The views expressed are those of the author and do not reflect the official policy or position of the Department of War or the U.S. Government.

A Machine Learning-Driven Digital Twin for Real-Time Coolant Flow Estimation in Power Electronics Cooling Systems

Josiah Worch[†], Kerry Sado[*], Austin R.J. Downey[†‡],

[*]*Dept. of Electrical Engineering*
[†]*Dept. of Mechanical Engineering*
[‡]*Dept. of Civil and Environmental Engineering*
University of South Carolina
Columbia, SC 29208 USA
austindowney@sc.edu

***Abstract*—Providing consistent and reliable cooling for power electronics is an important consideration in the design and operation of electrified aircraft. In this work, a machine-learning-enabled digital twin architecture is proposed for branch coolant flow estimation in liquid-cooled power electronics using temperature measurements. Predictions are enabled by utilizing either Support Vector Regression or Neural Network models, and training them on experimental data. On a test set spanning nominal and obstructed operating conditions, the neural network achieved an overall NRMSE of 9.40% and 19.9% for the Neural Network and Support Vector Regression methods, respectively. These results show that temperature only flow estimation can provide noninvasive support for sensor validation, anomaly monitoring, and control oriented thermal management in liquid-cooled power electronics systems.**

***Index Terms*—machine learning, cooling system, real-time prediction, electric aircraft, power electronics, digital twin**

I. Introduction

Electrification in aviation has been explored nearly since the advent of flight, with an electric motor being first used in 1883 to propel an airship [1]. Since then, aircraft have developed from being almost purely mechanical systems, in the cases of the Wright flyer and other early aircraft, to being largely electronically governed since the advent of fly-by-wire and other advanced avionics. Beyond the electrical systems that are used to govern modern flight control systems, many aircraft manufacturers are designing and producing More Electric Aircraft (MEA) and All Electric Aircraft (AEA) to replace pneumatic, hydraulic, and even thrust generation systems with their electric counterparts [2], [3]. These electric counterparts produce heat through inefficiencies, and typically utilize liquid cooling processes to avoid temperature-related malfunctions [4].

In thermal management, a Digital Twin (DT) combines live sensor data, a physics-based model, and estimation logic to maintain a continuously updated virtual representation of the cooling loop. Branch coolant flow rate is a high-value latent state within such an architecture because local flow distribution directly affects convective heat removal and temperature rise. Direct measurement of every branch, however, adds packaging burden, pressure loss, maintenance demand, and exposure to sensor failure. A temperature-based flow estimator therefore supplies a practical virtual sensor for the digital twin and enables monitoring, fault detection, and supervisory control without additional in-line instrumentation. Recent work on the same testbed demonstrated digital-twin-based blockage detection and autonomous recovery, but branch flow estimation from temperature data has not yet been addressed [5].

Many different Machine Learning (ML) methods have been utilized for flow prediction and forecasting in various areas of ongoing and completed research. Researchers tested four ML models for the purpose of flow rate prediction and forecasting in the Des Moines River in Iowa, USA. The test performed with Random Subspace, M5 Pruned, Random Forest, and Bagging ML models resulted in reliable flow prediction and forecasting up to 7 days in advance using M5P [6]. Similar research conducted at the Maritsa River in Bulgaria utilized Long Short-Term Memory, and an Adaptive Neuro-Fuzzy Inference System with Fuzzy C-Means, Subtractive Clustering, and Grid Partition. Using these ML methods, the researcher produced high success in volumetric flow rate predictions [7]. Fonseca et al. utilized an Artificial Neural Network (ANN) with temperature training data to perform mass flow rate predictions in refrigeration systems by relying on the known relationships between temperature, pressure, density, compressor rotation, and specific volume for the refrigeration cycle with a known refrigeration fluid [8].

Existing literature largely addresses river flow forecasting or refrigeration systems, but temperature only estimation of branch coolant flow in liquid cooled power electronics remains minimally reported. This paper addresses that gap by evaluating a neural network and a support vector regression model within a digital twin framework trained on experimental testbed data. The develops an ML-enabled digital twin approach to estimate coolant flow in real-time for power electronic cooling loops using measured component temperatures. By solely using component temperature data, this methodol-

This material is based upon work supported by the Office of Naval Research (ONR) under contract numbers N00014-23-C-1012 and N00014-24-C-1301.

