Reinforcement learning in Magnesium Nanoparticle synthesis with Pulsed Laser Ablation in Liquid using Powders

Balaji Rachamadugu, Dr.Abhishek Kaushik, Dr.Anesu Nyabadza
Dundalk institute of technology

Abstract

The synthesis of Magnesium (Mg) nanoparticles with Physical laser ablation in liquid‘ (PLAL) is the upcoming green technology which produces high purity nano colloids without hazardous chemicals. However, the challenging task is the optimization of laser processing parameters such as ablation time, laser scan speed, and fluence based on the user desired nanoparticle size. With the help of Reinforcement learning (RL) framework couples with machine learning surrogate model for efficient optimization of Mg nanoparticle synthesis.

Since the experimental dataset obtained is small, with the help of surrogate modeling increased the size of dataset. When performed EDA on the new dataset it showed similar results of experimental dataset, the Machine learning (ML) model development on the synthesized dataset using random forest model showed perfect fit with R2 value of 0.99 which can be used as environment in the RL framework for better prediction. Implementation of Deep Deterministic Policy Gradient (DDPG) and Twin Delayed DDPG (TD3) algorithms to search for pre-processing parameters of laser like time (min), scanspeed (mm/s) and fluence J/cm^2 based on the user defined nano particle sizes. When compared the DDPG model converged faster but it lacked stability and showed overestimation bias. To overcome some of these challenging tasks TD3 outperformed DDPG, by using double clipping, target policy smoothing and delayed policy update helped in achieving stability.

Contents

0.1 Introduction

Magnesium (Mg) is one of the crucial element in the world, it has found to exist in many materials (physical) used for die-casting, making various types of alloys and in living organisms (biological) for essential living and further more it applications is not limited. It is very much strong and light-weight in its purest form Abbas & Adim (2023). Also manufacturing of Magnesium is not complicated but also essential for daily human needs. Mg Nanoparticle (MgNP) when combined other elements has profound significance in scientific, industrial, technological, bio-medical and other applications. Nanoparticles when combined with other materials increase its usage and effectiveness in applications by reduction in weight, increased strength, conductivity, thermal stability Fazio et al. (2020).

The process of manufacturing Mg nanoparticles include various step and can be done in many ways like biological Suresh et al. (2014), Physical Laser ablation in Liquid (PLAL) Nyabadza, Vazquez, Coyle, Fitzpatrick & Brabazon (2021) which is upcoming method and widely used because ease of varying laser type, solution medium, laser properties to produce different type of Magnesium Nanoparticles (MgNPs) Saedi et al. (2023), or combustion methods which are not environmental friendly.

Few applications of Mg nanoparticles (Mg NPs) in bio-medical have high healing properties promoting the cell growth, cholera detection, pharmaceuticals, and wearable devices to track health Nyabadza, Vazquez, Coyle, Fitzpatrick & Brabazon (2021). Metallic nanoparticles are used in 3D and 4D printing Nyabadza, Kane, Vázquez, Sreenilayam & Brabazon (2021). Mg batteries are in verge of replacing Li batteries because Mg having high volumetric capacity of 3833 mA h/cc while Li has 2046 mA h/cc with environmental friendly nature Saha et al. (2014).

Pulsed laers systems can be which can be briefly classified as self seeded or manually controlled and generation of short pulses like nanosecond (ns), picosecond (ps), femtosecond (fs) Schimkowitsch (2024). Use of Reinforcement learning (RL) in laser can be autonomous for ease in synthesis of nanoparticles.

In present work Mg nanoparticles are manufactured using the PLAL method which is highly sensitive to parameters like fluence, scan speed and ablation time with the help of powders. Variance in these results in range of nanoparticle sizes and morphology, laser type used is picosecond Nd:YAG Nyabadza, Vazquez, Coyle, Fitzpatrick & Brabazon (2021). Further data is collected and analyzed for the nanoparticle size and distributions. Machine learning (ML) is used to predict the nanoparticle size, count and particular model is used as environment for reinforcement learning (RL).

Reinforcement learning is computational method to find the optimal process in stochastic environment. RL involves agent rewards that receives the signal from the environment in the from of states and rewards. The agent send the actions to environment. Advancements in RL like Deep Q-learning (DQN), Deep deterministic policy gradient (DDPG) and Twin delayed deep deterministic (TD3) Schimkowitsch (2024).

RL is a key factor in shaping the smart technology in the upcoming manufacturing industry i.e industry 4.0 and 5.0, promoting or use of Deep reinforcement learning (DRL) techniques and algorithms improves the factories production and efficiency del Real Torres et al. (2022).

0.1.1 Research questions

- Modeling of machine learning techniques to predict the magnesium nanoparticle synthesis?
- How Reinforcement learning can be integrated to yield nanoparticle size and distribution?

Problem definition : The process of manufacturing which involves lot of trial and error methods to understand the output of nanoparticle size and characteristics Costa et al. (2025), hence optimizing them sometimes becomes crucial for the desired output with help of reinforcement learning and machine learning modeling techniques will be ease to achieve them.

This work tries to integrate material science of manufacturing nanoparticles with reinforcement learning from data analytics and machine learning. Mainly reinforcement learning involves the "decision making" based on the "actions" in sequential "environment" to maximize the reward to achieve precise output Zhou et al. (2017). Deep neural networks or reinforcement learning can be trained for single-shot experiments which sometimes becomes hard in conventional approach which involves repeating Tani & Kobayashi (2022).

0.1.2 Technology and Research process used

Pulsed Laser Ablation in Liquid (PLAL) which involves ablating by Nd:YAG laser system with pulses at 1064 nm in isopropyl alcohol for synthesis of Mg NPs. Data is obtained from the experiment by varying 3 * 3 factors like fluence, ablation time, scan speed making 81 samples overall and output measured is nano particle diameter, count and distribution, whereas Dynamic light scattering (DLS) gives the nanoparticle size distribution and mean diameter UV-Vis gives the count of nanoparticles, colloidal density and absorbance are measured.Nyabadza, Vazquez, Coyle, Fitzpatrick & Brabazon (2021).

Using machine learning model for regression analysis which involves using models like Random Forest, Linear regression, Support vector machines, Gradient boosting Butler et al. (2018) also ensemble methods like Support Vector Regression (SVR) which combines the many models to improve the performance Mobarak et al. (2023) to predict the nanoparticle size and count. Models are evaluated based on the Root mean square error (RMSE), Mean Square error (MSE) and R-square values

Reinforcement learning algorithms like Deep Deterministic Policy Gradient (DDPG) or Deep Q-learning (DPQ) can be used to allow agents maximize the rewards based on the environment defined on the machine learning model which involves lot of sequential action Zhou et al. (2017). It develops near-optimal behavior based on the rules provided to system Mobarak et al. (2023). RL agent select the parameters like fluence, ablation time, scan speed where each action lead to experimental output which improves the decision policy over time.

0.1.3 Hypothesis for a solution

Main Research hypothesis: Reinforcement learning agent trained on the surrogate model of the Physical laser ablation in liquid process which automatically optimizes the laser pa-

rameters to synthesize magnesium nanoparticles with required or desired properties.

RL excels in controlling the tasks like flow reactions in chemistry Zhou et al. (2017), autonomous film synthesis like pulsed laser deposition Harris et al. (2024) these domain used non linear experimentation.

Since laser system is controllable by setting the boundaries for the parameters, RL can reduce the cost by making to set optimal conditions for experiments based on decision policy and integrate with the physical system Butler et al. (2018).

If the certain hypothesis becomes validated then the autonomous synthesis of Magnesium nanoparticles can be achieved.

Structure of report includes,

contains data analysis, model building, using the particular model as environment in reinforcement learning.

1. Introduction to Mg NPs and its synthesis, Machine learning and reinforcement learning with technologies used and hypothesis.

2. Literature review, all about topics in introduction.

3. Exploratory data analysis (EDA)

4. Conclusions

0.2 Literature Review

This paper focuses on the four key areas like Laser parameters, characterization techniques, Machine learning working and sub divided into models and reinforcement learning working and further sub divided into models, also discussed the future work and gaps in few research papers. The below mentioned relevance tree gives the information about the literature review structure in brief with the steps involved.

