

Machine Learning for the prediction of the dynamic behavior of a small scale ORC system

Laura Palagi^a, Apostolos Pesyridis^b, Enrico Sciubba^c, Lorenzo Tocci^{c,d,*}

^a*Department of Computer, Control, and Management Engineering, La Sapienza University of Rome, Via Ariosto 25, 00185 Rome, Italy*

^b*Department of Mechanical, Aerospace and Civil Engineering, Center of Advanced Powertrain and Fuels (CAPF), Brunel University, London, UK*

^c*Department of Mechanical and Aerospace Engineering, La Sapienza University of Rome, Via Eudossiana 18, 00184 Rome, Italy*

^d*Entropea Labs, 2a Greenwood Road, E81AB London, UK*

Abstract

Dynamic modeling plays a crucial role in the analysis of Organic Rankine Cycle (ORC) systems for waste heat recovery, which deal with a highly unsteady heat source. The efficiency of small scale ORCs (i.e. below 100 kW power output) is low ($< 10\%$). Therefore, it is essential to keep the performance of the system as stable as possible. To do so, it is helpful to be able to predict the dynamic behavior of the system, in order to perform a maximization of its performance over the time.

In this paper, Feedforward, Recurrent and Long Short Term Memory networks have been compared in the prediction of the dynamics of a 20 kW ORC system. Feedforward Neural Network is the simplest among the architectures developed for machine learning. Recurrent and Long Short Term Memory networks have been proved accurate in the prediction of the performance of dynamic systems. This study demonstrates that the three architectures are capable of predicting the dynamic behavior of the ORC system with a good degree of accuracy. The Long Short Term Memory architecture resulted as the highest performing, in that it correctly predicts the dynamics of the system, showing an error prediction lower than 5 % and 10 % respectively for what concern the prediction 10 and 60 seconds ahead.

Keywords: Artificial Neural Networks, ORC, Dynamic system, Experimental ORC

1. Introduction

The increase in the energy demand of the last decades and in the level of pollutants in the atmosphere pose challenges on the future of the energy production. Interest is growing

*I am corresponding author

Email addresses: laura.palagi@uniroma1.it (Laura Palagi), apostolos.pesyridis@brunel.ac.uk (Apostolos Pesyridis), enrico.sciubba@uniroma1.it (Enrico Sciubba), lorenzo.tocci@uniroma1.it (Lorenzo Tocci)

for those technologies that are able to produce electrical energy out of wasted sources, increasing the overall efficiency of existing power plants. A statistic that refers to the EU and reported in [1] reveals that 800 TWh are discharged every year to the environment as wasted thermal energy from industrial processes. Among other technologies, Organic Rankine Cycle (ORC) represents a promising method to produce electricity in waste heat recovery (WHR) applications. An ORC for WHR extracts the thermal energy from a wasted heat source and converts it to electricity by means of a thermodynamic cycle. Typically, the heat that feeds the ORC system is derived from wastes of industrial processes or from the exhaust gas of internal combustion engines (ICEs).

The performance of ORC plants for waste heat recovery applications is highly dependent on the fluctuations of the mass flow rate and temperature of the heat source, which varies according to the duty cycle [2]. Therefore, the ability to predict the dynamic behavior of the system is crucial for the implementation of a control system capable of optimizing the performance of the system over time.

The dynamic modelling of ORC systems has been extensively studied in the literature. Quoilin et al. [3] presented a numerical model to simulate the dynamic performance of an ORC system that is built interconnecting sub-models of the components. Zhang et al. [4] proposed a physical model to mimic the dynamics of a 100 kW ORC system. Twomey et al. [5] investigated the dynamic performance of an ORC system for solar thermal cogeneration. Authors provided different modelling approaches to characterize the transient behavior of the components. Particular emphasis is given to the heat exchangers, since their time constant is the highest among the different components of an ORC system. Desideri et al. [6] compared the moving boundary and the finite volume method in the characterization of the dynamic performance of ORC evaporators. Wei et al. [7] presented a comparison between discretization and the moving boundary technique to assess the performance of the two methods. They stated that both models have good accuracy. Bracco et al. [8] presented a numerical method to dynamically simulate a double-pipe heat exchanger. Vaja [9] proposed a dynamic model of the ORC system based on the discretization of the differential equations that are representative of the dynamics of the system. Quoilin et al. [10] addressed the issues related with numerical problems in modelling the transient behavior of evaporators, highlighting those related to the dynamic simulation of fluids in the two phase region. The models listed aim at simulating the dynamics of the process via a physical interpretation of the transformations that occur in the ORC components, solving the differential equations that are representative of the processes of energy transfer.

Machine learning techniques (MLTs) constitute a different approach to model the dynamic behavior of energy systems. MLTs are based on the possibility to learn the behavior of the system using experimental data. Among MLTs, Artificial Neural Networks (ANNs) can be used to derive the output of the system from examples [11]. The use of ANNs in energy applications has already been proved successful in the past. Kalogirou [12] presented a review which reported the use of neural networks in different renewable energy applications. He highlighted the potential of such tool in energy system's modelling. Narendra and Parthasarathy [13] discussed the use of neural networks in the identification and control of non linear dynamic systems. Mohanraj et al. [14] performed a review of the use of neural

networks for heat exchanger's modelling.

The application of ANNs to the design, optimization and control of ORC systems is recent. Some authors used ANNs for the static optimization of ORC systems. Yilmax et al. [15] used neural networks to investigate over the efficiency of ORC cycles using R410a and R407c as the working fluids. Massimiani et al. [16] proposed the use of neural network within a highly non-linear optimization model to allow the use of standard optimization algorithm and hence reduce the computational time required for deriving its solution. They applied their method to the optimization of an ORC system for low temperature applications. Rashidi et al. [17] performed an optimization of the thermal efficiency, the exergy efficiency and the specific work of ORC systems using a combination of artificial bees colony and artificial neural networks. Recently, some papers dealing with the dynamic modeling of ORCs using ANNs have been published. Liu et al. [18] used Radial Basis Function (RBF) networks to model an ORC for waste heat recovery applications. Yang et al. [19] performed a static optimization, looking for the optimal values of the inputs to the network that maximizes the power of the turbine. To the authors' knowledge, recurrent networks, such as RNN and LSTM, have never been applied to the dynamic characterization of ORC systems.

This study proposes the comparison of three different ANN architectures, namely Feed Forward (FF) networks, Recurrent Neural Networks (RNN) and Long Short Term Memory (LSTM) networks, in the prediction of the dynamic performance of a 20 kW ORC system. A wide experimental campaign to get experimental data for training the networks has been performed. FF networks has been proved successful in many different applications. RNNs and LSTMs present more sophisticated network's architecture, which make them suitable for the prediction of transient signals. The novelty of this study relies on the use of RNN and LSTM for the dynamic characterization of the performance of ORC systems and in their comparison w.r.t. FF. RNN and LSTM networks became popular in the past decade mainly in the field of language detection but are now widely applied to prediction of temporal sequences as dynamics of physical system. This paper investigated the suitability of the methodology for the prediction of the dynamic modelling of ORC systems and it is intended as a first step towards the definition of an optimal controller for ORC applications based on ANNs. The paper is organized as follows. In Section 2 the three neural network architectures are described. In Section 3 the experimental set up of the 20 kW ORC system is presented, which has been developed by Entropea Labs [20] in partnership with Brunel University London. In Section 4, the experimental data are analysed and put into the correct setting useful for training ANN. Finally in Section 5, the capability of the different ANN architectures in predicting the dynamic behavior of the ORC system is assessed. Based on the results obtained, the three ANN architectures analyzed resulted capable of predicting the dynamic behavior of the ORC system. LSTM networks outperform the other two configurations analyzed, showing their ability to keep track of the information of previous time steps.

