REALIZATION OF ARTIFICIAL NEURAL NETWORK IN SIMPLE MATHEMATICAL PENDULUM EXPERIMENT

Vijo M Joy[1*], Sivaraj K S[2], Tharanath R[3], Haripriya M[4]

[1]*Assistant Professor, Department of Electronics, Aquinas College Edakochi, Cochin, Kerala*
[2,3]*Assistant Professor, PG Department of Physics, Aquinas College Edakochi, Cochin, Kerala*
[3]*PG Department of Physics, Aquinas College Edakochi, Cochin, Kerala*
[*]*vijomjoy@gmail.com*

ABSTRACT

The objective of this study is to apply Artificial Neural Networks (ANNs) in straightforward mathematical pendulum experiments for gravity determination. The simple pendulum is a classic physics experiment that involves a mass attached to a string and it exhibits simple harmonic motion. The acceleration due to gravity is a fundamental constant in physics, and has various scientific and engineering applications. By collecting experimental data including the pendulum's parameters such as length, angle, and time period, we construct a dataset for training and testing ANNs. Four datasets were initially collected and subsequently interpolated to generate additional data. These augmented datasets were used for training each model. Training is done with the help of Neural Network tool available in MATLAB. Each model underwent testing to derive gravity values, which were then compared to reference gravity values. The Neural Network model exhibited excellent accuracy. The same modelling can be applied on any physical pendulum in which oscillation is generally performed by any material body suspended on a horizontal axis and allowed to rotate about the axis.

Keywords: Artificial Neural Network, Simple Pendulum, Gravity, Backpropagation

I. INTRODUCTION

The measurement of gravity is a fundamental aspect of physics and plays a pivotal role in understanding the behaviour of objects under its influence. One of the classic experiments used to determine the value of gravity is the simple mathematical pendulum experiment [1]. In this experiment, a mass is suspended from a fixed point, forming a pendulum, and allowed to oscillate freely under the influence of gravity [2]. By measuring the period of oscillation of the pendulum, we can derive the value of gravitational acceleration (g) with high precision [3] [4].

Traditionally, this experiment has been conducted using classical physics formulas and precise measurement devices. However, in recent years, there has been a growing interest in utilizing artificial intelligence techniques, particularly ANNs, to enhance the accuracy and efficiency of such experiments [5]. A Neural Network is a computational system designed to replicate the functioning of the human brain using simplified neuron models and their interconnections [6]. These networks are mathematically modelled to simulate human-like intelligence and find applications in various technologies [7]. Neural Networks adapt their free parameters through environmental stimuli, a process known as learning, and the type of learning is determined by how these parameters change [8]. Neural Networks consist of numerous processing elements interconnected to create a network with adjustable weighting functions for each input. Typically organized into three or more layers, they include input layers for data presentation, output layers for response generation, and one or more intermediate or "hidden layers." ANNs have demonstrated their ability to solve complex problems and make accurate predictions in various fields, including physics. Neural Networks offer several advantages: firstly, they excel in constructing high-level nonlinear function estimation models, and secondly, they do not impose limitations on the number of features [9]. In this work, we investigate the use of artificial neural networks to estimate the value of gravity in the simple pendulum experiment. A neural network can be created using MATLAB's neural network toolbox.

A model for predicting the gravitational acceleration with high accuracy can be created by training a neural network on a dataset of pendulum oscillation times and accompanying gravitational accelerations. This method has a number of benefits, such as the possibility to cut down on experimental errors and the capacity to take into consideration a number of variables that might have an impact on the behaviour of the pendulum. The subsequent sections of this research will delve into the methodology, data collection, and analysis, as well as the results obtained from the ANN-based approach. We will also compare the accuracy and reliability of our neural network model with traditional methods of gravity measurement, demonstrating the potential of artificial intelligence in revolutionizing experimental physics [10].

Research method: The mathematical pendulum experiments involve the collection of period data from pendulum swings, measuring the time for a single swing. Utilizing the known string length, the gravitational (g) value is computed using a specific equation. To enhance the dataset, experimental data is interpolated using Newton's interpolation method, generating additional data points not originally obtained. This expanded dataset serves as valuable training data. The combined dataset, comprising both interpolated and experimental data, is employed to train Linear Regression models and machine learning algorithms. These models are utilized to predict gravity values beyond the scope of the experimental results. The performance of each model is assessed by analysing error values, standard deviations, and percentage accuracy in comparison to reference values [11].

The proposed work involves four main phases:

- Data Collection: Experimental data is gathered by varying the length and initial angle of the pendulum and measuring the resulting time periods of oscillation. These data points are carefully recorded and used to create a comprehensive dataset.
- Model Development: Artificial neural networks are designed and trained using the collected data. The ANN is trained to learn the complex relationship between the

pendulum parameters and the acceleration due to gravity. Different ANN architectures and hyper parameters are explored to find the best-performing model for predicting the time period of the pendulum under various conditions using the software MATLAB.

- Evaluation and Prediction: The trained ANN model is subsequently used to predict the value of g based on new data from pendulum experiments. The accuracy of the model is assessed through comparisons with known gravitational acceleration values and conventional calculation methods.
- Validation: Assessing the accuracy and reliability of the ANN's predictions by comparing them with experimental data.

The outcomes of this project provide a practical example of utilizing ANNs for predicting physical phenomena, showcasing their versatility beyond traditional data analysis applications. The results contribute to a deeper understanding of the simple pendulum system and the potential of artificial neural networks in solving complex prediction problems in physics and engineering.