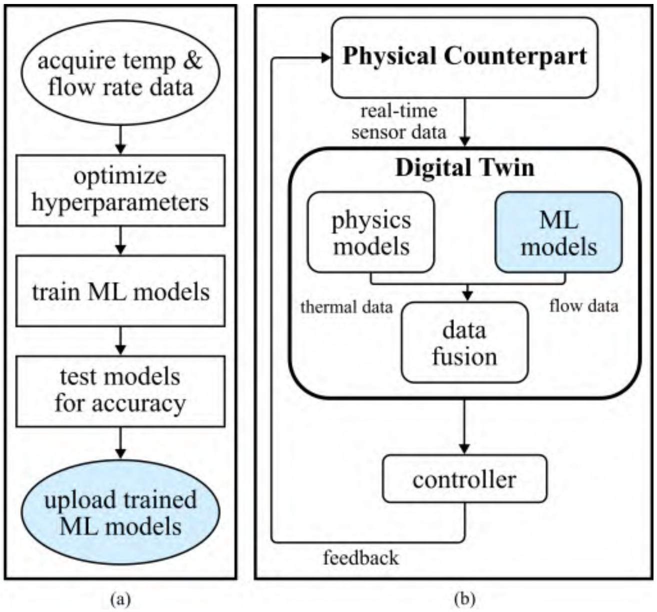

Fig. 1. Conceptual DT architecture and model development workflow for temperature-based branch flow estimation. Panel a shows offline training and validation. Panel b shows intended deployment linking the PT, DT, learned flow estimator, and supervisory controller.

ogy can remove the need for flow meters, reducing in-line space requirements and maintenance frequency, or provide a redundant method to validate physical sensor measurements.

Section II describes the methodology of the digital twin framework, experimental testbed, preprocessing, and regression models. Section III reports held out test results under nominal and obstructed conditions. Section IV summarizes the main findings and outlines future work.

II. Methodology

In this section, the use of ML models for coolant flow estimation is discussed within a DT framework using experimental data gathered from a physical testbed. The DT framework includes three key components: The first is a digital, physics-based model of the physical system-under-study that mirrors the thermal operation of the DT's corresponding testbed. The model is referred to as a digital shadow, and the corresponding testbed is referred to as the Physical Twin (PT). The second component consists of data analysis and communication between the digital shadow and the PT. While not the focus of this paper, these two components enable forecasting of PT thermal behavior when properly calibrated and have been used to support non-invasive obstruction detection [5]. The third component, and the focus of this paper, is a set of ML regression models used to estimate coolant flow rate from temperature measurements. The inclusion of the ML models enables the DT to estimate branch (module) coolant flow in real-time using temperature-derived features. Figure 1 shows the general operation of the DT‘s fluid flow prediction pipeline. Machine learning models are first trained offline on pre-existing temperature and flow-rate data. The DT then feeds the pre-trained ML models real-time thermal data gathered from the testbed. The ML models then provide flow rate predictions, which allow for general system feedback from a controller. This feedback can be used with adaptive controls and decision making systems for use in parameter optimization, anomaly detection, general system forecasting, and more. The ML techniques discussed in the following subsections focus on regression, which predicts a continuous target variable from a set of input features. In this work, both Neural Networks (NNs) and Support Vector Regression (SVR) are evaluated for this application. Both methods are grounded in optimization principles but differ significantly in their structure, flexibility, and interpretability. This section provides a mathematical and conceptual overview of these techniques.

A. Neural Networks for Regression

Neural networks are parametric models that approximate a function $f : \mathbb{R}^n \to \mathbb{R}$ by composing multiple layers of transformations. Each layer consists of a variable number of neurons that apply a linear transformation followed by a non-linear activation function. Including both linear and non-linear methods enables neural networks to capture complex relationships between input data and the resulting outputs.

1) Architecture and Forward Propagation: For a feed-forward neural network with a number of layers denoted by L, input vector $\mathbf{x}$, and output prediction $\hat{y}$, forward propagation is:

$$\mathbf{a}^{(0)} = \mathbf{x}, \tag{1}$$
$$\mathbf{z}^{(l)} = W^{(l)}\mathbf{a}^{(l-1)} + \mathbf{b}^{(l)}, \tag{2}$$
$$\mathbf{a}^{(l)} = \sigma(\mathbf{z}^{(l)}),\ l = 1, \dots, L \tag{3}$$

where:

- $W^{(l)} \in \mathbb{R}^{n_l \times n_{l-1}}$ is the weight matrix for layer l,
- $\mathbf{b}^{(l)} \in \mathbb{R}^{n_l}$ is the bias vector,
- $\sigma(\cdot)$ is an activation function such as Rectified Linear Unit (ReLU) or sigmoid.