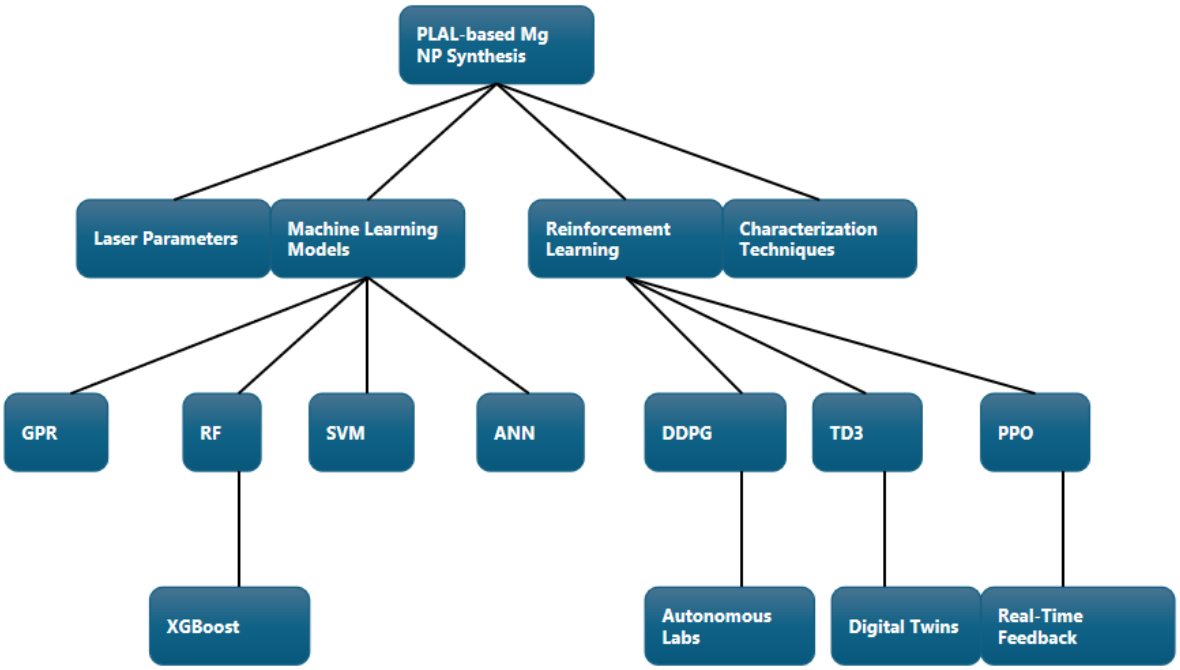

Figure 1: Relevance Tree

0.2.1 Synthesis of nanoparticles from Physical laser ablation in liquid

Among various methods used for synthesis of nanoparticles Physical laser ablation in liquid is well known for environmental friendly and biodegradable nature Phuoc et al. (2008). This process often produced more pure and homogeneous nanoparticlesCosta et al. (2025) .As the term suggests it indicates the process involves ablating of laser with certain parameters like ablation time, fluence, scanspeed Nyabadza, Vazquez, Coyle, Fitzpatrick & Brabazon (2021) also varying or keeping fixed repetition rate Costa et al. (2025). Most of the common laser system used was Nanosecond Nd:YAG 1064nm Nyabadza et al. (2023) Fazio et al. (2020). The solution medium used for synthesis of nanoparticles were different like Distilled water (DW) Nyabadza et al. (2023), isopropyl alcohol (IPA) Nyabadza, Vazquez, Coyle, Fitzpatrick & Brabazon (2021), acetone, Propanol, DI water + SDS Phuoc et al. (2008), Triton X-100 Saedi et al. (2023) and contact pulse duration in nanoseconds (ns) where a solid target submerged in liquid medium is ablated using the intense laser pulses, hence it avoids the forming of toxic materials. Resulted in manufacturing of Mg NPs and Mg alloys. And PLAL is highly flexible which can be used to produce monometallic and bimetallic nanoparticles as well.

The synthesis of nanoparticles by PLAL includes a few majority texts as absorption of laser by target material like Mg, Cu (Copper)Miao et al. (2023), Ag (Gold) Nyabadza et al. (2023) and ejection of ablated material which is mixture of liquid and vapor forming plasma plume. Final stage where the NPs can further grow and coalesce Miao et al. (2023).

0.2.2 Characterization of Mg nano particles

It is crucial to understand the physical and chemical properties of Magnesium nanoparticles synthesized. Most of the common techniques include Dynamic light scattering (DLS) to measure the particle size distribution . It gives information about hydrodynamic diameter and polydispersity index, which is critical for assessing colloidal stability. Transmission Electron Microscopy (TEM) Fazio et al. (2020) and Scanning Electron Microscopy (SEM) Nyabadza, Vazquez, Coyle, Fitzpatrick & Brabazon (2021) showed direct visualizations of nanoparticles morphology and structure. X-Ray diffraction (XRD) showed patterns and crystalline phase of Mg Nanoparticles Costa et al. (2025) also exposure to air showed oxide layer formation Butler et al. (2018). UV-Vis spectroscopy was used for particle size and concentration, other optical properties Nyabadza et al. (2024). UV-Vis absorbance spectra was able to identify peak positions and intensities for particle concentration Nyabadza, Vazquez, Coyle, Fitzpatrick & Brabazon (2021).

Accurate characterization helps to validate the synthesized Mg NPs results and analyzing the parameters chosen for the particular output. The data can be used to train the machine learning model for better prediction. This can also work as feedback for reinforcement learning as well.

0.2.3 Machine learning for Predictive Modeling

Machine learning (ML) emerged as tool to model non linear and complex relationships which was crucial for prediction in chemistry and PLAL techniques involved. There are chances of future breakthroughs in machine learning for automation and discovering different molecules and materials Butler et al. (2018).

Although machine learning require huge datasets to learn efficiently from the data, in process where chemistry and experiments are involved data is limited to hundreds or few sample of experimental data which act as high quality of data points Butler et al. (2018). With the help of methods like Neural Turing Machines (NTM) aid in the prediction Graves et al. (2014).

The training of ML model can be supervised, unsupervised and semi-supervised depending on the data available. In supervised, model training consists of providing the input data and goal of algorithm is to fit a function that predicts the output variable which can be single or multiple. Whereas, unsupervised is used to predict the underlying pattern in the dataset. Semi-supervised is used when large amount of input data which doesn't become relevant this work.

Some of the model that are used in this field include,

- **Gaussian process regression (GPR)**: This machine learning model was used to integrate into robotics for thin film deposition making it fully autonomous by optimizing the parameter space. The system fabricates thin silver films with an average of 2.3 attempts which reduced the labor costs Zheng et al. (2024)

- **Multiple linear regression (MLR)**: It is statistical method to find the relationship between one dependent variable and two or more independent variables, it is similar to linear regression but with multiple predictors. It adjusts the data to linear equation,

but the non linear nature of the dependent variable might affect the model. It is often used as a baseline comparison for the other non linear models like Random forest, XGBoost, Support vector regression Nyabadza & Brabazon (2025).

- **Support Vector Machines (SVM)**: This models used when the tasks or classification of data to determine the outcome by predefined quality. It projects the high dimensional data spaces from smaller dimensions makes it suitable for PLAL experiment data.

- **Support Vector Regression (SVR)**: A regression model similar to SVM, which works by mapping input features to high dimensional space with the help of kernel features. It is highly efficient for small datasets and non-linear relationships. SVR was used to analyze RC computation where the model was trained both on the scanning electron microscope (SEM) images and the experimental data parameters like laser beam diameter Liu et al. (2022).

  The integration of ML model with molecular dynamics is used for femtosecond laser precision drilling. The SVR model is trained by the data obtained from the molecular dynamics which includes laser parameters and quality machinery. And radial bias function (RBF) kernel is used which is suitable for non linear problems with grid search method Wang et al. (2022).

- **Random Forest Regression(RF)**: It is a ensemble learning algorithm which works by constructing a large number of decision tress and picks up the best one which has the maximum votes and trains the model. It is essential in capturing feature importance, and making it valuable for nanoparticle characteristics. RFR is used to find the best correlation with the experimental output features and identify the equation for LAL results. With the help of RFR Cu nanoparticle were synthesized Miao et al. (2023).

- **XGBoost**: It represents extreme gradient boosting, which is ensemble learning algorithm known for speed and performance. It works similar to random forest by building decision tress, in a sequential manner but this model corrects the error of previous one. Methods like regularization and parallel processing increases its efficiencies and effectiveness in various machine learning tasks with user specified loss-function Tsai & Yiu (2023).

- **Artificial neural Networks (ANNs)**: ANNS and deep neural networks which works similar to how human brain works with neurons. It includes arranged input, output and hidden layers. In hidden layers, where multiple neurons interact with each other and result in computation Butler et al. (2018). With the help of deep neural networks material properties and forecast size distributions of PLAL synthesis MgNPs. Data simulations and images from SEM can be used to train the model. The ANN works without the need of modeling of physics, the mathematical model build was successfully able to predict the nanoparticle size of polymers which where used in pharmaceutical by identifying just the surface activity, viscosity and hydrophobicity Youshia et al. (2017).

- **k-nearest neighbor (k-NN)**: This model takes the input and predicts the output with the nearest one by calculating the distance between samples and training data. The nearest point is called the k nearest neighbor. Hence the name k nearest neighbor. The output prediction depends on the input points, many methods are used to calculate the distance between them but euclidean distance is most commonly used Nyabadza & Brabazon (2025).

Each models offer unique strengths depending on data quality and size. Hybrid approaches and combining models like GPR and RFR with Bayesian optimization can predict output more reliably. Models like RF and XGBoost are well suited for small to medium sized datasets similar to PLAL experiments. SVR perform well but fails at interpreting Nyabadza & Brabazon (2025). But unfortunately ANNs are suitable for large datasets.Further steps involves the integration of machine learning model with the reinforcement learning.

0.2.4 Surrogate data modeling

In case of smaller datasets obtained from the experiment methods like surrogate modeling can help achieve the larger datasets that mimics the experimental datasets. Hence to tackle limitations non linearity, less data and high dimensionality the use of RF to build surrogate dataset is a feasible which explores the design space of experiments Dasari et al. (2019). A similar method of data augmentation was used in Nyabadza & Brabazon (2025) to increase the dataset size to attain high performance, where the samples augmented to 213 samples from 81 samples by which the model showed high accuracy.

0.2.5 Reinforcement learning for optimizing synthesis of Mg Nanoparticles

Reinforcement learning focuses mainly on trial and error method and optimal control. Recently, RL is considered to be one of the three main machine learning process, with supervised and unsupervised learning methods del Real Torres et al. (2022). RL learns through iterative process, by interacting between agent and environment through actions which is the decision taken by agent based on the signal received from the states based on the current state and reward gained from that state Schimkowitsch (2024) this is also called Markov Decision Process (MDP) as shown in the figure 2. The overall task is to maximize the cumulative reward in the long run, it is done in two ways by state value function for policy and action value function for policy based on the accumulative reward of state and action state del Real Torres et al. (2022). With the increase in higher dimensionality and complexity in data, deep neural networks were developed leading to deep reinforcement learning (DRL). The figure 3 shows the various algorithms used in DRL. In PLAL the approximate goal can be to maximize the reward by achieving desired nanoparticle size.

