Proceedings of the International Conference on Heat Exchanger Fouling and Cleaning - 2024 (Peer-reviewed)
April 21–26, 2024, Lisbon, Portugal; Editors: H.U. Zettler and E.M. Ishiyama

PREDICTIVE FOULING DETECTION IN FOOD PRODUCTION USING MACHINE LEARNING MODELS BASED ON REAL DATA

*** J. Yin[1], N. Jarmatz[2], M. Mauermann[1], W. Augustin[2] and S. Scholl[2]**
[1] Institut for Process Engineering and Packaging IVV, Division Processing Technology, Heidelberger Str. 20, 01189 Dresden, Germany
jialiang.yin@ivv-dd.fraunhofer.de (corresponding author)
[2] Technische Universität Braunschweig, Institute for Chemical and Thermal Engineering, Langer Kamp 7, 38106 Braunschweig, Germany

ABSTRACT

Quantitative monitoring and prediction of fouling in heat exchangers could help to avoid product quality and food safety issues. Production facilities that do not have inline sensors to monitor operating conditions are prone to fouling-related problems. We introduce an approach using a non-invasive temperature acquisition method and deep learning technology to predict fouling in a heat exchanger. First, an Encoder-Decoder long short-term memory (LSTM) network is proposed to predict the output temperature of the product based on the temperature measured on the outside of a heat exchanger pipe. A random forest (RF) regression model utilizes the predicted temperature to estimate the fouling mass. The dataset used was acquired from several laboratory fouling experiments, consisting of two categories for: i) predicting product output temperature and ii) fouling estimation. The models achieve a temperature prediction accuracy of 98.5% and an approximate accuracy of 85% in estimating the fouling mass on test data from the laboratory fouling experiments.

INTRODUCTION

Heat exchangers (HE) play a pivotal role in the food industry by facilitating the transfer of thermal energy from a hot flowing stream to a cold flowing stream [1]. One of their most prevalent applications is in the pasteurization of milk. However, due to the complex composition of milk, marked by the presence of proteins and minerals, a sequence of chemical reactions takes place [2]. During the heating process, proteins are susceptible to denaturation and precipitation, resulting in the undesirable deposition of material on the heat exchanger surfaces. This leads to higher energy consumption, increased water and cleaning chemicals consumption, extended downtime for facilities and a potential risk to food safety. To guarantee operational efficiency and food safety, fouling deposits are removed by cleaning processes [3], [4]. The aim of the industry is to increase production, shorten cleaning times and ensure food safety at the same time. Methods and knowledge for quantitative monitoring and prediction of fouling could help to avoid high production costs and increase food safety [5], [6].

Given these characteristics, researchers have proposed various approaches for monitoring fouling in heat exchangers. Ingimundardóttir et al. [7] employed temperature changes for numerical fouling determination, while Riverol et al. [8] utilized pressure drop as an input for artificial neural networks (ANNs) to identify fouling instances. Chen et al. [9] conducted experiments that incorporated stainless steel electrodes within the heat exchanger to investigate the correlation between heat flux and electrical resistance. In a separate investigation, Withers et al. [10] employed an acoustic sensor to assess the thickness of the fouling layer. This method relies on the calculation of the time of flight (TOF) from the transmitter, through the pipe wall, the product, and the opposite wall, with potential fluctuations attributed to the presence of a fouling layer.

As the temperature of product output is a significant indicator for fouling detection, numerous researchers have endeavored to predict fouling based on this parameter [11], [12]. Riverol et al. [8] developed an artificial neural network (ANN) that can establish a relationship between pressure drop and fouling thickness. However, these techniques, including the ANN, are unable to predict the future development of deposit thickness. To reduce energy losses and cleaning efforts, it becomes imperative to predict deposit evolution beforehand, avoiding any adverse impact on production [13]. While some classical algorithms, such as ARIMA [14], can be employed for forecasting tasks, their accuracy often falls short of acceptability [15]. Nevertheless, with the rapid advancements in deep learning witnessed in recent times, a prediction may become achievable. Deep neural networks acquire the ability to predict by means of a sequence of nonlinear layers to create intermediate feature representations [16]. This characteristic holds great promise for time series forecasting. One of the key technologies is

ISBN: 978-0-9984188-3-4; Published online www.heatexchanger-fouling.com

Recurrent Neural Networks (RNNs) [17]. They can propagate each layer's activations sequentially across adjacent time steps. With this specific character, RNN-like networks are particularly adept at modeling sequential data, and they have proven indispensable in the field of time series analysis due to their inherent capacity for capturing temporal dependencies.

Long Short-Term Memory (LSTM) networks, a modern variant of RNNs, which has introduced many components to overcome the gradient explosion and vanishing, have made a significant impact by enhancing the modeling of long-range dependencies within time series data [17]. This advancement makes LSTM networks an essential tool for numerous time series forecasting tasks. In the realm of sequence prediction, the Encoder-Decoder LSTM is also an advanced powerful network [18]. It first encodes the input sequence into a fixed-dimensional vector, and the Decoder leverages this encoded representation to produce the desired output sequence. Convolutional Neural Networks (CNNs), traditionally used for image analysis [19], have also found their place in time series analysis [20]. By treating time series data as one-dimensional signals, CNNs have demonstrated efficacy in capturing spatial patterns within the temporal domain, a feature that holds great promise in diverse time series applications [21].

The aim of this study is to develop and validate an Encoder-Decoder LSTM network to predict the product outlet temperature of a heat exchanger. The predicted values are used by another regression algorithm that estimates the mass of fouling. Its development and validation are also presented here. The results may contribute to the development of a novel method for predicting fouling in heat exchangers using a non-invasive sensor and deep learning techniques.

In the following sections, the general strategies for time series forecasting will be outlined, independent of specific models. Subsequently, relevant deep learning network architectures will be introduced. The experimental setup for generating training data through systematic variation of process parameters will then be described. The model training process, including various hyperparameter configurations, will be explained using the generated data. Following model training, the results will be analyzed to assess the performance of the models used. Finally, the advantages and limitations of the results will be discussed.

PROBLEM EXPLAINATION

In this study, our objective is to forecast the temperature profile based on real time sequential temperature measurements. To achieve this goal, time series values from several sensors are recorded. The measurement values are organized with corresponding timestamps, transforming the problem into a time series forecasting task.

