International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

Intelligent Thermal Performance Modelling Of Silver-Enhanced Graphene Oxide Hybrid Nanofluids Using Digital Twin

G.M. Subhani[1]
Assistant Professor
Department of Computer Science and Engineering – Artificial Intelligence and Machine Learning
CMR Technical Campus
Hyderabad, Telangana, India

S. Sreeja[2]
Undergraduate Student
Department of CSE (AI & ML)
CMR Technical Campus
Hyderabad, Telangana, India

R. Sai Rishik[3]
Undergraduate Student
Department of CSE (AI & ML)
CMR Technical Campus
Hyderabad, Telangana, India

T. Jeevan[4]
Undergraduate Student
Department of CSE (AI & ML)
CMR Technical Campus
Hyderabad, Telangana, India

***Abstract:-*Effective thermal management is one of the crucial demands of contemporary heat exchangers, cooling systems and electronics but existing thermal conductive fluids like water, ethylene glycol and oils have inherently low thermal conductivity. Hybride nanofluids consisting of multiple nanosized particles in a base fluid are a solution to this problem and Ag-GO (Silver-Enhanced Graphene Oxide) is one of the promising combinations of materials that provides high thermal conductivity of silver combined with high specific surface area and dispersibility of graphene oxide. Experimental study of such materials is time-consuming and costly process, which does not allow quick exploration of different operation modes. In this paper the design of the AI-powered thermal performance predicting framework for Ag-GO hybrid nanofluids and Digital Twin-oriented dashboard are presented. The following regression algorithms are used: Linear Regression, Random Forest, Support Vector Regression and Gradient Boosting. Comparison is performed on the basis of MAE, RMSE and R2.The temperature, flow rate, and Ag-GO concentration are regarded as the inputs, while the effective thermal conductivity, convective heat transfer coefficient, and Nusselt number are regarded as the outputs depending on the data that can be supported by the datasets. A Streamlit dashboard will be used for prediction and visualization purposes and can also serve as the basis for future synchronization using sensors. For now, the present scope involves a software prototype only as there are no sensors yet.**

Keywords: - Silver enhanced graphene oxide, Hybrid nanofluids, Thermal performance prediction, Machine learning, Regression, Digital twin.

I. INTRODUCTION

1.1 Background of Thermal Management

Thermal management is essential for the efficiency, reliability, and safety of systems as different as power generation, process heat exchangers, radiator coolers in vehicles, air conditioning circuits, and microelectronics packages. For all of these, there is a need for moving heat from one location to another by means of a fluid medium, and the transport process depends on both the thermophysical properties of the fluid and the flow characteristics [1]. With increasing powers per unit area and shrinking size of devices, the permissible temperature difference is very small, and hence the characteristics of the heat transfer fluid become a limiting parameter.

There are two approaches for improving the heat transfer performance: modification of the geometry of the heat exchange surface (through fins, micro-channels, and insertions) and improving the thermal properties of the heat transfer fluid itself. The first approach has seen extensive development and is nearing its practical limitations, prompting a long-term interest in heat transfer fluids with better transport properties [2], [3].

1.2 Traditional Heat-Transfer Liquids

Water, ethylene glycol, propylene glycol, mineral oils, synthetic oils and their blends continue to dominate the coolant market. Water has the advantage of a high specific heat combined with

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

low price and low toxicity, glycol-water mixtures can be operated in subzero conditions, oils withstand high temperatures in conditions of low vapor pressure. Their main disadvantage is the relatively poor thermal conductivity, being in the range of 0.1–0.6 W/m·K around room temperature, compared to a significantly higher conductivity of metals and carbon compounds, where silver, for instance, has the conductivity of about 400 W/m·K [1]. It was thought that this difference might be bridged by blending of solid conductors in a liquid [4].

1.3 Limitations of Classical Fluids

As the convective heat transfer coefficient is dependent on the conductivity of the fluid, low-conducting fluids need more surface area, high flow rates, or increased temperature differences in order to perform the required task. However, each of the above approaches incurs some disadvantages in terms of dimensions, energy, or power requirements. Initial efforts towards improving the conductivity using millimeter or micrometer-sized particles were unsuccessful because of high sedimentation rate, clogging of channels, erosion, and pressure losses [5].

1.4 Nanofluids and Hybrid Nanofluids

"Nanofluids" is a term coined by Choi and Eastman to describe fluids that contain stably dispersed nanoparticles with diameters of less than 100 nm [5]. The small size of the particles makes it possible to reduce significantly their sedimentation and erosive nature and to achieve noticeable increases in thermal conductivity for metallic and oxide particles at very low concentrations [6],[7]. The improvement becomes stronger when increasing the concentration of the particles and in many cases with rising temperature [8]. The increase in convective heat transfer during flowing through tubes has also been observed [9]. The physics involved (i.e., Brownian motion, liquid layers near the particle surface, aggregation and ballistic transport) were widely discussed in the literature [10]. In addition, significant variability in the experimental data obtained in different laboratories is noted [3].

Hybrid nanofluids consist of the mixture or composite of two or more types of nanoparticles in one base fluid in order to combine the beneficial features of each component such as high conductivity, stability and rheological behavior [11],[12]. A review on the topic highlights the problems of the area (viscosity, long-term stability and cost).

1.5 Silver-Doped Graphene Oxide Nanofluids

Silver is a promising component for nanofluids due to its extremely high thermal conductivity and conductivity increase of nanofluids containing silver and gold nanoparticles at very small volume fractions [15]. Graphene oxide (GO), which is obtained via oxidative delamination of graphite [16], contains hydroxyl, epoxy and carboxyl groups, which provide the sheets with good water solubility despite the reduced intrinsic conductivity compared to pure graphene. Nanofluids based on GO have been observed to exhibit increased thermal conductivity depending on volume fraction and temperature [17], [18]. Another attempt to combine two effects is the use of graphene oxide sheets doped with silver nanoparticles, which can serve as both a dispersion matrix and a highly conducting component [19]. In an Ag-GO system the sheets provide the matrix that minimizes agglomeration of silver nanoparticles, and the latter contribute highly conducting clusters. The experimental behavior strongly depends on the synthesis process, size, concentration of Ag nanoparticles and type of the base liquid. The present research does not focus on the synthesis process but on modeling of nanofluids. Confirm whether the authors prepared Ag-GO samples or used published/external data.

