Predicting Surface Heat Flux on Complex Systems via Conv-LSTM

Yinpeng Wang  
School of Electronic and Information  
Beihang University  
Beijing, 100191  
16081072@buaa.edu.cn

Nianru Wang  
School of Electronic and Information  
Beihang University  
Beijing, 100191  
17373024@buaa.edu.cn

Qiang Ren  
School of Electronic and Information  
Beihang University  
Beijing, 100191  
qiangren@buaa.edu.cn

July 7, 2021

ABSTRACT

Existing algorithms with iterations as the principle for 3D inverse heat conduction problems (IHCPs) are usually time-consuming. With the recent advancements in deep learning techniques, it is possible to apply the neural network to compute IHCPs. In this paper, a new framework based on Convolutional-LSTM is introduced to predict the transient heat flux via measured temperature. The inverse heat conduction models concerned in this work have 3D complex structures with non-linear boundary conditions and thermophysical parameters. In order to reach high precision, a forward solver based on the finite element method is utilized to generate sufficient data for training. The fully trained framework can provide accurate predictions efficiently once the measured temperature and models are acquired. It is believed that the proposed framework offers a new pattern for real-time heat flux inversion.

1 Introduction

Real-time identifications of the heat flux distribution at a given domain are essential in many fields like engine design, geological prospecting and so on. Direct ways of heat flux measurements to the external surface of a hypersonic aircraft are expensive and difficult to maintain a high accuracy due to the severe condition [1]. As a result, there is a strong demand to develop a more precise and more efficient measurement technique for determining the heat flux distribution mediately. One reasonable approach is to take advantage of the numerous temperature measurement on the interior surface of the capsule to invert the heat flux on the exterior surface. Measuring the temperature is more economical and robust than directly measuring the heat flux. Hence, we need to solve the inverse heat conduction problems (IHCPs) to acquire the anticipated heat flux.

An IHCP generally refers to the problems in identifying the unknown boundary or the thermal properties in a thermal system. As this problem is super sensitive to the measurement variation, it is mathematically ill-posed [2]. Therefore, many valuable techniques have been proposed to transform it into a relatively well-posed problem. Traditional algorithms used in the field of IHCPs include sequential function specification method (SFSM) [3], iterative techniques [4,5,6,7,8], Tikhonov regularization [9,10], singular value decomposition with model-reduction [11], digital filter method [1,12,13], genetic algorithm [14,15], Bayesian framework [16], neural networks [17,18,19,20,21,22,23] and so on.
Many of the inverse techniques mentioned above perform well in heat conduct systems with simple geometries (1D or 2D models) and linear thermal parameters. However, in many practical cases, the problems are usually more sophisticated. Some works about those IHCPs with nonlinearity have been published. A regular way is to divide the nonlinear model into several linearized sub models. Keanini et al. [24] proposed a method based on the non-iterative finite element algorithms and successfully reconstruct the time-varying heat flux. The sequential function specification algorithm was adopted in the study of Huang et al. [10] predict the spatial and temporal related heat flux distribution by supposing that changes are linear at every time step while the sensitivity coefficients were obtained by finite element method calculations. Lv et al. [25] proposed a multiple model adaptive inverse method to forecast the heat flux condition of a nonlinear IHCP. In fact, the accuracy of sub-model algorithms relies on how many sub-models are constructed. It is hard for these techniques to solve IHCPs with severe nonlinearity, as they should use a considerable amount of sub-models and may cost unaffordable computational resources. A novel approximate Bayesian computation algorithm is proposed by Zeng et al. [16] for both linear and non-linear systems, while massive samples are needed. Therefore, it is hard for this method to solve problems within seconds. Duda et al. [26, 27], efficiently solved the nonlinear IHCPs by dividing a complex domain into sample parts. However, the cases are limited to 2D models and the number of sensors needs to be specified in different cases.