To achieve this, there are two approaches. The first one involves the reconstruction of the time series by aggregating N (N>1) values into one, sequence to one. The original and predicted values serve as input for the model to predict the next one, constituting a single-step recursive method.

However, this method exhibits reduced performance when the forecasting horizon exceeds the dimensional embedding size. In such cases, as the forecasting horizon lengthens, all subsequent values are generated by the forecasting process. Consequently, with each iteration, residuals from the Mean Squared Error (MSE) accumulate in the input sequence, rapidly diminishing prediction accuracy. Using the second approach, models can be trained to reconstruct the time series from sequence to sequence, utilizing forecasted sequence values as input for the subsequent sequence. This approach introduces some accumulated error, but it remains lower than the single-step recursive method [22]. Thus, the problem can be represented with the following diagram, and the sequence of output can be formulated using the subsequent Fig. 1.

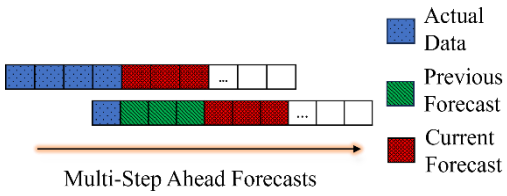

Fig. 1. Time Series Forecasting Multi-step Ahead

NETWORK ARCHITECTURES

In this work, we focus on Recurrent Neural Network (RNN). Unlike conventional feed-forward networks [23], whose activation flows only in one direction, RNNs allow the output from last step to be treated the input for next step, influencing subsequent steps [24]. The Long Short-Term Memory (LSTM), a modern RNN variant effective in handling sequential data, is applied in this work because the introduction of LSTM addressed the challenges associated with gradient vanishing and exploding problems [25]. Consequently, it enables the model to handle situations where the context is lengthy, addressing a limitation in conventional models.

In the subsequent sections, we will introduce the network architectures of LSTM, Encoder-Decoder LSTM, where Encoder and Decoder consist of LSTM cells, and Convolutional neural network LSTM (CNN-LSTM) indicating a combination of CNN and LSTM. These networks are utilized for time series forecasting, and their performances are compared in this study.

ISBN: 978-0-9984188-3-4; Published online www.heatexchanger-fouling.com

Long Short-Term Memory

As shown in Fig. 2, the Long Short-Term Memory (LSTM) architecture is composed of following fundamental components: the Forget gate F_t, Input gate I_t, Input Node C_t and Output gate O_t . Its primary function is to briefly store preceding data in "Short-Term Memory". Firstly, the initial hidden state H_{t-1} and input x_t are concatenated and processed through the Forget gate F_t, which employs a sigmoid function Eq. (1) with values ranging in [0, 1]. The gate's output is then multiplied by the preceding internal state $\tilde{C}_{t-1}$, determining whether the internal state should be considered.

$$\sigma(x) = \frac{1}{1+e^{-x}} \quad (1)$$

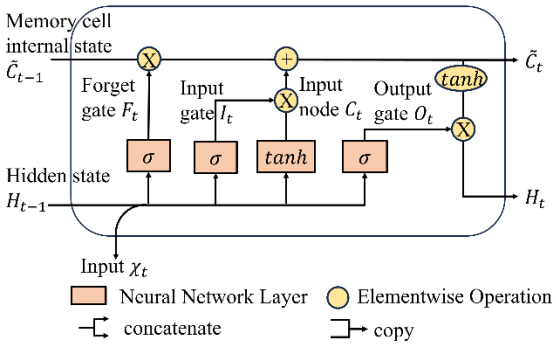

Fig. 2. Long Short-Term Memory Cell Architecture

The subsequent components involve the Input gate I_t and input node C_t. The $tanh$ function generates a new candidate internal state $\tilde{C}_t$, with I_t deciding whether to retain this new state. Once created, it is merged with the prior internal state $\tilde{C}_{t-1}$. The final internal state $\tilde{C}_t$ is confined to the range [-1, 1] and multiplied by the values from the output gate O_t. Notably, H_t serves not only as the state between steps but also as the output of this layer. This LSTM Cell is implemented in the Pytorch framework, which was mainly used in this work [26].

Encoder-Decoder LSTM

The Encoder-Decoder LSTM architecture illustrated in Fig. *3* represents an advanced variant of the traditional LSTM. The Encoder-Decoder model is a comprehensive framework for sequence-to-sequence (seq2seq) prediction, introduced in [18]. The components employed to construct the encoder and decoder can include various types of Recurrent Neural Networks (RNNs). In our work, we concentrate on the LSTM-powered Encoder-Decoder, as suggested in [18].

The restriction arises where LSTM passes only the last hidden state H_t to the next layer. In contrast, the Encoder, a sequence of LSTM cells, captures and embeds the entire sequence of information across its layers, enabling the comprehensive transfer of knowledge to the decoder, which consists also a series of LSTM cells. This intrinsic property theoretically enhances model performance, allowing for a more effective representation of sequential data and improving the overall predictive capability of the Encoder-Decoder LSTM architecture.

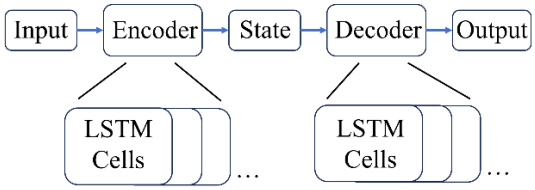

Fig. 3. Encoder-Decoder Long Short-Term Memory Architecture

Convolutional Neural Networks -LSTM

Convolutional Neural Networks (CNNs) represent a widely employed deep learning technology [27], [28]. A typical CNN architecture consists of convolutional layers, pooling layers, and fully connected layers [19]. The convolutional layer employs convolutional operations, utilizing specific masks [29] to extract features from the input data, enabling tasks such as edge detection in signals. Pooling layers play a crucial role in reducing the spatial size of the output from the preceding layer, thereby decreasing the computation, and required weights. In this work, the max pooling layer is selected to compute the maximal value within a kernel. Following this, the fully connected layer maps the extracted features to produce the final output. Despite CNNs traditionally being associated with image processing, their adaptability to 1D time series data has been explored by several researchers for forecasting tasks [30], [31]. However, CNNs are not usually adapted to correctly manage complex and long dependencies [32].