II. LITERATURE REVIEW

A simple mathematical pendulum can be described as a mechanical system exhibiting simple Harmonic motion. A point mass hanging from a rigid support by an inextensible string makes up the ideal simple pendulum. In actual use, a round metallic bob of a specific mass is suspended using a cotton thread and mounted to a retort stand [12]. Galileo Galilei was the first scientist to study about the properties of Simple Pendulum. His study started from 1602.He found out that time period of the pendulum is independent of the amplitude of oscillations, mass of the bob etc. and depends only on the length of the pendulum and acceleration due to gravity. The motion of simple pendulum follows predictable patterns, making it a valuable tool for studying concepts like periodic motion, gravity and harmonic motion. Its behaviour can be described mathematically using equations that relate its period, length and gravity acceleration [13]. Simple pendulum has a variety of daily life applications the most important being the use of the pendulum in the measurement of time. Since the time period of the pendulum remains constant it is employed in clocks in ancient times. Pendulums are also used in seismometers to measure the magnitude of earthquakes and have many other applications. In the laboratory we are focused mainly in measuring the acceleration due to gravity [14].

III. MATHEMATICAL MODELLING OF SIMPLE PENDULUM

Consider a simple pendulum executing simple harmonic motion. Let ‘ l’ be the pendulum’s length, ‘m’ be the bob’s mass and θ represents the angle the string makes with the equilibrium position when it oscillates.

The weight of the bob towards the downward direction and the tension T along the string in the upward direction are the two forces acting on the bob. Weight mg can be divided into two perpendicular halves $mgCos\theta \cdots and \cdots mgSin\theta$ which are both perpendicular to the string.

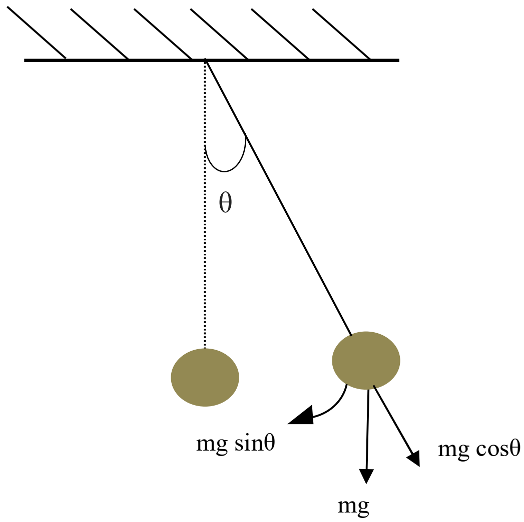

Figure 1: Simple pendulum executing simple harmonic motion

$mgCos\theta$ is balanced by tension T.

$$T = mgCos\theta \tag{1}$$

$mgSin\theta$ is balanced by restoring force

$$F = -mgSin\theta \tag{2}$$

Since pendulum is oscillated for small angle,

$$Sin\theta \approx \theta$$

Thus, Equation 2 becomes,

$$F = -mg\theta \tag{3}$$

Applying Newton's Third law and the concept of simple harmonic motion,

The, time period of the simple pendulum is obtained as

$$T = 2\pi\sqrt{l/g} \tag{4}$$

Squaring and rearranging we get the acceleration due to gravity,

$$g = 4\pi^2(l/T^2) \tag{5}$$

IV. ARTIFICIAL NEURAL NETWORKS

ANNs are stimulated by the functionality and edifice of the human brain. It seems like a network consisting of a large number of interconnected components called neurons The processing elements or the artificial neurons are the basic functional units of ANN. The ANN is designed by programming computers to behave like brain cells so that computer will have an ability to understand things and make decisions in a human like manner [15] [16].

Three different layers of neurons are used for processing. The layers are the input layer, hidden layer, and output layer. The layers are interconnected to each other by weights. Each artificial neuron takes an input from the input sources, applied an activation function to this input, and generates the net result. [17] [18].

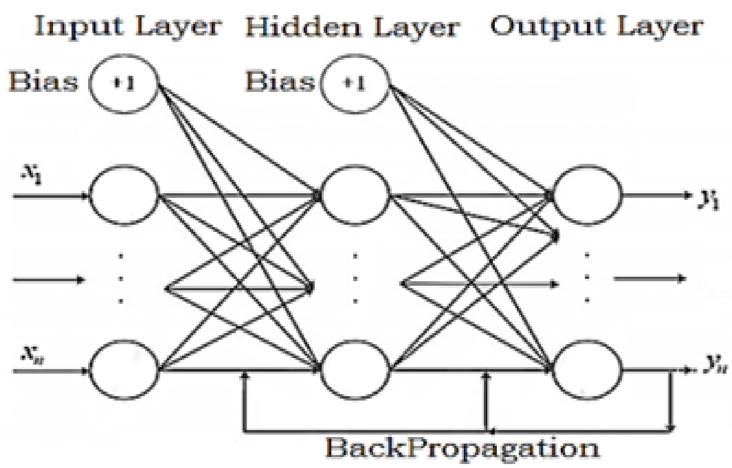

Figure 2 .Structure of BPNN

The Backpropagation Neural Network (BPNN) method is used for ANN training. It is a multilayer feed-forward neural network. The important steps for the training of a NN are as follows;

- Input vectors are feed forwarded
- The computer error is then backpropagated
- Updates the weight function to minimize the error.

This BPNN algorithm targets the reduction of root mean square error of the response. The delta rule is used for weight updation.The NN training performance depends on the initial weights, learning rate, update interval, and the hidden layers.