For regression tasks, the output layer typically uses a linear activation:

$$\hat{y} = \mathbf{a}^{(L)}. \tag{4}$$

An example of the typical structure of a neural network can be seen in Fig. 2, highlighting its distinct layers.

2) Loss Function: In this work, the neural network is trained to minimize the normalized root-mean-square error (NRMSE). Using a normalized value of root-mean-square-error (RMSE) removes scale dependence in prediction accuracy, and provides error in a percentage format. The RMSE between predicted and actual flow values is defined as

$$\mathrm{RMSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(\hat{y}^{(i)} - y^{(i)}\right)^2}, \tag{5}$$

where m is the number of training samples and $\hat{y}^{(i)}$ and $y^{(i)}$ are the predicted and true values, respectively.

To remove dependence on the absolute scale of the data, RMSE is normalized by the dynamic range of the target variable:

$$\mathrm{NRMSE} = \frac{\mathrm{RMSE}}{y_{\max} - y_{\min}}, \tag{6}$$

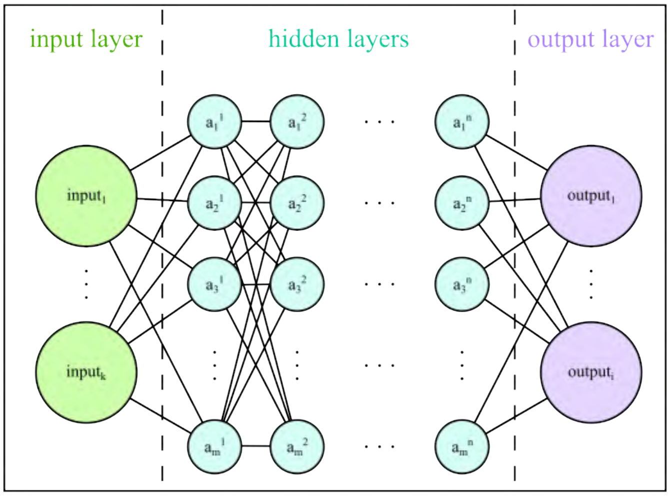

Fig. 2. Figure showing the input, hidden, and output layers for a given neural network. The neurons in the hidden layers apply weighted transformations and activation functions, allowing for the neural network to provide an accurate output for a given set of inputs.

where $y_{\max}$ and $y_{\min}$ denote the maximum and minimum observed flow rates in the dataset.

The training objective is therefore

$$J(\Theta) = \text{NRMSE}, \tag{7}$$

where $\Theta = \{W^{(l)}, \mathbf{b}^{(l)}\}_{l=1}^{L}$ contains all network parameters.

3) Optimization via Gradient Descent: Training involves iterative updates to minimize $J(\Theta)$ using gradient descent:

$$\Theta \leftarrow \Theta - \eta \nabla_{\Theta} J(\Theta), \tag{8}$$

where η is the learning rate. Gradients are computed using backpropagation:

$$\frac{\partial J}{\partial W^{(l)}} = \delta^{(l)} (\mathbf{a}^{(l-1)})^T \tag{9}$$

$$\delta^{(l)} = \left((W^{(l+1)})^T \delta^{(l+1)}\right) \odot \sigma'(\mathbf{z}^{(l)}) \tag{10}$$

This process propagates errors backward through the network, adjusting weights to reduce prediction error.

4) Regularization: To reduce the risk of overfitting, a regularization term is added:

$$J_{\text{reg}}(\Theta) = J(\Theta) + \lambda \sum_{l=1}^{L} \|W^{(l)}\|_2^2, \tag{11}$$

where λ controls the strength of regularization.

B. Support Vector Regression (SVR)

SVR utilizes Support Vector Machines to perform regression tasks by fitting a function within a tolerance margin, called the ε-tube.