1.6 Role of Machine Learning in Thermal Predictions

Traditional models like the effective medium theory, including Maxwell [4] and Hamilton-Crosser [20] formulations, model the conductivity of the composite in terms of fraction of particles and constituent materials' properties but do not take into account the role of temperature, particle shape and interface effects and thus generally under-predict enhancements in conductivity [6], [3]. Instead of such models, data-driven regression can be employed to learn the mapping from operating conditions to the thermal properties from the data itself. The prediction of the conductivity enhancement ratio of several nanofluids using artificial neural networks with thousands of literature data points has been reported [21], and various neural network topologies have been tried for oxide-water systems [22]. For smaller tabular datasets, the traditional regressor algorithms in scikit-learn [23], which include linear, tree-based, kernel and boosting regressors, constitute an interesting and interpretable alternative to deep learning.

1.7 Digital Twin Technology

The term “Digital Twin” refers to a virtual replica of a physical asset connected to it by data and is utilized for monitoring, predicting and optimizing its operation. The concept was introduced by Grieves and Vickers [24], and then made known to the aerospace vehicle community by Glaessgen and Stargel [25]; the state-of-the-art technology in the area and its applications are outlined in the surveys for industry [26]. In the field of thermal engineering, Digital Twins were envisaged for heating, cooling and energy storage systems [27] and for shell-and-tube heat exchangers [28]. The difference between the offline digital model, the digital shadow that collects data from the live asset in one direction, and the full Digital Twin with two-way information flow, is generally accepted. The system considered in this paper is intentionally put at the first stage: it is a Digital Twin-based software model and dashboard, that may be extended with sensors' data later on.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

1.8 Problem Statement

The thermal characteristics of Ag-GO hybrid nanofluids cannot be evaluated efficiently by experimental methods in relation to temperature, flow rate, and concentration parameters since it takes time and costs much money. There is a need for a rapid and reliable method which is capable of learning from existing thermal data and predicting some thermal quantities at new parameters, comparing the suggested models, and visualizing the obtained results.

1.9 Objectives of the Research Project

The objectives are: (i) Analysis of thermal characteristics of Ag-GO composite nanofluids; (ii) Processing the thermal data sets; (iii) Application of machine learning regression models; (iv) Prediction of thermal parameters based on the data sets; (v) Model performance assessment with respect to MAE, RMSE and R^2 metrics; (vi) Development of interactive prediction dashboard; and (vii) Creation of groundwork for Digital Twin technology implementation.

1.10 Scope and Contributions

The scope is limited to offline modelling via dataset and dashboard. The contributions include an end-to-end workflow of dataset to dashboard (Figures 1 and 2), comparative analysis of four regression algorithms using common criteria, and a Digital Twin Architecture (Figure 5), which distinguishes the current from the future by segregating between available data and future sensor data. The rest of the paper is organized as follows: Related Work in Section II, Analysis and Design in Section III, Requirements in Section IV, Implementation in Section V, Testing and Results in Section VI, and Conclusions and Future Scope in Section VII.

II. LITERATURE SURVEY

2.1 Nanofluid Conductivity: Past Research

Choi and Eastman [5] suggested adding nanoparticles of metals or oxides into ordinary liquids to improve their electrical conductivity without the problem of clogging present in earlier slurries; the contribution is mainly conceptual and laid out the foundations of the field without providing a predictive model. Lee et al. [6] studied oxide nanofluids (Al_2O_3, CuO) in water and ethylene glycol by means of transient hot-wire method and found enhancement of conductivity increasing with the amount of nanoparticles in the suspension, which is not explained by effective medium theory. Eastman et al. [7] found a conductivity enhancement due to copper nanofluids in ethylene glycol significantly exceeding prediction even at very low concentrations. Das et al. [8] investigated Al_2O_3 and CuO nanofluids in varying temperatures and discovered that enhancement grows with temperature increase, which directly relates to choosing the latter as one of input features.Xuan and Li [9] investigated the convection of Cu-water nanofluids in tube flow, thereby validating the usage of the flow rate as another operating parameter. Keblinski et al. [10] provided a theoretical analysis of possible enhancement mechanisms, while Yu et al. [3] made comparisons between the thermal conductive and convective studies, thus showing that there is considerable inconsistency in the data and the inadequacy of the model-based approach.

2.2 Research on Hybrid Nanofluids

Hybrid nanofluids comprising Al_2O_3-Cu/water have been made by Suresh et al. [11] via two-step synthesis technique, and they investigated the thermophysical properties of the nanofluids, revealing increased conductivity along with increased viscosity. Heat transfer enhancement, alongside increased friction factor, was observed for MWCNT-Fe_3O_4/water hybrid nanofluids in tube flow, as reported by Sundar et al. [12]. Preparation, properties and applications of hybrid nanofluids have been reviewed by Sarkar et al. [13] and Babar & Ali [14]. However, the research works consider some particular material combinations that do not include silver-graphene oxide and suffer from stability issues and the dilemma of conductivity and viscosity.

2.3 Silver Nanoparticles for Thermal Applications

Conductivity behavior of nanofluids with naked and monolayer-protected gold and silver nanoparticles was studied by Patel et al. [15], and it was found that conductivity increases at very low volume fractions with temperature and surface chemistry dependence. This study is experimental and limited to specific kinds of nanoparticles, but it can be used as a basis for choosing silver as a material with high conductivity.