In research paper Zhou et al. (2017), DRL agent is used to determine coupled laser parameters and laser duration with Deep Q-Networks (DQNs) and LSTM policies to optimize chemical reactions in reinforcement learning utility for sequential decision tasks.

In research paper Schimkowitsch (2024), using RL in pulse generated laser system with cavity dumping. To benchmark, RL approach is compared with the adaptive non linear

control. Overall, RL algorithms performed well in small samples while controller worked in large samples. But RL were able to stabilize the set-point with the help of optimal decision policy without any prior knowledge about the system and its working.

RL was used to stabilize the self adjusting laser system by optimizing the laser radiation controlling the laser plasma source and system autonomously adapting to changing environmental conditions. The feedback was given from the second harmonic of femtosecond laser radiation, and adjusting with the help of neural networks Mareev et al. (2023).

In manufacturing industry which involves decision making by operators. Tasks involved are rescheduling, supply chain management, process control and monitoring all of which are complex and time restricted. With the help of ML and RL process optimization can be achieved ahead of difficulties like adaption to environment, high -dimensionality, extensive prior knowledge of the system. In case of RL issued like stability, sample efficiency, sparse reward del Real Torres et al. (2022).

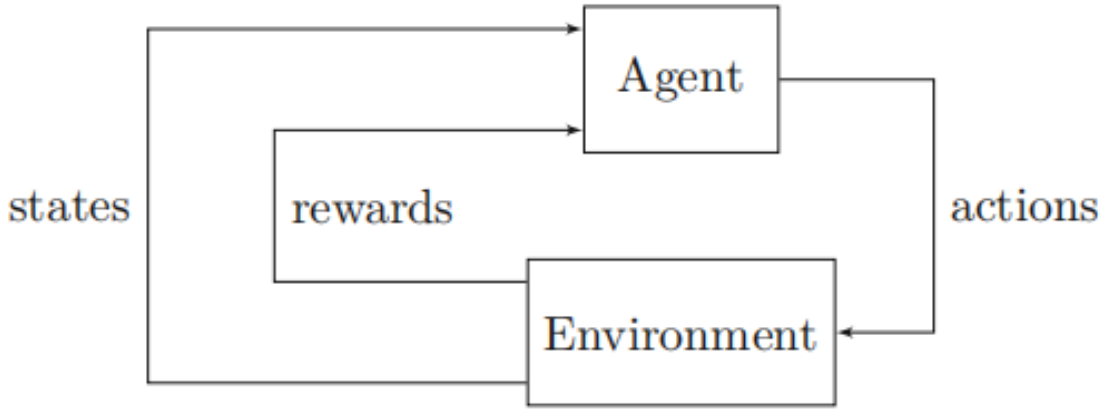

Figure 2: Reinforcement learning framework Schimkowitsch (2024)

Actor critic algorithm, which is used in models like DDPG works by splitting the tasks of RL. The critic gets trained from the state, reward and error. The actor tries to optimize the decision policy parameters which is shown in the figure 4.

Some of the algorithms that are useful in PLAL are discussed,

- DDPG (Deep Deterministic Policy Gradient): It is used commonly for control tasks which involves continuous action space like chemical process Ma et al. (2019) and laser system.It is a model free algorithm that can be used to control fluence, scan speed, duration and repetition rate.

- TD3 (Twin Delayed Deep Deterministic Policy Gradient): It is considered to be optimized version of DDPG, like where multiple agents can be trained to control multi loop process which optimizes the overestimation of value function of DDPG Yifei & Lakshminarayanan (2022).

In reference to chemical industry, DDPG and TD3 have outperformed other algorithms in process control in improving the stability and sample efficiency del Real Torres et al. (2022).

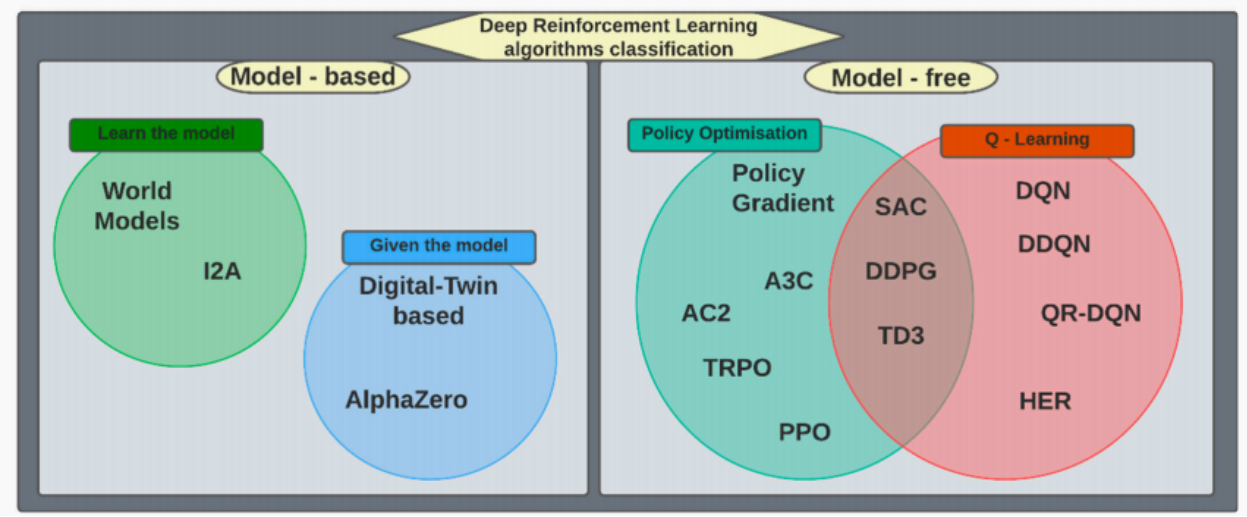

Figure 3: Deep reinforcement learning (DRL) algorithms
del Real Torres et al. (2022)

In summary,ML and RL can outperform human ability in synthesis of Magnesium nanoparticles with Physical laser ablation in liquid. With the model trained on various algorithms and parameters, also improving the RL algorithm in decision making through optimization this can lead a stepping stone in manufacturing industry 4.0 and 5.0.

0.3 Data and Methods

0.3.1 Experimental Setup

Pure Mg powder of 99 % was ablated with the isopropyl alcohol (IPA) rather than water as Mg powder when reacted with water forms the foaming or bubbles but in IPA it becomes non reactive which aids the further experiment. The beaker of 60 millimeter (mm) was taken and evenly spread the Mg powder of about 2 mm and covered the powder with approximately 3 mm IPA. The laser system used was Nd:YAG with constant operating it at 1064 nanometer, 10 Kilo-Hertz repetition rate and 600 ps pulse width. Later the design of experiments used 3*3 set of designs varying the ablation time (2, 5, 25 minutes), fluence (1.83, 1.88, 1.91 J/cm^2), scanspeed (3000, 3250, 3500 mm/s) and repeated them 3 times which resulted in total of 81 samples Nyabadza, Vazquez, Coyle, Fitzpatrick & Brabazon (2021).

Characterization: By using the Zetasizer Nano ZS system on each sample the output of raw data of nanoparticle size distribution was obtained, and Ultraviolet–Visible Spectroscopy (UV–Vis) was used to measure the nanoparticles count Nyabadza, Vazquez, Coyle, Fitzpatrick & Brabazon (2021).

0.3.2 Exploratory Data Analysis (EDA)

The dataset obtained from the experiment includes features like ablation time in minutes, laser scanspeed (mm/s), Fluence (J/cm^2), repetition rate and pulse-width is which is kept

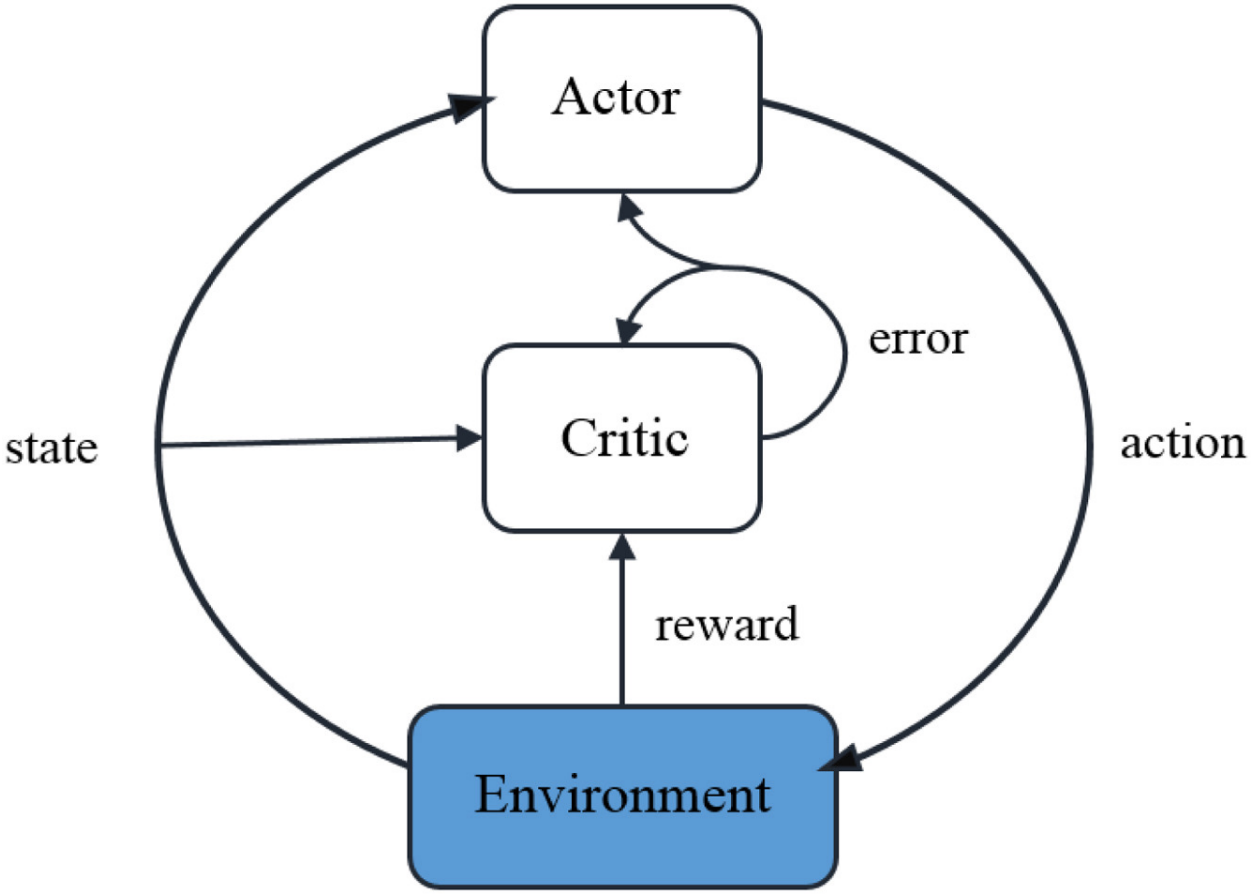

Figure 4: Actor-critic framework
Ma et al. (2019)

constant of 10 (kHz) and 0.6 respectively, pulse-width, Power %.The output features include DLS(nm), UV peak(nm) and UV Vis.