1) Primal Formulation: Given training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^{m}$, SVR solves:

$$\min_{w,b,\xi_i,\xi_i^*} \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{m} (\xi_i + \xi_i^*) \tag{12}$$

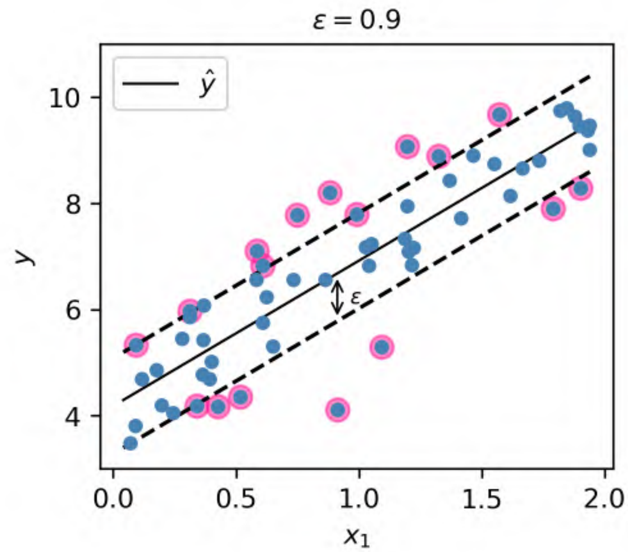

Fig. 3. Image demonstrating a linear example of support vector regression, highlighting the data points (blue dots) and those used as support vectors (pink circles).

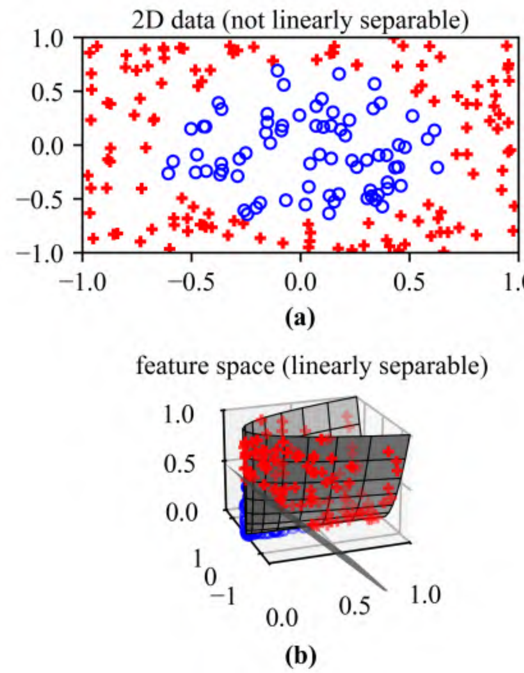

Fig. 4. RBF kernel demonstration, showing: (a) scatter plot of linearly inseparable data in 2D space, and; (b) data transformed by RBF kernel to be separable by a plane in 3D space.

subject to:

$$y_i - (w^T \mathbf{x}_i + b) \le \varepsilon + \xi_i, \tag{13}$$

$$(w^T \mathbf{x}_i + b) - y_i \le \varepsilon + \xi_i^*, \tag{14}$$

$$\xi_i, \xi_i^* \ge 0. \tag{15}$$

Here, w and b define the regression function, ϵ specifies the margin width, and C controls the trade-off between margin width and error tolerance.

Figure 3 illustrates the fitting of SVR to an example with linear data.