2.4 Nanofluids Made with Graphene Oxide

The process for making graphitic oxide by oxidation, popularly known as Hummers method, was described by Hummers & Offeman [16] and forms the basis for many GO preparations. Yu et al. [17] prepared an ethylene glycol based nanofluid using GO nanosheets and showed an increase in electrical conductivity with loading while Hajjar et al. [18] studied an increase in electrical conductivity of water-based GO nanofluids with varying concentrations and temperatures. Silver nanoparticles embedded in graphene was made by Baby & Ramaprabhu [19] and analyzed the nanofluid made from the material and is the nearest published material to the current material under study. They have not compared machine learning algorithms nor provided a dashboard.

2.5 Thermal Conductivity Predictions by Means of Machine Learning

In Ahmadloo and Azizi's work [21], a multi-input artificial neural network was developed for the prediction of the conductivity ratio of nanofluids based on 776 experimental data points obtained from 21 scientific publications. The same problem was addressed using self-organizing maps by Elsheikh et al. [22] where the conductivity ratio of Al_2O_3, CuO and TiO_2 water nanofluids was predicted based on particle size, temperature and volume fraction. The main advantage of such

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

data-driven models is that they may be superior to or complementary to classical correlations, while their limitations consist in dealing only with a single oxide or metal-oxide nanoparticles and relying on non-homogeneous literature data.

2.6 Regression Algorithms for Thermal Engineering

The four algorithms used in this study have been well-developed. Simple linear regression is easy to understand and use. The random forest algorithm [29] constructs an average of many decorrelated decision trees and deals with non-linear relationships without requiring much tuning. The support vector regression [30] finds a function inside an insensitive tube of width ε and can with kernels model smoothly any non-linear response from small data. Finally, the gradient boosted algorithm [31] constructs an additive model by fitting trees to residuals in an iterative way and usually gives the best accuracy among the four, although with more parameters. They are all used in scikit-learn [23], while NumPy [32] and pandas [33] take care of array and tabular data, and Matplotlib [34] does plotting. The references prove the methods in general but not on Ag-GO data specifically, which is our goal in Section VI.

2.7 Applications of the Digital Twin in the Field of Thermodynamics

The Digital Twin concept is explained as the interaction of a physical product, the virtual product, and the data between them by Grieves and Vickers [24] and as an integrated multi-physical simulation that reflects the life of a vehicle by Glaessgen and Stargel [25]. Digital Twin research in industry is reviewed by Tao et al. [26]. As regards thermal equipment, Singh et al. [28] review the Digital Twin in relation to a shell-and-tube heat exchanger through comparing finite element analysis with the Industry 4.0 approach while Maiorino and Petruzziello [27] describe prospects for Digital Twins in heating, cooling, and thermal energy storage. All these sources either are general or related to the equipment and not to the nanofluid itself, and they need a sensor network infrastructure which is lacking in our dataset-only project.

2.8 Comparison of Current Literature

To summarize, the current research papers [5], [6], [7], [8], [15], [17], [18] prove that the conductivity is influenced by concentration, temperature, and material; however, the scope of each paper is rather narrow. The literature dedicated to hybrid-nanofluids [11], [12], [13], [14] proves both advantages and difficulties in using the combination of elements. Prediction papers [21], [22] use mostly neural networks and conductive ratio. Digital Twin literature [28], [27] focuses on equipment and systems. No existing literature uses Ag-GO nanofluids dataset, a comparison of classical regressors, and the Digital Twin approach in the form of a dashboard. This statement is limited only to the above-listed papers; conduct a broader analysis before submission.

2.9 Research Gap

The following research gaps are identified. Firstly, the existing literature lacks predictive models for Ag-GO hybrid nanofluids that incorporate operational parameters such as temperature and flow rate, in addition to concentration. Secondly, the relative performance of interpretable regressors in relation to one another on similar datasets is hardly ever compared using consistent metrics. Thirdly, prediction models have not been bundled with visualization capabilities for later coupling with sensor readings.

III. ANALYSIS AND DESIGN

3.1 The Existing System

At the present time, there are three methods to assess the thermal properties of nanofluids: measuring conductivity and convective coefficients in the lab, calculating analytical relations, including the Maxwell relation [4], and computational simulation. Data is often stored in spreadsheets and tables, and investigation of any other conditions involves re-measurement or recalculation.

3.2 Limitations of Current Methods

The measurements involve special equipment, preparation of samples, and time; the correlations neglect effects of temperature, flow, and interface and may not correlate well with the measurements [3]; the simulations involve meshes, properties and computation; and none of these approaches offer an easy-to-use interactive interface.

3.3 Proposed System

The proposed system is a Python pipeline that reads the available thermal datasets, preprocesses them, trains four regression models, evaluates the models using MAE, RMSE and R2 metrics, and serves the models via Streamlit dashboard that takes operating conditions and gives back predicted values of the thermal parameters with visualizations. The structure of the system is such that it is possible to use live sensor data as the input in the future.

3.4 Benefits of Proposed System

The system works quickly, because it needs only seconds to do predictions based on new circumstances when the models have already been learned; it is reproducible, because the same code and input always produce the same output; it is extensible, because new data sets and models can be added to the system, and it is accessible, because the dashboard does not require programming skills. Comparison of four algorithms allows choosing the best one for prediction.

3.5 System Architecture

Figure 1 below represents the system architecture. The input data sets (dphx_dataset.csv and heatpipe_dataset.csv as per the project brief) go through preprocessing and feature selection to get to the model library and the output is presented via the dashboard. The layers of the architecture ensure that each layer

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

can be modified independently. The implementation or otherwise of each block in the architecture should match the project description by the authors and should be confirmed from the app.py file.

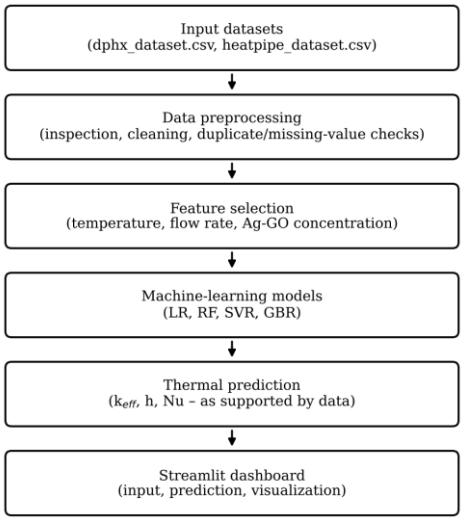


Figure 1. System Architecture Overview.