In industrial applications, the nonlinearity of the IHCP are more severe due to the following reasons: (1) The geometry of the model can be three-dimensional with intricate shapes and curved surfaces. In most practical cases, it is not appropriate to simplify the model into 1D or 2D models as it will lead to unbearable errors. (2) The nonlinearity of the model should be considered, especially the sophisticated radiation boundary condition and temperature-dependent materials. (3) The heat flux computation should be efficient, especially in a long-period inversion process. In such a case, the iteration-based methods with finite element methods solving forward problem [5, 7] are improper because of their unbearable time cost. In summary, handling IHCPs involving all the above aspects are much more meaningful but more complicated. Therefore, an efficient inverse algorithm is proposed to solve IHCPs with those characteristics.

The neural network has two advantages compared to the traditional method. One is that a fully trained framework spends less time to forecast the heat flux. The other is that the neural network can simulate the nonlinearity of the IHCP better than the traditional methods. Most of the studies have applied the artificial neural network (ANN) to solve IHCPs and used a backpropagation algorithm to optimize the framework. Ghadimi et al. [19] combined the Sequential Function Specification method and ANNs to handle the heat flux inversion of a locomotive brake disc. Via two layers of the fully connected network, he succeeded to forecast the linearized heat flux which is homogeneously distributed. The temporal-related heat flux is also predicted via perturbed temperature by Cortés et al. [20]. They also used the fully connected network to capture the temporal features of the IHCP and simplified the question by assuming that the heat flux is homogenously located. However, in real industrial scenarios, it is impractical to utilize fully connected networks to solve complex IHCPs since the trainable parameters are too large. Thanks to the development of the research on deep learning, some special neural networks have been proposed. The long short-term memory networks (LSTM) proposed by Hochreite et al. [23] is an efficient framework handling these temporal-related systems. LSTM has been proven to be stable and powerful for modeling long-range dependencies in various precious studies as a special RNN structure [29, 30, 31]. One superiority of employing the memory cell and gates to bridle information flow is that the gradient will be saved in the cell and be protected from fading too quickly while this is an essential problem for the vanilla RNN model [29, 30]. Multiple LSTMs can be coded and temporally linked to propose more complex structures, which have already been applied to solve many real-time sequence modeling problems [32]. However, such a network can only accept 1D tensors, so it could only solve problems that contain information about time. Very recently, a new method is proposed by Shi et al. [33] for precipitation nowcasting which modifies the common LSTM and puts the convolution into the layer to change the input tensor and adds the spatial information. Hence, the convolutional LSTM (ConvLSTM) is introduced as the base of our framework.

This paper is organized as follows. The second section firstly introduces the physical model of the IHCP to be investigated. Then the workflow and the method to generate the data set are illustrated. Furthermore, the architecture of the framework is also exhibited in detail. The third section displays and analyzes the outcome of the proposed framework. A brief conclusion is provided in the last section.

2 Methodology and Formulation

The whole workflow of the inversion process is demonstrated in Fig. 1 Here, we first generate various geometries in MATLAB and combine the thermal properties with the geometries to produce the physical models. Then, the FEM solver utilizes the physical models with the heat flux generated by MATLAB to calculate the temperature distribution. After that, the generated geometries and temperatures in each time step are packed together to serve as the input of the framework. The framework is then optimized in the training process where the output is supervised by the generated heat flux.
Table 1: Nomenclature

| Heat conduction | Coefficient matrix |
|-----------------|--------------------|
| $\Omega$        | Solid body domain  |
| $r$             | Space coordinate   |
| $S_{1,2,3}$     | Boundary surface   |
| $T_0$           | Initial temperature|
| $t$             | Time               |
| $q$             | Heat flux          |
| $T_c$           | Measured temperature|
| $k(T)$          | Thermal conductivity|
| $C_p(T)$        | Heat capacity      |
| $\rho$          | Mass density       |
| $\sigma$        | Stefan Boltzmann constant |
| $\varepsilon$   | Emissivity         |
| ConvLSTM        | Inputs of the ConvLSTM |
| $\chi_1, \ldots, \chi_t$ | Cell outputs of the ConvLSTM |
| $C_1, \ldots, C_t$ | Hidden states of the ConvLSTM |
| $H_1, \ldots, H_t$ | Inside gates of the ConvLSTM |
| $W$             | Coefficient of the ConvLSTM |
| Subscript       | Sensor index       |
| $j$             | Temperature on the radiation boundary |
| $w$             | Input index        |
| $x$             | Gates index        |
| $i, f, o$       | Hidden states index|
| $H$             | Cell output index  |
| $C$             |                     |