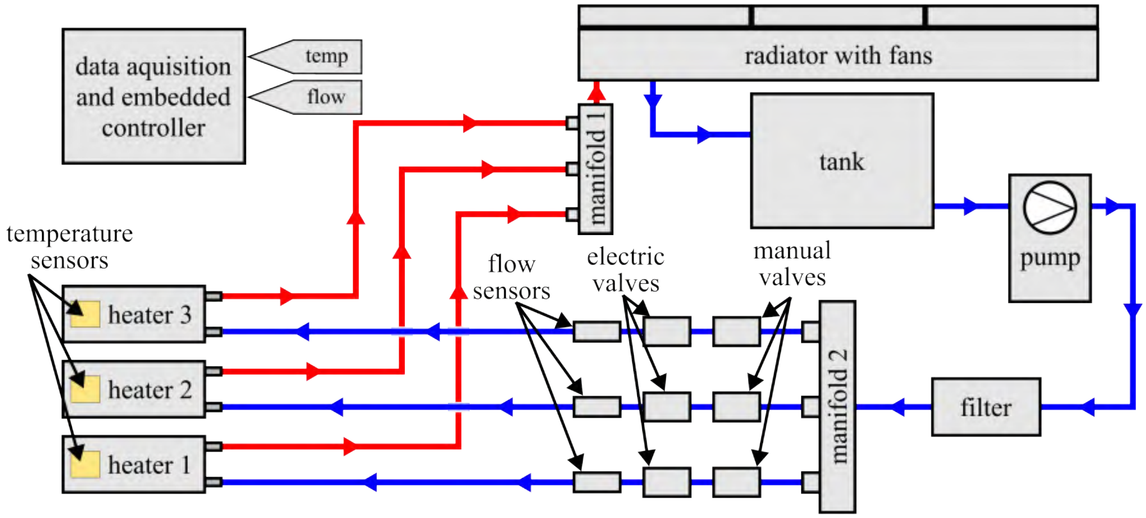

Fig. 5. A schematic diagram showing the design of the thermal-liquid system of the testbed used in this work.

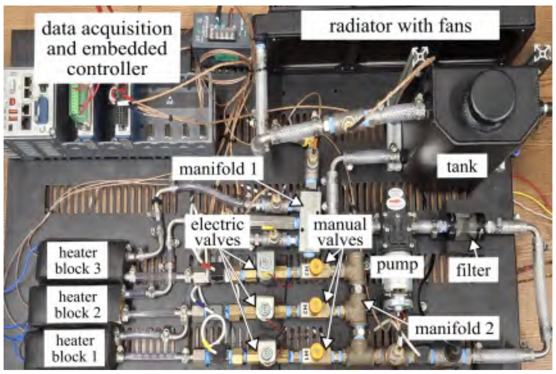

Fig. 6. The experimental Physical Twin testbed designed to emulate a liquid-cooled power electronics system.

2) Dual Formulation and Kernel Trick: The dual problem introduces Lagrange multipliers α_i, α_i^*:

$$\begin{aligned} \max_{\alpha_i,\alpha_i^*} \; & -\frac{1}{2}\sum_{i,j}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)K(\mathbf{x}_i,\mathbf{x}_j) \\ & -\varepsilon\sum_i(\alpha_i+\alpha_i^*)+\sum_i y_i(\alpha_i-\alpha_i^*), \end{aligned} \tag{16}$$

subject to:

$$\sum_i(\alpha_i-\alpha_i^*)=0, \quad 0\le\alpha_i,\alpha_i^*\le C. \tag{17}$$

The introduction of a kernel function $K(\mathbf{x}_i,\mathbf{x}_j) = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$ allows for SVR to perform nonlinear regression, providing regression capabilities of more complex data such as data that is linearly inseparable. Common kernels include:

$$\text{Polynomial: } K(\mathbf{x},\mathbf{x}')=(\mathbf{x}^T\mathbf{x}'+r)^d, \tag{18}$$

$$\text{RBF: } K(\mathbf{x},\mathbf{x}')=\exp(-\gamma\|\mathbf{x}-\mathbf{x}'\|^2). \tag{19}$$

For the purposes of this work, the RBF kernel is used. The plot shown in Fig. 4 provides an example use-case for the RBF kernel on linearly inseparable data.

3) Prediction Function: The SVR prediction is:

$$\hat{y}=\sum_{i=1}^{m}(\alpha_i-\alpha_i^*)K(\mathbf{x}_i,\mathbf{x})+b. \tag{20}$$

C. Comparison and Practical Considerations

Neural networks excel at modeling highly nonlinear relationships and scale well to large datasets, but they require careful tuning of architecture and hyperparameters. SVR training, in contrast, is a convex optimization problem with a global optimum and often performs well on smaller datasets, especially when combined with appropriate kernels. However, SVR can become computationally expensive and lose efficacy for large datasets due to its quadratic programming formulation [9].