Given the intrinsic property of Long Short-Term Memory (LSTM) networks in handling time series data, a combination of CNNs and LSTMs, shown in Fig. *4*, presents a promising approach for improved performance [15], [33]. With the help of the CNN network, the time series data is filtered through a mask, before they are sent to LSTM. Such combination leverages the strengths of both architectures, enhancing the capability to find local pattern from time series data and consequently improving forecasting outcomes.

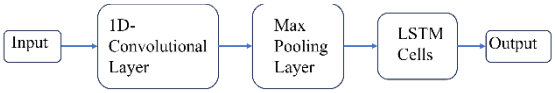

Fig. 4. CNN-LSTM Architecture

CLASSICAL REGRESSION MODELS

In this study, the product output temperature profile is forecasted using a seq2seq model. Subsequently, the profile is utilized to predict fouling mass. Despite efforts to design a comprehensive dataset, the limited number of 19

ISBN: 978-0-9984188-3-4; Published online www.heatexchanger-fouling.com

fouling experiments raises concerns about the dataset's representativeness. With only 19 samples available for model training and testing in fouling mass estimation, using small models, whose number of parameters is limited, on this small data sets is regarded as appropriate. To overcome this problem and enhance generalization and reliability, algorithms such as Support Vector Regression (SVR), linear regression (LR), and an ensemble model Random Forest (RF) [34] are considered.

SVR is a prevalent statistical learning method to commonly establish relationships between input and output variables. Its objective is to minimize prediction errors while managing the complexity of the regression function [34].

LR, another tool for establishing relationships between input and output, seeks to determine coefficients for a linear equation, while minimizing the sum of squared differences between predicted and actual values, thereby providing a straightforward method for predictions [34].

RF constructs a robust predictive model by generating multiple decision trees during training and aggregating the average prediction of individual trees. Based on decision trees, which recursively partition data into subsets according to the most significant attributes at each node, a tree-like structure is formed where leaves signify predicted outcomes. A random forest, comprising numerous decision trees, serves to mitigate overfitting and enhance generalization [34].

DATA ACQUISITION FROM FOULING EXPERIMENTS

Within the laboratory setting, a plate heat exchanger (PHE) (VT04 CD-16, GEA Ecoflex GmbH, Germany) was employed to simulate a common fouling process in dairy industry applying whey protein concentrate (WPC) as a classic substance system. The schematic representation of this facility, where the fouling classically occurs, is shown in Fig. 5. A total of 19 experiments were conducted. The clamp-on sensor type (EGT311F031, Sauter-Cumulus GmbH, Germany) selected for this work exhibited a selected measurement range set between -10 °C and 120 °C due to the parameter area of the process. The sensors were placed on the pipe surface of the heating water and product inlet as well as outlet, respectively. To determine the offset induced by the setup, inline temperature sensors (PT1000, class AA, Therma Thermofühler GmbH, Germany) were also integrated at the same position to detect the bulk fluid temperature. Also, the pressures at in- and output channels were recorded by a pressure transmitter (PA-33X, KELLER AG, Winterthur, Schweiz).

To collect the data generated by the sensors, a dedicated sensor reader application was developed and deployed a Raspberry Pi 4, a compact single-board computer. This device was chosen for its lightweight design and suitability for integration into the facility. Additionally, the inline temperature data were recorded using a commercially available data logger (Agilent 34970A, Keysight Technologies, Inc., USA).

For data acquisition, a series of experiments were carried out, each executed under various conditions: WPC concentration C_{WPC} = 70, 75, 80 g·L^{-1}, experiment duration t = 60, 120, 180 min, and input product temperature T_Bulk = 50, 55, 60 °C. In preparation for each experiment, the WPC powder was solved on a heating plate while the test rig was preheated. To start the experiments, the volume flow of the product side was set to 0.08 m^3·h^{-1} and the heating water circle to 1.508 m^3·h^{-1}. The test plates of the PHE were cleaned externally before every experiment and the dry as well as wet fouling mass was determined by a laboratory scale (Signum 3, Sartorius, Germany) after each experiment. Finally, a total of 8115 data points of temperature profiles and 19 of fouling mass were obtained. The collected data points cover several hours. As the time series was forecasted for a maximum of three hours ahead, around 400 data points were created per experiment, 8115 data points in total, which are sufficient [35]. However, since the 8115 data points are collected from only 19 fouling experiments, there are just 19 data points for fouling mass estimation. Consequently, only small models will be considered [36]. The small or simple models imply that the number of parameters is limited, making the model more interpretable. The parameters mentioned are those that make up the trained model, not the hyperparameters that can be set before model training. For example, linear regression has no hyperparameters, but the trained model consists of N (the sample size) + 1 parameters. Support Vector Regression (SVR) also has a limited number of parameters, specifically n (the feature size) + 1 + number of support vectors. In contrast, a random forest is essentially an ensemble of decision trees, and the number of parameters depends on the number of nodes in the trees. This is unlike deep learning networks, which can have millions of parameters.

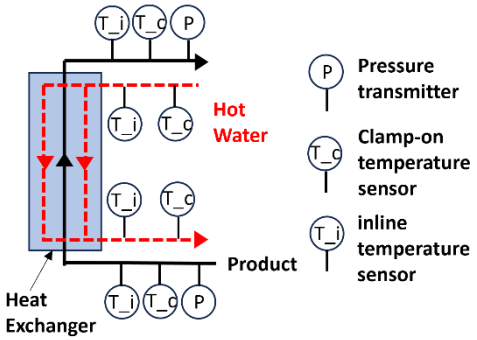

Fig. 5. Schematic representation of the experiment facility with clamp-on and inline temperature sensor

ISBN: 978-0-9984188-3-4; Published online www.heatexchanger-fouling.com

COMPREHENSIVE MODEL TRAINING AND VALIDATION PROCESS

After acquiring the dataset, was prepared for model training and validation. The following steps outline how the model training experiments are executed:

1. Preparation of the input and output data using sliding window method
2. Splitting of the prepared data into training, validation and test datasets
3. Buildup of models as described above
4. Run of the training, validation and test experiments using the selected metrics

The models LSTM, Encoder-Decoder LSTM and CNN-LSTM introduced in the Network Architectures section are adapted to the time series forecasting task and compared for their feasibility in fouling detection implemented to solve the time series forecasting task. The input and output of these models will be described using diagrams in the following section. The implementation code was composed in Python 3.10 and PyTorch [26] on a personal computer equipped with two Nvidia Quadro P5000 GPUs, which can significantly accelerate the model training process compared to CPU execution. All instantiated models underwent training for both 100 and 150 epochs, with an early stopping criterion implemented: training ceased once the validation loss fell below a predefined threshold. The optimization process employed the ADAM optimizer [37] to compute backpropagation, utilizing the batch size of 64. During the training and validation phases of the model, Mean Squared Error (MSE) loss values are recorded and stored.