Figure 1 depicts the BPNN structure. There are three different kinds of layers - Input, Hidden and Output Layers. The neurons in the input layer receive input signals from outside sources and transmit them to the neurons in the subsequent layers. In the hidden layer stage, no calculations are made. The input or hidden layer sends signals to the neurons in the output layer. Between the input and output layers hidden layer neurons are connected [19] [20].

(a) . LEVENBERG–MARQUARDT [LM] ALGORITHM

Many enhanced learning algorithms have been presented by the researchers to surmount the limitations of gradient descent-based systems. All these algorithms are derivatives of steepest gradient search, so ANN training is relatively slow. Second order learning algorithms have to be used for fast and efficient training; the most effective method is LM algorithm, which is a derivative of the Newton method. In NN model LM is one of the efficient and well-organized optimization algorithms. It is a combination of Gradient descents method and Gauss-Newton method. This combined technology is the best practice to explain a different range of optimization problems. This is a typical technique for the non-linear least square problem, and this iteration technique locates to the minimum of a function that is expressed as the sum of squares of non-linear functions. Gradient descent and Gauss-Newton methods use a series of calculations to find the solutions for non-linear problems [21].

This is a hybrid technique, such that Gradient descents method and Gauss-Newton method converge for an optimum solution. To solve the different optimization problems this, type of hybrid technology is the best practice. The system is very effective to handle small and medium sets of data and solve non-linear equations. The LM algorithm was developed only for layer-by-layer ANN topology, which is far from optimal. LM algorithm is increasing the convergence speed of ANN. The system performance is calculated in Mean Square Error (MSE). The LM algorithm is still unable to avoid the drawback of local minimum. To surmount this problem a bio-inspired optimization algorithms was combined with the LM algorithm to train the NN [22] [23].

V. METHODOLOGY

Preparation of data using simple pendulum experiment

1. Measure the diameter of the spherical metallic bob using vernier callipers. From it calculate the radius r of the bob.
2. Tie the bob to one end of an inextensible cotton thread. The other end is then passed through a split cork fixed to a retort stand.
3. The length of the pendulum that is the length from the point of suspension to middle of the bob is set to a desired value.
4. Displace the bob from the equilibrium position to one side with a small angular displacement and with the angle of oscillation being 10^{o} and then release it gently such that bob executes to and fro motion in a straight line, without spinning and revolving elliptically.

5. Now start counting the oscillation. Stop watch is started when pendulum crosses the equilibrium position to one side either left or right and the time for 20 oscillations is noted. Repeat the observation one more time for the same length.
6. Calculate the time taken for 1 oscillation i.e., the time period using the equation
   $T = t / 20$
7. Repeat the experiment by changing length.
8. Record the observation in tabular form.
9. Calculate T^2 and L/T^2 in each case and take the mean of L/T^2.
10. Find acceleration due to gravity g using,
11. Now change the angle of oscillation θ to 15^{0} and then to 20^{0} and repeat the same procedure.
12. Repeat the whole experiment for different masses of the bob The experimentally obtained value of g is compared with its theoretical value and the percentage error is calculated.

Prediction of data using ANNs

1. Preparation of data
2. Open the MATLAB
3. Add data to workspace.
   i. Select New
   ii. Rename the data
   iii. copy the variable to workspace
   iv. Transpose the variables. [Input, Target, Sample].
4. In command window type the command nn tool.
5. Import the variables to neural networks.
6. Create the neural network
   i. Network type- Feed forward back propagation.
   ii. Set the training parameters.
7. Train the network.
8. After the training is completed plot the regression analysis graph.
9. Stimulate this trained system to predict the future response of the simple pendulum.

VI. RESULTS AND DISCUSSION

The training architectures of networks with varying numbers of hidden layers are depicted in the figures given below. Initially, a configuration with 10 hidden layers is employed for training, utilizing four input parameters, including angle, length, and time. Figure 3 illustrates the structure of the training process, where a neural network is constructed with four input nodes at the input layer, 10 hidden layers, and a single-output layer responsible for predicting acceleration due to gravity.

The iterative training process continues until the optimal result is achieved, which, required 301 iterations. After the training is completed, the best validation performance, training state, and regression results are obtained.

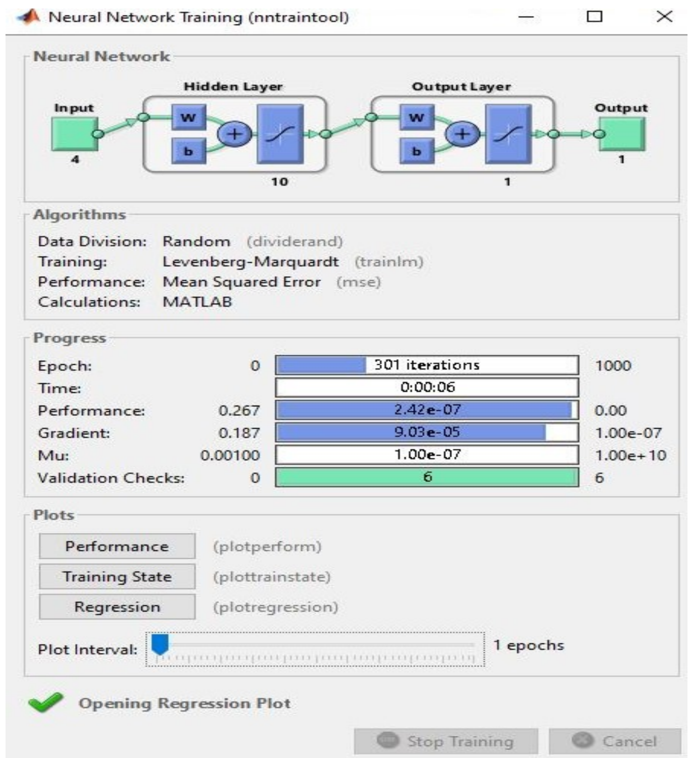

Figure 3: Training structure for 10 hidden layers

To get a better result the number of hidden layers changes to 25 and the training structure is shown in Figure 4. The iterative training process continues until the optimal result is achieved, which, in this case, required only 30 iterations. After the training is completed, the best validation performance, training state, and regression results are obtained.