D. Data Collection for Model Training

To allow for an ML model to properly make predictions, it must first be properly trained. An experimental testbed was constructed based on the schematic in Fig. 5 to provide training data for the ML models. The testbed was designed to emulate a typical liquid-cooled power electronics cooling system with heat rejection to ambient air. To this end, the testbed was assembled with three heaters simulating waste heat generation from power electronics inefficiencies, a pump, a tank, a debris filter, and a radiator. The details of the design can be seen in Fig. 6. The system also includes electrically actuated control valves and manual balancing valves to distribute and perturb flow across the three heater branches; the electrically actuated valves are controlled by a programmable logic controller (PLC) to automatically adjust coolant flow rate during testing. This testbed was constructed to validate the framework presented in this paper. The temperature and flow sensors utilized in this work, as highlighted in Fig. 5, have relatively low-precision and high-noise [10], [11]. Thus smoothing factors were applied to the data prior to use in model training to reduce the effect of noisy input data.

III. Results

The NN and SVR models were each trained on 16 labeled data sets of temperature and flow rate, with a final data set reserved for testing. These sets ranged from 0.3 h to 3.9 h, and contained time-series temperature and flow rate data recorded at a rate of 1 Hz. The testing data set, shown in Fig. 7, is a section of data gathered near the steady-state operation of the

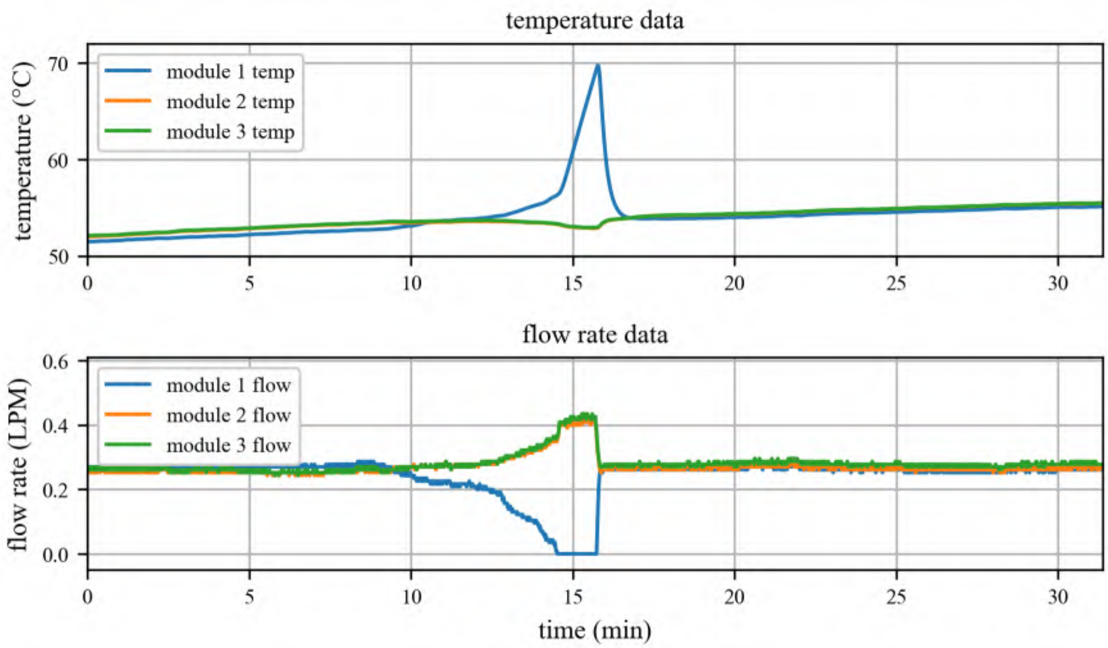

Fig. 7. Plotted data from the data set used for NN and SVR testing. Data begins near steady-state operation and demonstrates the implementation of a full flow obstruction.

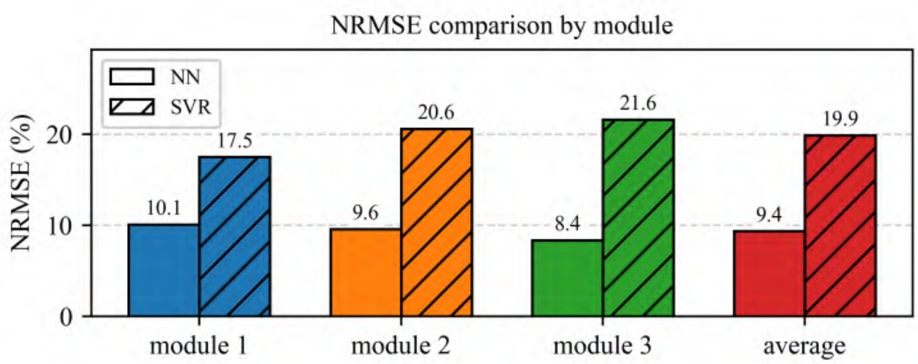

Fig. 8. Bar graph presenting NRMSE values for NN and SVR testing. The graph displays the per-module NRMSE averaged across the full testing data set and the module-averaged NRMSE averaged across the full testing data set.

