Neural Network based automated hot water mixture

F.N.M Firsan[1], G.M Herath[2] and T.D Thilakanayake[3]
[1]School of Civil and Mechanical Engineering, Curtin University Kent St, Bentley WA 6102, Australia
[2,3]Faculty of Engineering, Sri Lanka Institute of Information Technology New Kandy Rd., Malabe 10115, Sri Lanka

ABSTRACT

In the present day and age, most residential spaces comprise a shower system and generallya conventional system of hot water showers. Throughout history, showering has developed as anessential need in a person's life. Nevertheless, a typical hot water shower system comprises delays in hot water mixing and usually requires an average of 2 to 4 minutes to mix the cold and hot waterto deliver the appropriate shower temperature. The delay in mixing provides less comfort and poorsatisfaction affecting people's lifestyles. Due to these disadvantages, a system incorporating artificial Intelligence can be utilized to enhance the performance of mixing which can offer an automated hot watermixture system with improved efficiency and effectiveness. Recently, significant research has been focusedon utilizing deep learning technology due to its multiple breakthroughs in fabricating a broad range of automated novel applications since Neural Networks comprise the capacity to learn from data to offerefficient and accurate systems. In this research project, the hot water mixture is employed by an ArtificialNeural Network model integrated with the combination of an embedded system of the proposed systemof hot water mixture. Furthermore, the proposed system comprises temperature and flow sensors along with controllable flow valves. The tested system indicated acceptable accuracy between the actualand desired output flow rate and temperature.

KEYWORDS: *neural networks, deep learning, hot water mixture, embedded system*

1 INTRODUCTION

Throughout history, showering has evolved into an essential need for people. The primary reason showering was considered a fundamental need is to maintain a satisfactory degree of hygiene and other positive effects with respect to people's health. Hot shower systems have developed and improved throughout history to improve the quality of showering. However, the existing conventional hot water shower control systems cause a delay of approximately 1 to 2 minutes in hot water mixing.

In general, there are two fundamental shower control systems. The pressure balance shower system comprises one lever which performs the on/off function, and the water temperature can be regulated even though the volumetric flow of water cannot be controlled with this type of system (Shower Systems Explained). When the valve of this system is turned on, it cannot be adjusted to the desired position since the valve can only function in completely open or closed positions. Moreover, the outputvolume of water cannot be regulated, nevertheless, an adjustable shower head may be employed to alterthe pressure. On the other hand, the thermostatic shower system comprises two levers, where one leveris capable of controlling the water temperature (thermostatic valve) and the other lever regulates the volume of water with the inclusion of an on/off function (volume control valve). This system enables the userto set the temperature permanently so that whenever the shower functions it provides the water temperature to the given value. However, when both basic shower systems are analyzed it can be said that they comprise drawbacks and lack versatility.

Therefore, this study offers a system of Artificial Intelligent hot water mixture that provides an improved duration of hot water mixing and a reliable system with increased versatilityto improve the lifestyles of people by delivering a comfortable shower experience. The study prioritizes the development of a hot water mixture system providing peoplewith an easy-to-use and versatile system to bring comfort to their lifestyle. The primary objective of thisstudy is to fabricate an automated hot water mixture system by implementing Neural Networks to provide the user the ability to input the desired shower temperature and flow rate to the automated hot water mixture system. In addition, a good automatic control mixture must be established, in simple terms,the proposed system must be an advanced and user-friendly system. Moreover, the secondaryobjectives were to construct a comparison utilizing learning strategies of Artificial Neural Networks (ANN), collect data to train the ANN model, develop the script of data collection and ANN model,and lastly develop the overall algorithm to function the automated system. Furthermore, utilizing Deep Neural Network technology provides a faster rate of hot water mixing as computations occur rapidlyto provide the best possible results and since this technology comprises the capacity to learn, the hot water mixing system can be improved over time by integrating more features and functions to the system. Hence, when constructing a comparison between Neural Network and conventional methods, Neural Network offers more advantages to the system. Therefore, the study examined the ways of utilizing current shower control systems to introduce an innovative and novel approach.

The study design comprises implementing a Neural Network based system by utilizing a significant amount of dataset to train the Neural network model successfully to acquire the best-performing model by employing several techniques, particularly hyperparameter optimization. The results of the best-performing models were then utilized by the microcontroller to provide the necessary instructions to the controllable flow valve when the user inputs the desired temperature and flow rate. This enables good automatic control and versatility for the user to insert the desired temperature and flow rate.

1.1 Literature background

Artificial Neural Networks (ANN) contain units of computation called neurons(Charu C. Aggarwal, 2018). Every input to a neuron consists of a scaled weight that affects the unitcomputation of that function. The propagation of computed values is transferred from the input tooutput neurons utilizing weights as the intermediary parameter in an ANN which evaluates thefunction of inputs. The learning process generally occurs by adjusting the weights connected withthe neurons and these weights are adjusted in an ANN in response to errors from predicted results.