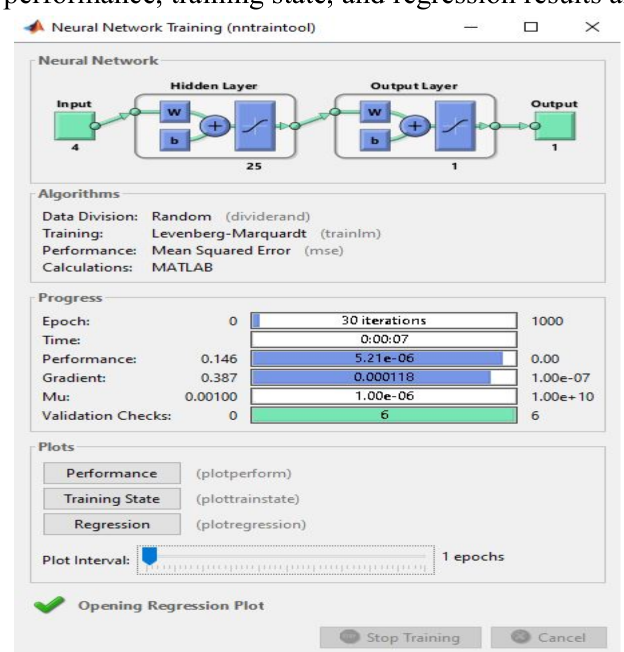

Figure 4: Training structure for 25 hidden layers

Finally, the number of hidden layers changes to 50 and the training structure is shown in Figure 5. The iterative training process again continues until the optimal result is achieved, which, in this case, required lesser iteration, i.e., 27 iterations and the time taken to complete the execution is 01 sec. After the training is completed, the best validation performance, training state, and regression results are obtained.

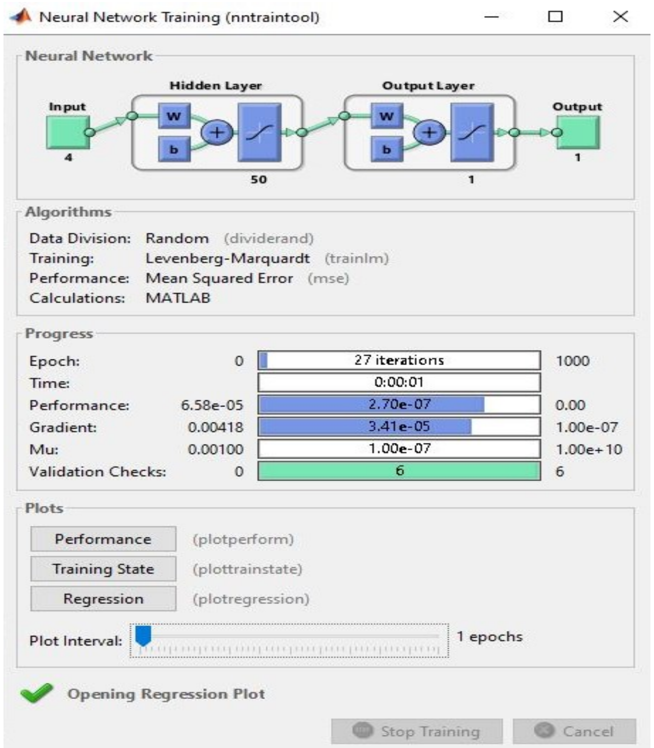

Figure 5: Training structure for 50 hidden layers

Figures 6-8, below illustrate the Regression (R) analysis of the system. It is an arithmetical procedure for approximating the relationship between the variables. The regression standard scale is the association between goals and system response. The significance of the regression is '1' means an adjacent association and is '0' means random relationship between the parameters. Here, 70% of the data is used for the training, 15% is employed for validation and the remaining 15 % is used for testing. The regression analysis of network with different hidden layers such as 10, 25 and 50 are shown in Figures 6 - 8.