2. Neural networks

Machine learning techniques are used as regression tools in all those cases in which examples of an unknown process are available but there is no known analytical model that correlates these quantities. Among learning machines, Artificial Neural Networks are widely used for their simplicity, ease of implementation and for their capacity to achieve good performance in many different tasks [21]. Section 2 introduces the three different network architectures considered in this study, presented in order of complexity.

The definition of the architecture of a neural network consists in choosing the number of parameters that define the vector of inputs $\mathbf{x} \in \mathbb{R}^n$, the number of hidden layers of the network and the number of neurons of each layer. Neural network architectures differ in the way they treat the information of the input vector. Generally speaking, the higher the number of layers and the number of neurons in each layer, the more complex is the function the network can predict with a good degree of accuracy. However, the use of a network that is too complex for the information it needs to learn, could lead to the phenomenon of over-fitting of the training data. If this happens, the network will perform extremely well in predicting the known data of the training set but it will not be able to make good predictions on unseen instances, which is ultimately the goal of a neural network. The correct network architecture must be defined through a trial and error process.

The aim of the ANN model is to define a function $h : \mathbb{R}^n \rightarrow \mathbb{R}$ which is a good approximation of the unknown function underlying the process. The type of function and the parameters' setting is defined through a learning procedure as briefly described below.

The design of a neural network is performed in two different steps:

- Definition of the architecture of the network;
- Training of the network.

Once the architecture is defined, the network is trained on a number of existing instances $(\mathbf{x}^p, \bar{h}^p)$, for $p = 1, \dots, P$, where $\mathbf{x}^p \in \mathbb{R}^n$ is a vector that represents the features of the input and $\bar{h}^p \in \mathbb{R}$ the corresponding output. The training process consists in solving an optimization problem, in which the mean squared error (MSE) of the network in predicting known instances is minimized.

At the end of the training process, the network is tested on unknown instances, to verify its generalization properties, i.e. the ability of the ANN to make good predictions on unseen data. If the results in the testing phase are not satisfactory, the architecture of the network is modified in a trial and error process, until the generalization properties satisfy the user.

It is widely considered that one hidden layer is enough for the resolution of almost any regression problem. Hence, the use of the so called deep learning networks has not been considered in this work. The choice of the number of neurons in the hidden layer represents an unsolved question. Most of the scientists use empirical methods to define them, which lead to the selection of a number of hidden neurons comprehended between the number of the input nodes and that of the output nodes [22]. Another factor that affects the choice of the number of neurons in the hidden layer is the number of instances available in the

training set. A high number of neurons in the hidden layer increases the risk of over-fitting with respect to the training data. However, the higher the number of data in the training set, the higher the number of hidden neurons needed to obtain satisfactory performance of the network. The objective is to define a network that presents a similar error between the test and the training set. If this happens, the ANN has good generalization capabilities.

In this work, different architectures of ANNs have been designed to predict the dynamic behavior of the 20 kW ORC system presented in section 3. To this end, a training set and a test set have been constructed, which are made up of pairs $(\mathbf{x}^p, \bar{h}^p)$ obtained collecting measurements from the sensors installed on the ORC test rig. Six different experiments have been considered in the analysis. Four experiments define the training set while two experiments have been used to test the performance of the network.

The network has been designed and tested using the open source software Keras [23] implemented in Python [24]. Section 2.1, 2.2 and 2.3 present the different network architectures considered in this work.

2.1. Feed Forward Neural Networks (FF)

Feed Forward (FF) neural networks represent the most basic ANN architecture. FF networks are composed by units (neurons) organized in layers forward connected with each other. A graphical representation of a FF network is shown in Figure 1 (a) where the nodes represent the neurons and the arcs represent the weighted synaptic connections among neurons. The basic FF configuration is based on the following layers:

- an input layer, that receives the training examples;
- an intermediate layer, called *hidden* layer;
- an output layer, which consists of a single neuron and produces the output of the network.

A single unit of the hidden layer is characterized by the activation function g , which acts as an on-off trigger on a weighted combination of the outputs of the neurons in the preceding layer, and a bias b_j ; each connection from input i to neuron j of the hidden layer is characterized by a weight w_i^j as shown in Figure 1 (b); the arcs from the hidden neuron j to the output neuron are weighted by w_o^j .

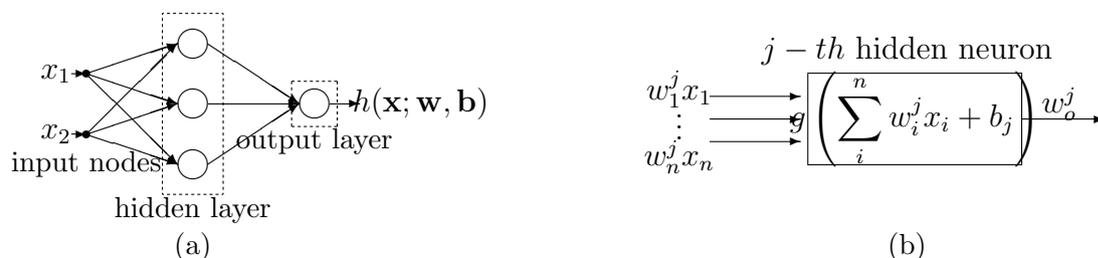


Figure 1: (a) Feed Forward network configuration with one hidden layer and one output; (b) internal structure of the j -th neuron of the hidden layer.

The number of neurons m in a hidden layer together with the activation function g of the neurons are user-dependent parameters. Once they are fixed, the output of the FF network takes the following form:

$$h(\mathbf{x}; \mathbf{w}, \mathbf{b}) = \sum_j^m w_o^j g \left(\sum_i^n w_i^j x_i + b_j \right) + b_o ; \quad (1)$$

where $\mathbf{w} = \{w_o^j, w_i^j\}_{i=1, \dots, j=1, \dots, m}$ and $\mathbf{b} = \{b_j, b_o\}_{j=1, \dots, m}$.

All the parameters \mathbf{w}, \mathbf{b} that appear in Eq. (1) are “learned” through a *training* process which consists in the resolution of the minimization problem of Eq. (2):

$$\min_{\mathbf{w}, \mathbf{b}} E(\mathbf{w}, \mathbf{b}) = \frac{1}{2} \sum_p^P \|\bar{h}_p - h(\mathbf{x}_p; \mathbf{w}, \mathbf{b})\|^2 . \quad (2)$$

2.2. Recurrent Neural Networks (RNN)

Recurrent neural networks (RNN) differ from FF ones in that the output of the network is fed back to the network, as depicted in Figure 2. The output of the network is cyclically fed back to the hidden layer, after being weighted. This feature makes RNNs useful for temporal sequences, because they are able to maintain information from previous inputs.

