An LSTM control Network

ABSTRACT

Super conducting linear Accelerators(linac) use Proportional Integral derivative(PID) as one of the most common control methods used to maintain the Cryogen level inside cryogenic vessels constant. Operating the linac cryostats in manual mode has proven to be challenging because of the significant unpredictable fluctuations in the cryogen level caused by continuous changes in RF load and input flow. With a deep neural network model based on LSTM (Long-Short Time Memory), this work offered a novel model-based solution for the autonomous control of level control systems. A recurrent neural network(RNN) is developed and the model is trained on a dataset derived from the outputs of a Python-programmed TCLab PID framework.An in-house built embedded http server-device is deployed near each sensor-actuator site, which is remote from the control room PC that runs RNN standalone. LSTM commands this device via web packets for read-write operations between any one level-sensor and a proportional-valve. The technique employed produced a hand-free operation with an error value of +/- 1% of level with respect to a requested set point for days together. The results that are displayed demonstrate how well the trained model predicts the level control system's outputs in relation to the target levels. This has been tried for automation and control of a successfully installed cryogenic control network of a central control room at Inter University Accelerator Centre in New Delhi, India.

Keywords— deep learning, level control, LSTM, neural network, , PID, RNN, superconducting linac.

I. INTRODUCTION

Artificial intelligence and deep learning applications are becoming increasingly common in a variety of scientific domains today. Long Short Term Memory (LSTM) networks are a part of Recurrent Neural Network that is designed to recall and forecast based on long-term dependencies learned from time series data. As illustrated in Fig. 1, the Inter University Accelerator Centre employs superconducting cavities to accelerate high energy ion beams using a linear accelerator[1]. These cavities are kept cool by liquid helium at temperatures below the superconducting temperature of ~4.2 K. Each of the five linac cryostats of the linachas a separate liquid helium dewar, and keeping the level stable inside is critical. This has been a particularly difficult task because to the time-varying RF loads into helium that are unpredictable during linac tests. This is accomplished through the use of rapid closed-loop real-time sensor-actuator control systems that employ the Proportional Integral Derivative approach. [2].

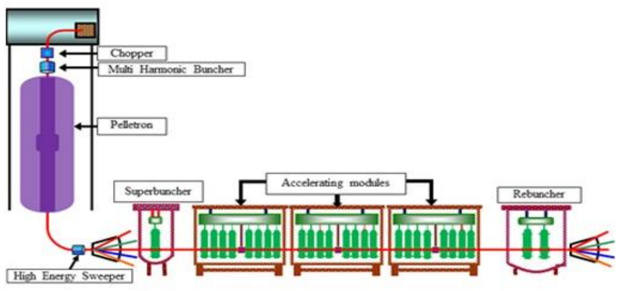

Fig.1.Super conducting linacand five cryostats at IUAC

The most common control algorithms used in closed loop control systems are implemented primarily with standalone PID controllers[RSI]. The difference between a set point (SP) and a measured process variable (PV) is used to automatically modify a control output with Proportional Integral Derivative (PID) control. As the system input, the value of the controller output u(t) is sent.

$$u(t) = u_{bias} + K_c e(t) + \frac{K_c}{\tau_I}\int_0^t e(t)dt - K_c \tau_D \frac{d(PV)}{dt} \quad (1)$$

$$\text{where } e(t) = SP - PV$$

In many situations, PID-based control systems have been seen to successfully give the intended outputs.[3-7].However, because operators prefer completely hands-free operations over fine-tuning P.I,D settings depending on operational conditions, machine learning-based control methods have been developed in a variety of systems. Fuzzy logic is one of the most used machine learning techniques. In one study, a fuzzy logic-based control system was combined with PID control[8-12]. In another work, a single-factor fuzzy logic controller for temperature and humidity regulation was developed[13]. In addition to these investigations, deep learning models are commonly used in classification-based studies and have yielded positive results[14-16].Because of its capacity to retain information for a long length of time, handle time series data, and perform prediction and classification tasks, the Long Short-Term Memory (LSTM) algorithm is a good deep