The objective of adjusting the weights was to reshape the computed function in order to securemore accurate predictions in future iterations. Hence, weights are altered delicately in a mathematically validated pathway to minimize the error in the particular computation of a model. With successful alterations to the weights between a myriad of input and output pairs of neurons, thefunction computed by ANN gets refined over a period of time to provide further accurate predictions.For example, in this research project, the ANN was trained with a significant amount of datasets, as eventually, itwill be able to predict and provide the desired output flow rate and temperature accuratelywhen a user inputs his desired temperature and flow rate whereas in most situations the desired valuesare outside the dataset provided to the model that has been trained. This key ability to accurately evaluate or compute functions of concealed inputs by training with a specific dataset which consistsof a finite collection of input and output pairs is known as *model generalization.* The prime effectiveness of all deep learning models is acquired by their potential to generalize their learning process from visible training data to concealed examples or situations.

The experiments and computations for a particular fine-tuned neural network will not be suitable for another neural network. This is because it may consist of different input and output parameters, and a distinct complexity, and the objective of the results required may vary. Hence, for a given neural network, many parameters exist that are in need of optimization to acquire the best solution. A few of the common parameters to be tweaked in the neural network are, the number of hiddenlayers, number of neurons, batch size, and epoch number (Antonio Guili et.al, 2019). These listed parameters are limited as the number of parameters to be optimized will vary significantly, depending on the complexity of the neural networks. Furthermore, these parameters that are to be optimized in the neural network are defined as hyperparameters. This term is provided in order to distinguish between the system parameters of the network such as weights and biases. During the process of training, Hyperparameter tuning is the approach of discovering the optimal set of hyperparameters used to obtain improved cost functions (reduced cost functions). There are several hyperparameter tuning methods offered by Python in the form of libraries that may beutilized to fine-tune the hyperparameters of an ANN model such as Hyperband, Keras Tuner and Scikit optimize. These libraries aid in tuning the hyperparameters to obtain a set of best models thatcan be utilized to train the ANN model with the relevant dataset.

In any Hyperparameter tuner, the initial hyperparameter that will be considered is the number of hidden layers, and by constructing more layers the depth of the neural network may increase and eventually form a deep neural network that provides the ability to further examine complex features(Agrawal, 2021). Therefore, for simple problems, one or two hidden layer networks will work relatively well whereas for more complicated problems the number of hidden layers can be increased until the point of overfitting the training dataset is reached. Following this hyperparameter, the number of neurons in each existing hidden layer must be considered. The number of neurons is typically determined by the sort of input and output task required. In general, a fundamental size isto establish a pyramid-like pattern of neuron numbers, that is fewer neurons in each adjacent layer (Geron, 2019). However, this concept was discarded as utilizing the same number of neurons in each hidden layer performs fairly well or even better in the majority of the cases. Nevertheless, all of this depends on the dataset as it can provide an idea to construct the neurons in the hidden layers.Similar to the number of layers, the number of neurons may also be increased gradually to the point where the network begins to overfit, but this approach may be tedious and it is more efficient and straightforward to survey a model with more hidden layers and neurons first and thereafter to utilize options such as early stopping and regularization strategies to avoid overfitting. A scientist named Vincent Vanhoucke called this approach the “stretch pants approach” (Vanhoucke et al., 2011). This indicates that, instead of spending time to examine pants that precisely match the size, utilizing large stretch pants which will narrow down to the required size is more efficient. This approach avoids bottlenecking layers which can possibly ruin the model architecture and performance.

Likewise, the hyperparameter called learning rate is a salient hyperparameter that controls how one regulates variations of the model with respect to the computed error each time the weights of the model are updated (Patterson, Josh, and Gibson, 2017). The basic concept of identifying a reasonable learning rate is to train the network for several hundred iterations, initially with a low learning rate and slowly increasing it to a larger value. Subsequently, the best learning rate discoveredthrough this process can be utilized to train the network again whereas, the Optimizer dictates howthe network may be updated such as weights depending on the loss function (Chollet François, 2019). Several optimizers are available to offer high-performance to train models of deep learning. For example, stochastic gradient descent, AdaGrad, Adam, and AdaDelta (Soydaner, 2020).

Batch size may also have a substantial impact on the performance and time taken for the training of the model. A paper by Dominic Masters and Carlo Luschi in 2018, deduced that utilizingsmall batches typically between 2 to 32 was recommended since small batches supported providingbetter models in a short training period (Masters & Luschi, 2018). Nevertheless, a paper written in 2017 by Elad Hoffer et al. and Priya Goyal et al. illustrated a high probability of utilizing large batch sizes typically up to 8192, employing a variety of techniques such as increasing the learning rate, by warming it up from low to high learning rate value which is guided to a short period of training and without any generalization gaps. (Hoffer et al., 2017). Hence, two approaches are available either to utilize a large batch by warming up the learning rate and afterward if the training seems to be unstable or the performance of the model is unacceptable then it is recommended to try utilizing a smaller batch size instead.

Lastly, the objective of the activation function is to introduce nonlinearity to each neuron. Many activations functions are available for selection such as ReLU, Sigmoid, and variants of ReLU (suchas Leaky ReLU). Moreover, it is important to choose the ideal activation function for the neural network model since they are used on all neurons available in the neural network of the given model and backpropagation utilizes their derivates. Consequently, the function and its derivative must comprise less computational complexity (Agrawal, 2021).

2 EXPERIMENTAL PROCEDURE

2.1 Development stage

The primary phase of the study was to perform an in-depth and detailed study of research findings related to the topic. The discovered research was then examined carefully to obtain a fundamental understanding and visual of the study. The research sources comprised books, articles, or papers that provide principal ideas, designs, and novel strategies and how engineering techniques are employed to overcome the complications that emerge in the existing research findings. The priority of the findings was mainly on the shower control systems, Neural networks, and microcontrollers.