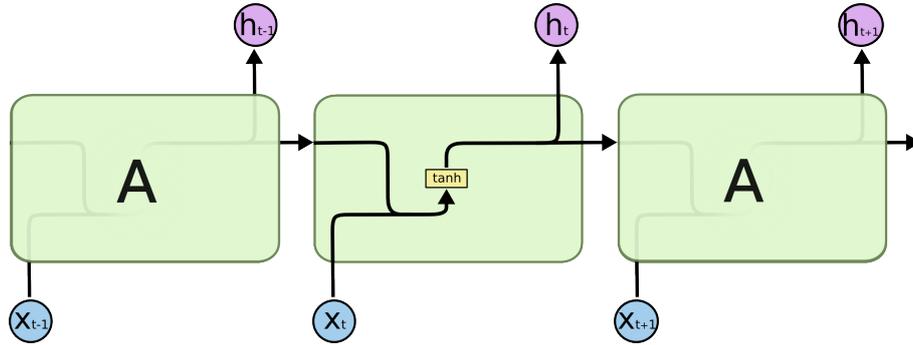


Figure 2: Recurrent Neural Network (RNN) structure from [25]

The output of the network h_t (see Figure 2) depends on the new input to the network x_t and on the output of the network at the previous time step h_{t-1} and it can be expressed as shown in Eq. (3):

$$h_t = \tanh(W_h[h_{t-1}, x_t] + b_h) \quad (3)$$

where the vector of weights W_h is composed of the weights that refer to the input of the network and of the weights that refer to the output of the previous time step. The training phase consist in solving an optimization problem similar to that introduced by Eq. (2) in which the vector of weights W_h and the vector of bias b_h comprehend also the values that refer to the feedback loops. Since the feedback loop occurs at each time step, RNNs push forward information from time steps in the past as long as the memory can persist.

2.3. Long Short Term Memory Networks (LSTM)

Long Short Term Memory (LSTM) networks represent an advancement with respect to the simple RNNs presented in Section 2.2. LSTMs were introduced in 1997 [26] to solve the problem of long term dependencies. This is of particular importance for ANNs used for time series prediction, in which the dependencies of the output of the network from values of the input of old time steps is not known *a priori*. The neurons in LSTMs are replaced by memory cells, as the one presented in Figure 3.

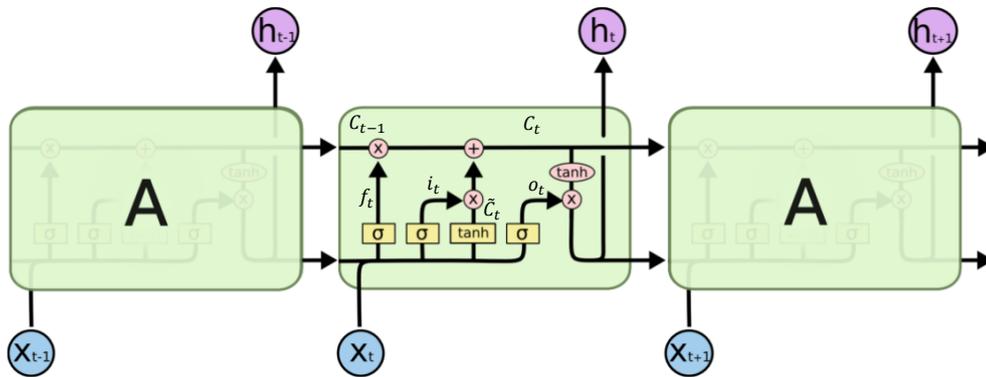


Figure 3: Long Short Term Memory (LSTM) cell from [25]

LSTM cells are capable of filtering the information present in the input layer, discarding or memorizing instances of the past. The input of the network x_t and the output of the previous time step h_{t-1} are manipulated in the LSTM cell to produce the output h_t . As opposed to traditional neural units, LSTM cells present a cell state that takes into account the relationship between the inputs of the network and the information previously elaborated.

An in depth explanation of LSTMs is out of the scope of this work. The interested reader can refer to [25] for a thorough explanation. The high number of weights that characterize each cell of LSTM networks makes the training phase more time consuming with respect to that of the other network architectures presented.

The experimental data obtained from the ORC test rig are presented in section 3, which have been used to build the training and test sets used to compare the ANN architectures presented in section 2.

3. Experimental set-up of the 20 kW ORC system

Section 3 shows the description of the experimental set-up considered to gather the data used for the training and test of the ANNs. The ORC plant has been designed by Entropea

Labs [20], a London based company that develops waste heat recovery technology based on the Organic Rankine Cycle. Figure 4 shows the schematic of the ORC plant including the position of the sensors in the test rig.

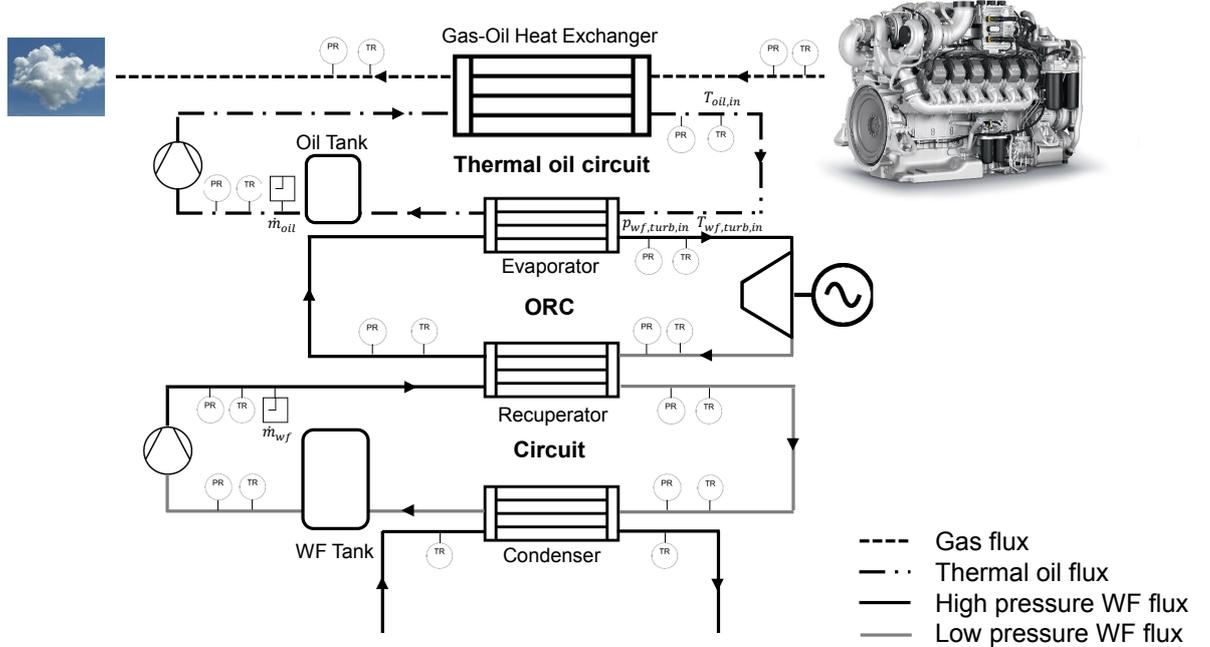


Figure 4: Scheme of the ORC test rig, including the sensor installed

Figure 4 reports the main components of the ORC together with the pressure (PR), temperature (TR) and flow meter transistors installed in the experimental rig.

3.1. Heat transfer loop

In the thermal oil circuit (see Figure 4) the thermal energy content of the exhaust gas of the engine is transferred to a thermal oil. The engine of the experimental set up is a 206 kW engine by Yuchai (engine model YC6A280-30), installed in the engine lab of Brunel University London. The heat transfer between the exhaust gas and the thermal oil takes place in a shell and tube heat exchanger, which is the only component that is not installed in the ORC skid. The gas/oil heat exchanger has been designed keeping the back pressure on the engine low enough to avoid the deterioration of the engine performance. A volumetric pump is installed to circulate the oil in the loop. The frequency of the motor of the oil pump is controlled by means of an inverter. The maximum volumetric flow rate that the oil pump can supply is 40 l/min .