3.6 Proposed Methodology

This methodology consists of 8 steps which are illustrated in Fig. 2 below. These steps include data collection, inspection, data cleaning, feature selection, training, evaluation, prediction and visualization. Inspection includes examination of names, types, ranges and number of records in columns. Data cleaning involves handling of missing and duplicate records. Feature selection sets input and output columns. Training includes training of algorithms using training dataset while evaluation is performed by using a separate dataset.

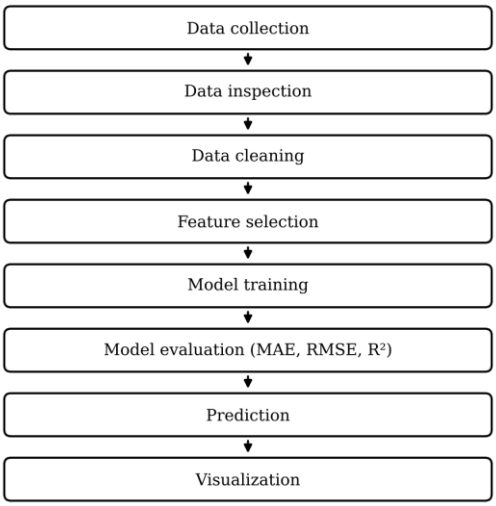


Figure 2. Flowchart for methodology.

3.7 Analysis of the Datasets

The dataset mentioned in the project description include dphx_dataset.csv and heatpipe_dataset.csv. The number of records in each file, the name of columns, units, ranges and source of data have not been confirmed as the data files were not available. Generic attributes that may be expected include those shown in Table III (Section IV). No statistics have been provided as these could be unconfirmed figures.

3.8 Data Preprocessing

Figure 3 displays the process flow of data preprocessing, including loading the dataset, missing value analysis, duplicate detection, cleaning of data and preparation of features. Missing value analysis involves counting the number of empty values in each column, while duplicate detection involves removing duplicate rows. Cleaning involves handling the missing values by either removing them or filling them in, as well as fixing any issues regarding the types and units. Feature preparation involves constructing the input matrix and the target vectors, and scaling may be done at this stage since the support vector regression is scale-sensitive.

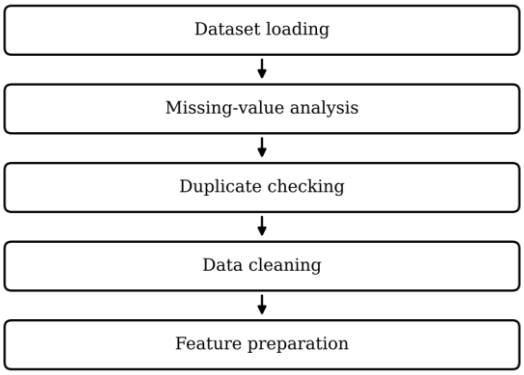


Fig. 3. Dataset preprocessing workflow.

3.9 Feature Selection

The following three candidates are selected for input variables: temperature, flow rate, and Ag-GO concentration. The inputs are based on physical reasoning: enhancement of conductivity is dependent on both temperature and loading [8], [6], while the convective coefficient is a function of flow rate [9], [1]. Only those inputs that exist as features in the datasets will be used.

3.10 Target Selection

Target candidates include thermal conductivity, convective heat transfer coefficient, and the Nusselt Number. The Nusselt number expresses the relationship between the coefficient and the conductivity as:

$$Nu = \frac{hD}{k_f} \qquad (1)$$

Nu being the Nusselt number, h the heat transfer coefficient (W/m²·K), D a representative length such as the pipe diameter (m) and kf the thermal conductivity of the fluid (W/m·K) [1]. In case you want a theory reference, the Maxwell suspension of spheres model is

$$\frac{k_{eff}}{k_{bf}} = \frac{k_p+2k_{bf}+2\varphi(k_p-k_{bf})}{k_p+2k_{bf}-\varphi(k_p-k_{bf})} \qquad (2)$$

where keff is the effective thermal conductivity of the suspension, kbf is the thermal conductivity of the base fluid, kp is that of the particles, and φ is the particle volume fraction [4].

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

Equation (2) is a classical equation and is not the output of the present work; it is not used for Ag-GO data in this manuscript. What are the target columns in the datasets and whether h and Nu are measured or calculated needs to be confirmed.

3.11 Design of Machine Learning Models

Every model is considered a regression function which maps an input vector into a target:

$$\hat{y} = f(x;\theta), x = (T, Q_f, \varphi) \qquad (3)$$

where ŷ denotes the predicted value of target, x the input vector composed of variables such as Temperature (T), Flow rate (Qf) and Concentration (φ), and θ the model parameters which can be estimated using training data set. As illustrated in Fig. 4, the data is split into training and test sets, and the model is trained using the training set while predictions are made using the test set. Comparison of four algorithms: Linear Regression, Random Forest [29], Support Vector Regression [30] and Gradient Boosting [31] is provided in Table IV (Section V).

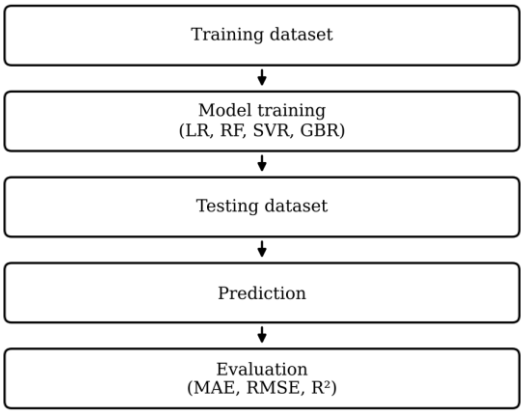


Figure 4: Machine learning process.