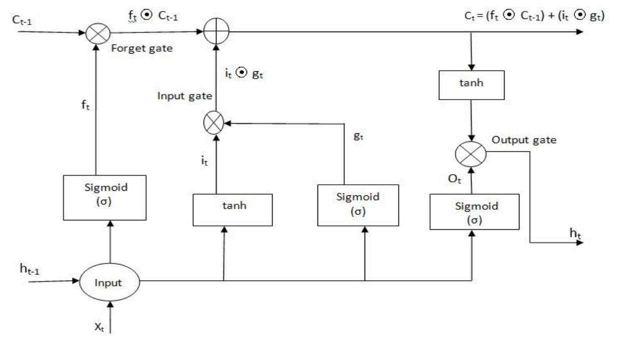

Fig.2.LSTM cell structure

learning technique for time-dependent control applications[17]. Prediction-based studies based on the LSTM algorithm can be found in the literature[18].In one of these investigations, LSTM was used to anticipate the controller for room temperature in a residence based on user patterns[19]. The LSTM algorithm also produced successful results in a study targeted at optimizing indoor temperature for energy-saving goals in building energy[20].

LSTMs are recurrent neural networks (RNNs) that can learn long-term dependencies. They feature a cell structure that is similar to a chain, with four layers within each cell. Data is pre-processed, labelled, and produced in Python using the Keras package [21], which was utilised to build the sequential LSTM model.The LSTM is an upgraded cell of recurrent neural networks (RNN) [22] that was created to address the gradient vanishing issue that the vanilla RNN presented. RNN has a limited memory in its structure to keep the current iteration output. It then uses this output as an input in the next iteration. This makes RNN useful for processing time series data since it remembers previous iterations and recognises dependencies between them.In comparison to the RNN's regular short-term memory, LSTM includes long-term storing capabilities and gates that allow the algorithm to reflect on the long-term dependencies from previous iterations' input and also accelerate the learning process. As a result, LSTM is better suited to deal with sequential problems, particularly those involving time series data. The LSTM memory's gates determine whether information is retained or deleted. Figure 2 depicts the LSTM cell layout, which includes four collaborating gates: input (I), forget (F), memory cell (M), and output (O) gates, which replace the hidden neurons of the regular RNN.The forget gate computes Equation (3) based on previously hidden state information $h_{t-1}$and current iteration $X_t$at time t, to decide whether information in the memory state to keep or discard. The forget gate 'ft' determines which information from the previous cellular state Ct-1 should be removed and replaced with information from the current cellular state Ct.

$$f_t = \text{sigmoid}(W_f \cdot X_t + W_h \cdot h_{t-1} + b_f) \tag{3}$$

The sigmoid layer output varies between 0 and 1 and if the output is closer or equal to zero, all information is thrown out. After the forget layer decides, the input gate decides on the new candidate data for the next layer. It performs Equation (4) using the sigmoid function to quantify the new information from the new iteration to update the memory state (C_t). The input gate'I' decides how much new information from the current input$X_t$is to be added to the current cellular state $C_t$based on the previous hidden state h_{t-1}.

$$g_t = \text{sigmoid}(W_i \cdot X_t + W_h \cdot h_{t-1} + b_i) \tag{4}$$

$$i_t = \tanh(W_c \cdot X_t + W_h \cdot h_{t-1} + b_c) \tag{5}$$

The (tanh) layer in the memory gate, on the other hand, generates a vector of fresh memory values (i_t)using Equation (5). In Equation (6), the outputs of the two gates are combined and point-wise multiplied to produce an update to the cell memory state (Ct).

$$C_t = (f_t \odot C_{t-1}) + (i_t \odot g_t) \tag{6}$$

The output gate applies the (tanh) function to the cell memory and then solves Equation (7) to determine what information to output. The secret information for the following iteration is then determined using Equation (8). The output gate O decides how the current cellular state C_t influences the current output hidden state h_t by

$$O_t = \text{sigmoid}(W_o \cdot X_t + W_o \cdot h_{t-1} + b_o) \tag{7}$$

$$\begin{aligned} h_t &= O_t \cdot \tanh(C_t) \\ &= \text{sigmoid}(W_o \cdot X_t + W_o \cdot h_{t-1} + b_o) \cdot \tanh(C_t) \end{aligned} \tag{8}$$

The ability to achieve the pre-set target levels with the desired precision and within the required time is the most significant condition for cryogenic level regulating Accelerators. As a result, this study proposed a unique approach for controlling a proportional valve-based system with a user-input set-point utilizing an LSTM-based deep neural network model. The generated model was trained with data and learned with the help of a TCLab PID controller library. The traditional PID controller technique is replaced in this approach by an RNN-based LSTM technique that uses the history of level data to forecast the required liquid to maintain a constant level from a time series sensor-data. This paper describes the hardware and software design details, as well as numerous test results. Sections II describe the training technique, LSTM algorithm hyper parameters, and pertinent inputs. Section III presents the acquired results.