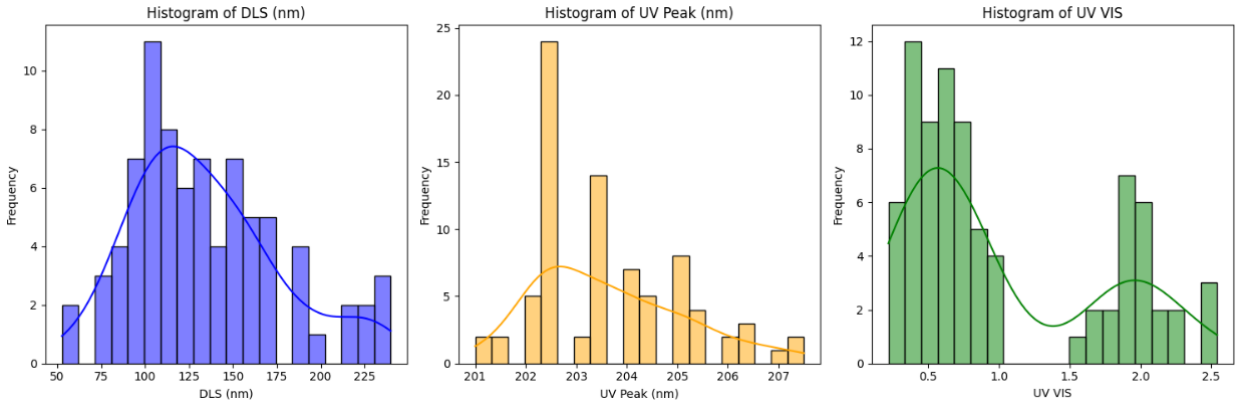

Figure 5: Histogram of DLS, UV peak and UV Vis

By analyzing the histogram plots 5, DLS is slightly right skewed with normal distribution, UV Peak with right skewed and UV Vis is bimodal distribution.

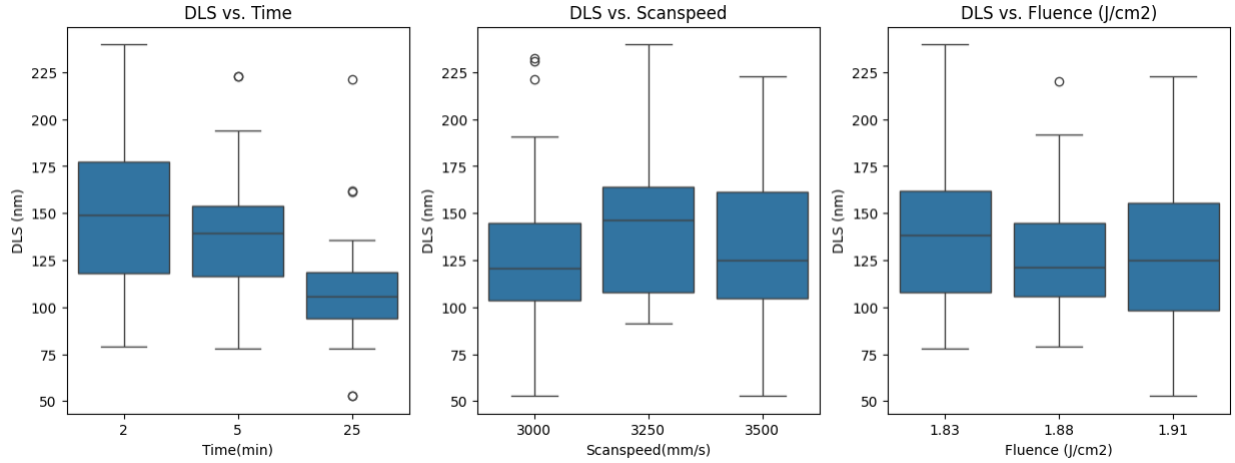

Figure 6: Boxplot of nanoparticle size with laser parameters

By analyzing the boxplot of Nanoparticle size with other parameters as shown in the fig 6,

- **With Ablation time**: For 2 min it showed normal distribution, as the ablation increased from 2 to 5 min it became slightly skewed and for 5 to 25 min distribution is highly right skewed. And the overall mean diameter size also decreases.

  So as the ablation time increases the overall nanoparticle size decreases.

- **With Scanspeed**: At 3000 mm/s, distribution is slightly skewed with outliers but shows normal distribution. When increased to 3250 mm/s, distribution becomes right skewed with increase in mean diameter. At 3500 mm/s, the distribution becomes normal with mean of around 125 nm.

- **With Fluence**: When fluence is 1.83 J/cm^2, it shows rightly skewed, when increased to 1.88 J/cm^2, skewness decreases and at maximum of 1.91 J/cm^2 normal distribution is observed.

Other than ablation time which displayed slight trend in the nanoparticle size other parameters did not have trend. Hence, the nanoparticle size can be affected by other parameters as well.

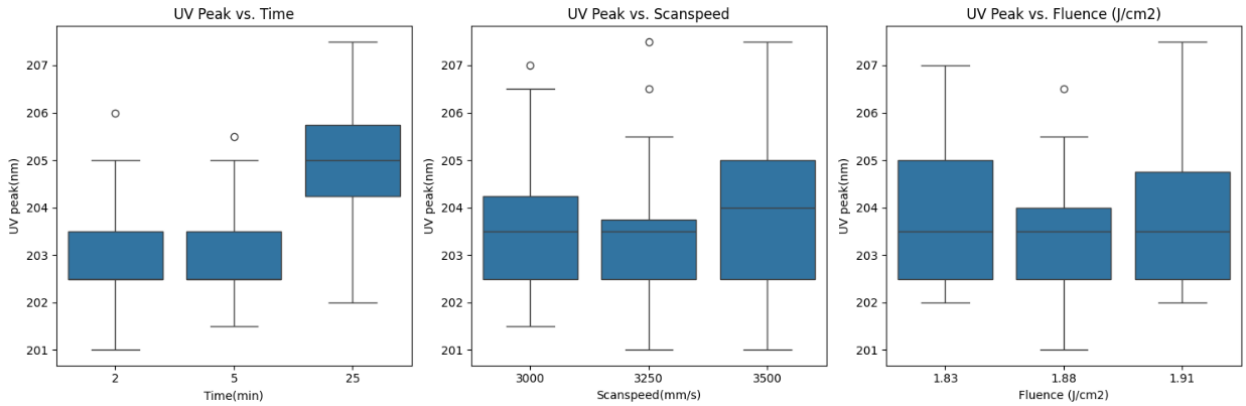

Figure 7: Boxplot of UV peak with laser parameters

By analyzing the boxplot of UV peak shown in the figure 7, ablation time at 2 and 5 min shows slightly right skewed and at 25 min normal distribution. For scanspeed it does not show any trend. In case of fluence the overall mean remains same with varying distribution of right and normal.

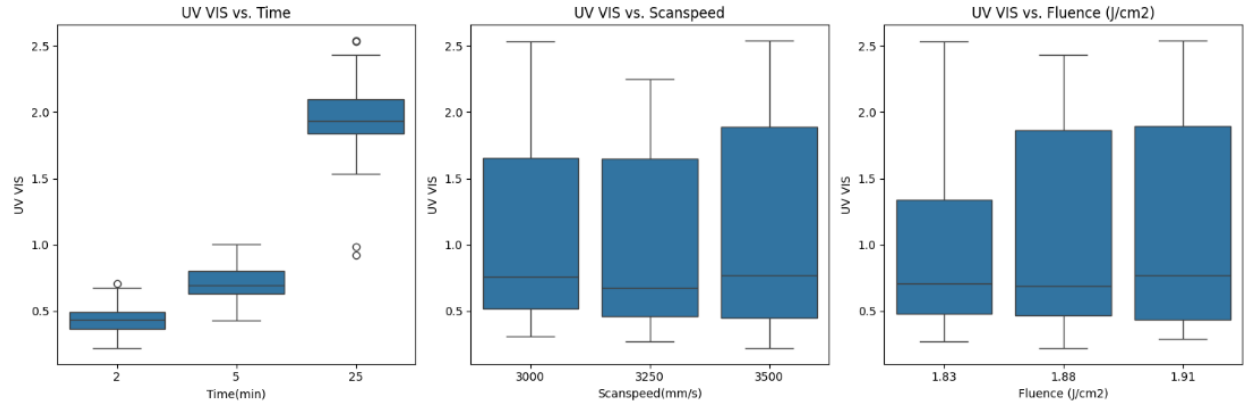

Figure 8: Boxplot of UV Vis with other laser parameters

By analyzing the boxplot as shown in the figure 8, the mean of UV Vis with ablation time increases from 2 to 5 min, and 5 to 25 min shows significant increase with the normal distribution. Whereas, scanspeed and fluence does not show any better variation or significant trend.