3.12 Digital Twin Architecture

Figure 5 provides an illustration of the Digital Twin architecture. The continuous line (data input, computational model, prediction engine, visualization dashboard) denotes the software capabilities only. The dotted lines for sensor integration, real-time data synchronization, and experimental validation denote future capabilities. This means that the current architecture must be referred to as a Digital Twin-based prototype rather than a Digital Twin in real time, as there is no physical asset associated with the current system [24], [26].

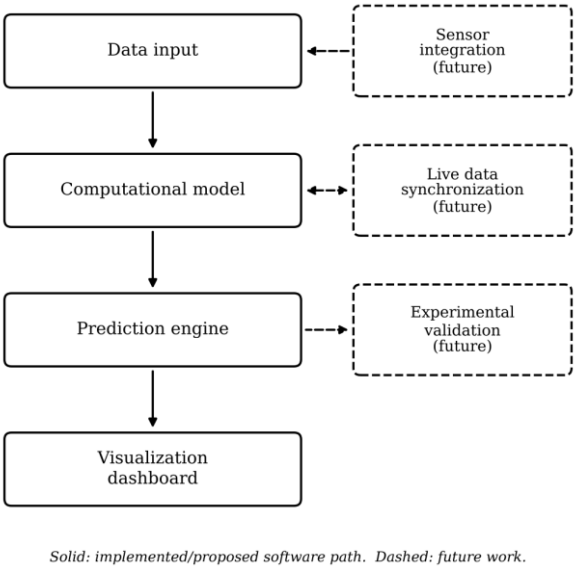


Figure 5. Digital Twin architecture (dashed: future work).

3.13 Design of Dashboard Module

The process in the dashboard is based on the flow chart shown in Fig. 6. Users input operating conditions, the validation module ensures that the inputs are numerical and fall into the range of the training data in order for the model not to extrapolate without notifying the user; the model generates prediction for the output and the prediction result with the units is visualized together with graphs that display relationships between inputs and the prediction, and where possible comparison of the models.

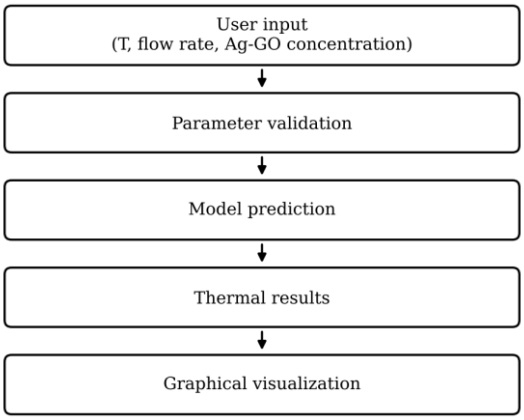


Fig. 6. Dashboard workflow.

3.14 System Workflow

Overall, the data workflow process works as follows. The datasets are imported using pandas data frame; they are reviewed and cleaned; input and target variables are identified; data splitting occurs; the models are trained and evaluated; the trained models are stored in memory or stored on disk; and finally, the dashboard imports them, gets user inputs, checks them, makes predictions and plots. The current components include those mentioned in the authors 'project overview, while the rest are planned for the future.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

IV. SYSTEM REQUIREMENTS

4.1 Software Requirements

Software Requirements are listed in Table I along with their functions. Version Numbers will be left for verification from requirements.txt.

Table I SOFTWARE REQUIREMENTS

Component	Purpose	Version
Python	Primary programming language	*pending*
Streamlit	Web dashboard framework	*pending*
Pandas	Loading and data cleaning	*pending*
NumPy	Number arrays	*pending*
Scikit-learn	regression analysis, measures	*pending*
Matplotlib	Static storylines	*pending*
Plotly	Interactive graphs (if any)	*if used*
Internet browser	Access to the dashboard	*pending*
operating system	Execution environment	*pending*

4.2 Hardware Requirements

It is necessary to have only a standard PC, because the datasets are small and the models used are classical. Table II provides minimum recommendations for reproducing our results, but not actual measurements of the authors' hardware setup.

TABLE II NECESSARY HARDWARE

Component	Suggested minimum	Purpose
Processor	Dual-core CPU	Training of model and inference
Memory	4–8 GB RAM	Data Frames, Model Fitting
Storage	1 GB free	Coding, data, saved model
Display	1366×768	Dashboard display
Network	Optional	Package installation exclusively

4.3 Operating Environment

The application works on various platforms like Windows, Linux, and macOS along with a compatible Python interpreter and a modern web browser. Dashboard works as a local web server accessible via web browser. Operating system and version of Python used by the authors is still under confirmation.

4.4 Programming Language

Python is selected due to the presence of well-established libraries for doing data science, as well as an easy-to-read syntax and Streamlit.

4.5 Libraries and Frameworks

Pandas [33] for reading CSV files and working with table data; NumPy [32] for arrays manipulation; scikit-learn [23] for regressors, splitting data functions and metrics; Matplotlib [34] for visualizing plots; Streamlit to render the dashboard; Plotly is only mentioned if it is used in implementation.

4.6 Datasets Specifications

Datasets have to be tabulated CSV datasets with numerical inputs and targets in the same unit of measure and adequate number of records to ensure train/test split is valid. Table III provides the list of attributes required where columns with "pending" should be replaced with real attributes.

TABLE III DATASET CHARACTERISTICS

Attribute	Role	Unit	Status
Temperature	Input	*pending*	*verify column*
Flow rate	Input	*pending*	*verify column*
Ag-GO content	Input	*pending*	*verify column*
Good heat conductivity	Target	W/m·K	*verify column*
Coefficient of convective heat transfer	Target	W/m²·K	*verify column*
Nusselt Number	Target	–	*verify column*

4.7 Functional Requirements

The system should be able to: load the data sets; provide information on data quality; deal with missing and duplicated data; train four regression models; calculate MAE, RMSE and R^2; take in user-defined operating conditions; verify those conditions; output results with units; and plot the output.