The research findings were then utilized to develop ideas on constructing the hot water mixture system design by employing existing components and technologies, particularly from previous related works. The design was developed employing AUTOCAD software and Figure 1 illustrates the design of the automated hot water mixture system.

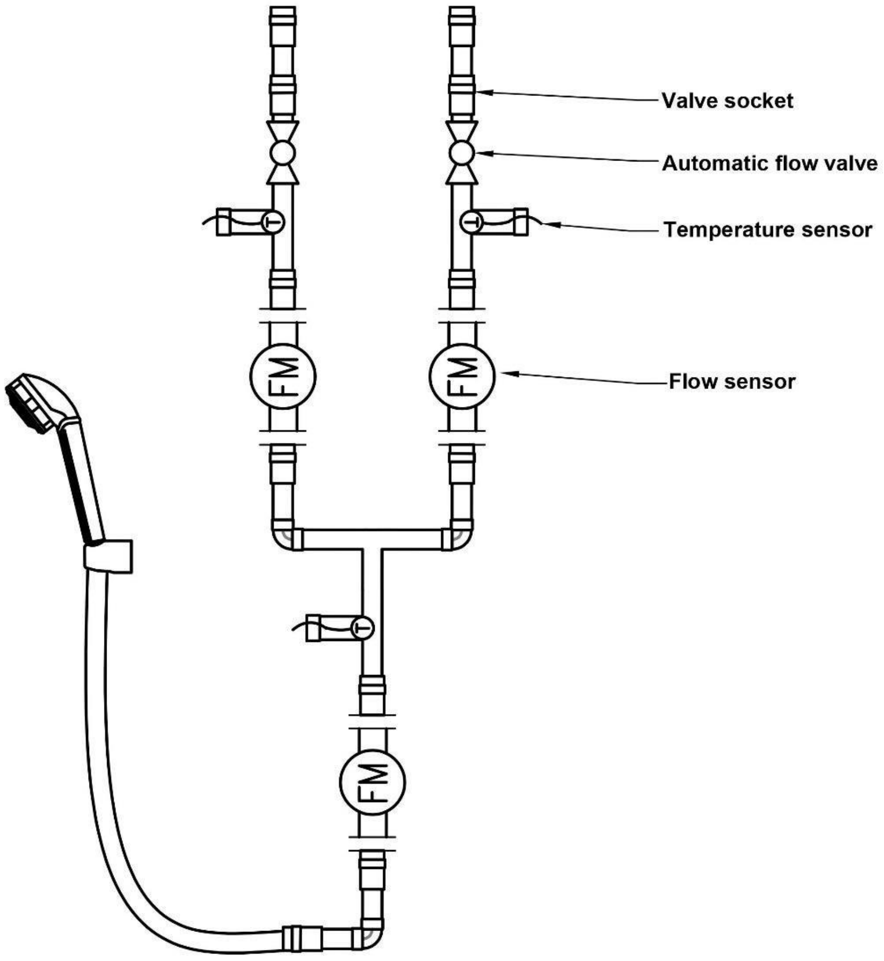


Figure 1 Automated hot water mixture design

After the design was developed, the priority was given to constructing the Neural Network architecture by investigating the parameters of the hot water mixture system, and by identifying the input and output parameters of the system the architecture of the Neural Network was fabricated. The parameters for the ANN input layer were considered as output temperature and output flow rate. The parameters of the output layer were considered as hot and cold water flow rates and temperatures. Figure 2 illustrates the architecture of the constructed ANN model here the number of neurons in the first hidden layer was declared as $n^{(1)}$ and the term l denotes the number of hidden layers in the ANN model. The values of the number of neurons in hidden layers and the number of hidden layers were given in the arbitrary form. This was due to the hyperparameter optimization technique utilized in the script developmentstage to identify these hyperparameter values.

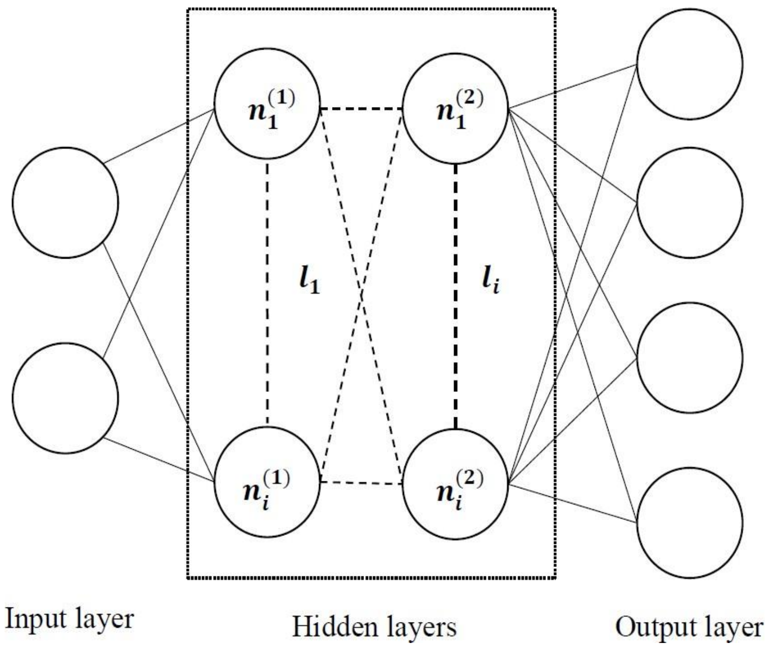


Figure 2 Architecture of Artificial Neural Network model

2.2 ANN and physical model implementation stage

The list of components selected is illustrated in Table 1. These components are essential to fabricate the modelrequired for data collection and the final proposed system of the study.