Training

Supervised

TensorFlow

Data Generation

Heat Flux

Generation

FEM Solver

Temperature

Geometry

Generation

Thermal Properties

Assigned

Figure 1: The workflow to train and test the framework.

2.1 Nonlinear heat conduction model

The developed method aims to forecast a time-varying heat flux distribution of a nonlinear 3D heat conduction model with temperature-dependent heat conduction parameters. Fig. 2 displays the schematic of a typical example of such a model, where the solid body is a isotropic homogeneous domain denoted as $\Omega$. $S_1$ is a curved surface where the unknown temporal and spatial varying heat flux $q(r, t)$ is imposed. The bottom surface is designated as $S_2$ and the physical boundary is radiation boundary conditions. The rest of the surfaces are insulated and symbolized as $S_3$. The temperature sensors are uniformly scattered on the temperature surface $S_2$ which are denoted as $T_j$ ($1 \leq j \leq I$), where $I$ is the total number of the sensors.

The governing equations, related boundaries and initial conditions of the system are

$$k\nabla^2 T = \rho C_p \frac{\partial T}{\partial t}, r \in \Omega, t \geq 0$$  \hspace{1cm} (1)

$$-k \frac{\partial T(r; t)}{\partial z} = q(r; t), r \in S_1, t \geq 0$$  \hspace{1cm} (2)

$$-k \frac{\partial T(r; t)}{\partial z} = \sigma \varepsilon (T^4_w(r; t) - T^4_0), r \in S_2, t \geq 0$$  \hspace{1cm} (3)
Figure 2: A typical example of the 3D nonlinear heat conduction systems.

\[-k \frac{\partial T(r; t)}{\partial n} = 0, r \in S_3, t \geq 0\]  \hspace{1cm} (4)

\[T(r; 0) = T_0, r \in \Omega, t = 0\] \hspace{1cm} (5)

where \(T(r; t)\) represents the temperature and \(\rho\) designates the mass density of the solid body. Temperature-dependent heat capacity and thermal conductivity are designated as \(C_p(T)\) and \(k(T)\). And \(\varepsilon\) is the emissivity while \(\sigma\) is the Stefan Boltzmann Constant.

The physic boundary settings, the detailed geometry parameters and thermal properties expressed in Table 1 will be the foundation of the test case. Fig. 3 shows the temperature-dependent thermal conductivity \(k(T)\) and heat capacity \(C_p(T)\) of an alloy used on the aircraft. Both the initial temperature of the solid body and ambient temperature are 273.15K. 900 temperature sensors are uniformly located on the bottom surface \(S_2\).

Figure 3: The temperature-dependent thermal conductivity and heat capacity.

\[2.2 \text{ Data set generation}\]

After constructing the 3D nonlinear model, we evolve COMSOL based on the finite element method to generate sufficient data for training and testing. The detailed procedure of this algorithm is as follows: Firstly, the 3D model is divided into \(E\) sub grids, each of which contains \(p\) nodes. The temperature at any given position \((x, y, z)\) can be expressed approximately by the interpolation of node temperature.