4.8 Non-functional Requirements

The system should interactively react to the input for prediction, should be reproducible using fixed random seeds and packages, be modular so that the model or dataset can be easily replaced, should be user friendly even for non-programmers, and should safeguard against invalid input.

4.9 Future Hardware Requirement

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

None of the sensors or instruments have been mentioned as existing. In order for Digital Twin technology to work in future, there will be requirement for the following items: Temperature sensors (thermocouples/RTD's) at the inlet, outlet and wall; Flowmeter; Heater and pump in a test loop system; Data acquisition unit/microcontroller; and a method of transferring information to the dashboard.

V. IMPLEMENTATION

The process of implementation is explained in this part. The code snippets provided are only demonstrative and have to be substituted by validated snippets from app.py and training scripts.

5.1 Development Environment

Development is performed in Python utilizing libraries from Table I within a separate virtual environment on an editor or a notebook-like environment, where dependencies are listed in requirements.txt. Information on editor, Python version, and operating system is still to be determined.

5.2 Project Folder Structure

The expected folder structure includes the dashboard script (app.py), one or more training scripts, the CSV data sets (dphx_dataset.csv, heatpipe_dataset.csv), and optionally model files and requirements.txt. The actual folder structure will be confirmed later and should be placed here.

5.3 Dataset Loading

The datasets are loaded into data frames using pandas as follows in the first few lines of the following fragment.

```
df = pd.read_csv("dphx_dataset.csv")
X = df[FEATURES]      # e.g. temperature, flow rate,
concentration
y = df[TARGET]        # e.g. k_{eff}, h or Nu
X_tr, X_te, y_tr, y_te = train_test_split(
    X, y, test_size=TEST_FRACTION, random_state=SEED)
```

5.4 Data Inspection

Inspection provides the structure, column names, data types, descriptive statistics, and counts of nulls. This step ensures that the correct columns are present and that the values are within the realistic range before any model training.

5.5 Data Preprocessing

Preprocessing takes place as per Figure 3, which includes removing missing values from rows, dropping duplicate data, and ensuring unit consistency for numerical variables. For support vector regression, feature scaling is done so that no variable becomes dominant in terms of distance through the kernel.

5.6 Feature and Target Selection

The FEATURES and TARGET in the above fragment are derived from the validated columns of each dataset. A different model is trained for each target, unless it is a multi-output regression algorithm. Pending final choices.

5.7 Machine Learning Algorithms

Table IV presents a comparison between the four algorithms based on qualitative criteria from the literature [29], [30], [31], [23]; the table does not include any measurements for this project.

TABLE IV COMPARISON OF MACHINE LEARNING ALGORITHMS

Algorithm	Strengths	Limitations	Key settings
Linear Regression	clear-cut, simple baseline	Cannot represent non-linearity	Few; feature scaling optional
Random Forest	Non-linearity management; sturdy	Cannot draw inference; big model	Trees, depth, leaf size
Support Vector Regression	Good for small smooth data	Needs optimization; kernel tweaking	C, ε, kernel
Gradient Boosting	Frequently most accurate in tables	Sensitivity to tuning; overfitting	Learning rate, trees, depth

5.8 Training of Models

The models are trained using the training partition by applying the same interface fit to the four algorithms, making it easy to compare them to each other.

```
models = {"LR": LinearRegression(), "RF":
RandomForestRegressor(),
          "SVR": SVR(), "GBR":
GradientBoostingRegressor()}
for name, m in models.items():
    m.fit(X_tr, y_tr)
    pred = m.predict(X_te)
```

5.9 Model testing

Held out data, which the models haven't been exposed to during training, will be used for making predictions and comparing them with the actual values. The use of held out data prevents memorization.

5.10 Model Evaluation

The models will be assessed based on the criteria provided in Table V, given below for n test observations where yi represents the actual value, ŷi the predicted value, and ȳ the average actual value. The Mean Absolute Error is

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \qquad (4)$$

The Root Mean Square Error is

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \qquad (5)$$

the coefficient of determination is

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \qquad (6)$$

The formulas of MAE and RMSE have the same units as the target; however, RMSE gives higher weights to larger errors

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

than MAE. $R^2 = 1$ when the predictions are perfect; otherwise, the closer the R^2 value is to 0 or less, the worse the model is compared to the mean model.

TABLE V EVALUATION CRITERIA

Metric	Symbol	Interpretation
Mean Absolute Deviation	MAE	Average error magnitudes; smaller values are preferred
Root Mean Squared Error	RMSE	Punishes large mistakes; smaller is better
Coefficient of Determination	R^2	Proportion of variation explained; nearer to 1 is better

5.11 Prediction Module

The prediction module takes an accepted feature vector, invokes predict on the chosen trained model and provides the output target in physical units. In case the model has been trained using scaled feature vectors, the same scaler must be used for the inputs as well. It is yet to be seen whether models will be retrained or loaded from the saved files.

5.12 Dashboard Design

The dashboard is created using Streamlit, which input widget components naturally correlate to the operating parameters:

```
import streamlit as st
T  = st.number_input("Temperature")
Qf = st.number_input("Flow rate")
phi = st.number_input("Ag-GO concentration")
if st.button("Predict"):
    st.write(model.predict([[T, Qf, phi]]))
```

5.13 Data Visualization

Charts show the relationship between the predicted target and the input variable, predicted vs. observed on the test data set, and comparison of models' performance. Static plots are generated using Matplotlib [34]. Plotly is used only if installed. Charts to be finalized from app.py and screen shots.

5.14 Application Execution

After the installation of dependencies from requirements.txt file, the application is executed using Streamlit command (streamlit run app.py). This starts the server and displays the dashboard in the browser. Screenshots and execution outputs will be provided soon.

VI. TESTING, DEBUGGING AND RESULTS

6.1 Goals of Testing

The objective of testing is to validate whether data are being read and processed correctly, whether models are training and predicting without errors, whether the dashboard is functioning as expected for both correct and incorrect data, and whether metrics are calculated accurately. This draft hasn't been tested at all, hence the tests described below are pending.