2. EXPERIMENTAL DESIGN

The goal of this study was to achieve level control using an LSTM model trained with a TCLab framework programmed in Python. The work flow included identifying the goal level set-points, constructing the dataset, extracting features, designing and training the LSTM model, and assessing the model's performance. LSTM models based on recurrent neural networks, for example, have been shown to be suitable for learning data with temporal properties or sequences [23-24]. An LSTM network is made up of several hidden layers and an output layer. Its distinguishing feature is the memory cells in the hidden layers, which learn facts by preserving or modifying the state.

This system is linked to an Ethernet-based microcontroller hardware that runs an embedded HTTP server and also functions as a current driver circuit for proportional-valve that can be controlled via IoT. This hardware eliminated the need for commercial TCLabrecommended custom boards for LSTM, which are difficult to obtain in India. Furthermore, the indigenous device is a less expensive design, and such a large number of devices can be distributed anywhere. The system was evaluated under crucial linear accelerator settings, with online trials using a highly variable RF power. This AI approach, when applied on a PC, may maintain a consistent level. The installed LSTM on a control-terminal is operated by Cryogenic control room operators, and a visual interface using Python is given for hand-free operation. The benefit of an LSTM-based control is that it may run numerous such channels from a single computer or GPU, and it can dynamically interconnect sensors and actuators during experiments. As a result, the goal of this development is to automate an Accelerator's cryogenic level control process using an LSTM network for automation and control of a functional Cryogenic control network of a linear Accelerator set-up.

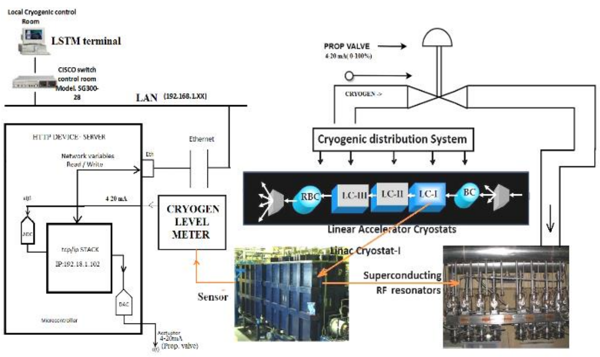
Local Cryogenic control Room
LSTM terminal
CISCO switch control room Model. SG300-28
LAN (192.168.1.XX)
PROP VALVE
4-20 mA(0-100%)
CRYOGEN ->
HTTP DEVICE- SERVER
Network variables Read / Write
Eth
Ethernet
Cryogenic distribution System
4 20 mA
ADC
tcp/ip STACK
IP:192.18.1.102
CRYOGEN LEVEL METER
RBC
LC-III
LC-II
LC-I
BC
Linear Accelerator Cryostats
Linac Cryostat-I
Superconducting RF resonators
Sensor
DAC
Microcontroller
Actuator 4-20mA (Prop. valve)

Fig.3. Block diagram of a single channel LSTM-Accelerator Control network

The block diagram of a single channel LSTM- Accelerator Control network is shown is Figure 3 and the steps followed to design and implement such a system is given below in figure 4.