\[T = \sum_{i=1}^{P} N_i(x, y, z)T_i^e = [N]\{T\}^e\] \hspace{1cm} (6)

where \(N_i\) is the weight of the \(i_{th}\) node. Applying the functional theory, the heat conduction equation can be rewritten as

\[\begin{bmatrix} K_1^e \end{bmatrix}\{T\}^e + \begin{bmatrix} K_2^e \end{bmatrix}\{T\}^e + \begin{bmatrix} K_3^e \end{bmatrix}\{\dot{T}\}^e - \{P\}^e = 0\] \hspace{1cm} (7)
where \([K_1^{(e)}], [K_2^{(e)}], [K_3^{(e)}]\) and \([P]^{(e)}\) denote the heat conduction, the boundary condition, the unsteady and the load matrix, respectively. By assembling each sub matrix, it can be obtained that
\[
[K] \{P\} = \{P\}
\]
where the coefficient matrix are
\[
[K] = \sum_{e=1}^{E} \left( [K_1^{(e)}] + [K_2^{(e)}] + [K_3^{(e)}] \right)
\]
\[
\{P\} = \sum_{e=1}^{E} \{P\}^{(e)}
\]
After imposing (2)-(5) to (8), the temperature at the \(S_2\) can be acquired. With various models generated by MATLAB, it is feasible to produce sufficient data for training and testing.

2.3 Architecture of the framework

The 3D mode in our model has two tough points: (1) there are 900 sampling points of heat flux to be predicted, so that it is inefficient to use fully connected-LSTM to predict one by one; (2) there are various shapes of the solid body, so that the temperature distribution is influenced by both the heat flux and the body shapes. To address these problems, the ConvLSTM [33], containing both convolutional structures in input-to-state and state-to-state transitions, is introduced to the framework. By assembling multiple ConvLSTM layers with LSTM and fully connected layers, we could build a network model for 3D IHCPs.

A major drawback of fully connected-LSTM in handling spatiotemporal data is that it uses only fully connected layers in transitions so that it does not encode any spatial information. However, since all the inputs \(\chi_1, \ldots, \chi_t\), cell outputs \(C_1, \ldots, C_t\), hidden states \(H_1, \ldots, H_t\) and gates \(i_t, f_t, o_t\) of the ConvLSTM are 3D tensors, it is feasible to store different spatial information in different channels. This is achieved by using a convolution operator in the transitions. The key equations of ConvLSTM are shown below, where ‘*’ denotes the convolution operator and ‘o’ denotes the Hadamard product:

\[
i_t = \sigma(W_{xi} * \chi_t + W_{hi} * H_{t-1} + W_{ci} * C_{t-1} + b_i)
\]
\[
f_t = \sigma(W_{xf} * x_t + W_{hf} * H_{t-1} + W_{cf} * C_{t-1} + b_f)
\]
\[
c_t = f_t \circ C_{t-1} + i_t \circ \tanh(W_{xc} * x_t + W_{hc} * H_{t-1} + b_c)
\]
\[
o_t = \sigma(W_{xo} * x_t + W_{ho} * H_{t-1} + W_{co} * C_t + b_o)
\]
\[
H_t = o_t \circ \tanh(c_t)
\]
ConvLSTM can serve as a layer for a more complex framework to capture spatial-temporal characteristics in practical scenarios. In this work, the proposed framework shown in Fig. 4 with three layers of ConvLSTM, three layers of FC-LSTM and one layer of fully connected network is applied to solve the complex IHCPs. By minimizing the kernel size gradually in different layers, this structure could capture the different sizes of spatial relationships efficiently. We set two kernel filters of the first layers as \(4 \times 4\) to separately keep the geometry shapes and temperature distributions. Considering the nonlinearity of the system, we add an extra layer in which the kernel filter size is \(2 \times 2\) to better fit the spatial influence of the heat flux. And the last layer of ConvLSTM is to unify the result of two filters into one. By minimizing the kernel filters and unifying two channels, the output of the last ConvLSTM layer is a combination of geometry shape and temperature. Therefore, the input of LSTM could be imagined as a 3D temperature which is similar to the heat flux in spatial so that we only need to capture the temporal information. The first two layers of LSTM are set to return the sequence in each time step, as this will help find the relationship between the temperature and heat flux more precisely. At the last layer, the LSTM is used to set the sensitive coefficient for every time step to make sure the main heat flux influence the temperature at this time. To avoid missing the information included in former layers, the
number of the neurons is the same as that of the sampling points of heat flux. After the LSTM, we also add a fully connected layer to resize the output of the LSTM as it may be unable to match the supervised data. Since the network has multiple stacked ConvLSTM layers, it has striking power to give predictions in complex dynamic systems like IHCP in 3D complex models.