Table 1 List of selected components of the final model of study

Component name	Model Number
Microcontroller	Raspberry Pi 4 Model B (2gb ram)
Power Supply	Switch mode power supply (SMPS)
Servo motor	DS3230 Metal gear servo
Temperature Sensor	DHT22/AM2302
Flow sensor	YF-S201 hall effect sensor
Ball valve	SLON two pieces ball valve

The system is based on an Artificial Neural Network model, and in order to train the model, a significantamount of dataset must be acquired to obtain accurate predictions. Initially, an automatic and custom flow valve must be designed and tested prior to the collection of datasets. The custom flow comprises two key components, which are the servo motor and the ball valve. Figure 3 illustrates the fabrication of the automated flow valve which includes the other essential components such as brackets and metal plates for mounting purposes to ensure the proper functioning of the automated flow valve.

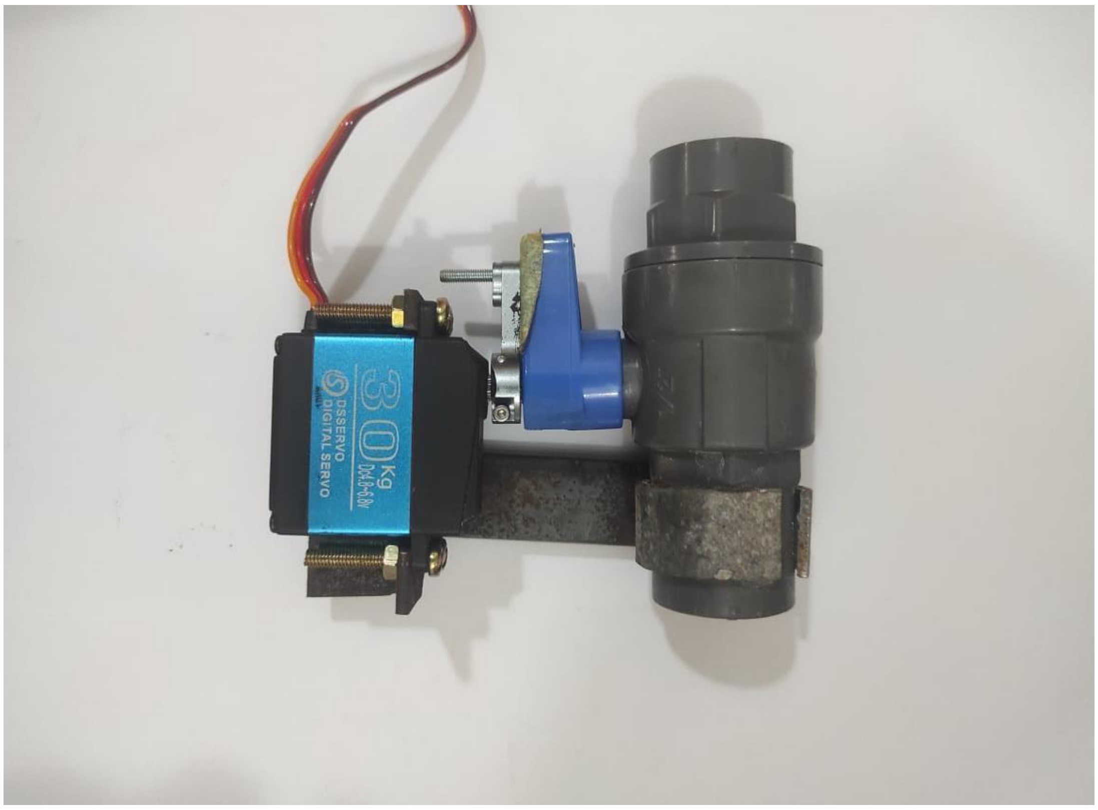


Figure 3 Automated flow valve

Due to the novel approach being performed by employing an Artificial Intelligence embedded systemintegrated into a hot water mixture system, the relevant dataset was not readily available. Hence, to collect relevant and accurate data, the hot water mixture system was fabricated, and data was acquired by executing the developed data collection script to collect the data at the main locations of a pipeline whichare hot, cold, and output lines.

The data collection script typically comprises the setup of the temperature sensors, flow rate sensors, and servo motors. After the setup, the primary loop of the script was developed to acquire readings at 3 main locations of the system which are hot water temperature and flow rate (hot pipeline), cold water temperature and flow rate (cold pipeline), output water temperature and flow rate (output pipeline). When the script was executed, the instruction was given to both servo motors at hot and cold pipelines to rotate between the specified range of 30 to 90 degrees. Then at each iteration, the servo motor regulates the volume of water flowing in both hot and cold pipelines, and the temperature and flow rates at input pipelines were measured after 3 seconds of waiting time after the servo motors rotate. The output temperature and flow rates were measured after 4 seconds of waiting which includes the delay of the readings measured by the hot and cold flow sensors since the sensors read the pulse within the 1-second timeframe when the counter is activated and deactivated. Hence the system was functioning to collect data until the script was terminated. Furthermore, the readings were written to a file after the completion of each iteration of the primary loop. In addition, the file consists of a comma-separated value (CSV) format.