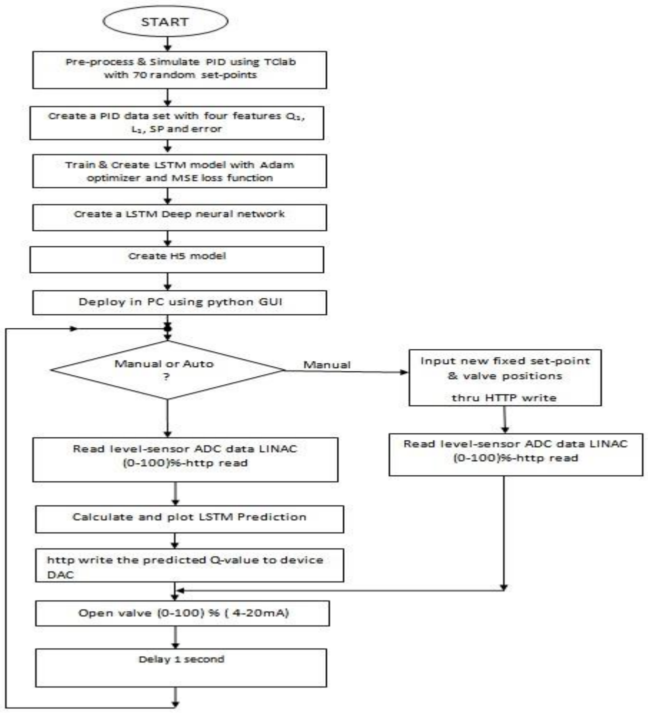
START
Pre-process & Simulate PID using TClab with 70 random set-points
Create a PID data set with four features Q₁, L₁, SP and error
Train & Create LSTM model with Adam optimizer and MSE loss function
Create a LSTM Deep neural network
Create H5 model
Deploy in PC using python GUI
Manual or Auto ?
Manual
Input new fixed set-point & valve positions thru HTTP write
Read level-sensor ADC data LINAC (0-100)%-http read
Read level-sensor ADC data LINAC (0-100)%-http read
Calculate and plot LSTM Prediction
http write the predicted Q-value to device DAC
Open valve (0-100) % (4-20mA)
Delay 1 second

Fig.4. Flow diagram of LSTM implementation

2.1 DATA SET

Beforethemodeltrainingphase,thedatasettobegeneratedfortrainingismadereadywithdatapreprocessing.As a result, the training dataset for the LSTM network was created by leveraging input-output data from the proposed PID controller model using the TCLab PID framework. The level control system controlled by the PID controller must successfully reach the desired level values in order for the training dataset to not introduce training errors and for the LSTM network to perform with high accuracy. To do this, a set of target level set-points were generated using the TCLAB library and random values. Figure 5 depicts the developed PID set-points for training and testing, with 80% utilised for training and 20% used for testing.

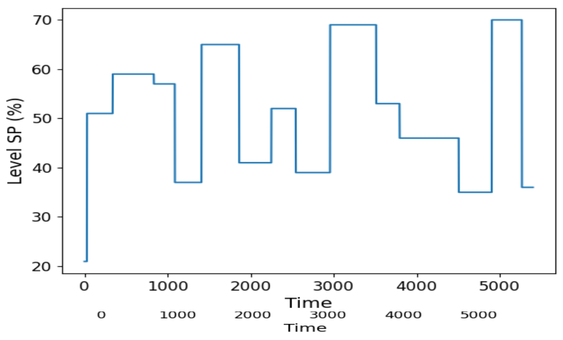

Fig.5.Targetlevel-Timegraph.

The target range is determined at random between 30% and 70%. The dataset was constructed with 14 random set-points over a 90-minute (5400s) period. To speed up the preprocessing process, set the SPEEDUP option to 100, which means that the data will be created and trained 100 times faster than the default SPEEDUP of 1. Each set-point will be retained for 240-600 seconds at random. The results of the tests revealed that the output of the Python programutilizingTCLab correctly captured the intended values. Figure 6 depicts the output performance of the PID controller. Q1, L1, and SP are the three parameters listed in the graph. In the developed system, Q1 represents the output valve's expected percentage representation. L1 is the current linac level value (%) of the PID-controlled system. The SP option specifies the desired level (%).

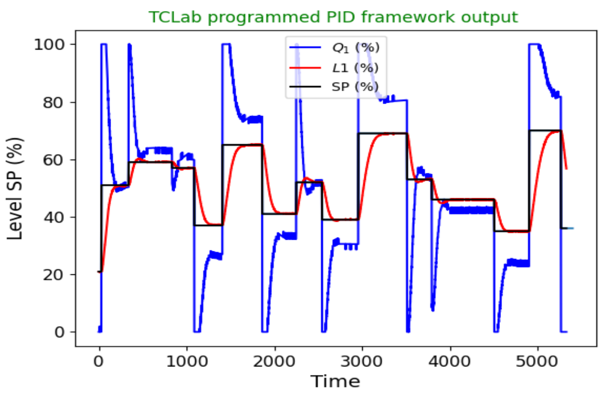

Fig.6.PIDcontrollertestresultgraph.

This dataset is saved in CSV format, and it is used for training and testing the LSTM model. A LSTM neural network developed in Python with the Tensorflow and Keras frameworks is used to represent the given PID system and the generated dataset. The following section goes into greater detail about the model's hyper-parameters and standard parameters.