Data preparation for model training and evaluation

As the formation of the fouling layer is a gradual process that occurs over time, the data are sampled down to one measurement per second. The dataset includes preparation time before and cleaning time after the experiment, which were removed manually.

$$\{y_t\}_{t=M}^{N} = F(\{x_t\}_t^M) \quad (2)$$

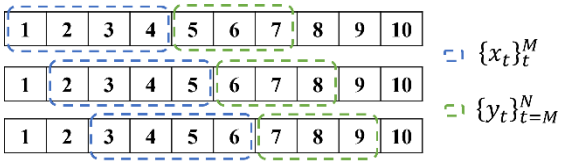

Fig. 6. Sliding Window Method

Time series forecasting is a form of multi-input and multi-output regression, involving the mapping of features through a function F to targets, as depicted in following series after x_t. In this context, the features consist of values from the last few time stamps, and the targets are the values in the near future. The collected data need to be split into three datasets for training, validation, and testing of the model. To achieve this, a sliding window algorithm was employed. Fig. *6* illustrates the concept of this method, where the blue and green rectangles with rounded corners represent a fixed-size window. The series in the blue rectangle represent input series, while those in the green rectangle stand for output series. The window advances one step at a time from the beginning to the end of the time series. Thus, inputs and outputs are extracted. For generalization of model training and validation, a random seed is varied from 0 to 2, this can randomly select train, validation, and test dataset.

Input Selection

Since the dataset consists of data from various sensors and experimental conditions, the process for selecting input for model training needs to be discussed. Five combinations of factors have been chosen as inputs for the model. Subsequent sections will enter a comparative analysis of the accuracy achieved by these five combinations. Table 1 summarizes the relevant input parameters and corresponding combinations for the model. Other parameters do not contain sufficient information. The input parameters are selected based on background knowledge, which could contribute to the model's performance.

Table 1. Combination of relevant parameters as input for model training

parameters	Combinations				
	C1	C2	C3	C4	C5
C_T_product_out	■	■	■	■	■
I_T_product_out	■	■	■	■	■
C_T_hw_p_out_diff	■	■		■	■
P_p_in_out_diff	■			■	
WPC				■	■
T_bulk				■	■

Hyperparameter Tuning

Hyperparameter tuning plays a crucial role in model building, contributing to enhanced performance. In our study, we conducted tuning on the hyperparameters as presented in

Table 2. While the input and output lengths are flexible, their sum must not exceed the minimum experiment duration. From a forecasting standpoint, the forecasting window holds significance for the prediction accuracy of the RNN model. This window represents the number of preceding time series data that the model considers for forecasting values in the near future. For this study, the input

ISBN: 978-0-9984188-3-4; Published online www.heatexchanger-fouling.com

and output windows were selected from the sets {50, 70, 100} and {50, 70, 100} respectively.

Table 2. Hyperparameter under different categories

window size	Learning parameter	LSTM constructor	CNN constructor
Input sequence length	Number of epochs	Number of hidden layers	Number of hidden layers
Output sequence length	Teacher forcing ratio	Number of neurons in hidden layers	Number of neurons in hidden layers
	Learning rate		Kernel size

Moreover, our study involves the comparison of three models, each with its unique set of hyperparameters. Despite the differences, all three models incorporate LSTM networks, sharing some common hyperparameters. In the LSTM network, the number of hidden layers and neurons in one hidden layer requires tuning. In the CNN-LSTM model, the input undergoes processing by the CNN network before entering the LSTM, necessitating the tuning of relevant parameters, such as the kernel size of convolution. To retain all features, the convolution method "padding" is employed. For the Encoder-Decoder architecture, a specific model training strategy called teacher forcing is employed. Teacher forcing aims to enhance training efficiency and stability by providing the decoder with the correct or ground truth output from the previous time step during training, with a certain probability. This approach prevents the model from accumulating errors during training, leading to improved convergence speed and stability. In this work, the teacher forcing probability is chosen from {60%, 80%} [38]. To mitigate overfitting during model training, a dropout layer is also incorporated into each model with dropout probability 0.2 and 0.4.

Evaluation Metrics

To assess the performance of the model, the following metrics are employed. Given that time series forecasting aligns with regression, the conventional metric, mean squared error (MSE) [34], is utilized during training and validation, as depicted in Eq. ($MSE= \frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2$ (*3*). During the model testing process, the mean absolute percentage error (MAPE) [39], presented in Eq. ($MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i-\hat{y}_i}{y_i}\right| \times 100\%$ (4), is employed.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i-\hat{y}_i\right)^2 \qquad (3)$$

In both equations, y_i represents the true values from the validation dataset, and $\hat{y}_i$ corresponds to the predicted values. n denotes the total number of time series. In this study, the metric – 1 - MAPE is applied as a performance indicator to enhance the illustration of model performance. The greater the resultant value, the more reliable the model performance is deemed to be. In the paper, we refer to this 1 – MAPE loss as resultant MAPE.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i-\hat{y}_i}{y_i}\right| \times 100\% \qquad (4)$$

Model selection for fouling mass estimation

For the regression models, the input parameters are WPC, Experiment duration, input product temperature and product output temperature. The target is fouling mass. Then the hyperparameter needs to be optimized. For Support Vector Regression (SVR), the critical hyperparameters include the kernel and C. The kernel, serving as a function transforming input data into a higher-dimensional space, facilitates the identification of nonlinear decision boundaries. In this context, the choices include "linear" and "rbf" ,Radial Basis Function, which computes the similarity of two points [34]. The parameter C governs the model's generalization and is selected from the set {0.1, 1, 10}. Additionally, epsilon, denoting the tolerance between actual and predicted data, is subject to variation within the range {0.01, 0.1, 0.5}.