testbed. This testing data includes a flow obstruction beginning development at 10 min, 31.3% through the 32 minute test, and reaching full obstruction at 16 min. Each model was trained five times to ensure accuracy in the results. The average training time for the NN and SVR models was 9m 58s and 7m 18s respectively. While the SVR model shows a slight training time advantage, it also produced an average NRMSE of 19.9% across the three modules, while the NN model produced an average NRMSE 9.40% across all three modules. The average and module specific NRMSE values produced from this testing can be found presented graphically in Fig. 8 and fully plotted in Fig. 9 and Fig. 10. Since the models were trained and tested on data with nominal and abnormal operating conditions, the NRMSE data provided was also divided into nominal and abnormal NRMSE to demonstrate and present the functionality of each model during different conditions. These values, as well as the total NRMSE values, are displayed in Table I.

TABLE I
NRMSE VALUES FOR NN AND SVR TESTING UNDER NOMINAL, ABNORMAL, AND OVERALL CONDITIONS, AVERAGED ACROSS ALL MODULES.

	NN NRMSE (%)	SVR NRMSE (%)
Nominal	7.51	9.27
Abnormal	11.92	31.16
Overall	**9.40**	**19.9**

IV. Conclusion & Future Work

The work presented in this paper focuses on the development of accurate ML models for use in flow-rate prediction for power electronic liquid-cooling systems when trained with time-series temperature data. As seen in Table I, the NN model consistently outperformed the SVR model when applied to the same testing data. This conclusively demonstrates the superiority of the NN model over the SVR model for use in this work. Thus, NN modeling will be used for regression purposes in future work. The NN model produced an average

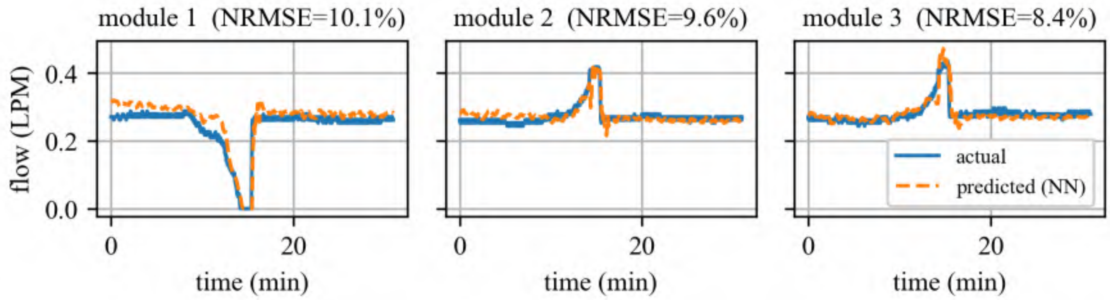

Fig. 9. Plotted data comparing the flow rates predicted by the NN model for each module to the recorded values.

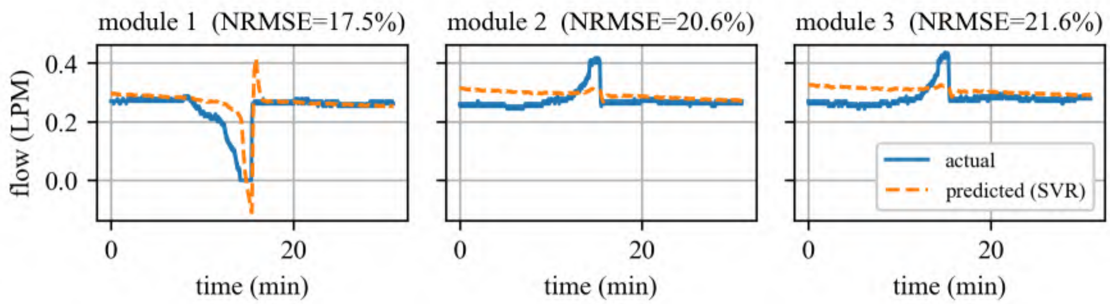

Fig. 10. Plotted data comparing the flow rates predicted by the SVR model for each module to the recorded values.