![Architecture of the framework in solving inverse heat conduction problems.](image)

**Figure 4:** Architecture of the framework in solving inverse heat conduction problems.

### 2.4 Training process

The training process is performed on an NVIDIA RTX 2080 Ti graphic card on Dell Precision 7920 Tower. During the process, the loss of the training set (training loss) and that of the test set (validation loss) decrease gradually. After 300 iterations, the loss reduced to an acceptable level. The loss curves are exhibited in Fig. 5.

![Loss curve of the training process.](image)

**Figure 5:** The loss curve of the training process.

### 3 Results

In this part, some specimens with various heat flux distributions are chosen from the complex 3D data set to assess the performance of the proposed framework. Here, the ground truth of heat flux distribution is defined by the data generated through MATLAB.

#### 3.1 Regular 3D models

Before starting to handle 3D complex models, we first build up a simple 3D model with flat surfaces to evaluate the ability of the proposed framework in capturing the temporal and spatial features. In such models, as the heat flux surface and the temperature measured surface are parallel, the spatial distribution of the temperature is similar to that of the heat flux and this kind of models is denoted as regular. The main difficulty is capturing the temporal features. Here we
use three heat flux cases to test the network and the outcome is visualized in Fig. 6. We randomly choose two different
time steps to assess the framework and the result exhibits that the difference in each model is slightly affected by the
time steps which demonstrates the proposed framework is good at predicting the temporal variation of the heat flux. To
show the precise difference between the ground truth of heat flux and predicted ones, we add the distribution of relative
error in each case which is defined as

$$\text{error} = \left| \frac{q_{\text{predicted}} - q_{\text{real}}}{q_{\text{real}}} \right|$$  \hspace{1cm} (16)

Figure 6: Samples in the regular 3D test set. (a) The predicted heat flux, (b) the ground truth of heat flux and (c) the
relative errors.

### 3.2 Complex 3D models

After the 3D regular models, we change the training data of the framework and measure the ability of the ConvLSTM
to reconstruct the distribution of heat flux combining the features of complex 3D models and temperature distributions.
In this kind of models which is symbolized as complex, the heat flux surface is no longer parallel to the temperature
measured surface and contains curved surface. To demonstrate the performance of the ConvLSTM framework in a
better way, we randomly choose twelve cases to exhibit the temporal and spatial inversion ability separately. Fig.6
shows three cases that include different complex models and heat flux distributions. The geometry shapes of the models
varies in many different aspects. The max height, min height and curvature are different while the relative error is stable.
Therefore, we could figure out that the proposed framework performed pretty well on complex 3D models and have a
good generalization ability in geometry shapes.

In Fig. 8 there are three complex models and three different heat fluxes for each model. These heat fluxes are unevenly
distributed in space, and we randomly select three points to illustrate the temporal fitting capability of the framework.
The three points are located on the diagonal of the surface and have different ranges of the heat flux variation. The
result shows that the range of the heat flux variation and location of the point have few influence on the prediction and
demonstrates that our framework has a good generalization ability in predicting temporal and spatial related heat flux.

### 3.3 Calculation accuracy and speed

We compute the relative error for the test case in Fig. 9. As mentioned before, the regular refers to the regular models of
which the two surfaces are parallel while the complex denote the complex models where two surfaces are not parallel.
Figure 7: Samples in the complex 3D test set. (a) The geometry of the complex models, (b) the forecast of the heat flux and the measured temperature (t=70s) (c) the ground truth of the heat flux and the measured temperature (t=70s) and (d) the relative errors.

Figure 8: Samples in the complex 3D test set. (a) The geometry of the complex models, (b), (c) and (d) three different heat flux distributions imposed on (a). $R_i (i = 1, 2, 3)$: ground truth, $P_i (i = 1, 2, 3)$: prediction.
Since the regular models offer more information about the heat flux, the difficulty of the IHCP for regular ones is less and the loss distribution is better than the complex ones. These figures illustrate that for regular 3D models it is easier to predict the heat flux more precisely and stable.