6.2 Dataset Testing

Dataset tests ensure that each file opens successfully, that columns with numeric types are present in the dataset and that there are no discrepancies in the number of records and value ranges.

6.3 Preprocessing Testing

Preprocessing tests ensure that the number of missing values after preprocessing is as expected, that no duplicate records are present, and that the shapes of the feature matrix and target variable match.

6.4 Model Training Testing

Model training tests ensure that all four models fit successfully, that fixed random seeds produce reproducible results and that the output of the predictions for the test partition has the right shape.

6.5 Prediction Testing

Prediction tests provide inputs within the training boundaries and verify that the outputs are numeric, finite and of the expected size; inputs outside the training boundary should produce warnings or errors.

6.6 Dashboard Testing

Functional tests of the data pipeline and dashboard are listed in Table VI along with the expected output of each test. The actual output of the tests and their current statuses are marked pending.

TABLE VI
FUNCTIONAL TESTING

ID	Test Case	Expected Output	Actual Output	Status
T1	Loading dataset	DataFrame with expected columns	Pending	Pending
T2	Checking missing values	Count shown per column	Pending	Pending
T3	Same essay	Duplicates eliminated	Pending	Pending
T4	Four model training	Model fitting done	Pending	Pending
T5	Effective forecasting	Numeric answer with units	Pending	Pending
T6	Error input	Error or warning displayed	Pending	Pending
T7	Start dashboard	Browser loads page	Pending	Pending
T8	Plot representation	Chart presentations	Pending	Pending

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

6.7 Debugging and Error Handling

Common errors in these pipelines include column name inconsistency, non-numerical value, missing values in estimators that do not support them, failure to scale using the train scaling on predictions and unit inconsistency. These will be solved using column checking, type conversion, imputation or deletion, stored scaler and input checking on the dashboard. The bugs detected and the fixes used will be added by the authors.

6.8 Evaluation Metrics

These are the metrics given in Table V and Eqs. (4) – (6). They will be computed exactly the same way for the four models and all targets for fairness in comparison.

6.9 Comparison of Models Performance

Table VII is the table format to be used in reporting the results. Numerical values are not given because there is no output of the models' training and evaluation at this time.

TABLE VII
MODEL PERFORMANCE RESULTS

Model	MAE	RMSE	R^2
Linear Regression	Under review	Under review	Under review
Random Forest	Under review	Under review	Under review
Support Vector Regression	Under review	Under review	Under review
Gradient Boosting	Under review	Under review	Under review

6.10 Results and Discussion

Discussion of results is impossible before Table VII is completed. Once done, discussion shall mention which model has the least errors and maximum R^2 for each target, if the ranking satisfies expectations (for instance, if tree ensembles perform better than a linear one in case of non-linearity of the response variable), if there are signs of overfitting based on the comparison of training and test sets, and the most influential input variable. Any result of modeling should be described as a property of a specific data set and not as a general characteristic of Ag-GO nanofluids. Model development and analysis is pending; experimental verification is pending.

6.11 Limitations

The study is constrained by the size, scope and origins of the datasets used without being verified here; by the lack of any physical experiments; by the fact that data-based models work reliably only in the range that was covered during training; and by the software-only nature of the Digital Twin-oriented dashboard without any sensor connection.

VII. CONCLUSION AND FUTURE SCOPE

This study aimed at designing an AI framework to forecast the thermodynamic properties of Ag-GO nanofluid hybrids. The methodological approach involved loading and pre-processing of data sets, selecting temperature, flow rate, and concentration of candidate inputs, training Linear Regression, Random Forest, Support Vector Regression, and Gradient Boosting regressors, assessing their performance based on MAE, RMSE, and R2 values, and producing outputs via a Streamlit web app. The machine learning element in this study is a reliable, replicable evaluation of four different models. The dashboard component represents a web tool that was built as a basis of a Digital Twin concept. Possible applications could involve pre-screnning of operating conditions for heat exchangers, cooling circuits, and electronic cooling.

Current status. This study includes only the design and description of the software prototype. There are no available dataset statistics, model performance measures, and screenshots, as well as physical sensors, real-time synchronization, and experimental validation. The overall result of this study thus includes only a designed and extendable framework.

Future directions. The future developments may be outlined as follows: integration of sensors providing real-time data on temperature and flow; physical experiments involving prepared samples of Ag-GO material; synchronization of the collected data to turn the proposed dashboard into a fully-fledged Digital Twin; better predictive models using hyperparameters and confidence bounds; thermal parameters like viscosity, specific heat and pressure drop; and expansion of the application area to industry-scale cooling systems and electronics thermal management.

REFERENCES

[1] F. P. Incropera, D. P. DeWitt, T. L. Bergman, and A. S. Lavine, *Fundamentals of Heat and Mass Transfer*, 6th ed. Hoboken, NJ, USA: Wiley, 2007.

[2] S. K. Das, S. U. S. Choi, W. Yu, and T. Pradeep, *Nanofluids: Science and Technology*. Hoboken, NJ, USA: Wiley, 2007.

[3] W. Yu, D. M. France, J. L. Routbort, and S. U. S. Choi, "Review and comparison of nanofluid thermal conductivity and heat transfer enhancements," *Heat Transf. Eng.*, vol. 29, no. 5, pp. 432–460, 2008.

[4] J. C. Maxwell, *A Treatise on Electricity and Magnetism*, 1st ed. Oxford, U.K.: Clarendon Press, 1873.

[5] S. U. S. Choi and J. A. Eastman, "Enhancing thermal conductivity of fluids with nanoparticles," in *Proc. ASME Int. Mech. Eng. Congr. Expo.*, San Francisco, CA, USA, 1995, FED-vol. 231/MD-vol. 66, pp. 99–105.

[6] S. Lee, S. U.-S. Choi, S. Li, and J. A. Eastman, "Measuring thermal conductivity of fluids containing oxide nanoparticles," *J. Heat Transf.*, vol. 121, no. 2, pp. 280–289, 1999.