For Random Forest, a key hyperparameter is the number of estimators, determining the quantity of decision trees within the ensemble. It assumes values from the set {50, 100, 150}. Another significant hyperparameter is the maximum depth, delineating the permissible depth of an individual tree and assuming values from the set {2, 3, 4}. The tuning of these hyperparameters is instrumental in achieving optimal model performance.

RESULTS

In this section, the performance of the proposed Encoder-Decoder model is assessed in comparison to that of LSTM and CNN-LSTM, each configured with different hyperparameters. **Error! Reference source not found.** summarizes the averaged training and validation losses during training for three models, considering different random states for the random selection of train and test datasets for cross validation. The training and validation loss curve for LSTM model barely changes during the training phase, shown in **Error! Reference source not found.**a. The MSE loss for training is around 0.05 and for validation around 0.02 is significantly larger than expected and remains relatively constant thereafter. Ideally, after each backpropagation step, the MSE loss should undergo some degree of change, ultimately leading to model convergence. However, in the case of the LSTM model, the parameters appear to remain unaltered throughout

ISBN: 978-0-9984188-3-4; Published online www.heatexchanger-fouling.com

the training process. This lack of significant change suggests that the LSTM model has not effectively learned from the training dataset.

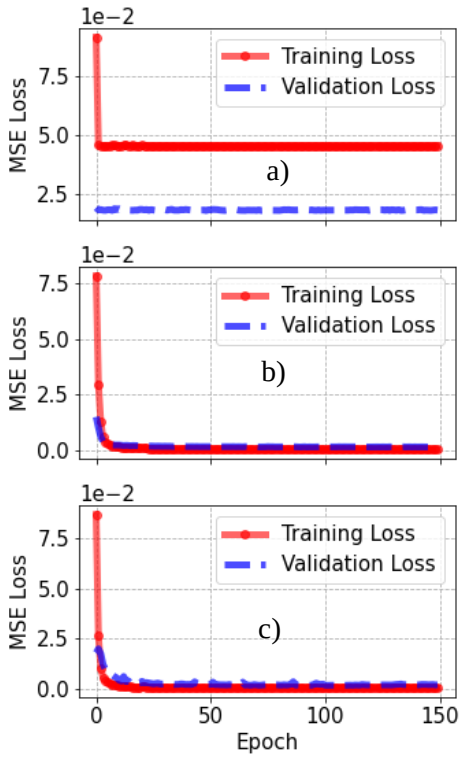

Fig. 7. Validation MSE Loss profile for the examined models: a) Validation data LSTM, b) Validation data CNN-LSTM, c) Validation data Encoder-Decoder LSTM

Comparatively, the CNN-LSTM and Encoder-Decoder LSTM models present distinct behavior in their training and validation phases **Error! Reference source not found.**b, 7c. In both models, the validation loss initiates at 0.015 and 0.02, respectively, and converges after around 70 epochs, similar to the training loss started from around 0.075 and converged to nearly 0. The CNN-LSTM model achieves stabilization within an acceptable tolerance range after approximately 50 epochs, while the Encoder-Decoder LSTM model requires nearly 80 epochs. Despite these disparate convergence epochs, both models demonstrate a validation loss of around 0.0014. It is noteworthy that, at the beginning of training, the CNN-LSTM model exhibits a lower initial loss compared to the Encoder-Decoder LSTM model. Additionally, the validation loss curve of the CNN-LSTM model appears smoother than that of the Encoder-Decoder LSTM model. This behavior can be attributed to the property of CNN, where time series features are extracted using a mask, enabling earlier convergence than the Encoder-Decoder LSTM model. Additionally, the training loss variation in the CNN-LSTM model is smaller than that in the Encoder-Decoder LSTM model.

Fig. *8* represents the test losses**Error! Reference source not found.**, showcasing a box plot resultant MAPE for three models. The LSTM model displays the smallest interquartile range, indicating consistent behavior attributed to stable parameters throughout training. CNN-LSTM and Encoder-Decoder LSTM models have mean accuracies of 98.53% and 98.68% respectively. The interquartile range for CNN-LSTM spans 97.97% to 99.27%, wider than Encoder-Decoder LSTM's range of 98.41% to 99.17%. Notably, the narrower range for Encoder-Decoder LSTM and higher mean accuracy suggest greater stability and precision of prediction in test results compared to CNN-LSTM.

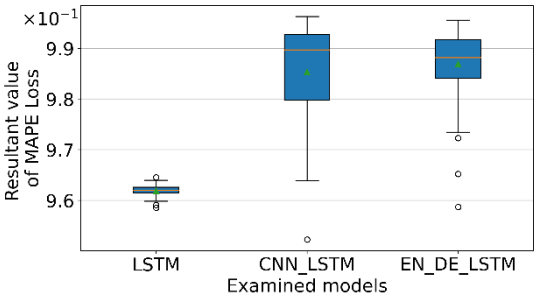

Fig. 8. Resultant value of MAPE Loss by exanimated modes by testing

To provide a clearer view of the performance of different models, a test dataset was selected and applied to the trained models to forecast the product outlet temperature based on this dataset. The results are shown in Fig. 9. As observed, the pure LSTM model's results remain constant, similar to what is shown in **Error! Reference source not found.**, where the MSE Loss does not converge. It leads to the poor performance in predicting sequence data. In contrast, the Encoder-Decoder and CNN LSTM models perform quiet well. Both models can forecast the decreasing trend of the product outlet temperature, with the maximum predict error remaining below 1°C.