Figure 9: Statistical distribution for test cases separately produced by two kinds of generators. The regular one represents the flat heat flux surface while the complex one represents the complex surface.

Also, since the framework is based on the GPU and CUDA is employed to forecast the heat flux, the average time cost to predict a case which contains 71 time steps and represents a 35s period is only 1.27s. In other words, for each time step, it takes time within 20 ms.

4 Conclusion

In this work, we have proposed a new framework to forecast the heat flux of complex 3D models, which have temperature-dependent thermal parameters and nonlinear physical boundaries. We introduce a new variation of ConvLSTM which is designed to solve spatial and temporal-related problems. Compared to the regular iterative methods, such a framework can work efficiently after being trained properly. As it usually uses CUDA to calculate in parallel, it could figure out the heat flux precisely within 20 ms, which is extremely superior to those traditional algorithms. Also, as this framework combines ConvLSTM with LSTM, it takes few epochs to reach convergence. The result illustrate that our framework have a good generalization ability in forecasting temporal and spatial related heat flux. Hence, the proposed framework can replace the direct heat flux measurements in real-time heat flux inversion problems with advanced temperature measurements and have multiple applications in many industrial scenarios like engine design or furnace operation in the future.

Acknowledgments

This work is supported in part by the National Nature Science Foundation (Grant No. 61801009), in part by the National Science and Technology Major Project (Grant No. 2019-VIII-0009-0170), in part by Defense Industrial Technology Development Program (Grant No. JCKY2020601B011) and in part by Stable Operation Project.

References

[1] Hamidreza Najafi and Keith A Woodbury. Online heat flux estimation using artificial neural network as a digital filter approach. International Journal of Heat and Mass Transfer, 91:808–817, 2015.
[2] James V Beck, Ben Blackwell, and Charles R St Clair Jr. *Inverse heat conduction: Ill-posed problems*. James Beck, 1985.

[3] S Chantasiriwan. Comparison of three sequential function specification algorithms for the inverse heat conduction problem. *International communications in heat and mass transfer*, 26(1):115–124, 1999.

[4] OM Alifanov. Solution of an inverse problem of heat conduction by iteration methods. *Journal of Engineering Physics*, 26(4):471–476, 1974.

[5] Mattia Bergagio, Haipeng Li, and Henryk Anglart. An iterative finite-element algorithm for solving two-dimensional nonlinear inverse heat conduction problems. *International Journal of Heat and Mass Transfer*, 126:281–292, 2018.

[6] Miao Cui, Kai Yang, Xiao-liang Xu, Sheng-dong Wang, and Xiao-wei Gao. A modified levenberg–marquardt algorithm for simultaneous estimation of multi-parameters of boundary heat flux by solving transient nonlinear inverse heat conduction problems. *International Journal of Heat and Mass Transfer*, 97:908–916, 2016.

[7] Piotr Duda. A general method for solving transient multidimensional inverse heat transfer problems. *International Journal of Heat and Mass Transfer*, 93:665–673, 2016.

[8] JV Beck, B Litkouhi, and CR St Clair Jr. Efficient sequential solution of the nonlinear inverse heat conduction problem. *Numerical Heat Transfer, Part A Applications*, 5(3):275–286, 1982.

[9] Andrey N Tikhonov and Vasily Y Arsenin. Solutions of ill-posed problems. *New York*, 1:30, 1977.

[10] Shuwen Huang, Bo Tao, Jindang Li, Yajun Fan, and Zhouping Yin. Estimation of the time and space-dependent heat flux distribution at the tool-chip interface during turning using an inverse method and thin film thermocouples measurement. *The International Journal of Advanced Manufacturing Technology*, 99(5):1531–1543, 2018.

[11] JR Shenefelt, R Luck, RP Taylor, and JT Berry. Solution to inverse heat conduction problems employing singular value decomposition and model-reduction. *International Journal of Heat and Mass Transfer*, 45(1):67–74, 2002.