3.2. ORC loop

The ORC loop consists of a fluid tank, a volumetric pump, a recuperator, an evaporator, a turbine and an electric generator. The pump sucks the working fluid from the tank and it

pressurizes it. The fluid flows in the recuperator and in the evaporator where it is pre-heated and vaporized. The turbine extracts energy from the fluid, producing mechanical energy. The fluid at the outlet of the turbine is circulated through the hot side of the recuperator and through the condenser to revert back to its initial thermodynamic conditions. The mechanical power produced by the turbine is converted into electrical energy in the high speed generator and it is dissipated in a resistor load back. The volumetric flow rate of the working fluid is controlled via an inverter, which allows the user to control the frequency of the motor coupled to the pump. The maximum volumetric flow rate that the working fluid pump can supply is 80 l/min . The heat exchangers of the ORC loop, namely evaporator, recuperator and condenser are brazed plate heat exchangers, boarded by insulating panels. The expansion machine is a single stage radial turbine developed in house coupled with a high speed generator.

3.3. Cooling loop

The lab is equipped with a water tank, which supplies the cold water to condensate the working fluid. Two flexible hoses connect the water tank of the facility to the condenser installed in the ORC skid. The temperature of the cooling water varies in the range $5 - 25^\circ\text{C}$.

3.4. Transducers

The test rig is equipped with transducers, as depicted in Figure 4. Pressure, temperature and mass flow rate levels are recorded during operations, to characterize the performance of the system. The thermodynamic library COOLPROP [27] is used to calculate the enthalpy and entropy values at each thermodynamic point to derive the efficiency of the components, the power absorption of the pumps and the power output of the system.

4. Analysis of the training and test sets

This work investigates the possibility to use ANNs to predict the dynamics of the system. To this end, the first step to take is the selection of the variable to monitor, which represents the output of the network named h_t in section 2. The purpose of the neural network is to predict the temperature of the working fluid at the inlet of the turbine, which plays a crucial role in the definition of the ORC performance. In fact, the temperature of the working fluid at the inlet of the turbine greatly affects the efficiency of the turbine and hence, the power output produced by the ORC system. This work is intended as a first step towards the definition of an optimal controller for the ORC system based on the use of Neural Networks. The choice of predicting the temperature of the working fluid at the inlet of the turbine has been made to demonstrate the validity of the proposed approach. The same technique can be used to predict other variables of interest for the control of the system. The different configurations of the neural network, namely FF, RNN and LSTM, have been trained to predict the turbine inlet temperature respectively 10 seconds, 60 seconds and 120 seconds ahead. The predictions 120 seconds ahead are considered sufficient to actuate the control system. Eq.(4) reports the output variable of the network h_t .

$$h = [T_{wf,turb,in}(t + n)]_{n=10,60,120} \quad (4)$$

Once the outlet of the network is defined, the user should define the vector of inputs, named x_t in section 2. It is necessary to include in the vector of the inputs all those variables that have an effect in the fluctuation of the output vector h_t . In this case, for simplicity, the gas/oil heat transfer loop has been neglected, considering the thermal oil as the unsteady heat source of the ORC system; this assumption allows to extend the results of this analysis to ORC systems for different applications, where the gas/oil loop is absent. The mass flow rate and temperature of the thermal oil at the inlet of the evaporator (see Figure 4) have been included in the input vector x_t , together with the mass flow rate, the temperature and the pressure of the working fluid at the inlet of the turbine. A trial and error process led to the decision of feeding the network with a set of five different values for each of the aforementioned variables, recorded in five consecutive seconds during the experiments. Eq. (5) reports the input vector of the network x_t .

$$x = \begin{bmatrix} \dot{m}_{oil}(t - 5)_{t=1,\dots,5} \\ T_{oil,in}(t - 5)_{t=1,\dots,5} \\ \dot{m}_{wf}(t - 5)_{t=1,\dots,5} \\ p_{wf}(t - 5)_{t=1,\dots,5} \\ T_{wf,turb,in}(t - 5)_{t=1,\dots,5} \end{bmatrix} \quad (5)$$

The total number of inputs to the neural networks is 25, five different values for each of the five variables considered.

The dynamic tests on the experimental rig have been conducted with the internal combustion engine that feeds heat to the ORC operated at part load. Table 1 reports the conditions of different variables under which the tests have been performed.

Table 1: Testing conditions

$\dot{m}_{oil}[kg/s]$	$T_{oil,in}[^{\circ}C]$	$\dot{m}_{wf}[kg/s]$	$p_{wf}[bar]$	$T_{wf,turb,in}[^{\circ}C]$	$P_{engine}[kW]$
0.3 - 0.5	75 - 255	0.1 - 0.6	1 - 12	40 - 225	60 - 80

A broader range of values for the operating conditions of the engine would have led to a broader set of information for the training of the neural network. However, this work aims at demonstrating that the proposed methodology allows for the prediction of the dynamic performance of ORC systems, regardless of the operating conditions.

The selection of the input and output vectors of the ANNs has been followed by the definition of the training set and of the test set. The networks have been trained considering 4 different experimental tests. Each variable has been recorded every second. The training set is formed by 10995 instances. The network has been tested on two additional data sets obtained from two experiments, Test set 1 and Test set 2, to verify the capability of the three networks to generalize on unseen examples. The test sets count for a total of 8934 instances. Table 2 reports the statistical description of the overall available samples.

Table 2: Analysis of the data in the training and test sets

	$T_{oil,in} [^{\circ}C]$	$\dot{m}_{oil} [kg/s]$	$P_{wf} [bar]$	$\dot{m}_{wf} [kg/s]$	$T_{wf,turb,in} [^{\circ}C]$
count	19929	19929	19929	19929	19929
mean	184.311879	0.523552	4.385740	0.261017	168.298370
std	29.906591	0.063264	3.557430	0.172044	38.167201
min	73.135353	0.219002	0.815605	0.000000	40.660940
25 %	167.567898	0.495215	1.562166	0.078287	155.341028
50 %	184.191427	0.539358	3.418055	0.263291	172.239101
75 %	202.022090	0.563886	5.336384	0.385660	192.458527
max	253.724995	0.614862	13.858494	0.617379	224.317270

Table 2 shows the following values:

- count: The number of instances present in the set
- mean: The mean value of of the variables in the set
- std: Standard deviation that expresses by how much the values differ from the mean value
- min: The minimum value it is possible to find in the set for each variable
- 25, 50 and 70 %: The percentile that indicates the value below which a given percentage of observations can be found
- max: The maximum value it is possible to find in the set for each variable

The model proposed would accurately predict the temperature of the working fluid at the inlet of the turbine when the input values fall in the min-max interval shown in Table 2. The min and max values demonstrate the dynamics of the experiments that have been carried out. In fact, for every variable in input to the network there is a wide variability. In addition, the percentile values allow the reader to understand the distribution of the values of the collected data. For clarification, the reader is referred to Figure 5, where the frequency of appearance of different values of the turbine inlet temperature is reported.

Figure 5 shows that the temperature values in the training set (left) cover a broader range with respect to those of the test set (right). This is of importance in that the network will be able to accurately predict values in the range of those covered by the training set. Furthermore, Figure 5 shows that the turbine inlet temperature of the working fluid during testing has been kept in the range 130 - 210 °C for most of the time.