The overall fabricated model of the study is illustrated in Figure 4 which indicates the arrangement of the key devices. Where the custom automated flow valves are initially installed, followed by the temperature and flow sensors in each input pipeline. The flow sensors comprise threaded ends to be easily installed in the hot water mixture system. However, the installation of the temperature sensors was relatively complex comprising fixing a T joint connector and fitting an end cap which was drilled at the center to enable the wires to reach out of the system and be connected to the Raspberry Pi. Furthermore,the temperature sensor was installed perpendicular to the flow inside the T joint connector.

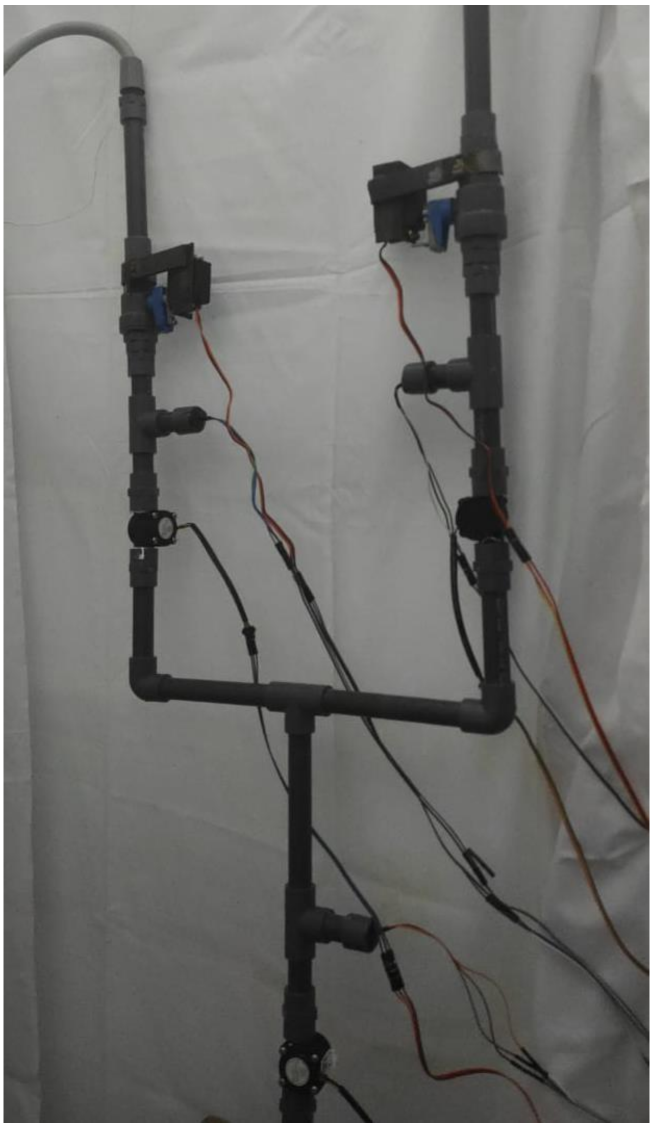


Figure 4 Fabricated physical model of study

The Neural network script was developed gradually, and the collected dataset was utilized to train the model. Before training the model, a crucial step was involved in obtaining the best hyperparameters for the model (O'Malley et al., 2019). This was employed by utilizing hyperparameter optimization techniques offered by Keras API. The tuner known as Hyperband was employed in this study along with the Random Search tuner to perform hyperparameter optimization. The best tuner was selected by monitoring the training loss (must be low) of the model since the hyperparameters found from both tuner algorithms are distinct and as a result, provide models with different performances. Hence, the best model was acquired from the tuner which provided the least training loss and then the best model with the ideal set of hyperparameters was trained. The set of hyperparameters this study prioritized to obtain was the number of hidden layers, the number of neurons in those hidden layers, the optimizer, and the dropout. Moreover, several other combinations of hyperparameters were experimented with prior to the final selection of the hyperparameters such as activation functions and learning rate. After the training procedure, the best model was evaluated with the test data, and predictions were made to examine the performance of the model. Furthermore, visualization strategies were utilized to visualize the performance of the model in a graph, particularly the graphs of training and validation losses. The library utilized to visualize was matplotlib.

2.3 Physical model testing stage

A relationship between valve angles and flow rates was established using the data from the acquired data collection process. The respective data of hot and cold angles and flow rates were read utilizing the pandas library. The variables were called to establish two linear equations for the regression graphs utilizing the polyfit function. The linear equation comprises the angles of hot and cold being the independent variables and hot and cold flow rates as the dependent variables. From the equation, the gradients and intercepts were obtained. Hence, when the user inputs the desired temperature and flow rate, the model predicts the hot and cold water flow rates and temperatures considering the new user inputs as new instances. The predicted flow rates are utilized to find the angle required to turn in the hot and cold pipelines. Then the data of hot and cold angles are utilized by the Raspberry Pi to send the control signal to the servo motor in the form of pulses to rotate to the angle computed, which will result in deliveringthe desired output flow rate and temperature. The accuracy of delivering the desired parameters depends on the performance of the Neural Network model.