The correlation heatmap as shown in figure 9 is used to analyze the interaction between features.

1. As discussed from the box plot, time is negatively correlated with nanoparticle size.
2. UV Vis is strongly correlated with time which can be used to predict the count of nanoparticles.

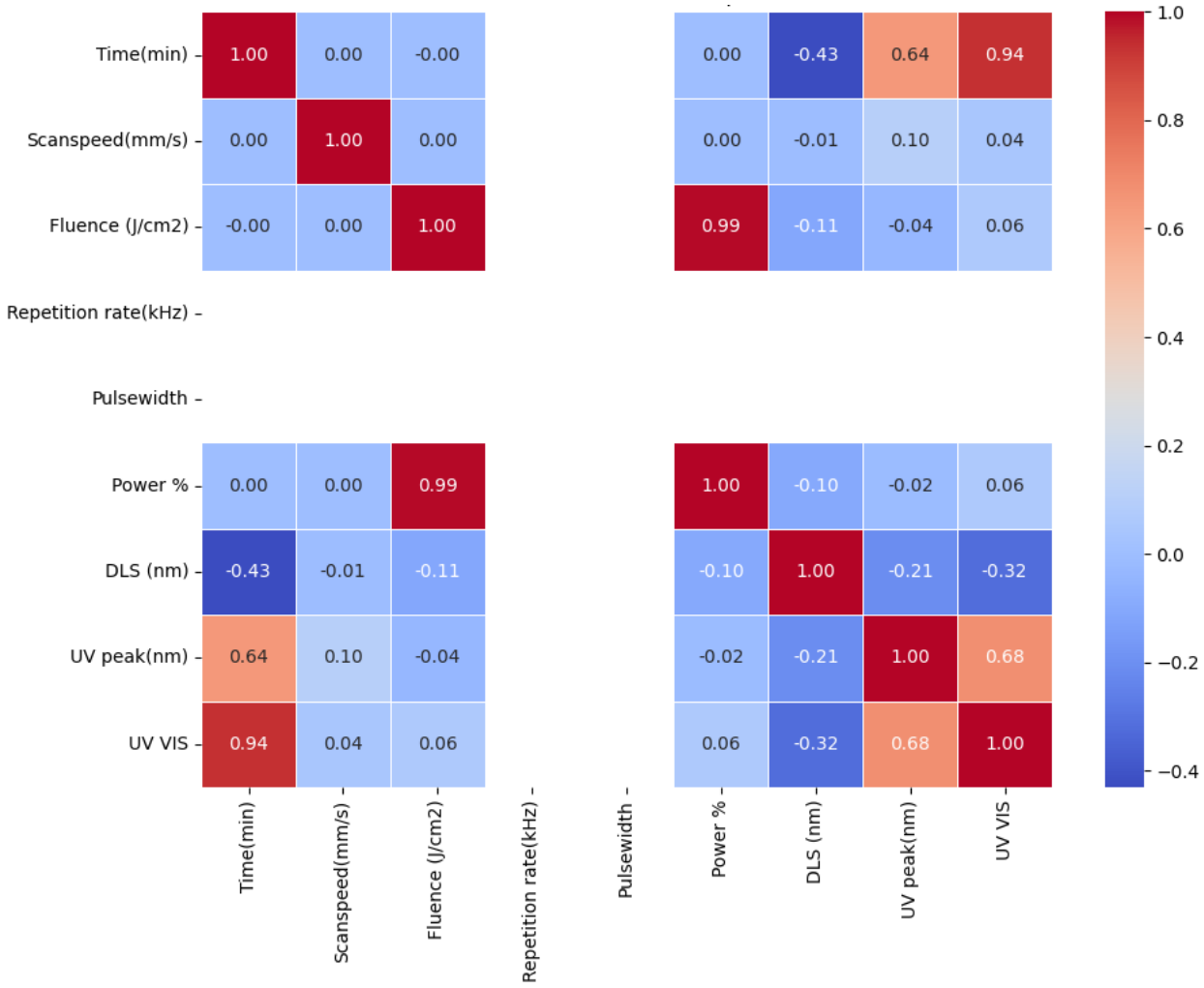

Figure 9: Correlation heatmap of features

In the figure 10, shows the absorbed UV Vis, according to nanoparticle size and power.

In the figure 11, shows the nanoparticle size vs ablation time with UV Vis. From the figure, higher the ablation time of particle the more absorbance is seen.

In the figure 12, it can be depicted that nanoparticle size increase in size from 2 min to 5 min ablation time for highest fluence 1.91, but for the other 2 it showed decrease in trend. For 5 min to 25 min it showed decreased in trend in all cases.

0.3.3 Feature selection

Performing tests like Principal component analysis and Random forest regressor to see the feature importance. Similar results is shown in the figure 13. From the feature elimination

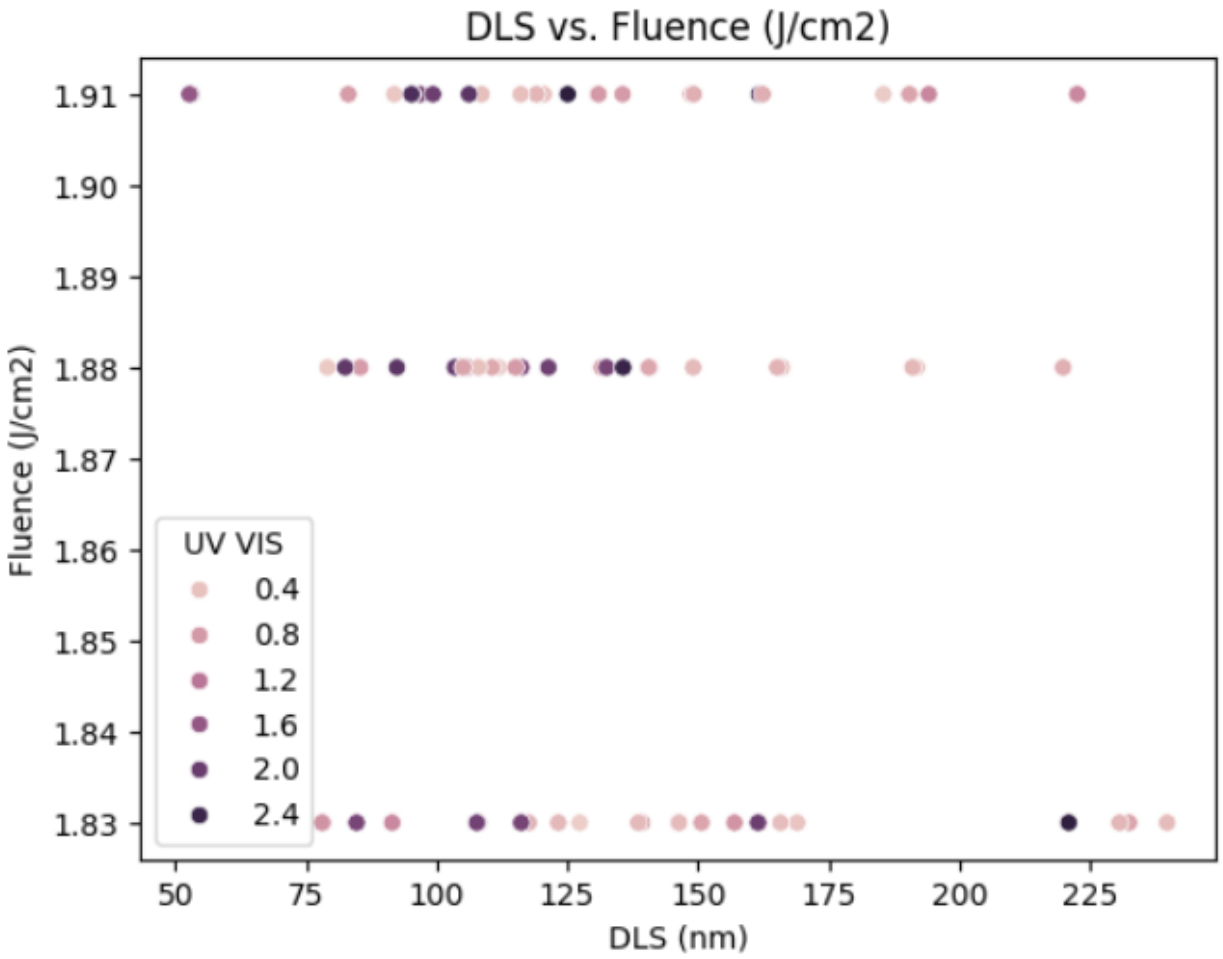

Figure 10: DLS vs Fluence by UV Vis

it states only 3 features are responsible for prediction

0.3.4 Machine learning Algorithms

After performing the EDA, next step involves the choosing of machine learning model the can predict the output most accurately. In total 4 types of model was used to train the model which are MLR, RF, XGBoost and SVR. Many types of evaluation metrics can be used to validate the output but in this study R2 or R square or R-sq is used. Each model was trained on the same set of data points which were taken from the original dataset.

The first step is to drop all the output features in the input features used to train the model. Setting the target feature as DLS to predict the nanoparticle size, the data is split into test train data, where 80 percent is train data and 20 percent is test data. After splitting the data, all the machine learning models are trained on the train data using the for loop which helps to compare the results of each model in a simple way. The 5 fold cross validation with R-square is used to check the accuracy. The same process is repeated for the other features which are UV Vis and UV peak, to see how input features are related to it.