A correlation matrix is constructed by evaluating the Pearson’s coefficient among the different inputs of the network and it takes values in $[0;1]$ being 1 perfect correlation. The correlation matrix has been designed using the software Keras [23]. Figure 6 reports the correlation matrix.

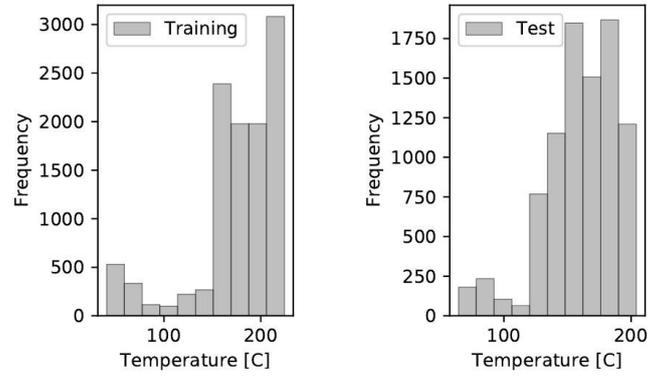


Figure 5: Frequency of the values of temperature of the working fluid at the inlet of the turbine $T_{wf,turb,in}$

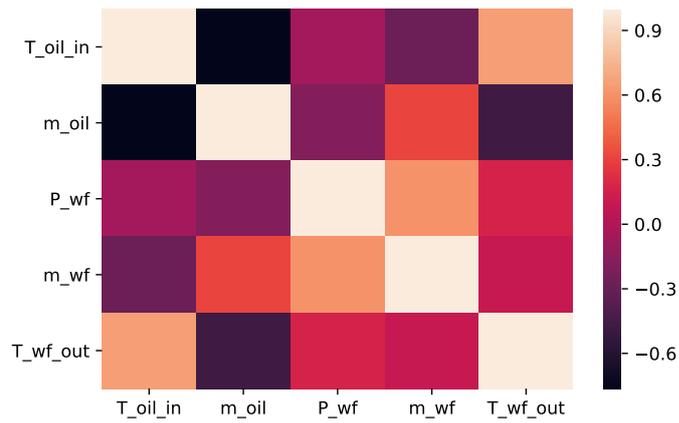


Figure 6: Correlation matrix of the data

A correlation matrix is representative of the correlation existing among the different inputs of the network. The purpose of the correlation matrix is twofold. First, the correlation matrix can be used to withdraw from the input vector some inputs which are redundant, i.e. linearly dependent on other inputs. Second, it allows the user to make sure that the data collected are consistent. Values of light colors imply high correlation, 1 revealing perfect correlation. The values in dark colors imply low correlation. For example, Figure 6 shows that there is high correlation between the mass flow rate and the pressure of the working fluid. This can be physically explained. An increase in the mass flow rate of the working fluid has the effect of increasing the back pressure produced by the turbine, and hence, the pressure on the high pressure side of the system. On the contrary, the correlation between the temperature of the oil at the inlet and the pressure of the working fluid is close to zero. In fact, the pressure of the working fluid is affected by the parameters on the working fluid side but it is not dependent on the fluctuation of the parameters on the oil side. Finally, the correlation between the temperature and the mass flow rate of the oil is low. In fact, the mass flow rate of the oil has been kept fairly constant during testing. The variation of the thermal input from the exhaust gases and the thermal output to the working fluid lead to the fluctuation of the temperature of the oil. Figure 7 shows the frequency of appearance of the different mass flow rate values in the training set and in the test set.

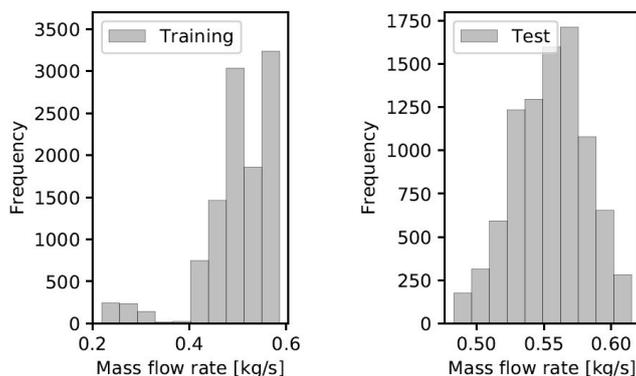


Figure 7: Frequency of the values for the mass flow rate of the oil

Figure 7 clarifies that the mass flow rate of the oil has been kept in the range $0.5\text{--}0.6\text{ kg/s}$ throughout the experimental campaign. Following the data pre-processing, the different configurations of the neural network have been trained and tested to evaluate their predicting performance.

5. Results

In this section the results obtained from the training of the different neural networks and those of the predictions obtained on the test set are presented.

The training of a network consists of the minimization of the Mean Squared Error (MSE) on the instances of the training set. The MSE is defined as in eq (6)

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{h}_i - h_i)^2 \quad (6)$$

where n is the number of instances of the dataset, \hat{h}_i is the value predicted by the network and h_i is the experimental value corresponding to the i -th input. The algorithm used for the training of a network is a modified stochastic gradient descent algorithm as implemented in Adam algorithm [28] of the platform Keras. The training phase has been performed using 15 epochs and considering a batch size of samples of 300.

The performance of the networks have then been compared using the root mean squared error (RMSE) which is defined as the squared root of the MSE of the network in predicting experimental values and it is expressed as in eq. (7):

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{h}_i - h_i)^2} \quad (7)$$

In particular the RMSE has been considered for the training set and the test sets. The closer the RMSE value to zero, the higher is the accuracy on the chosen data set. However when the RMSE on the training set is a much lower with respect to those of the test sets, this may be a warning that the network is not able to generalize on instances it has never seen before. This phenomenon is known as over-fitting. Actually this is a well known trade-off to account when solving the optimization training problem. Indeed tricks like early stopping or k -fold cross validation to individuate the best configuration of the network are often used. In this specific case, a 10-fold cross validation has been performed over a grid of parameters that characterized the networks.

Indeed the networks have been trained multiple times with different number of neurons and parameters to define the most performing configurations, i.e. those that allow for the minimization of the error in the prediction of instances of the test sets. A single hidden layer with 20 neurons has been considered for the FF and for the RNN networks. A single layer with 5 memory cells has been considered for the LSTM network.

The different network's configurations have been trained using the same training set, for the purpose of comparison.

Table 3 reports the RMSE of each of the networks tested.

The RMSE is calculated for each of the sets (i.e. training set, test set 1 and test set 2) comparing the value predicted by the trained network and that recorded experimentally from the test rig. Table 3 shows that every network has been trained successfully, displaying comparable values between the training and the test sets. The internal structure of the memory cells allows the LSTM network to keep track of the information from previous time steps that influences the output of the network. In fact, LSTMs have been conceived to let information last in the "memory" of the cell for a long time. On the contrary, FF networks

Table 3: RMSE for each of the network design

Network	Training Set	Test Set 1	Test Set 2
FF 10 s	0.03	0.04	0.05
FF 60 s	0.06	0.06	0.09
FF 120 s	0.08	0.07	0.11
RNN 10 s	0.02	0.02	0.03
RNN 60 s	0.05	0.04	0.08
RNN 120 s	0.07	0.07	0.11
LSTM 10 s	0.05	0.03	0.03
LSTM 60 s	0.08	0.06	0.05
LSTM 120 s	0.11	0.09	0.08

do not take into account the consequentiality of the instances provided during the training process; for this reason, they are more suitable for steady state predictions. Finally, RNNs have the ability to learn the relationship existing between subsequent inputs but they are not good in learning the relationship among inputs far apart. From a computational stand point, FF networks display the shortest training time (about 15 seconds per epoch), followed by RNNs (about 120 seconds per epoch) and LSTMs (about 200 seconds per epoch). The computer resource used to train the ANNs is an Intel Core i5-421-U CPU 1.7 GHz, RAM 8GB.