3 RESULTS AND DISCUSSION

3.1 Data collection

Tests conducted with the customized automatic flow valve to check for proper rotation of the ball valve indicated positive results where both automated flow valves rotated to the specified angle accurately. The data collected from the physical model was successful and a sample of the data collected isillustrated in Table 2.

Table 2 Results of data collected of study

Hot Angle $(degrees)$	**Cold angle** $(degrees)$	**Hot flow** (l/min)	**Cold Flow** (l/min)	**Output Flow** (l/min)	**Hot temp** **℃**	**Cold Temp** **℃**	**Output Temp** **℃**
53	33	3.6	3.6	7.267	41	27.2	34.1
71	69	3.333	3.467	6.867	41.4	27.2	34.2
39	69	3.733	3.467	7.133	41.6	27.2	34
66	37	3.467	3.6	7	42.4	27.2	35.1
76	71	2.133	3.333	5.533	41.8	27.2	34.4
69	81	2.533	1.73	4.2	42.2	27.2	33.1
40	75	3.733	1.6	5.4	42.2	27.2	37.6
68	44	3.467	3.733	7	47.5	27.2	41.8
88	36	0	4.133	4.067	47.5	27.2	34.5

The sample of collected data in Table 2 indicates the results were within acceptable bounds for flow rate readings with respect to the ball valve angle. However, there were some data points with inaccurate measurements. This can be possibly improved by increasing the waiting time between the devices to measure the reading during each iteration of the loop, which means after the ball valves rotate, a settlingtime of water must be taken into consideration. The waiting time considered in this study for each iteration was roughly 8 seconds and a total of 3000 datasets were collected here for 1 hour only 450 datasets can be collected. Furthermore, during data collection, other factors must also be considered such as giving rest for the system since the devices can possibly fail or overwork, and in this study, a heater unit was utilized as a hot water source as the heating unit also requires substantial time to rest.

3.2 Artificial Neural Network training

The dataset was utilized to successfully train the ANN model and prior to the training, the Hyperband tuner of KerasTuner library was employed to find the best hyperparameters ideal for this study’s ANN model. In addition, the Hyperband tuner algorithm was selected over RandomSearch algorithm due to better training and validation loss provided by Hyperband tuner algorithm. Unfortunately, due to KerasTuner library being released in 2019, false information was provided with respect to the number of neurons in the hidden layers. However, other hyperparameters tuned were provided correctly as illustrated in Table 3.

Table 3 Summary of selected hyperparameters

Hyperparameter	**Selected choice or value**
Number of hidden layers	8
Optimizer	adam
Loss	Mean Absolute Error (MAE)
Activation function	ReLU
Output layer Activation function	Linear
Dropout in first hidden layer	0.4
Dropout in other hidden layers	0.2

The choices in the tuner search space for optimizer were given as adam, adagrad and rmsprop and the dropout choice was given between 0.1 to 0.5 in the tuner search space. As illustrated in Table 3 the Hyperband tuner algorithm selected adam and other hyperparameters to be the most appropriate choice or value for this ANN model. Furthermore, the loss and activation function were not tuned since this study comprises a regression model and the type of activation function and loss function were considered asprovided in Table 3. After acquiring the best model from the tuner the best model was trained for 120 epochs to monitor the training and validation loss as illustrated in Figure 5.

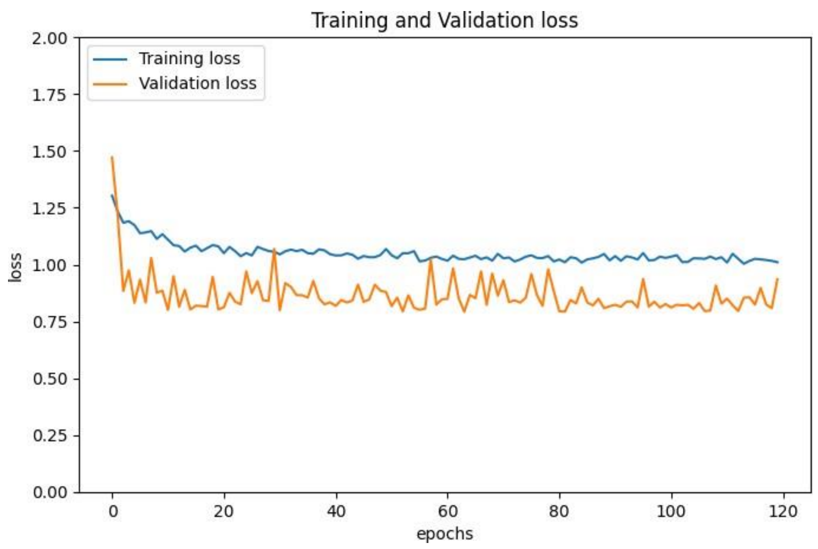
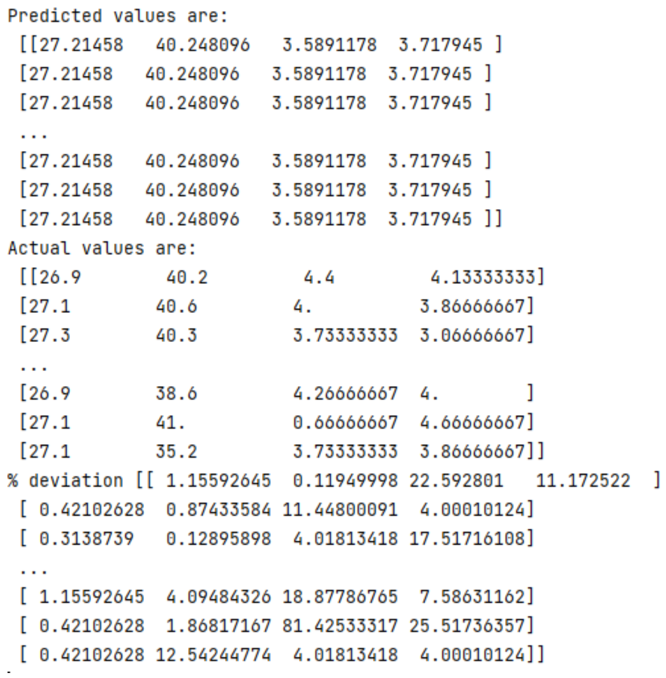