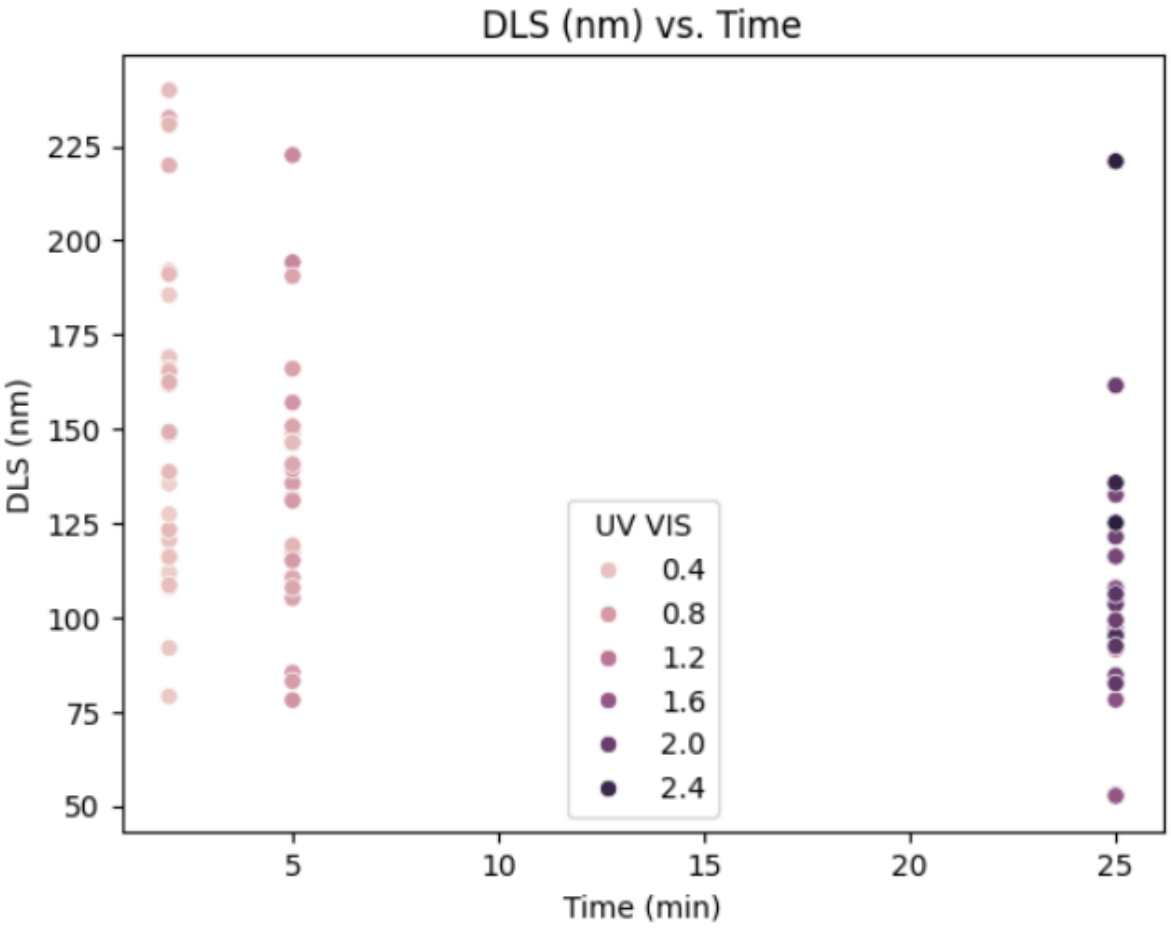

Figure 11: DLS vs Time by UV Vis

MLR

It works by forming the relation by the help of below mentioned mathematical equation. It works well for variables that are linearly related but faces few drawbacks in the non linear relationships and in cases of multicollinearity. It assumes the residuals to be normally distributed.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \epsilon \tag{1}$$

where:

- y = dependent variable (response).
- β_0 = intercept (the expected value of y when all predictors are equal to 0).
- β_j = regression coefficient associated with predictor x_j; it represents the effect of a one-unit change in x_j on y, while holding all other predictors constant.
- x_j = independent variables (predictors).
- ϵ = error term, where the variability in y is not explained by any of the predictors.

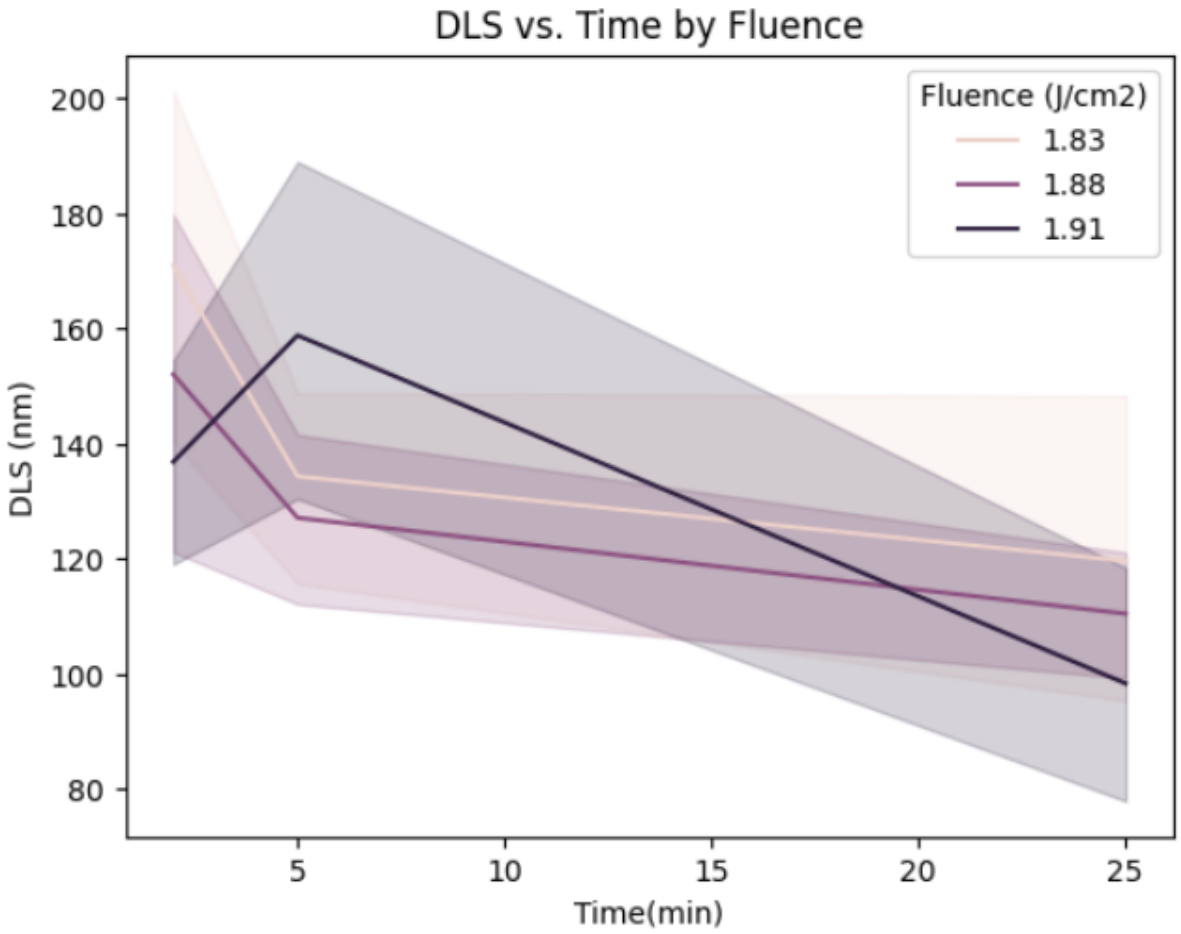

Figure 12: DLS vs Time by fluence

Random forest

As explained this model picks the best decision tree correcting the previous errors, this methods becomes highly effective in the regression tasks like this, are expected to make low errors Nyabadza & Brabazon (2025). It effective in handling non linear relationship well with the robustness.

$$\hat{y} = \frac{1}{T} \sum_{t=1}^{T} f_t(x) \tag{2}$$

where:

- T = total number of trees in the forest,
- $f_t(x)$ = prediction from the t-th decision tree,
- $\hat{y}$ = final averaged prediction across all trees.

XGBoost

It similar to RF but this does the building of trees sequentially instead of averaging the outputs and each tree hence evaluating both works is crucial for this work Nyabadza & Brabazon (2025).

The evaluation metrics R-square is used which is statistical measure of variance of dependent variable by the predictor or predictors. It is calculated as one minus (RSS/TSS), where RSS is the residual sum of squares and TSS is the total sum of squares. But in case of multiple linear regression it reflects the fraction of variance explained by all the input features used.

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}} \tag{3}$$

where

$$SS_{\text{res}} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{4}$$

$$SS_{\text{tot}} = \sum_{i=1}^{n} (y_i - \bar{y})^2 \tag{5}$$

0.3.5 Surrogate modeling using Random forest regressor

In the engineering field, the experimental data to obtain sometimes requires huge amount of resources depending upon the materials to be used. Since to reduce the cost of procedure, model simulations are used to generate the synthetic dataset from the given dataset. A similar approach was used with the help of data augmentation to increase the size of dataset to achieve higher accuracy Nyabadza & Brabazon (2025).

In PLAL method, to try many number of trails on the parameters like time, scan speed and fluence to determine the outputs like DLS(nm) which are non linear in nature is often the challenging task.

Surrogate model provides the data by imitating the complex pattern in the data. Providing the large design of experiments from the limited data but the model needs to be trained on the specified model to learn the underlying pattern in the data. A similar method was used in exploring the data of aerospace turbine Dasari et al. (2019) and flood inundation simulations Sasanapuri et al. (2025).