A graphical representation will clarify the results presented in Table 3. From a control perspective, it is important to be able to predict values farther in time, to activate the controller in advance. For this reason, the three different network topologies have been compared with respect to their ability to make predictions on the test sets. Finally, the predictions of the LSTM 10 seconds ahead for the two test sets are shown, as they result in the most accurate predictions (see Table 3).

Figure 8 shows the pressure and the mass flow rate of the working fluid of the test sets used to demonstrate the ability of the method proposed to accurately predict the dynamics of the system.

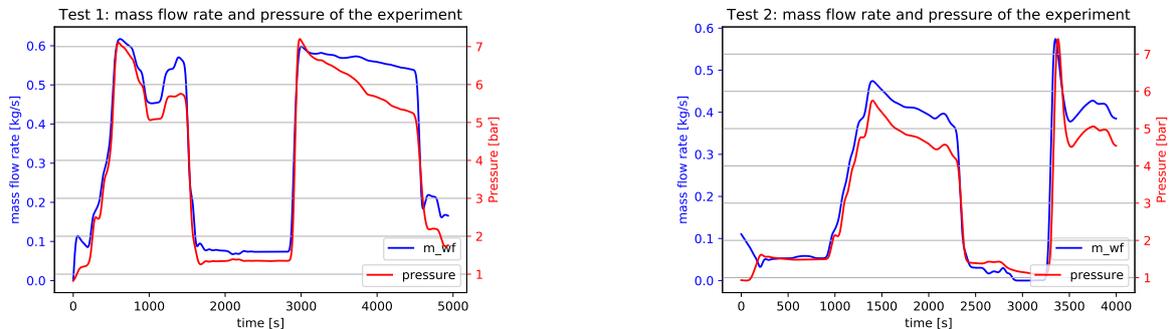


Figure 8: Mass flow rate and pressure of Test 1 and Test 2

Figure 8 shows that the mass flow rate and the pressure vary over a wide range during testings. Furthermore, it can be seen how the two values are correlated. In fact, as more mass flow rate is drawn into the system, the back pressure created by the turbine increases, causing an increase in the pressure of the system.

Figure 9 reports the prediction of the FF network 60 seconds ahead.

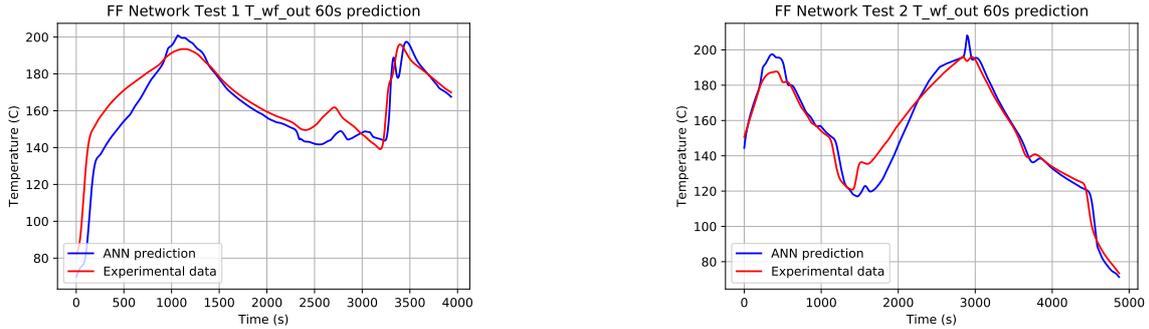


Figure 9: FF Network prediction of the turbine inlet temperature 60 seconds ahead

As shown in Figure 9, both tests have been predicted fairly well by the network. However, the predictions are not good enough to be used to activate a control system. Some delay in the prediction can be observed, particularly in those intervals of time in which the temperature fastly fluctuates over time.

Figure 10 reports the prediction of the RNN network 60 seconds ahead.

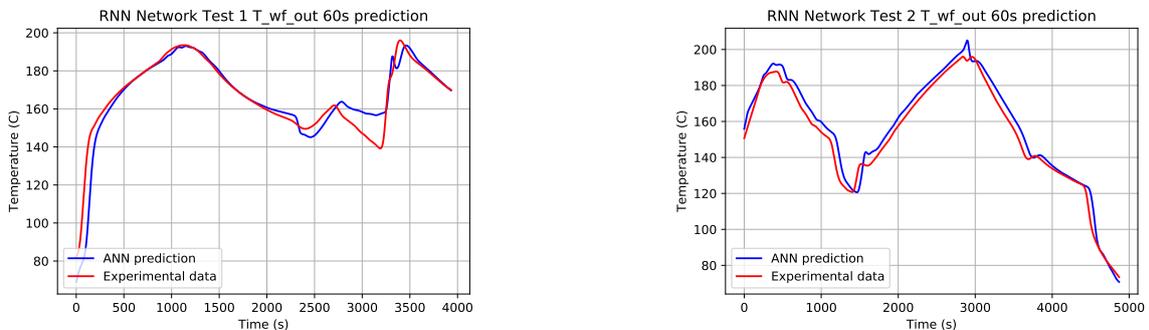


Figure 10: RNN Network prediction of the turbine inlet temperature 60 seconds ahead

From Figure 10 it is possible to notice that the RNN predicts with a higher accuracy the temperature of the working fluid at the inlet of the turbine with respect to the FF network, in accordance with the results of Table 3. In particular, regarding test 2, the prediction of the RNN in the interval of time 1000 - 2000 s, where the temperature fluctuates more, results in more accurate predictions than those of the FF network.

Figure 11 shows the prediction of the LSTM network 60 seconds ahead.

From Figure 11 it is possible to notice that the LSTM network follows accurately every dynamic present in the system, showing a slight time delay in the prediction. Although

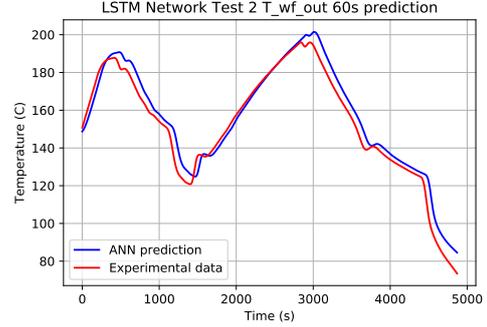
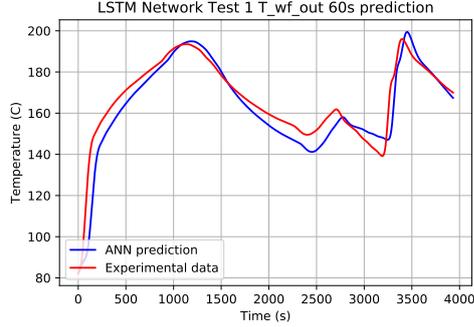


Figure 11: LSTM Network prediction of the turbine inlet temperature 60 seconds ahead

the RMSE of the RNN and of the LSTM network are comparable (see Table 3), the latter configuration follows more precisely the picks during severe fluctuations. In fact, despite a similar training and test error for the RNN and LSTM networks (see Table 3), the LSTM network's ability to keep track of information from previous predictions make it possible to accurately follow the fast dynamics of a system. Ultimately, this is the end goal of a dynamic model.