2.2 LSTMIMPLEMENTATION

LSTM networks were chosen for PID output prediction due to its robustness, better handling of vanishing gradients, and capacity to learn longer ranges between events, all of which are important for neural networks in comparison to standard RNNs. As shown in Fig. 2, each cell represents a time stamp, and adjacent cells are connected by two vectors: the Cell State (C_t) and the Hidden State (h_t), which help in learning temporal dependencies. The data was collected at a rate of one data point per second. As a result, the 5400-data-point dataset was divided into training and testing, with 80% of the data allotted for training and 20% for testing. The data-window of the preceding 15 seconds was used for prediction in the LSTM algorithm. Two LSTM layers and a fully connected hidden layer were used to build the model.

A dense layer, which is a feature of feed forward neural networks, is an example of an internal layer that is added to the neural network in addition to LSTM layers. Furthermore, regularisation procedures are typically used to limit the likelihood of over-fitting. Dropout is the preferred strategy for preventing over-fitting in the generated model (resulting in a 2% reduction in error). Furthermore, the activation function is a basic component of all neural networks, including LSTMs. To facilitate processing, the activation function nonlinearly alters the data. LSTMs have sigmoid and hyperbolic tangent functions as standardised activation functions.The neural network was tested with different parameter values, and the parameter values used for the model were evaluated. Table1 represents the most optimal values and methods used in our design.

Table 1: Hyper-Parameters Values

Hyper-parameters	Value/Method
Layers and activationfunctions	LSTM Layer (100, (tanh), sigmoid)Dropout (0.1) LSTM Layer (100, (tanh), sigmoid)Dropout (0.1) DenseLayer(15, tanh)
Timestep (History dimension)	15
Optimizer	Adam
Dropout rate	0.1
Costfunction	MeanSquareError(MSE)
Numberofiterations	300(with earlystopping)
epochs	25
Batchsize	100
Return sequence	True

The neural network's learning is carried out in batches of a certain number of iterations, with each step comprising a batch size of samples. An iteration is a loop in which all of the training data is handled. The cost function values for the training and testing data are evaluated after each iteration. The mean squared error (MSE) is used for the cost function. With a history size of 15, the Adam optimizer with a learning rate of 0.005 was chosen since it produced the best results in terms of learning speed and accuracy. Early stopping was used to improve performance during the model's design stage, as shown in table 1. All hyper-parameters were tested experimentally, and optimal values for model development were identified.

3.0 MODEL PERFORMANCE

Figure7 depicts the model performance produced as a result of training with the chosen parameters. The error values reduced over time reaching a minimal loss level of low 10e-2.

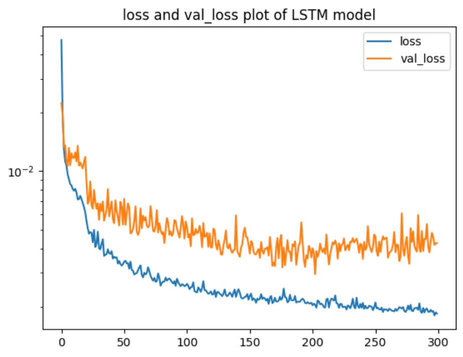

Fig.7. Trainingperformancegraph.

The LSTM model was initially tested on the data set aside as the test dataset after being trained with PID controller outputs. Figure 8 displays the PID controller and LSTM network outputs for the test data on the same graph.

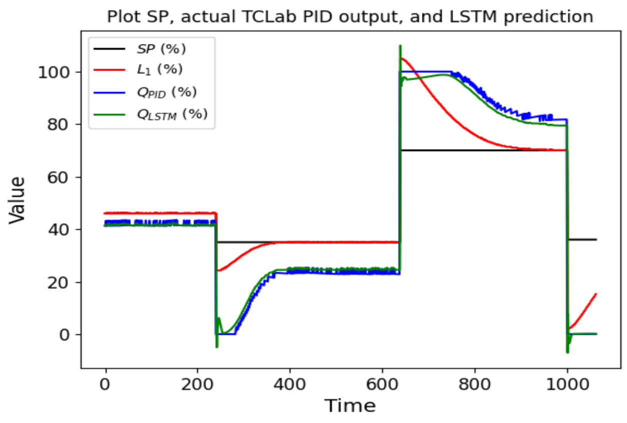

Fig.8.Test result graph for PID and LSTM outputs.