Figure 5 Graph of training and validation loss

The training loss in the last epoch which is 120th epoch was 1.0104 and the validation loss was 0.9354. The graph illustrated in Figure 5 indicates that the training loss (top) and validation loss (bottom) are alleviating and both losses were gradually converging with each other from the starting epoch during training. A factor must also be considered that the tuning algorithm implemented provides us with the best model to train with. Hence, when the best model obtained was trained the degree of convergence wouldbe less visible. A possible reason the validation loss is less than the training loss is due to the training loss being consistently reported over the period of a complete epoch whereas the validation loss is evaluated over the validation set only once the current training epoch is completed. This suggests on average, training losses are measured half an epoch earlier. Furthermore, the training loss fluctuates less roughly after 90 epochs. Therefore, the trained model obtained can be considered good when comparing the average difference of both losses was fairly minor. The model was evaluated on the test data to examine the performance and the test loss from the evaluation was 0.938. This indicates that the training loss is relatively higher than the test loss which conveys that there is room for improvement in the ANN model. The most probable reason for the results of the training, validation, and test loss was due to the accuracy or noise comprised in the dataset collected. In addition, the overall dataset was split into20% tests and the test data were split into half to utilize for validation data.

Several strategies were employed to calculate the accuracy since the model is of regression type, the accuracy cannot be computed automatically from the compile method from the Sequential model since it is only available for classification problems. Therefore, the percentage deviation was computed after computing the predicted values of the ANN model with the test data and Figure 6 illustrates the comparison of actual and predicted results along with the deviation of the predicted result with respect to the actual result where the deviations were within 5% range for many of the results illustrating good predictions by the ANN model.

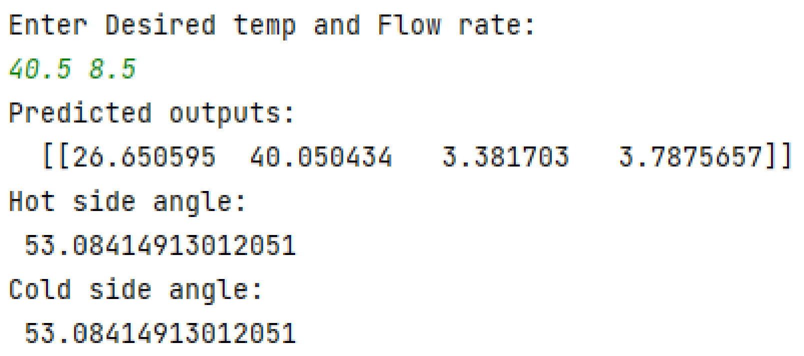


```
Predicted values are:
 [[27.21458   40.248096   3.5891178  3.717945 ]
 [27.21458   40.248096   3.5891178  3.717945 ]
 [27.21458   40.248096   3.5891178  3.717945 ]
 ...
 [27.21458   40.248096   3.5891178  3.717945 ]
 [27.21458   40.248096   3.5891178  3.717945 ]
 [27.21458   40.248096   3.5891178  3.717945 ]]
Actual values are:
 [[26.9        40.2         4.4         4.13333333]
 [27.1        40.6         4.          3.86666667]
 [27.3        40.3         3.73333333  3.06666667]
 ...
 [26.9        38.6         4.26666667  4.        ]
 [27.1        41.          0.66666667  4.66666667]
 [27.1        35.2         3.73333333  3.86666667]]
% deviation [[ 1.15592645  0.11949998 22.592801   11.172522  ]
 [ 0.42102628  0.87433584 11.44800091  4.00010124]
 [ 0.3138739   0.12895898  4.01813418 17.51716108]
 ...
 [ 1.15592645  4.09484326 18.87786765  7.58631162]
 [ 0.42102628  1.86817167 81.42533317 25.51736357]
 [ 0.42102628 12.54244774  4.01813418  4.00010124]]
```