NRMSE below 10%. While not satisfying fine precision levels, this NRMSE establishes the NN model as a medium-fidelity, non-invasive flow prediction tool for use in digital twins for thermal systems, allowing for greater oversight and monitoring capabilities without additional hardware.

This work provides a foundation from which to perform data fusion between ML and physics-based flow-rate modeling. Future work will show this fusion performed and used to provide comprehensive flow-rate prediction and forecasting in modeling using solely temperature data inputs. This comprehensive modeling will allow for a non-invasive method of providing better informed optimization and control in complex power electronic systems. Additionally, while the work presented in this paper utilizes training data produced from a physical testbed, the ML models are capable of receiving and training on simulated data as well. While the current system uses data provided by flow meters for initial training, using data from physics-based models to perform ML training would allow for the system to be completely independent from flow meters. Future work seeks to determine the efficacy, accuracy, and feasibility of utilizing data provided from the physics-based system model as training data for the ML models.

ACKNOWLEDGMENT

This work is supported by the Office of Naval Research (ONR) under ONR contract numbers N00014-23-C-1012 and N00014-24-C-1301. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessary reflect the views of the United States Navy.

REFERENCES

[1] M. Schrötter, R. Bréda, R. Andoga and L. Főző, "The Evolution of the Aircraft Electric Power Systems," 2018 International IEEE Conference and Workshop in Óbuda on Electrical and Power Engineering (CANDO-EPE), Budapest, Hungary, 2018, pp. 000159-000162, doi: 10.1109/CANDO-EPE.2018.8601156.

[2] A. Barzkar and M. Ghassemi, "Electric Power Systems in More and All Electric Aircraft: A Review," in IEEE Access, vol. 8, pp. 169314-169332, 2020, doi: 10.1109/ACCESS.2020.3024168.

[3] B. Sarlioglu and C. T. Morris, "More Electric Aircraft: Review, Challenges, and Opportunities for Commercial Transport Aircraft," in IEEE Transactions on Transportation Electrification, vol. 1, no. 1, pp. 54-64, June 2015, doi: 10.1109/TTE.2015.2426499.

[4] D. Pal and M. Severson, "Liquid cooled system for aircraft power electronics cooling," 2017 16th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm), Orlando, FL, USA, 2017, pp. 800-805, doi: 10.1109/ITHERM.2017.7992568.

[5] J. Worch, K. Sado, A. R. J. Downey, J. Khan and E. Santi, "Real-Time Blockage Detection and Autonomous Recovery in Liquid-Cooled Systems Using Digital Twins," 2025 IEEE Electric Ship Technologies Symposium (ESTS), Alexandria, VA, USA, 2025, pp. 543-549, doi: 10.1109/ESTS62818.2025.11152430.

[6] Elbeltagi, A., Di Nunno, F., Kushwaha, N.L. et al. River flow rate prediction in the Des Moines watershed (Iowa, USA): a machine learning approach. Stoch Environ Res Risk Assess 36, 3835–3855 (2022). https://doi.org/10.1007/s00477-022-02228-9

[7] Ilhan, A. Forecasting of river water flow rate with machine learning. Neural Comput & Applic 34, 20341–20363 (2022). https://doi.org/10.1007/s00521-022-07576-9

[8] V. D. Fonseca, W. M. Duarte, R. N. de Oliveira, L. Machado, and A. A. T. Maia, "Mass flow prediction in a refrigeration machine using artificial neural networks," *Applied Thermal Engineering*, vol. 214, 118893, Sept. 2022, doi: 10.1016/j.applthermaleng.2022.118893.

[9] R. F. Chevalier and et al.,"Support vector regression with reduced training sets for air temperature prediction: a comparison with artificial neural networks," in Neural Computing and Applications, vol. 20, pp. 151-159, 2011, doi: 10.1007/s00521-010-0363-y.

[10] The Sensor Connection (Alliance Sensors Group), "Surface Mount Stick-On Thermocouple, Type K, Data Sheet [Online]. Available: https://alliancesensors.com/sites/default/files/2021-01/Surface

[11] McMaster-Carr, "Variable-Area Flowmeter, Model 4386N201," [Online]. Available: https://www.mcmaster.com/4386N201/.