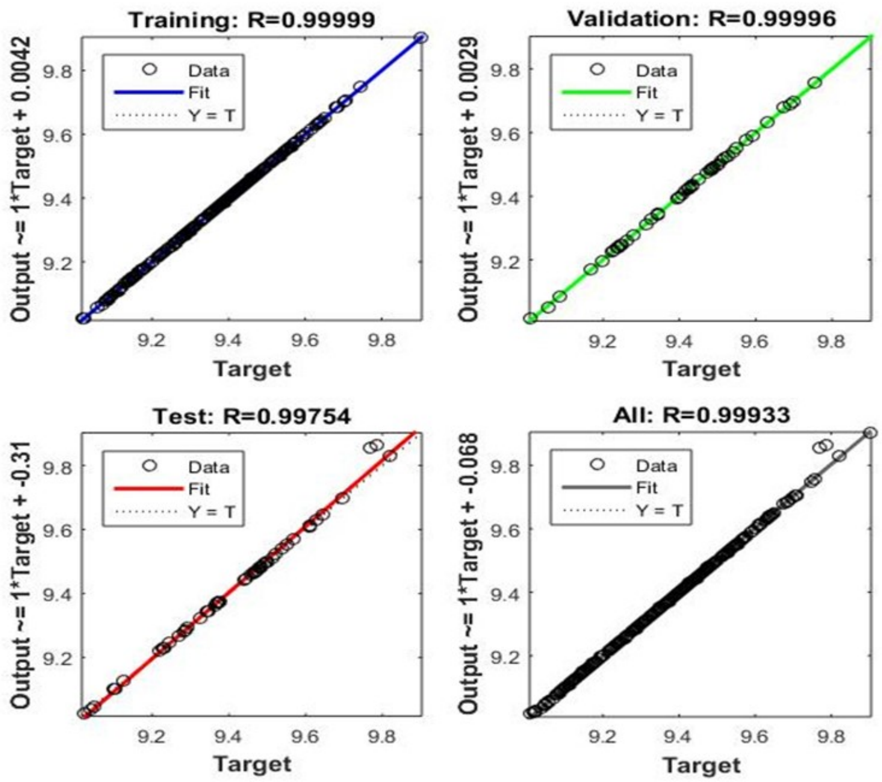

Figure 6: Regression analysis of network with 10 hidden layers

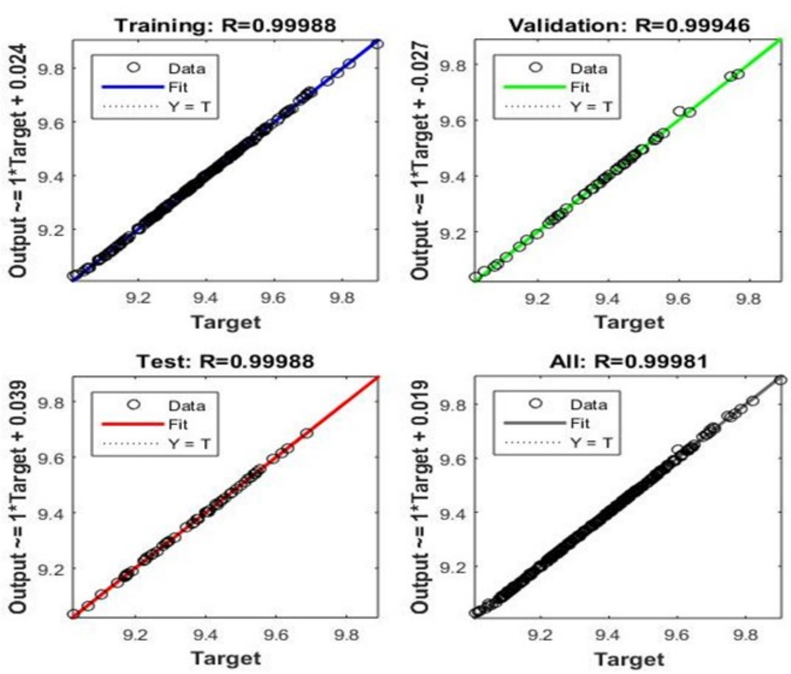

Figure 7: Regression analysis of network with 25 hidden layers

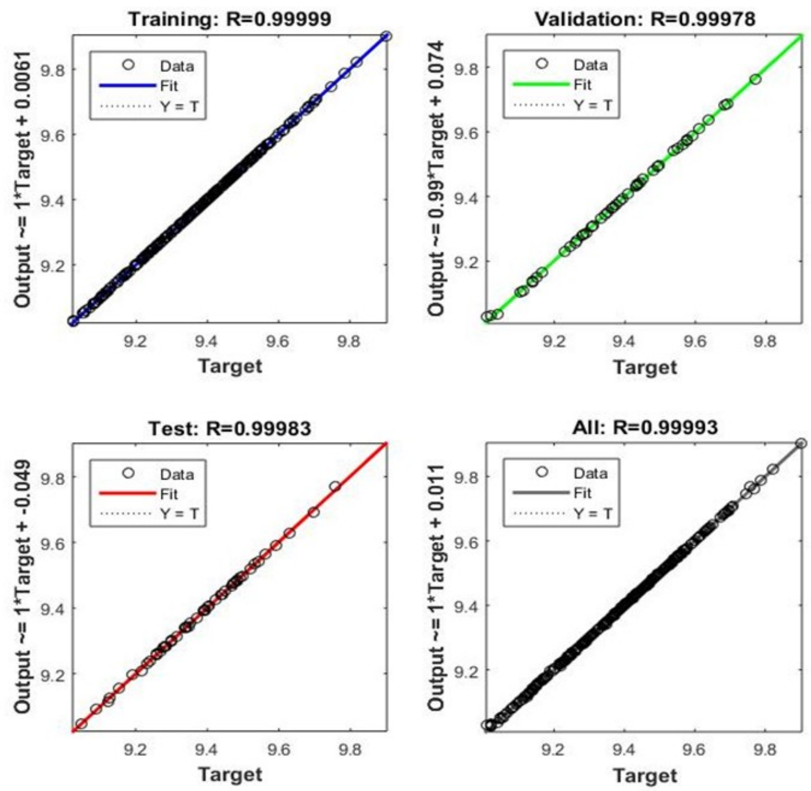

Figure 8: Regression analysis of network with 50 hidden layers

In the regression analysis, it becomes evident that R values are as follows: R = 0.99933 for 10 hidden layers, R = 0.99981 for 25 hidden layers, and R = 0.99993 for 50 hidden layers. Notably, the highest R value is achieved when employing 50 hidden layers, suggesting that accuracy improves with the augmentation of hidden layers.

The experiment has been carried out to determine the value of gravity using the pendulum swing method where the method involves collection of period data, T from pendulum swings. If we know the length, L of the rope used by the pendulum, the gravitational value can be obtained.