[7] J. A. Eastman, S. U. S. Choi, S. Li, W. Yu, and L. J. Thompson, "Anomalously increased effective thermal conductivities of ethylene glycol-based nanofluids containing copper nanoparticles," *Appl. Phys. Lett.*, vol. 78, no. 6, pp. 718–720, 2001.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

[8] S. K. Das, N. Putra, P. Thiesen, and W. Roetzel, "Temperature dependence of thermal conductivity enhancement for nanofluids," *J. Heat Transf.*, vol. 125, no. 4, pp. 567–574, 2003.

[9] Y. Xuan and Q. Li, "Heat transfer enhancement of nanofluids," *Int. J. Heat Fluid Flow*, vol. 21, no. 1, pp. 58–64, 2000.

[10] P. Keblinski, S. R. Phillpot, S. U. S. Choi, and J. A. Eastman, "Mechanisms of heat flow in suspensions of nano-sized particles (nanofluids)," *Int. J. Heat Mass Transf.*, vol. 45, no. 4, pp. 855–863, 2002.

[11] S. Suresh, K. P. Venkitaraj, P. Selvakumar, and M. Chandrasekar, "Synthesis of Al2O3–Cu/water hybrid nanofluids using two step method and its thermo physical properties," *Colloids Surf. A*, vol. 388, nos. 1–3, pp. 41–48, 2011.

[12] L. S. Sundar, M. K. Singh, and A. C. M. Sousa, "Enhanced heat transfer and friction factor of MWCNT–Fe3O4/water hybrid nanofluids," *Int. Commun. Heat Mass Transf.*, vol. 52, pp. 73–83, 2014.

[13] J. Sarkar, P. Ghosh, and A. Adil, "A review on hybrid nanofluids: Recent research, development and applications," *Renew. Sustain. Energy Rev.*, vol. 43, pp. 164–177, 2015.

[14] H. Babar and H. M. Ali, "Towards hybrid nanofluids: Preparation, thermophysical properties, applications, and challenges," *J. Mol. Liq.*, vol. 281, pp. 598–633, 2019.

[15] H. E. Patel, S. K. Das, T. Sundararajan, A. S. Nair, B. George, and T. Pradeep, "Thermal conductivities of naked and monolayer protected metal nanoparticle based nanofluids: Manifestation of anomalous enhancement and chemical effects," *Appl. Phys. Lett.*, vol. 83, no. 14, pp. 2931–2933, 2003.

[16] W. S. Hummers Jr. and R. E. Offeman, "Preparation of graphitic oxide," *J. Am. Chem. Soc.*, vol. 80, no. 6, p. 1339, 1958.

[17] W. Yu, H. Xie, and D. Bao, "Enhanced thermal conductivities of nanofluids containing graphene oxide nanosheets," *Nanotechnology*, vol. 21, no. 5, Art. no. 055705, 2010.

[18] Z. Hajjar, A. M. Rashidi, and A. Ghozatloo, "Enhanced thermal conductivities of graphene oxide nanofluids," *Int. Commun. Heat Mass Transf.*, vol. 57, pp. 128–131, 2014.

[19] T. T. Baby and S. Ramaprabhu, "Synthesis and nanofluid application of silver nanoparticles decorated graphene," *J. Mater. Chem.*, vol. 21, no. 26, pp. 9702–9709, 2011.

[20] R. L. Hamilton and O. K. Crosser, "Thermal conductivity of heterogeneous two-component systems," *Ind. Eng. Chem. Fundam.*, vol. 1, no. 3, pp. 187–191, 1962.

[21] E. Ahmadloo and S. Azizi, "Prediction of thermal conductivity of various nanofluids using artificial neural network," *Int. Commun. Heat Mass Transf.*, vol. 74, pp. 69–75, 2016, doi: 10.1016/j.icheatmasstransfer.2016.03.008.

[22] A. H. Elsheikh, S. W. Sharshir, and A. S. Ismail, "An artificial neural network based approach for prediction the thermal conductivity of nanofluids," *SN Appl. Sci.*, vol. 2, no. 2, Art. no. 235, 2020, doi: 10.1007/s42452-019-1610-1.

[23] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[24] M. Grieves and J. Vickers, "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," in *Transdisciplinary Perspectives on Complex Systems*, F.-J. Kahlen, S. Flumerfelt, and A. Alves, Eds. Cham, Switzerland: Springer, 2017, pp. 85–113.

[25] E. Glaessgen and D. Stargel, "The digital twin paradigm for future NASA and U.S. Air Force vehicles," in *Proc. 53rd AIAA/ASME/ASCE/AHS/ASC Struct., Struct. Dyn. Mater. Conf.*, Honolulu, HI, USA, 2012, Paper AIAA 2012-1818.

[26] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, "Digital twin in industry: State-of-the-art," *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 2405–2415, Apr. 2019.

[27] A. Maiorino and F. Petruzziello, "Unlocking the potential of digital twins in heating, cooling, and thermal energy storage technologies," *Int. J. AI Mater. Des.*, vol. 1, no. 3, pp. 1–3, 2024, doi: 10.36922/ijamd.4532.

[28] H. Singh, U. Mishra, P. Saxena, G. Shetiya, and Y. M. Puri, "Digital twin for shell and tube heat exchanger in Industry 4.0," in *Advances in Mechanical Engineering*, V. Kalamkar and K. Monkova, Eds., Lecture Notes in Mechanical Engineering. Singapore: Springer, 2021, doi: 10.1007/978-981-15-3639-7_76.

[29] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[30] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," in *Adv. Neural Inf. Process. Syst. 9*, 1997, pp. 155–161.

[31] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, 2001.

[32] C. R. Harris *et al.*, "Array programming with NumPy," *Nature*, vol. 585, pp. 357–362, 2020.

[33] W. McKinney, "Data structures for statistical computing in Python," in *Proc. 9th Python Sci. Conf.*, 2010, pp. 56–61.

[34] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 90–95, 2007.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)