In this work, outputs like DLS particle size, UV Vis and UV peak and inputs like time, scanspeed and fluence are continuous variables so using regressor works better than the classifier which is designed mainly for categorical outputs. Some other reasons to include the random forest regressor is to counter the relationship between laser input parameters and nanoparticle size distribution which are highly non-linear. Random forest regressor as surrogate model maps the 3 inputs to each output with 3 separate models. Each model creates 100 decision tress (n-estimators=100).

For each output, random forest regressor was fitted with the experimental data. Internally, the model learns to reduction the prediction error using the mean square error. In

regression, the predicted outcome is the mean of all individual tree predictions, often improving the performances in general. In summary, Random forest build the ensemble of models to map inputs to output specified with underlying function.

As from the EDA it was observed that longer ablation time and higher fluence affected the nanoparticle size to increase in experimental data, the RF model captured the other patterns and perform well on small datasets, it avoids overfitting and improve generalization than decision tree Nyabadza & Brabazon (2025). For default parameter of 100 decision trees the model performed well.

Exploration of design space

After fitting the model it allows to explore the parameter space and synthesize the data or predict it accurately, the synthetic dataset created has some of the parameters to consider:

- For each input feature, the minimum and maximum values of the original data is considered with command (np.linspace), by selecting 7 points for each variable including the extreme points.
- The cartesian product of all these combinations which is 7*7*7 = 343, which covers whole range of experimental conditions. So the model generated the synthetic dataset with certain time, scanspeed and fluence for the outputs DLS, UV Vis and UV peak.

The surrogate model interpolate the experimental values similar to dense grid of experiments, but these are model generated prediction by analyzing the trends in the given dataset which help us in training the model further on the generated dataset.

Some of the advantages of using this surrogate approach are:

- To capture the complex relationships, RF learns the underlying pattern between nanoparticle size and input features.
- The model after getting trained generates desired number of data which help in further training or simulations. Whereas, RF surrogate model achieved predictions with error of about 2% in the flood simulation model in comparison with physics model Sasanapuri et al. (2025).
- RF surrogate model was observed to perform well with small datasets provided by learning the key patterns, which gives enough reliability to proceed further in training the model Dasari et al. (2019).

Hence, during challenges like smaller dataset, to solve non-linear relationships, to create stable data for the model to train, to minimal pre-processing methods surrogate models are used Dasari et al. (2019). The merging of experimental data with the data driven modeling approach is useful and commonly used in material science and chemical based approach Nyabadza & Brabazon (2025).

EDA of the surrogate model dataset

After performing the surrogate method, to verify the distribution of data the EDA was performed again on the new dataset and observed the histogram in the figure 14 , distribution almost remained same to previous one.

The correlation map when compared with original dataset and the surrogate model dataset, the correlation between the variables remained same as shown in the fig 15.

0.3.6 Reinforcement learning

RL is computational method to find the optimal action in well defined unknown and stochastic environment which is pre defined by the user Schimkowitsch (2024). It works similar to unsupervised learning but unlike it relies on the labeled datasets while RL learn from the experiences by trial and error method basically by performing actions, observing the outcomes and receiving rewards. This is the natural learning process which is evolved by humans to play, walk or practice and improve based on the feedback like a child.

In our case, the random forest model trained on the surrogate dataset is used as the environment, the RL agent after receiving the command for the desired DLS proposes the parameter settings like time, scanspeed and fluence required to manufacture the nanoparticle and model evaluates it through reward function.

The parameter space defined by RL framework includes random forest model as environment,

- **State**: The space where continuous value vectors are defined such as time, scanspeed and fluence.
- **Action**: The adjustments to parameter values takes place, as the variables are continuous the action space is also continuous hence methods like Q-learning does not apply here.
- **Reward function**: It works as feedback signal as the negative absolute error between the predicted nanoparticle size and user defined size.

$$J(\pi) = \mathbb{E}_{s_t, a_t \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t \, R(s_t, a_t) \right] \tag{6}$$

π **(policy).** the parametric policy $\pi_\theta(a \mid s)$, typically like a neural network that adjusts the continous variables.

s_t **(state at time t).** Current process setting,

$$s_t = \begin{bmatrix} \text{Time (min)} \\ \text{Scan Speed (mm/s)} \\ \text{Fluence (J/cm}^2\text{)} \end{bmatrix} \in \mathbb{R}^3.$$

a_t **(action at time t).** Change applied to the setting (continuous control),

$$a_t = \begin{bmatrix} \Delta\text{Time} \\ \Delta\text{Scan Speed} \\ \Delta\text{Fluence} \end{bmatrix} \in \mathbb{R}^3.$$

$\mathbb{E}_{s_t,a_t\sim\pi}[\cdot\ ]$ Expectation over trajectories when following policy π, averaging over random initial states. For deterministic transitions $s_{t+1} = f(s_t, a_t)$,

$$P(s_{t+1} \mid s_t, a_t) = \delta\big(s_{t+1} - f(s_t, a_t)\big).$$

$R(s_t, a_t)$ **(immediate reward)** : Scalar feedback at step t.

Usually, in RL framework the agent receives the signals from environment and sends the action based on the current state while the policy of the agent is updates using the reward function Schimkowitsch (2024). Initially RL was studied with the discrete action space by directly applying the Q-learning but in some cases like laser parameter processing continuous action space Q-learning does not work hence the actor-critic algorithm has been developed Ma et al. (2019) Lillicrap et al. (2015).

0.3.7 DDPG

In the actor critic network the learning process is divided into two steps as,

- **Actor**: Optimizing the policy
$$\pi_\theta(a \mid s)$$
that maps the states to action by the values from the critic.
- **Critic**: It is trained to learn a state or state-action Schimkowitsch (2024). It basically criticize the actor based on the action.

Usually the DDPG which uses the Deep Q-network that solves various challenging tasks by creating the agents in high dimensional state spaces and discrete low dimensional action spaces. Deep Q-network can not be used for continuous action spaces Tan (2021). However, Lillicrap et al. (2015) adapted continuous action space to DDPG by using the replay buffers.

In case of PLAL, for high dimensional control of continuous variables shows we can use the optimization of deterministic policy Silver et al. (2014) as shown in equation below. DDPG works better when the dimensionality of action are less and the environment is deterministic .

$$\nabla_\theta J(\mu_\theta) = \mathbb{E}_{s\sim D}\Big[\nabla_\theta \mu_\theta(s)\ \nabla_a Q^\mu(s,a)\big|_{a=\mu_\theta(s)}\Big] \tag{7}$$

Silver et al. (2014)

The DDPG algorithm assumes the sample to be independent and identically distributed but when the exploration happens sequentially this assumption fails hence the parameter called replay buffer is introduced. The replay buffer uses first in first out memory that stores

the tuples of states and actions $(s_t,\ a_t,\ r_t,\ s_{t+1})$ in this way so the agent can learn off-policy in mini-batches which are drawn uniformly and break temporal correlations and this technique is also used in TD3 model as well. At each time step the actor and critic gets updated from the samples through replay buffer Lillicrap et al. (2015). The control variables are time, scanspeed and fluence which are continuous and bounded and environment is deterministic, while DDPG outputs continuous actions and learns quickly in these scenarios. The working algorithm of DDPG is shown in the below flowchart Tan (2021).

Steps involves in the prediction of inputs by DDPG

To build gym environment using the model:

Setting the RF model as environment obtained from training the model. To define the action space by setting the limits which helps the model to predict in the mentioned boundaries. Time is between 2 to 25 minutes, scanspeed is between 3000 to 3500 mm/s and Fluence is between 1.83 J/cm^2 to 1.91 J/cm^2. State is any current parameter set that has these three values. The reward function (r) is set with negative normalized absolute error,

$$r \ = \ - \frac{|\,\text{pred\_DLS} - \text{target\_DLS}\,|}{203 - 83}$$

the denominator is set with the DLS range to shrink the reward range and the error is converted into signal form. The reward tells the agent exactly how well the action was done.

Replay buffer:

It stabilizes the model by sampling random mini batches and breaks the pattern of temporal correlation by storing the actions and states. It discards the oldest item when pushing the new item by storing only one transition per cell which is usually s, a, r, s', d (state, action, reward, next-state, terminal flag). With the help of np array it converts the component by mapping them in mini batches and soft-update the targets. A multi layer perceptron (MLP) is used to create feed forward neural network while the actor maps the 3 dimensional (3-D) to 3-D action, while the critic maps state and action to $Q(s, a)$. The MLP has ReLU activations between hidden layers like

- Layer1: input-dim*256 + 256
- Layer2: 256*256 + 256 = 65,792
- Layer3: 256*output-dim + output-dim
- Total = 256(input-dim+1) + 65,792 + (output-dim)(256+1)

overall this helps the DDPG stability.

DDPG Agent:

At first initialization is made for actor to map states with the actions and critic takes state, action concatenated to scalar $Q(s, a)$. The target networks create clones at start and soft update to stabilize the target in bellman equation. For optimization the critic gets a higher learning rate of $1e-3$ while the actor learning rate is $1e-4$ as the critic must learn quickly so the actor can have the reliable gradient to follow the reward Lillicrap et al. (2015).

The gamma value is set to 0.99 for future discount returns and tau value 0.005 for soft target updates.

Action selection is made by computing the deterministic action, and adding gaussian noise to the action. Further the action is clipped with the mentioned bounds to perform well in the training and make DDPG explore.

Training the agent involves first to collect enough data to form the samples of mini batches from the buffer avoiding the small transactions. At this step temporal correlations is broken down in on-policy updates and off-policy learning is enable which make the core characterization of DDPG. After this, the shapes are transformed as (state, next state) to (B, state dim), action to (B, action dim), (reward, done) to (B, 1).