Figure 8 shows the output level (output of the PID control system), Q_{PID} as the percentage value of the proportional valve output controlled by the actual TCLAB PID, Q_{LSTM} as the predicted percentage value which will generate 4-20 mA output corresponding to 0 to 100% in valve, and *SP* as the target level value. When comparing the output values obtained with PID control and the output values obtained with the LSTM network, it can be observed that they yield quite similar results.

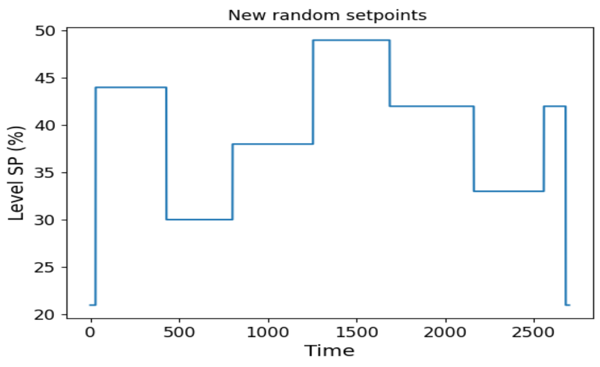

Fig.9. Test recipe created to test the LSTM network

Figure 9 shows the new tested set points generated for LSTM model testing.Figure10 displaystheresultgraphfortheoutputsofbothcontrolmethods,PIDcontroller,andLSTMnetwork,testedonthenewlycreatedtestdata.Inthisstageofthestudy,L_1represents the current level value controlled bythe LSTM network. It can be observed that the controlsystem successfully reproduces outputs for a different testdata.

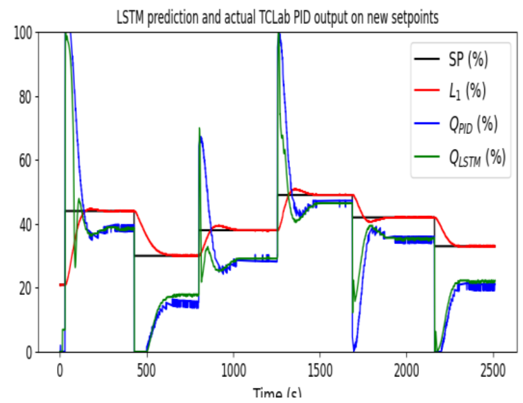

Fig.10.TestresultgraphforPIDandLSTMoutputsaccordingtonewtestrecipe.

The LSTM network model was tested independently of the PID controller in the last stage of the testing phase. In this scenario, the LSTM h5 model is solely used in standalone mode to test a new set-point. Figure.11 illustrates the outcomes of this test. The test output graph shows that the LSTM network successfully attained the level set-points and effectively regulated the proportional valve outputs.It is possible to infer that the LSTM-based neural network model achieved substantial success in level control, and its integration into different systems is possible regardless of PID parameters.

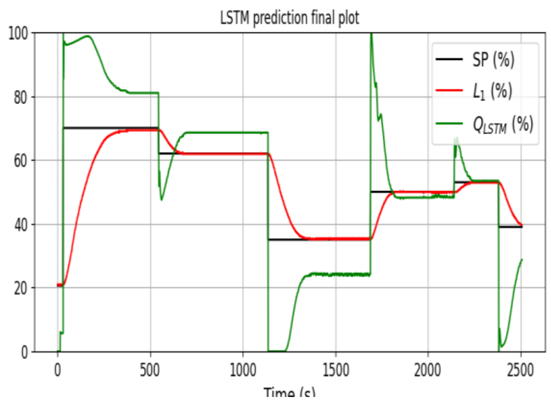

Fig.11.TestresultgraphforLSTMnetworkmodeloutputs.