Figure 12 reports the error of the different network's configurations in the prediction of the turbine inlet temperature 60 seconds ahead.

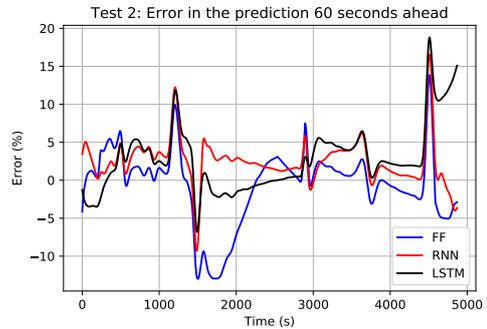
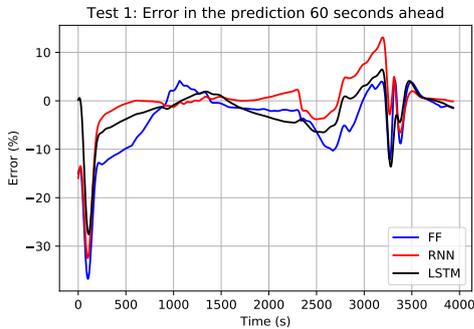


Figure 12: Percentage error of the different network configurations when predicting the temperature of the working fluid at the inlet of the turbine 60 seconds ahead

The error shown in Figure 12 is calculated using the following expression:

$$Err(\%) = \frac{Y_{ANN} - Y_{test}}{Y_{test}} * 100 \quad (8)$$

Figure 12 demonstrates that the LSTM network is the most performing in the prediction 60 seconds ahead. In fact, except for a couple of spikes per test, in which the temperature of the fluid changes sharply, the LSTM network is able to predict the temperature of the working fluid at the inlet of the turbine with an percentage error always smaller than 10 %. Furthermore, notice that the high error that the three configurations commit at the beginning

of test 1 is related to the way the error is calculated. In fact, the percentage error as defined in Eq. 8 displays higher values for lower values of the variable that is measured.

Since the prediction 60 seconds ahead of the LSTM configuration has been proved accurate enough, the behavior of the LSTM networks in predicting the turbine inlet temperature 120 seconds ahead has been also investigated . Figure 13 shows the results.

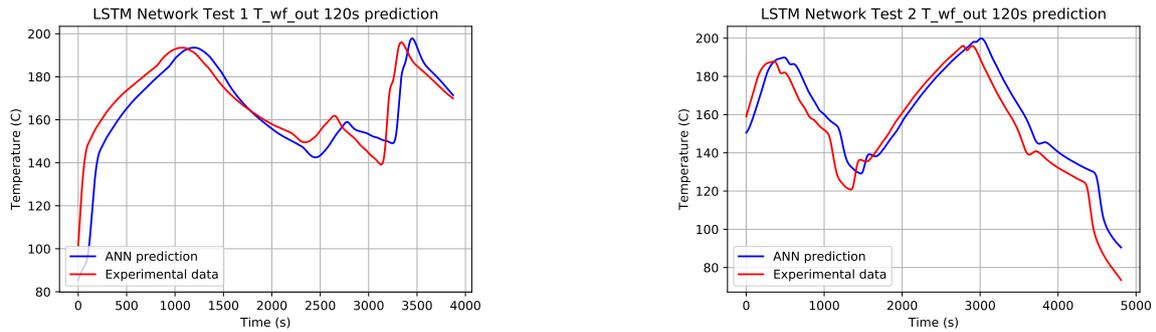


Figure 13: LSTM Network prediction of the turbine inlet temperature 120 seconds ahead

Figure 13 shows that the LSTM network provides a good prediction of the dynamics of the system 120 seconds ahead. However, the possibility to use the prediction 120 seconds ahead for control purposes needs to be further analyzed. In fact, the time delay of the prediction present in Figure 13 could result in poor control performance. For example, the prediction of the turbine inlet temperature of test 2 at time 4500 seconds presents an error of approximately 40 °C, not suitable for a proper control.

Finally, Figure 14 shows the prediction 10 seconds ahead of the LSTM network for test 1 and test 2.

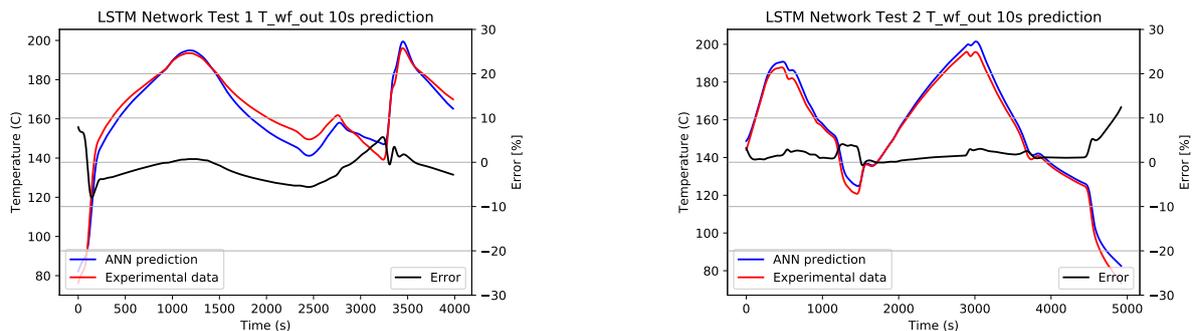


Figure 14: LSTM Network prediction of the turbine inlet temperature 10 seconds ahead

In comparison with the results obtained for predictions 120 seconds ahead (see Figure 13), Figure 14 demonstrates that the prediction of the networks 10 seconds ahead is practically in perfect accordance with the experimental data. The black line in Figure 14 demonstrates that the error that the LSTM configuration shows in predicting the temperature of the working fluid 10 seconds ahead is lower than that shown when predicting the temperature

60 seconds ahead. In fact, the prediction 10 seconds ahead shows an error always lower than 5 %, except for the last part of Test 2, where it reaches a value slightly greater than 10 %. As expected, the RMSE increases as the network tries to predict the temperature farther ahead. The RNN and LSTM networks result in comparable values for the RMSE whilst the FF network results as the worse performing among the three configurations tested, respectively for the prediction at 10 s, 60 s and 120 s.

This work demonstrates that the Neural Network approach for the prediction of the thermodynamic values during operations of an ORC system is a viable option. The results, although quite accurate, would benefit from the availability of a broader range of values to be used to train the network. In fact, the more information the neural network has on the system, the more accurate the prediction that it can achieve. Furthermore, it is worth mentioning that because of the nature of the proposed approach, one has to define a new training set anytime he wants to apply the methodology to a different ORC experimental rig. This paper demonstrate the advantage of using LSTM network's configuration over the other architectures tested for the dynamic prediction of the performance of ORC systems.

6. Conclusions

This work investigates the possibility to predict the dynamic performance of a 20 kW ORC system, using a Machine Learning approach. Different network architectures have been compared in the analysis, namely Feed Forward, Recurrent Neural Networks and Long Short Term Memory Networks.