Figure 6 Results of predictions and percentage deviation

3.3 Testing the automated hot water mixture system

The results of the tested system are shown in Figure 7, where the user can input the temperature andflowrate of required and the angles are computed to deliver the user desired parameters by sending the control instruction to the servo motor via the Raspberry Pi.



```
Enter Desired temp and Flow rate:
40.5 8.5
Predicted outputs:
  [[26.650595  40.050434   3.381703   3.7875657]]
Hot side angle:
 53.08414913012051
Cold side angle:
 53.08414913012051
```

Figure 7 Testing the system of automated hot water mixture

The tests performed on the final system indicated that the ANN model performs well for a certain range of temperatures and flow rates. Table 4 illustrates the time taken for the key procedures of the study where data collection is the time taken to collect the data required for Neural Network training of this study, The training time is the average time utilized to train the best model acquired from the hyperparameter optimization process followed by the training process. Furthermore, the convergence time is the time required for the system to converge the existing output temperature to the required user temperature and lastly, the response time is the duration the user has to wait for the final automated system to deliver the flow rate and temperature. In summary, the Neural Network approach offers a faster response time which comprises a significant percentage difference of 161% (13 seconds and 120 seconds) in hot water mixing than the existing conventional methods.

Table 4 Comparison between the research study approach and conventional methods

	Neural Network	**Conventional methods**
Data collection (hours)	≥7 hours	-
Training time (seconds)	25 minutes	-
Convergence time	<10 seconds	-
Response time	<13 seconds	60-120 seconds

4 CONCLUSION

The main goal of this study was to develop a system of Neural Network based automated hot water mixture to reduce the time taken for hot water mixing by the current conventional systems of hot water showers.

The procedure required to develop a path to employ an Artificial Intelligent embedded system consisting of automated and controllable flow valves along with sensors to compute the water temperature and flow rate. The Neural network model performance was significantly dependent on the data collection process and development of the ANN model by utilizing strategies for instance Hyperband tuning algorithm as a hyperparameter optimization technique.

The physical model was successfully fabricated which comprises the salient implementations for instance the fabrication of the custom automated flow valve, the setup, and installation of DHT22 series temperature and YF-S201 flow sensors along with the overall pipework fabrication which includes the valve sockets, T joint, and elbow connections. Moreover, the key findings identified from the study were: the inversely proportional relationship developed between the angle and flow rate of hot and cold water in experimental results, and the relationship between the predicted and actual results acquired from the trained ANN model.

In conclusion, acceptable results were attained due to the performance of the Neural Network modeland providing predictions to compute the angles essential to rotate the controllable flow valves in instances where the desired shower temperature and flow rate are input by the user. This process was viable due to the successful collection of datasets from the physical model. Although, certain complications arose during the process of data collection, it can be deduced that a substantial amount of the data collected was within the accepted bounds. Moreover, some potential benefits were identified such as achieving stable temperature with fewer fluctuations and providing good control for users to enter the desired temperature and flow rate, and faster response rates achieved in contrast to conventional methods. In addition, this study employed a system that ensures user-friendliness and comprises an advanced intelligent system to offer versatility to the system of automated hot water mixture.

REFERENCES

Agrawal, T. (2021). Hyperparameter Optimization in Machine Learning. In *Hyperparameter Optimization in Machine Learning*. https://doi.org/10.1007/978-1-4842-6579-6

Antonio Guili; Amita Kapoor; Sujit Pal. (2019). *Deep Learning with TensorFlow 2 and Keras: Regression, ConvNets, GANs, RNNs, NLP, and More with TensorFlow 2 and the Keras API*. O'Reilly Media, Inc.

Charu C. Aggarwal. (2018). *Neural Networks and Deep Learning. A Textbook*. Springer. https://doi.org/10.7551/mitpress/13811.003.0007

Chollet François. (2019). Chollet - 2018 - Deep learning with Python. In *Manning* (Vol. 53, Issue 9). http://faculty.neu.edu.cn/yury/AAI/Textbook/Deep Learning with Python.pdf

Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (2nd Editio). O'Reilly Media, Inc.

Hoffer, E., Hubara, I., & Soudry, D. (2017). Train longer, generalize better: closing the generalization gap in large batch training of neural networks. *Advances in Neural Information Processing Systems*, *2017-December*, 1732–1742. https://arxiv.org/abs/1705.08741v2

Masters, D., & Luschi, C. (2018). *Revisiting Small Batch Training for Deep Neural Networks*. https://arxiv.org/abs/1804.07612v1

O'Malley, Tom and Bursztein, Elie and Long, James and Chollet, Fran\c{c}ois and Jin, Haifeng and Invernizzi, L. and others. (2019). *KerasTuner*. https://keras.io/keras_tuner/

Patterson, Josh and Gibson, A. (2017). *Deep Learning: A Practitioner's Approach*. O'Reilly.

Shower Systems Explained. (n.d.). Retrieved November 13, 2021, from https://luxehomebydouglah.com/blog/shower-systems-explained/

Soydaner, D. (2020). A Comparison of Optimization Algorithms for Deep Learning. *International Journal of Pattern Recognition and Artificial Intelligence, 34*(13). https://doi.org/10.1142/S0218001420520138

Vanhoucke, V., Senior, A., & Mao, M. (2011). Improving the speed of neural networks on CPUs. *Proc. Deep Learning and ...*, 1–8. http://research.google.com/pubs/archive/37631.pdf