4.0 HTTP-SERVERCONTROLLER HARDWARE

As illustrated in Fig.12, the created LSTM network is established employing indigenously produced devices with Ethernet connectivity that serve as distributed nodes. The front-end of this hardware is an ARM MBED microcontroller based on NXP 1768 that runs an embedded HTTP server with a network-specific static IP address. The device accepts a 4-to-20-mA current-loop input coupled to a 12-bit ADC for level-sensor read-out data and another 4-20mA DAC output for proportional valve actuator interface, all of which are controlled via HTTP GET commands. The level-sensor data packet is generated from the level input, and the actuator-data drives a power driver circuit, both of which can be controlled via IoT. This hardware is an alternative approach that replaced apmonitor.com's reliance on USB-driven commercial TCLab custom boards[25], which we discovered were difficult to obtain in India. Furthermore, this indigenous device is a cost-effective option that can be located near valves that are remote from the central computer, and the main advantage is that such a large number of devices can be deployed across Ethernet anywhere in the world or within a private LAN. The hardware has been tested under critical conditions of linear accelerators in online experiments. The firmware of this device is created using ARM MBED studio. Each HTTP GET is easily programmed using python httpclientlibraries [26] which are freely available.

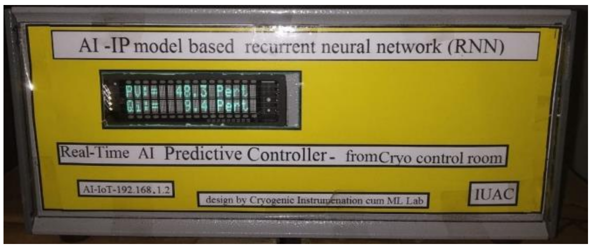
AI -IP model based recurrent neural network (RNN)
PV= 48.3 Pert
Qi= 9.4 Pert
Real-Time AI Predictive Controller - fromCryo control room
AI-IoT-192.168.1.2
design by Cryogenic Instrumenation cum ML Lab
IUAC

Fig.12.

The network-on-chip (NOC) firmware implementation resulted in a small TCP/IP stack within ARM microcontroller chip. Most application programs communicate with Ethernet via a Transmission Control Protocol/Internet Protocol (TCP/IP) stack. The read/write operation of network variables is accomplished by implementing the RPC-over-HTTP mechanism via remote procedure calls (RPC) built into ARM microcontroller chips. A device server is thus created by a meaningful set of http commands that are firmware-coded into each on-chip http-server. Such a vast number of command sets implanted throughout a big number of device-servers in the network transforms it into a full-fledged control network, where devices can directly send or receive data with a central PC.

RESULTS

This work aimed to developan innovative LSTM-basednetworkmodelfor control networks and can reach level values in the most optimal time, as an alternative to thesuccessofthePIDcontrolalgorithminlevelcontrol. Figure 13 depicts the outcome of a successful deployment of LSTM in a linac experiment where liquid nitrogen level (red) followed a setpoint-value (black) by automatically changing valve (green) when control is transferred from manual mode to LSTM mode. Using the TCLAB framework, the LSTM network was trained using PID system models. For training, the mean square error (MSE) approach was used.

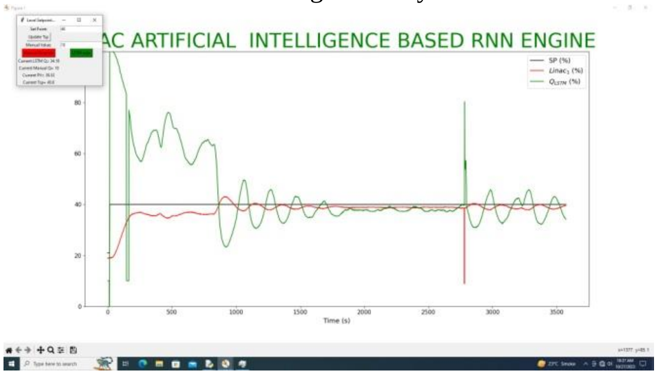

Fig 13. The successful LSTM in manual and auto mode

Finally, the PID system and subsequently the constructed LSTM network model were tested online using randomly generated target level set-points with the real linac experiment, yielding an error value of +/-1%. The graphs produced compare the outcomes of the LSTM network and the PID system. The LSTM network produced successful results when run independently of the PID system.

DISCUSSION

Artificial intelligence and deep learning applications are being used successfully and are becoming increasingly ubiquitous in a variety of fields nowadays. Deep learning algorithms' ability to learn and develop is an important innovation in addressing and improving closed loop control systems. In keeping with this goal, the creation of a real time control system that is trained on high-dimensional datasets and evaluated on real-world systems with optimal solutions can make a substantial contribution. Taking all of these issues into consideration, the development of a new control system based on deep learning contributes significantly to the industry and the literature.