As mentioned before, the impact of input selection on the model performance was also investigated, exploring whether additional experimental parameters enhance efficacy. Fig. 10 illustrates the analysis of five data combinations, highlighting suboptimal performance in c4 and c5 due to the constant nature of WPC and input product temperature. These parameters, resembling initial conditions, contribute minimally and even detract slightly from model performance. In contrast, model with c1, c2 and c3 as input have achieved better performance. c1, incorporating both temperature and pressure, contributed minimally to model training due to the limited variability in pressure difference compared to temperature. In contrast, c2 emerged as the top performer with a mean accuracy of 98.14% and superior stability, attributed to its inclusion of C_T_product_out, I_T_product_out, and C_T_hw_p_out_diff. The c3 combination,

ISBN: 978-0-9984188-3-4; Published online www.heatexchanger-fouling.com

featuring only C_T_product_out and I_T_product_out, displayed less favorable performance, lacking precision in the linear relationship between clamp-on and inline sensors. The optimal performance achieved by c2 can be attributed to the similarity in input parameters, enabling the model to learn more effectively from the three time series datasets. The data indicates that the C_T_product_out and I_T_product_out variables have the greatest influence on model performance. It would also be valuable to exclude the I_T_product_out sensor data from the model training process to evaluate how the trained model performs without this input. In contrast, c3 lacks the inclusion of difference values between hot water and product output temperature, resulting in the model missing important information. Moreover, the input parameter of pressure used in the c1 combination exhibits a significantly different data distribution from temperature, leading to a decrease in performance.

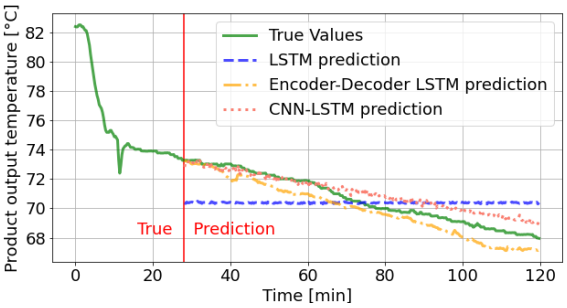

Fig. 9 Comparison of predicted values from different models

The CNN-LSTM model requires about 15 minutes to train, while the Encoder-Decoder LSTM completes training in only 10.46 minutes, which is nearly 1.5 times faster. The extended time consumption of the CNN-LSTM model may be attributed to the feature selection performed by the CNN. The CNN processes the entire dataset to extract meaningful features, which are subsequently passed to the LSTM cells.

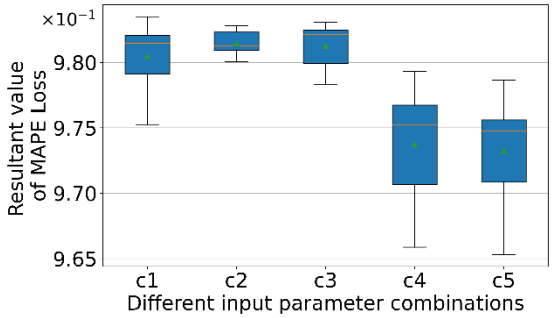

Fig. 10. Comparison of resultant MAPE Loss by different input parameter combinations as indicated in Table 1

For fouling mass estimation, classical regression models were used. Variability was introduced in the random seed during training for improved result generalizability. The results shown in Fig. *11* demonstrate that SVR achieved the lowest MAPE at 80.78%, while LR and RF achieved comparable accuracies of 83.06% and 83.48%, respectively. SVR encounters challenges due to the inefficiency of its hyperplane in separating high-dimensional data points. RF outperforms LR by 0.42%, leveraging a set of estimators for enhanced input-output relation representation. Despite RF's longer training time (50 times more), its predictive duration becomes less important once the model is built. LR boasts an average prediction time of 0.3ms, while RF requires 9ms, 30 times longer but still within acceptable timeframe. After model selection, optimal hyperparameters involve a maximal tree depth of 3 and 50 estimators, resulting in an 85.12% prediction accuracy.

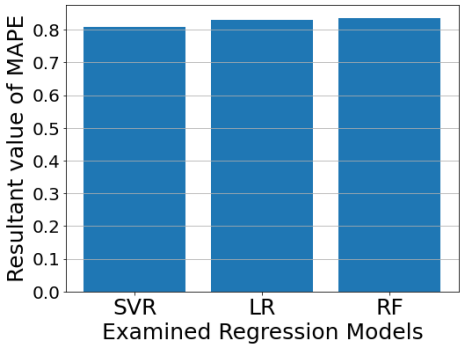

Fig. 11 Resultant value of MAPE Loss by exanimated regression modes by testing

CONCLUSION

This study introduces a new method for monitoring product output temperatures, as well as for forecasting product output temperature profiles and fouling mass. To compare both forecasting and regression models, the model training experiments using cross-validation was conducted to obtain a more accurate estimate of each model's performance on test data. The approach utilizes clamp-on temperature sensors and an Encoder-Decoder LSTM model trained on laboratory fouling experiment data. The results received from laboratory test data demonstrated an accuracy of 98.5% in predicting the product output temperature. The accuracy of the fouling mass estimate using a random forest regression model reaches 85 % for test data.

It is evident from the results that the LSTM model exhibits the lowest accuracy, which indicates its unsuitability for multi-step ahead predictions. Both the Encoder-Decoder LSTM and CNN-LSTM, however, demonstrate accurate predictions. Considering model robustness and accuracy, the Encoder-Decoder LSTM is the optimal choice among the alternatives. It is important to note that the performance of the model is dependent on the

ISBN: 978-0-9984188-3-4; Published online www.heatexchanger-fouling.com

dataset. Therefore, it is not practical to completely disregard CNN-LSTM. Despite CNN-LSTM requiring 1.5 times longer training than the Encoder-Decoder LSTM, CNN-LSTM converges rapidly, allowing for training with fewer epochs while observing changes in prediction accuracy. Among the methods tested, the RF regression model is the most accurate method for fouling estimation due to its composition of numerous sub-estimators. Although LR and SVR achieve over 80% accuracy, their limited ability to capture complex input-output relationships may hinder significant long-term improvement. The maximal size of constructed Encoder-Decoder LSTMs is only 460 KB, which refers to the storage space required to load the model. With a such small size, the model can be deployed on resource-constrained IoT devices.

The method presented has limited suitability for direct application in industry due to initial training based on limited data from laboratory fouling experiments. However, in the next step, authentic industrial data from a food manufacturer will be used to validate the reliability of the model prediction in production processes. This industrial dataset, including parameters like product as well as hot water temperatures, and pump speed, requires an expanded hyperparameter tuning range to adapt the intricacies of industrial data. This transition allows for a comprehensive examination of the proposed network's reliability in production settings. Recognizing the anticipated differences between laboratory and industrial data - particularly in maintaining constant product output temperatures dictated by food safety policies - the essential next step involves integrating real-time data from the process control system. This can enhance the model's predictive capabilities, aligning it more closely with the dynamic demands of industrial production environments.

