order of magnitude while adding only a minimal number of search mistakes in reality.

Because of this, the network gets too large when the dictionary contains more than tens of thousands of words.

There are now a variety of methods for coping with very big vocabularies. It's common for them to use multi-pass methods. The search space is shrunk with each iteration as data is prepared for the next.

Some systems use word lattices to store potential interpretations, while others use word lattices to store acoustic scores together with the placement of words, depending on how they're structured. Word lattice building may just need a Viterbi beam-search with word scoring and location memorised.

There are more accurate language models and occasionally more sophisticated audio models used to score the lattice words. Rescoring may involve fresh computations of HMM probability or may just use acoustic models that are not pre-calculated, depending on the needs of the application. It is possible to use a long-distance language model in the last stage, which is done by searching the word lattice and using a long-distance language model. Using trigram probabilities, a dynamic programming approach is used [13-22].

There is an approach that does not utilise the term lattice. Continuous speech recognition (CSR) is one of the earliest approaches suggested, and it combines sophisticated language modelling with thorough acoustic modelling in a single phase.

5.CONCLUSION

An technique termed back-propagation has been developed and implemented in this thesis to identify characters. The neural network is trained and evaluated on a total of sixty-two input patterns. A hidden layer of thirty nodes and an input layer of thirty-five nodes are used during training. Using the back Propagation method, an artificial neural network approach is used. There must be inputs and a goal output in order for this to work. When the mean squared error (MSE) of the artificial neural network is attained, the notion of training the network using input and target outputs is halted. The final weight is recorded in a file at this stage. Once we have determined the attributes of each character in a document, we compare them to the learnt set. Summation of the squared discrepancies between each attribute of the extracted character and the learnt character returns a "Confidence" for the comparison procedure.

the full linked list of learnt characters; however, in order to speed things up, we changed this somewhat to categorise characters based on their baselines. Below-the-baseline letters like "g," "j," and "y" form one group; tall letters like "l" and "T," short characters like "a," and floaty ones like "" and "" form another. Once a character has been categorised, it is compared to other characters in the same group that have been learnt. If a suitable match is not identified, the extracted character is compared to the other learnt characters, independent of their group.

References

1. X. Aubert, C. Dugast, H. Ney, and V. Steinbiss. Large vocabulary continuous character recognition of wall street journal data.

2.T. H. Applebaum and B. A. Hanson. Regression features for recognition of speech in quiet and in noise.

3. F. Alleva, X. Huang, and M. Y. Hwang. An improved search algorithm using incremental knowledge for continuous speech recognition.

4. T. Anastasakos, J. Makhoul, and R. Schwartz. Adaptation to new microphones using tied-mixture normalization.

5. Advanced Research Projects Agency. *Proceedings of the 1993 ARPA Human Language Technology Workshop*, Princeton, New Jersey, March 1993. Morgan Kaufmann.

6. Advanced Research Projects Agency. *Proceedings of the 1994 ARPA Human Language Technology Workshop*, Princeton, New Jersey, March 1994. Morgan Kaufmann.

7. Advanced Research Projects Agency. *Proceedings of the 1995 ARPA Human Language Technology Workshop*. Morgan Kaufmann, January 1995.

8. Advanced Research Projects Agency. *Proceedings of the ARPA Spoken Language Systems Technology Workshop*. Morgan Kaufmann, January 1995.

9. A. Acero and R. M. Stern. Environmental robustness in automatic speech recognition.

10. V. M. Alvarado and H. F. Silverman. Experimental results showing the effects of optimal spacing between elements of a linear microphone array.

11. Bishnu S. Atal. Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification. *Journal of the Acoustical Society of America.*

12. S.Kannadhasan, G.Karthikeyan and V.Sethupathi, A Graph Theory Based Energy Efficient Clustering Techniques in Wireless Sensor Networks. Information and Communication Technologies Organized by Noorul Islam University (ICT 2013) Nagercoil on 11-12 April 2013, Published for Conference Proceedings by IEEE Explore Digital Library 978-1-4673-5758-6/13 @2013 IEEE.

[13] Singh, D., Buddhi, D., & Karthick, A. (2022). Productivity enhancement of solar still through heat transfer enhancement techniques in latent heat storage system: a review. Environmental Science and Pollution Research, 1-34.

[14] Haseena, S., Saroja, S., Madavan, R., Karthick, A., Pant, B., & Kifetew, M. (2022). Prediction of the Age and Gender Based on Human Face Images Based on Deep Learning Algorithm. Computational and Mathematical Methods in Medicine, 2022.

[15] Jasti, V., Kumar, G. K., Kumar, M. S., Maheshwari, V., Jayagopal, P., Pant, B., ... & Muhibbullah, M. (2022). Relevant-based feature ranking (RBFR) method for text classification based on machine learning algorithm. Journal of Nanomaterials, 2022.

[16] Babu, J. C., Kumar, M. S., Jayagopal, P., Sathishkumar, V. E., Rajendran, S., Kumar, S., ... & Mahseena, A. M. (2022). IoT-based intelligent system for internal crack detection in building blocks. Journal of Nanomaterials, 2022.

[17] Chidambaram, S., Ganesh, S. S., Karthick, A., Jayagopal, P., Balachander, B., & Manoharan, S. (2022). Diagnosing Breast Cancer Based on the Adaptive Neuro-Fuzzy Inference System. Computational and Mathematical Methods in Medicine, 2022.

[18] Saroja, S., Madavan, R., Haseena, S., Pepsi, M., Karthick, A., Mohanavel, V., & Muhibbullah, M. (2022). Human centered decision-making for COVID-19 testing center location selection: Tamil Nadu—a case study. Computational and Mathematical Methods in Medicine, 2022.

[19] Kumar, R. R., Thanigaivel, S., Priya, A. K., Karthick, A., Malla, C., Jayaraman, P., ... & Karami, A. M. (2022). Fabrication of MnO2 Nanocomposite on GO Functionalized with Advanced Electrode Material for Supercapacitors. Journal of Nanomaterials, 2022.

[20] Karthick, A., Mohanavel, V., Chinnaiyan, V. K., Karpagam, J., Baranilingesan, I., & Rajkumar, S. (2022). State of charge prediction of battery management system for electric vehicles. In Active Electrical Distribution Network (pp. 163-180). Academic Press.

[21] Bharathwaaj, R., Mohanavel, V., Karthick, A., Vasanthaseelan, S., Ravichandran, M., Sakthi, T., & Rajkumar, S. (2022). Modeling of permanent magnet synchronous motor for zero-emission vehicles. In Active Electrical Distribution Network (pp. 121-144). Academic Press.

[22] Jayalakshmi, Y., Subramaniam, U., Baranilingesan, I., Karthick, A., Rahim, R., & Ghosh, A. (2021). Novel Multi-Time Scale Deep Learning Algorithm for Solar Irradiance Forecasting. Energies 2021, 14, 2404.

.