Gravitational values are taken which are closest to the reference gravity value for each different rope length. The data obtained is as shown in Table 1. This set of data was interpolated using network to obtain data which could not be measured in the experiment. This set of interpolated data is shown in Table 2. These data obtained experimentally and by interpolation are appropriate for training and application to machine learning system which can further predict the gravity values for rope length and pendulum swing periods that are greater than the variables and parameters possible for the experiments. The set of predicted data obtained is as shown in Table 3.

Table 1: Test result data

Sl. No	Angle	Length (m)	Time(s)	g(kg.m/s2)	Accuracy (%)
1	10	0.146	0.799	9.032	92.162
2	20	0.146	0.793	9.168	93.554
3	15	0.146	0.792	9.173	93.604
4	20	0.193	0.917	9.037	92.210
5	10	0.193	0.916	9.059	92.437
6	15	0.193	0.907	9.239	94.278
7	20	0.273	1.081	9.203	93.906
8	15	0.273	1.065	9.475	96.686
9	10	0.273	1.062	9.534	97.282
10	20	0.313	1.169	9.028	92.119
11	15	0.313	1.155	9.247	94.356
12	10	0.313	1.153	9.280	94.690
13	20	0.453	1.391	9.223	94.114
14	15	0.453	1.372	9.481	96.744
15	10	0.453	1.358	9.684	98.814
16	20	0.466	1.410	9.244	94.324
17	15	0.466	1.409	9.258	94.469
18	10	0.466	1.403	9.340	95.301
19	20	0.469	1.429	9.053	92.373
20	15	0.469	1.406	9.360	95.506
21	10	0.469	1.402	9.417	96.090
22	20	0.513	1.479	9.246	94.351
23	15	0.513	1.471	9.344	95.348
24	10	0.513	1.458	9.512	97.057
25	10	0.526	1.493	9.303	94.932
26	20	0.589	1.577	9.340	95.310
27	15	0.589	1.573	9.391	95.830
28	10	0.589	1.553	9.631	98.278
29	10	0.693	1.693	9.534	97.285
30	15	0.693	1.690	9.563	97.586
31	20	0.695	1.706	9.418	96.100
32	20	0.753	1.791	9.249	94.381
33	15	0.753	1.776	9.407	95.987
34	10	0.753	1.761	9.576	97.711
35	20	0.813	1.852	9.345	95.354
36	15	0.813	1.842	9.447	96.397
37	10	0.813	1.839	9.480	96.737
38	20	0.886	1.934	9.345	95.354
39	15	0.886	1.929	9.396	95.878
40	10	0.886	1.928	9.403	95.945
41	15	0.933	2.009	9.109	92.953
42	20	0.933	2.009	9.116	93.023
43	10	0.933	1.988	9.309	94.986
44	10	0.953	2.033	9.094	92.791
45	20	0.953	2.031	9.105	92.907
46	15	0.953	2.029	9.116	93.021

Table 2: Data interpolated & predicted

Sl. No	Angle	Length (m)	Time(s)	g(kg.m/s2)	Accuracy (%)
1	20	0.393	1.296	9.215	94.042
2	15	0.393	1.295	9.237	94.260
3	10	0.393	1.280	9.456	96.483
4	10	0.406	1.312	9.298	94.882
5	15	0.406	1.308	9.359	95.500
6	20	0.406	1.297	9.526	97.202
7	15	0.409	1.315	9.327	95.184
8	20	0.409	1.310	9.407	95.985
9	10	0.409	1.304	9.486	96.797
10	20	0.413	1.333	9.156	93.424
11	15	0.413	1.307	9.530	97.252
12	10	0.413	1.305	9.552	97.476
13	15	0.426	1.339	9.367	95.583
14	10	0.426	1.338	9.392	95.833
15	20	0.426	1.316	9.735	99.336
16	20	0.429	1.363	9.105	92.931
17	10	0.429	1.346	9.338	95.293
18	15	0.429	1.345	9.359	95.506
19	20	0.433	1.355	9.289	94.798
20	15	0.433	1.344	9.447	96.392

Table 3: Data on overall results of interpolation and prediction

Sl. No	Angle	Length (m)	Time(s)	g(kg.m/s2)	No. of Hidden Layers		
					10	25	50
1	20	0.393	1.296	9.216	9.216	9.220	9.215
2	15	0.393	1.295	9.238	9.237	9.235	9.237
3	10	0.393	1.280	9.455	9.455	9.455	9.456
4	10	0.406	1.312	9.298	9.298	9.299	9.298
5	15	0.406	1.308	9.359	9.359	9.360	9.359
6	20	0.406	1.297	9.526	9.526	9.522	9.526
7	15	0.409	1.315	9.328	9.328	9.328	9.327
8	20	0.409	1.310	9.407	9.407	9.403	9.407
9	10	0.409	1.304	9.486	9.486	9.485	9.486
10	20	0.413	1.333	9.156	9.156	9.155	9.156
11	15	0.413	1.307	9.531	9.531	9.530	9.530
12	10	0.413	1.305	9.553	9.552	9.552	9.552
13	15	0.426	1.339	9.367	9.367	9.368	9.367
14	10	0.426	1.338	9.392	9.392	9.391	9.392
15	20	0.426	1.316	9.697	9.655	9.625	9.735
16	20	0.429	1.363	9.107	9.106	9.103	9.105
17	10	0.429	1.346	9.339	9.339	9.338	9.338
18	15	0.429	1.345	9.360	9.360	9.360	9.359
19	20	0.433	1.355	9.290	9.290	9.293	9.289
20	15	0.433	1.344	9.446	9.447	9.448	9.447

Table 3 shows the network model with different number of hidden layers. Lowest error rate is obtained at 50 hidden layers. This shows that the accuracy of the NN model is best when using 50 hidden layers compared with the lower hidden layers for this case. When using a greater number of hidden layers, the training time increases. Next, we analyze the error rate. The average error rate for different model is shown in Figure 9.