[12] Hongchu Chen, Jay I Frankel, and Majid Keyhani. Nonlinear inverse heat conduction: Digitally filtered space marching with phase-plane and cross-correlation analyses. *Numerical Heat Transfer, Part B: Fundamentals*, 72(2):109–129, 2017.

[13] Farshad Kowsary, Morteza Mohammadzaheri, and Saeed Irano. Training based, moving digital filter method for real time heat flux function estimation. *International communications in heat and mass transfer*, 33(10):1291–1298, 2006.

[14] Balázs Czél and Gyula Gróf. Inverse identification of temperature-dependent thermal conductivity via genetic algorithm with cost function-based rearrangement of genes. *International Journal of Heat and Mass Transfer*, 55(15-16):4254–4263, 2012.

[15] Keith A Woodbury. Neural networks and genetic algorithms in the solution of inverse problems. *Bulletin of Brazilian Society for Computing and Applied Mathematics*, 2000.

[16] Yang Zeng, Hu Wang, Shuai Zhang, Yong Cai, and Enying Li. A novel adaptive approximate bayesian computation method for inverse heat conduction problem. *International Journal of Heat and Mass Transfer*, 134:185–197, 2019.

[17] FJCP Soeiro, Patrícia Oliva Soares, HF Campos Velho, and AJ Silva Neto. Using neural networks to obtain initial estimates for the solution of inverse heat transfer problems. In *Inverse Problems, Design an Optimization Symposium*, pages 358–363, 2004.

[18] J Krejsa, KA Woodbury, JD Ratliff, and M Raudensky. Assessment of strategies and potential for neural networks in the inverse heat conduction problem. *Inverse Problems in Engineering*, 7(3):197–213, 1999.

[19] B Ghadimi, F Kowsary, and M Khorami. Heat flux on-line estimation in a locomotive brake disc using artificial neural networks. *International Journal of Thermal Sciences*, 90:203–213, 2015.

[20] Obed Cortés, Gustavo Urquiza, JA Hernandez, and Marco A Cruz. Artificial neural networks for inverse heat transfer problems. In *Electronics, Robotics and Automotive Mechanics Conference (CERMA 2007)*, pages 198–201. IEEE, 2007.

[21] Ali Mirsepahi, Lei Chen, and Brian O’Neill. A comparative artificial intelligence approach to inverse heat transfer modeling of an irradiative dryer. *International communications in heat and mass transfer*, 41:19–27, 2013.

[22] Balázs Czél, Keith A Woodbury, and Gyula Gróf. Simultaneous estimation of temperature-dependent volumetric heat capacity and thermal conductivity functions via neural networks. *International Journal of Heat and Mass Transfer*, 68:1–13, 2014.
[23] Balázs Czél, Keith A Woodbury, and Gyula Gróf. Inverse identification of temperature-dependent volumetric heat capacity by neural networks. *International Journal of Thermophysics*, 34(2):284–305, 2013.

[24] Russell G Keanini, Xianwu Ling, and Harischandra P Cherukuri. A modified sequential function specification finite element-based method for parabolic inverse heat conduction problems. *Computational Mechanics*, 36(2):117–128, 2005.

[25] Cai Lv, Guangujun Wang, Hong Chen, and Shabin Wan. Estimation of the moving heat source intensity using the multiple model adaptive inverse method. *International Journal of Thermal Sciences*, 138:576–585, 2019.

[26] Piotr Duda and Mariusz Konieczny. A new algorithm for solving an inverse transient heat conduction problem by dividing a complex domain into parts. *International Journal of Heat and Mass Transfer*, 128:865–874, 2019.

[27] Piotr Duda and Mariusz Konieczny. Solution of an inverse transient heat conduction problem in a part of a complex domain. *International Journal of Heat and Mass Transfer*, 127:821–832, 2018.

[28] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[29] Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

[30] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318. PMLR, 2013.

[31] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215*, 2014.

[32] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR, 2015.

[33] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *arXiv preprint arXiv:1506.04214*, 2015.