To make the model adaptive, the strategy like the attention mechanism [40] could be used. This mechanism empowers models to selectively analyze relevant data segments of input sequences. This property allows for the selection of particular segments during training, enabling the model to address especially initial timeframes in experiments where the product output temperature remains constant, and to emphasize the temperature drop phase.

NOMENCLATURE

F_t Forget gate in LSTM Cell, symbol for function
e Euler's number, dimensionlessI_t Input gate in LSTM Cell, symbol for function
C_t Input node
$\tilde{C}_t$ Cell interval state in LSTM Cell
H_t Hidden state in LSTM Cell
O_t Output gate
σ Symbol for sigmoid function
T Temperature, °C
P Pressure, bar
C_T_product_out Temperature of product outlet from clamp-on sensor
I_T_product_out Temperature of product outlet from inline sensor
C_T_hw_p_out_diff Temperature difference between product and hot water outlet from clamp-on sensor
P_p_in_out_diff Pressure difference between product inlet and outlet from inline sensor
C_{WPC} Whey protein concentration
T_Bulk Input product temperature

Subscript

t an arbitrary point in time in time series data
M timestamp, where the input for model ends
N timestamp, where the output for model ends

REFERENCES

[1] S. Kakac and H. Liu, *Heat Exchangers Selection, Rating, and Thermal Design*, Second Edition (2nd ed.). CRC Press, 2002. [Online]. Available: https://doi.org/10.1201/9781420053746

[2] B. Bansal and X. D. Chen, 'FOULING OF HEAT EXCHANGERS BY DAIRY FLUIDS – A REVIEW', *Heat Exch Fouling Clean Challenges Opportunities*, vol. RP2, no. 23, 2005.

[3] S. Gottschall, R. Murcek, S. Städtler, and M. Mauermann, 'System for automated monitoring of local soil removal during cleaning in closed food processing lines with a quartz crystal sensor', *Heat Mass Transfer*, Jul. 2023, doi: 10.1007/s00231-023-03389-1.

[4] R. Murcek, J. Hölzel, H. Köhler, A. Boye, M. Hesse, and M. Mauermann, 'Development of a quartz crystal sensor system to monitor local soil removal during cleaning in closed food processing lines', *Food and Bioproducts Processing*, vol. 127, pp. 282–287, May 2021, doi: 10.1016/j.fbp.2021.03.011.

[5] M. Joppa, T. Hanisch, and M. Mauermann, 'Methodology for the assessment of cleanability and geometry optimization using flow simulation on the example of dimple-structured pipe surfaces', *Food and Bioproducts Processing*, vol. 132, pp. 141–154, Mar. 2022, doi: 10.1016/j.fbp.2021.12.004.

[6] T. Hanisch, M. Joppa, V. Eisenrauch, S. Jacob, and M. Mauermann, 'Optimizing the macrostructure of 3D-printed pipe surfaces to improve cleanability', *Heat Mass Transfer*, Jul. 2023, doi: 10.1007/s00231-023-03387-3.

[7] H. Ingimundardóttir and S. Lalot, 'Detection of Fouling in a Cross-Flow Heat Exchanger