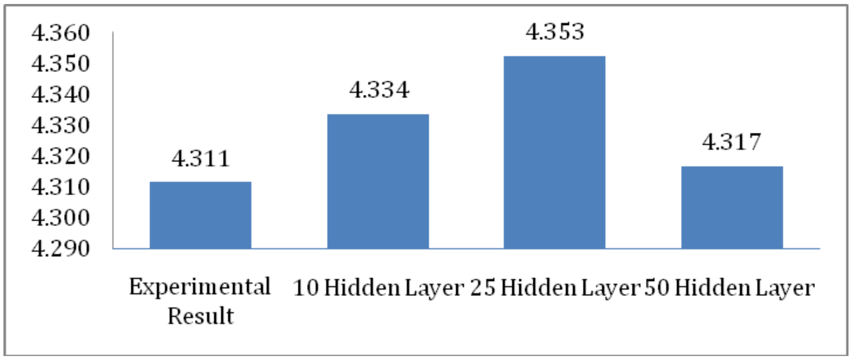

Figure 9: Average error rate

The Figure 9 displays an average error rate of 4.311 for experimental results. When employing 10 hidden layers, the predicted output yields an average error rate of 4.334. Upon adjusting the number of hidden layers to 25, an average error rate of 4.353 is observed. Interestingly, when employing 50 hidden layers, the average error rate decreases to 4.317, which is lower than the rates observed with fewer hidden layers. The average accuracy of the present system is 95.708. Comparison of error rate for different model is shown in Figure 10.

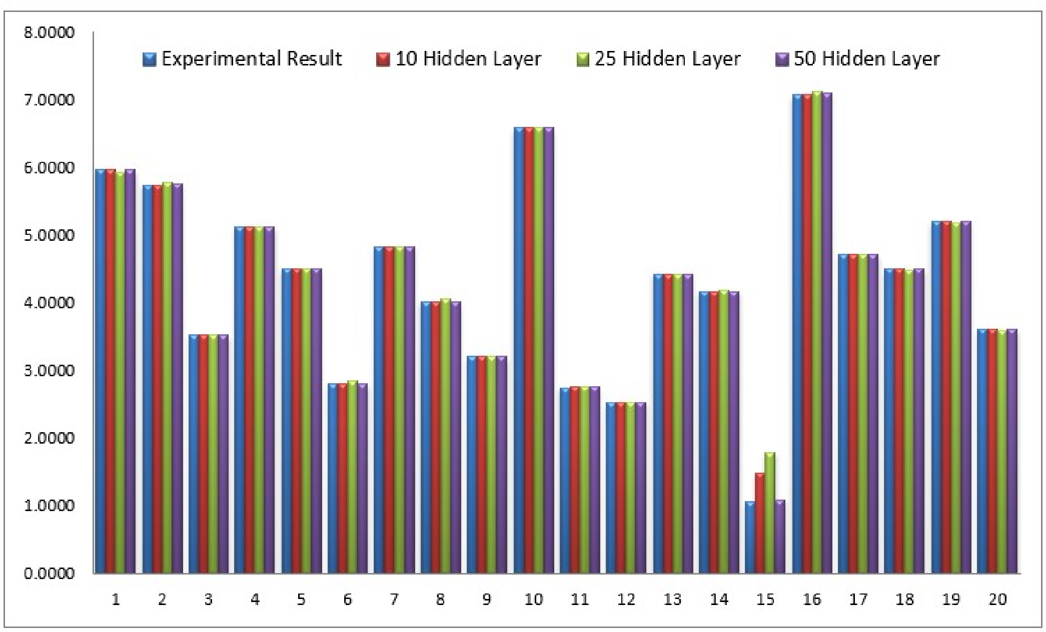

Figure 10: Comparison of error rates

VII. CONCLUSION AND FUTURE SCOPE

In conclusion, the project on the "Realization of Artificial Neural Network to Determine the Value of Gravity in Simple Mathematical Pendulum Experiment" has been a significant exploration into the integration of modern machine learning techniques with classical physics experiments. Throughout this project, we have achieved several important milestones and gained valuable insights. First and foremost, we successfully designed and conducted a simple mathematical pendulum experiment, collecting data on the pendulum's oscillation periods for various lengths. These empirical observations formed the foundation of our project, providing the necessary dataset for training and testing our artificial neural network.

The implementation of the artificial neural network to predict the value of gravity based on the pendulum's oscillation periods demonstrated the potential of machine learning in scientific research. By feeding the network with input data and fine-tuning its parameters, we were able to create a predictive model that offered accurate estimations of the gravitational acceleration constant. This project underscores the versatility of artificial neural networks, not only for complex tasks but also for scientific measurements and experimentation. It highlights the potential for machine learning to complement traditional scientific methods, offering new perspectives on data analysis and prediction. Moreover, the project has contributed to a deeper understanding of the relationship between the length of a simple pendulum and the period of its oscillation, as well as the practical application of neural networks in solving real-world physics problems.