The results obtained in this paper show that:

- the three different types of network predict well the dynamic behavior of the system;
- the quality of the results deteriorates as the time ahead of the prediction increases;
- The LSTM architecture outperforms FF and RNN networks in predicting the dynamic behavior of the 20 kW ORC system.
- The LSTM network shows a percentage error lower than 10 % in predicting the dynamics of the ORC system 60 seconds ahead;
- The LSTM network shows a percentage error lower than 5 % in predicting the dynamics of the ORC system 10 seconds ahead.

This study shows that, through the definition of a proper training set and input parameters, ANNs are able to predict the behavior of the dynamics of ORC systems, with a high grade of accuracy. This can lead to the optimization of the real time performance of the system, coupling ANNs with the optimal control theory.

Acknowledgments

This work was funded by the 2016 Scholarship of the Knowledge Center on Organic Rankine Cycle (KC ORC, www.kcorc.org), awarded to Lorenzo Tocci to work on this research project with Dr Pesyridis at Brunel University London. Entropea Labs is also acknowledged for the economic and technical support provided during the completion of this study.

Abbreviations

ANN	Artificial Neural Network
CPT	Crude Preheat Train
EU	European Union
FF	Feed Forward
ICE	Internal Combustion Engine
LSTM	Long Short Term Memory
MLT	Machine Learning Technique
MSE	Mean Square Error
NARX	Non-linear Auto Regressive network with Exogenous inputs
ORC	Organic Rankine Cycle
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
PI	Proportional-Integral
PID	Proportional-Integral-Derivative
WHR	Waste Heat Recovery

Nomenclature

b	Threshold
g	Activation Function
h	Experimental value
\hat{h}	Predicted value
\dot{m}	Mass Flow Rate [kg/s]
p	Pressure [Pa]
P	Power [kW]
σ	Sigmoid Function
\tanh	Hyperbolic Tangent
T	Temperature [°C]
x	Vector of inputs
w	Weight

Subscripts

<i>in</i>	Inlet
<i>m</i>	Number of neurons in the output layer
<i>n</i>	Number of neurons in the hidden layer
<i>t</i>	Time
<i>turb</i>	Turbine
<i>wf</i>	Working Fluid

- [1] F. Spigarelli, L. Curran, A. Arteconi, China and Europe's Partnership for a More Sustainable World: Challenges and Opportunities, Emerald Group Publishing Limited, 2016.
- [2] A. Karvountzis-Kontakiotis, A. Pesiridis, H. Zhao, F. Alshammari, B. Franchetti, I. Pasmazoglou, L. Tocci, Effect of an orc waste heat recovery system on diesel engine fuel economy for off-highway vehicles, Tech. rep., SAE Technical Paper (2017).
- [3] S. Quoilin, V. Lemort, J. Lebrun, Experimental study and modeling of an organic rankine cycle using scroll expander, Applied energy 87 (4) (2010) 1260–1268.
- [4] J. Zhang, W. Zhang, G. Hou, F. Fang, Dynamic modeling and multivariable control of organic rankine cycles in waste heat utilizing processes, Computers & Mathematics with Applications 64 (5) (2012) 908–921.
- [5] B. Twomey, P. Jacobs, H. Gurgenci, Dynamic performance estimation of small-scale solar cogeneration with an organic rankine cycle using a scroll expander, Applied Thermal Engineering 51 (1-2) (2013) 1307–1316.
- [6] A. Desideri, B. Dechesne, J. Wronski, M. Van Den Broek, S. Gusev, V. Lemort, S. Quoilin, Comparison of moving boundary and finite-volume heat exchanger models in the modelica language, energies 9 (5) (2016) 339.
- [7] D. Wei, X. Lu, Z. Lu, J. Gu, Dynamic modeling and simulation of an organic rankine cycle (orc) system for waste heat recovery, Applied Thermal Engineering 28 (10) (2008) 1216–1224.
- [8] S. Bracco, I. Faccioli, M. Troilo, A numerical discretization method for the dynamic simulation of a double-pipe heat exchanger, International Journal of Energy 1 (3) (2007) 47–58.
- [9] I. Vaja, Definition of an object oriented library for the dynamic simulation of advanced energy systems: methodologies, tools and application to combined ice-orc power plants, Ph.D. thesis, Università di Parma, Dipartimento di Ingegneria Industriale (2009).
- [10] S. Quoilin, I. Bell, A. Desideri, P. Dewallef, V. Lemort, Methods to increase the robustness of finite-volume flow models in thermodynamic systems, Energies 7 (3) (2014) 1621–1640.
- [11] C. M. Bishop, Neural networks and their applications, Review of scientific instruments 65 (6) (1994) 1803–1832.
- [12] S. A. Kalogirou, Artificial neural networks in renewable energy systems applications: a review, Renewable and sustainable energy reviews 5 (4) (2001) 373–401.
- [13] K. S. Narendra, K. Parthasarathy, Identification and control of dynamical systems using neural networks, IEEE Transactions on neural networks 1 (1) (1990) 4–27.
- [14] M. Mohanraj, S. Jayaraj, C. Muraleedharan, Applications of artificial neural networks for thermal analysis of heat exchangers—a review, International Journal of Thermal Sciences 90 (2015) 150–172.
- [15] F. Yilmaz, R. Selbaş, A. Ş. Şahin, Efficiency analysis of organic rankine cycle with internal heat exchanger using neural network, Heat and Mass Transfer 52 (2) (2016) 351–359.
- [16] A. Massimiani, L. Palagi, E. Sciubba, L. Tocci, Neural networks for small scale orc optimization, Energy Procedia 129 (2017) 34–41.
- [17] M. Rashidi, N. Galanis, F. Nazari, A. B. Parsa, L. Shamekhi, Parametric analysis and optimization of regenerative clausius and organic rankine cycles with two feedwater heaters using artificial bees colony and artificial neural network, Energy 36 (9) (2011) 5728–5740.
- [18] K. Liu, K. Li, J. Zhang, M. Lin, Modeling of organic rankine cycle for waste heat recovery using rbf

- neural networks, in: Computational Intelligence (SSCI), 2016 IEEE Symposium Series on, IEEE, 2016, pp. 1–8.
- [19] F. Yang, H. Cho, H. Zhang, J. Zhang, Y. Wu, Artificial neural network (ann) based prediction and optimization of an organic rankine cycle (orc) for diesel engine waste heat recovery, *Energy Conversion and Management* 164 (2018) 15–26.
 - [20] Entropea labs website, <http://entropea.com/> .
 - [21] C. M. Bishop, *Neural networks for pattern recognition*, Oxford university press, 1995.
 - [22] S. R. Shahamiri, S. S. B. Salim, Real-time frequency-based noise-robust automatic speech recognition using multi-nets artificial neural networks: A multi-views multi-learners approach, *Neurocomputing* 129 (2014) 199–207.
 - [23] F. Chollet, et al., Keras: Deep learning library for theano and tensorflow, URL: <https://keras.io/k>.
 - [24] G. Van Rossum, et al., Python programming language., in: *USENIX Annual Technical Conference*, Vol. 41, 2007, p. 36.
 - [25] C. Olah, Understanding lstm networks, GITHUB blog, posted on August 27 (2015) 2015.
 - [26] S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural computation* 9 (8) (1997) 1735–1780.
 - [27] I. H. Bell, J. Wronski, S. Quoilin, V. Lemort, Pure and pseudo-pure fluid thermophysical property evaluation and the open-source thermophysical property library coolprop, *Industrial & engineering chemistry research* 53 (6) (2014) 2498–2508.
 - [28] D. P. Kingma, J. L. Ba, Adam: A method for stochastic optimization, in: *Proc. 3rd Int. Conf. Learn. Representations*, 2014.