ISBN: 978-0-9984188-3-4; Published online www.heatexchanger-fouling.com

Using Wavelets', *Heat Transfer Engineering*, vol. 32, no. 3–4, pp. 349–357, Mar. 2011, doi: 10.1080/01457632.2010.495668.
[8] C. Riverol and V. Napolitano, 'Estimation of fouling in a plate heat exchanger through the application of neural networks', *Journal of Chemical Technology & Biotechnology*, vol. 80, no. 5, pp. 594–600, 2005, doi: 10.1002/jctb.1198.
[9] X. D. Chen, D. X. Y. Li, S. X. Q. Lin, and N. Özkan, 'On-line fouling/cleaning detection by measuring electric resistance—equipment development and application to milk fouling detection and chemical cleaning monitoring', *Journal of Food Engineering*, vol. 61, no. 2, pp. 181–189, Feb. 2004, doi: 10.1016/S0260-8774(03)00085-2.
[10] P. Withers, 'Ultrasonic sensor for the detection of fouling in UHT processing plants', *Food Control*, vol. 5, no. 2, pp. 67–72, doi: 10.1016/0956-7135(94)90088-4.
[11] W. Al Hadad, V. Schick, and D. Maillet, 'Fouling detection in a shell and tube heat exchanger using variation of its thermal impulse responses: Methodological approach and numerical verification', *Applied Thermal Engineering*, vol. 155, pp. 612–619, Jun. 2019, doi: 10.1016/j.applthermaleng.2019.04.030.
[12] J. W. Choi *et al.*, 'Temperature Difference-Based Fouling Detection in the Heat Exchanger of Gas-Solid Fluidized Beds', *Chem Eng & Technol*, vol. 45, no. 9, pp. 1623–1630, Sep. 2022, doi: 10.1002/ceat.202200188.
[13] S. Sundar *et al.*, 'Fouling modeling and prediction approach for heat exchangers using deep learning', *International Journal of Heat and Mass Transfer*, vol. 159, p. 120112, Oct. 2020, doi: 10.1016/j.ijheatmasstransfer.2020.120112.
[14] S. L. Ho, 'The use of ARIMA models for reliability forecasting and analysis', *Computers & Industrial Engineering*, vol. 35, no. 1, pp. 213–216, doi: https://doi.org/10.1016/S0360-8352(98)00066-7.
[15] I. E. Livieris, E. Pintelas, and P. Pintelas, 'A CNN–LSTM model for gold price time-series forecasting', *Neural Comput & Applic*, vol. 32, no. 23, pp. 17351–17360, Dec. 2020, doi: 10.1007/s00521-020-04867-x.
[16] B. Lim and S. Zohren, 'Time-series forecasting with deep learning: a survey', *Phil. Trans. R. Soc. A.*, vol. 379, no. 2194, p. 20200209, Apr. 2021, doi: 10.1098/rsta.2020.0209.
[17] A. Sherstinsky, 'Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network', *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, Mar. 2020, doi: 10.1016/j.physd.2019.132306.
[18] I. Sutskever, O. Vinyals, and Q. V. Le, 'Sequence to Sequence Learning with Neural Networks', *Advances in neural information processing systems*, vol. 27, 2014, [Online]. Available: https://doi.org/10.48550/arXiv.1409.3215
[19] K. O'Shea and R. Nash, 'An Introduction to Convolutional Neural Networks', Dec. 02, 2015, *arXiv*: arXiv:1511.08458. Accessed: Nov. 02, 2023. [Online]. Available: http://arxiv.org/abs/1511.08458
[20] M. Binkowski, G. Marti, and P. Donnat, 'Autoregressive Convolutional Neural Networks for Asynchronous Time Series', in *International Conference on Machine Learning*, 2018, pp. 580--589. [Online]. Available: https://doi.org/10.48550/arXiv.1703.04122
[21] R. Velastegui, L. Zhinin-Vera, G. E. Pilliza, and O. Chang, 'Time Series Prediction by Using Convolutional Neural Networks', in *Proceedings of the Future Technologies Conference (FTC) 2020, Volume 1*, vol. 1288, K. Arai, S. Kapoor, and R. Bhatia, Eds., in Advances in Intelligent Systems and Computing, vol. 1288. , Cham: Springer International Publishing, 2021, pp. 499–511. doi: 10.1007/978-3-030-63128-4_38.
[22] S. Suradhaniwar, S. Kar, S. S. Durbha, and A. Jagarlapudi, 'Time Series Forecasting of Univariate Agrometeorological Data: A Comparative Performance Evaluation via One-Step and Multi-Step Ahead Forecasting Strategies', *Sensors*, vol. 21, no. 7, p. 2430, Apr. 2021, doi: 10.3390/s21072430.
[23] P. Sharma, N. Malik, and N. Akhtar, 'FEEDFORWARD NEURAL NETWORK: A Review', *International Journal of Advanced Research in Engineering and Applied Sciences (IJAREAS)*, vol. 2, no. 10, pp. 25–34, 2013.
[24] G. Aurelien, *Neural networks and deep learning*. Beijing: O'Reilly Media, Inc., 2018.
[25] S. Hochreiter and J. Schmidhuber, 'Long Short-Term Memory', *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
[26] A. Paszke *et al.*, 'PyTorch: An Imperative Style, High-Performance Deep Learning Library', in *Advances in Neural Information Processing Systems 32*, Curran Associates, Inc., 2019, pp. 8024--8035. Accessed: Aug. 08, 2024. [Online]. Available: http://arxiv.org/abs/1912.01703
[27] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, 'Object Detection With Deep Learning: A Review', *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no.

ISBN: 978-0-9984188-3-4; Published online www.heatexchanger-fouling.com

11, pp. 3212–3232, Nov. 2019, doi: 10.1109/TNNLS.2018.2876865.
[28] Y. Xiong *et al.*, 'MobileDets: Searching for Object Detection Architectures for Mobile Accelerators', *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Mar. 2021, doi: 10.1109/CVPR46437.2021.00382.
[29] T. Bose and F. Meyer, *Digital Signal and Image Processing | Guide books*. USA: John Wiley & Sons, Inc., 2003. Accessed: Nov. 25, 2023. [Online]. Available: https://dl.acm.org/doi/abs/10.5555/861362
[30] A. P. Wibawa, A. B. P. Utama, H. Elmunsyah, U. Pujianto, F. A. Dwiyanto, and L. Hernandez, 'Time-series analysis with smoothed Convolutional Neural Network', *J Big Data*, vol. 9, no. 1, p. 44, Dec. 2022, doi: 10.1186/s40537-022-00599-y.
[31] B. Yang, Z.-J. Gong, and W. Yang, 'Stock market index prediction using deep neural network ensemble', in *2017 36th Chinese Control Conference (CCC)*, Dalian, China: IEEE, Jul. 2017, pp. 3882–3887. doi: 10.23919/ChiCC.2017.8027964.
[32] Y. Bengio, A. Courville, and P. Vincent, 'Representation Learning: A Review and New Perspectives', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013, doi: 10.1109/TPAMI.2013.50.
[33] W. Lu, J. Li, Y. Li, A. Sun, and J. Wang, 'A CNN-LSTM-Based Model to Forecast Stock Prices', *Complexity*, vol. 2020, pp. 1–10, Nov. 2020, doi: 10.1155/2020/6622927.
[34] R. Tibshirani, J. H. Friedman, and T. Hastie, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer. [Online]. Available: https://books.google.de/books?id=eBSgoAEACAAJ
[35] Hanke John E. and Dean W. Wichern, in *Business Forecasting*, Pearson Education, 2014, p. 80.
[36] P. Xu, X. Ji, M. Li, and W. Lu, 'Small data machine learning in materials science', *npj Comput Mater*, vol. 9, no. 1, p. 42, Mar. 2023, doi: 10.1038/s41524-023-01000-z.
[37] D. P. Kingma and J. Ba, 'Adam: A Method for Stochastic Optimization', *arXiv:1412.6980 [cs]*, Dec. 2014, Accessed: Sep. 18, 2017. [Online]. Available: http://arxiv.org/abs/1412.6980
[38] R. J. Williams and D. Zipser, 'A Learning Algorithm for Continually Running Fully Recurrent Neural Networks', *Neural Computation*, vol. 1, no. 2, pp. 270–280, Jun. 1989, doi: 10.1162/neco.1989.1.2.270.
[39] R. Madhu P.K., J. Subbaiah, and K. Krithivasan, 'RF-LSTM-based method for prediction and diagnosis of fouling in heat exchanger', *Asia-Pac J Chem Eng*, vol. 16, no. 5, Sep. 2021, doi: 10.1002/apj.2684.
[40] A. Vaswani *et al.*, 'Attention Is All You Need', Aug. 01, 2023, *arXiv*: arXiv:1706.03762. Accessed: Jan. 22, 2024. [Online]. Available: http://arxiv.org/abs/1706.03762

ISBN: 978-0-9984188-3-4; Published online www.heatexchanger-fouling.com