Future Scope

The "Realization of Artificial Neural Network to Determine the Value of Gravity in Simple Mathematical Pendulum Experiment" project has successfully bridged the gap between classical physics and modern machine learning. It represents a significant step forward in utilizing artificial intelligence to enhance scientific research and provides a valuable framework for further exploration and integration of these technologies in the field of experimental physics. This project has not only achieved its intended goals but has also opened doors to exciting possibilities at the intersection of physics and machine learning. Using ANN, the data obtained from a relatively simple pendulum experiment can be extrapolated to train the algorithms to predict the gravitational values for similar yet complex mechanical systems. We propose to extend this work to study more chaotic systems like double pendulum and utilize the versatility of ANN to obtain accurate values.

References

[1]. Nelson, Robert A., and M. G. Olsson. "The pendulum—Rich physics from a simple system." American Journal of Physics , vol. 54, no. 2, (1986), pp. 112-121.

[2]. Rowan, Sheila, and Jim Hough. "Gravitational wave detection by interferometry (ground and space)." Living Reviews in Relativity, vol. 3, no. 1, (2000), pp. 1-41.

[3]. Parks, Harold V., and James E. Faller. "Simple pendulum determination of the gravitational constant." Physical Review Letters, vol. 105, no.11 (2010), p. 110801.

[4]. Salehi, Hadi, and Rigoberto Burgueño. "Emerging artificial intelligence methods in structural engineering." Engineering structures, vol. 171, (2018), pp. 170-189.

[5]. Cuomo, Salvatore, Vincenzo Schiano Di Cola, Fabio Giampaolo, Gianluigi Rozza, Maziar Raissi, and Francesco Piccialli. "Scientific machine learning through physics–informed neural networks: Where we are and what's next." Journal of Scientific Computing , vol. 92, no. 3, (2022), p. 88.

[6]. Shanmuganathan, Subana. Artificial neural network modelling: An introduction. Springer International Publishing, 2016.

[7]. Nolfi, Stefano, and Domenico Parisi. "Learning to adapt to changing environments in evolving neural networks." Adaptive behaviour, vol. 5, no. 1, (1996), pp. 75-98.

[8]. Zayas Victor, Stanley S.L and Stephen AMahin, "A simple pendulum technique for achieving seismic isolation." Earthquake spectra, vol. 6, no. 2 (1990), pp. 317-333.

[9]. Woldseth, Rebekka V., Niels Aage, J. Andreas Bærentzen, and Ole Sigmund. "On the use of artificial neural networks in topology optimisation." Structural and Multidisciplinary Optimization, vol. 65, no. 10 (2022), p. 294.

[10]. Bender, Carl M., Darryl D. Holm, and Daniel W. Hook. "Complex trajectories of a simple pendulum." Journal of Physics A: Mathematical and Theoretical, vol. 40, no. 3 (2006), p.81.

[11]. Rodriguez-Galiano, V., M. Sanchez-Castillo, M. Chica-Olmo, and M. J. O. G. R. Chica-Rivas. "Machine learning predictive models for mineral prospectivity: An evaluation of neural networks, random forest, regression trees and support vector machines." Ore Geology Reviews, vol. 71, (2015), pp. 804-818.

[12]. Crossley, David, Jacques Hinderer, and Umberto Riccardi. "The measurement of surface gravity." Reports on Progress in physics, vol. 76, no. 4, (2013), p. 046101.

[13]. Lakshmanan, Muthusamy, and Shanmuganathan Rajaseekar. Nonlinear dynamics: integrability, chaos and patterns. Springer Science & Business Media, 2012.

[14]. Rothleitner, Christian, and Stephan Schlamminger. "Invited Review Article: Measurements of the Newtonian constant of gravitation, G." Review of Scientific Instruments , vol. 88, no. 11 , (2017).

[15]. Gupta, Tarun Kumar, and Khalid Raza. "Optimization of ANN architecture: a review on nature-inspired techniques." Machine learning in bio-signal analysis and diagnostic imaging, (2019), pp. 159-182.

[16]. Dongare, A. D., R. R. Kharde, and Amit D. Kachare. "Introduction to artificial neural network." International Journal of Engineering and Innovative Technology, vol. 2, no. 1, (2012), pp. 189-194.

[17]. Cirovic, Dragan A. "Feed-forward artificial neural networks: applications to spectroscopy." TrAC Trends in Analytical Chemistry, vol. 16, no. 3, (1997), pp. 148-155.

[18]. Cong, Tenglong, GuanghuiSu, Suizheng Qiu, and Wenxi Tian. "Applications of ANNs in flow and heat transfer problems in nuclear engineering: a review work." Progress in Nuclear Energy, vol. 62, (2013), pp. 54-71.

[19]. Vijo M Joy , Joseph John and S Krishnakumar, " Backpropagation neural network based adaptive load scheduling method in an isolated power system,", Bulletin of Electrical Engineering and Informatics, vol. 12, no. 5, (2023), pp. 3000-3007

[20]. Ranganathan, Ananth. "The levenberg-marquardt algorithm." Tutoral on LM algorithm, vol. 11, no. 1, (2004), pp. 101-110.

[21]. Vijo M Joy and S. Krishnakumar., "Optimal Design of Power Scheduling using Artificial Neural Network in An Isolated Power System," International J. of Pure and Applied Mathematics, vol. 118, (2018), pp. 289-294.

[22]. Vijo M Joy , Joseph John and S Krishnakumar, "Optimal Model for Effective Power Scheduling using Levenberg Marquardt Optimization Algorithm," SSRG International Journal of Electrical and Electronics Engineering, vol. 9, no. 10, (2022), pp. 1-6.

[23]. Lourakis, Manolis IA. "A brief description of the Levenberg-Marquardt algorithm implemented by levmar." Foundation of Research and Technology, vol. 4, no. 1 (2005), pp. 1-6.