With the help of Bellman equation, the target update for the critic is made by the equation mentioned below

$$y \;=\; r \;+\; \gamma(1-d)\, Q_{\phi'}\big(s',\, \mu_{\theta'}(s')\big) \tag{8}$$

where d is done, when d = 1 bootstrapping is turned off, this prevents the learning target to chase itself. Keeping the target motion slow the y becomes stable during training as actor and the critic change.

The critic update, computes the prediction $Q_\phi(s, a)$ for mini batch size of B by concatenation. The bellman regression loss of critic is calculated with the mean squared error against the target y.

The actor update, it aims the maximize the function $Q(s, \mu_\theta(s))$ and minimize its negative it implements the DDPG policy gradient

$$\nabla_\theta J(\mu_\theta) = \mathbb{E}_{s\sim D}\Big[\nabla_\theta \mu_\theta(s)\; \nabla_a Q_\phi(s,a)\big|_{a=\mu_\theta(s)}\Big] \tag{9}$$

the gradient flows through critic input and action back into actor hence its called auto-gradient function. Further polyak averaging is used to update the target values to stabilize the bootstrapped target networks $(\mu_{\theta'}, Q_{\phi'})$. At each training step, the critic target is computed with the target networks. The target parameters are updated with polyak averaging 10 these low pass filters the target by reducing the variance and stabilize the bellman equation,

$$\theta' \leftarrow \tau\,\theta + (1-\tau)\,\theta', \qquad \phi' \leftarrow \tau\,\phi + (1-\tau)\,\phi' \tag{10}$$

with $\tau \in (0, 1)$ (we use $\tau = 0.005$) Lillicrap et al. (2015).

In training loop, episode is initialized by setting the target DLS to 150 nm (any user defined value) then at time t the actor initiates the deterministic action $a_t = \mu_\theta(s_t)$ by adding gaussian noise and clips the boundaries while the environment returns (s_{t+1}, r_t, d_t), where d_t indicates termination. Reward function makes the agent closer to target DLS, replay buffer stores the experience and updates the values. Further stabilized with the help of polyak averaging.

0.3.8 TD3

TD3 works in a similar way of DDPG which is specialized in continuous action spaces, where the actor network learns from the policy and critic learns from the Q function, TD3 is like

a extension to DDPG in a data architecture way. However the DDPG has drawbacks like instability and overestimation bias Fujimoto et al. (2018), TD3 addresses these issues and makes key modifications by enabling the twin critic networks to stabilize the training and improve the efficiency:

Twin critic network policy: The DDPG model can have overestimation bias which is found in single critic models, hence the use of twin critic structure where each of the critic $Q_{\phi_1}(s,a)$ and $Q_{\phi_2}(s,a)$ receives the concatenated state action pair as input and produces two separate Q-value estimations Shen (2024). It is also known as clipped double Q-learning model, this may add the underestimation bias but it is better than the overestimation bias.

$$y \;=\; r \;+\; \gamma \min_{i=1,2} Q_{\phi_i'}\big(s',\, \mu_{\theta'}(s')\big) \tag{11}$$

Fujimoto et al. (2018)

For the same target considering $y_2 = y_1$, the same actor is optimized with Q_ϕ. If $Q_{\phi 1} > Q_{\phi 2}$ it detects the overestimation and reduces the values when $Q_{\phi 1} < Q_{\phi 2}$ the update is identical and no bias is introduced.

Target policy smoothing: It is the characteristic feature of TD3 model which adds stability to the model during the training. Sometimes the deterministic policy narrows the learning process causing the overfitting of the model hence the TD3 model introduces the accurate noise into the process during the action selection, addition of Gaussian noise in the critic updates by clipping thereby increasing the exploration. The noise helps in increases the convergence and smoothens the estimations Shen (2024).

$$\tilde{a}_{t+1} \;=\; \mu_{\theta'}(s') \;+\; \epsilon, \qquad \epsilon \sim \mathrm{clip}(\mathcal{N}(0,\sigma),\, -c,\, c)\,. \tag{12}$$

Fujimoto et al. (2018)

Delayed policy updates: In case of DDPG model the updates of critic and actor network at each step, while TD3 model updates the actor too quickly based on the critic that have not yet converged yet. The target networks cause the increase in volatility in convergent behaviors in fixed policy but in fast-updating environment it showed highly divergent behavior when this happens the overestimation caused the model to fail and policy will become poor with inaccurate value. Hence the model updating the policy network with lower frequency reduces the error Fujimoto et al. (2018).TD3 model delays the actor updates, typically updating the actor once over the two critic updates allowing more consistent and reliable updates to stabilize the learning and smooth convergence Shen (2024).

TD3 algorithm working

At first the model initializes the three sets of neural networks, two critic networks and actor network. The TD3 selects the actions with the help of actor network by adding the noise, selecting actions and observing the rewards obtained.

At each step the critic update is made, the sample batch of transitions like $s, a, r, s', d)$ are stored as tuples in the replay buffer for further training. The minimum value of Q is calculated for the target value y based on the two critics

$$y \;=\; r \;+\; \gamma \min_{i\in\{1,2\}} Q_{\phi_i'}(s',\, a').$$

, further the minimization of the mean squared error loss between $Q_{\phi_i}(s, a)$ and y Shen (2024).

The actor update takes place every d steps, it is done based upon the deterministic policy Q, the gradient policy is updated with respect to its parameters that maximizes the function Shen (2024). It also soft updates the target similar to DDPG model but with the delayed networks.

Steps involves in predicting the inputs by TD3

The action bounds for the time, scanspeed and fluence are set first to enforce the physical bounds. DLS range is set and to normalize reward, episode steps are given. Using the random forest model as environment, gives fast and safe which works as virtual environment. As usual replay buffer stores the transitions for off policy learning. The neural networks are defines with 256 units for the actor updates. The twin critic outputs concatenated state and action, outputting the Q value estimate. Learning rate, discount and soft target update is mentioned and noise is added for target policy smoothing. Then the TD3 agent does all the steps mentioned in the algorithm.

0.4 Results and Discussion

After applying the ML algorithms and techniques on the dataset it showed poor results which may be accounted for the smaller size of the dataset. The below table 1 shows the R2 value of DLS with ML models.

Table 1: Cross-validated R^2 scores by model for DLS

Model	R^2 **(mean CV)**
Linear Regression	-0.026289835060442378
Random Forest	-0.3458702816616305
Gradient Boosting	-0.5025714537488087
Support Vector Regressor	-0.16611060743615774

The below table 2 shows the R2 value of UV Vis with ML models.

Table 2: Cross-validated R^2 scores (UV–Vis target)

Model	R^2 **(mean CV)**
Linear Regression	0.805
Random Forest	0.759
Gradient Boosting	0.716
Support Vector Regressor	-0.280

The below table 3 shows the R2 value of UV peak with UV models.

Table 3: Cross-validated R^2 scores (UV-peak target)

Model	R^2 **(mean CV)**
Linear Regression	0.077
Random Forest	-0.041
Gradient Boosting	-0.055
Support Vector Regressor	-0.131

0.4.1 Model selection

Since the dataset we have is smaller the use of surrogate modeling increase the size of dataset. After performing the EDA on the surrogate model dataset, the data distribution is approximately similar to the original dataset. The ML model as explained random forest regressor was used with 80 % train data and 20 % test data, after training the model it performed very well with the R2 value of 0.99 which is perfect fit which is further used in the RL as environment.

0.4.2 DDPG

The trained model is set up to inference to check how well the input parameters are predicted for the desired DLS, actor with deterministic policy μ_θ maps the current state to a good action, the actor predicts the action deterministically by checking the errors for all trials with input adjustments.

The reward became more negative over the long run, where at early stages it showed around -1.26 but later showed -3.72, the model was far from the target.

Table 4: Representative training outputs for target DLS = 150 nm.

Episode range	**Mean reward**	**Final DLS (nm)**
1–6	−1.26	134.90
7–13	−4.12	199.48
989–1000	−3.72	105.40

Some of the reasons for this might be action clipping, overestimation due to single critic and no target policy smoothing, reward scale. To fix these errors TD3 model has been introduced.

0.4.3 TD3

The model got stuck at some point, where it proposes same setting like Time = 25.00, scanspeed = 3000, Fluence = 1.83, where the RF model predicts it to be 134.9 nm whereas the target is 150 nm with the error of approximately 15 nm but for some cases, the model got stuck at 120.86 nm with the error of approximately 29.14 nm.

Reward per step is found out to be 0.1258

$$\text{error} \;=\; |134.90 - 150| \;=\; 15.10,$$

$$r_t \;=\; -\frac{15.10}{120} \;\approx\; -0.1258.$$

The final DLS hardly changes this might due to replay buffer sees same samples all the time. Exploration is slightly weak, filtration of noise might be helpful.

Advantages of TD3 over DDPG,

- It overcomes the overestimation bias by methods of double clipping or twin critic.
- Avoids overfitting by target policy smoothing
- Ensure more stabilization by delaying the updates.

0.4.4 Challenges and Opportunities

- Since the data is obtained from the experiment which is expensive and limited. Hence simulations and model training becomes critical.
- Integrating RL with the physical system and testing is crucial for real world analysis.
- Due to ethical concern, RL or ML working remains a black box.

0.5 Conclusions

The study focuses on the integration of RL with the surrogate model to optimize the synthesize of magnesium nanoparticles via PLAL method. The experimental dataset which lacks in size did not succeed in ML model development, with the help of surrogate modeling using random forest regressor synthesized the dataset that mimics the experimental values. When the EDA performed on the new dataset it showed similar behavior to the experimental one. When the random forest model was developed for the synthesized dataset it showed highly effective and better predictive accuracy with R2 value of 0.99 and RMSE of 2.5 nm, which helps the RL agent to explore the design space.

Using this model as gym compatible environment in the RL, advanced RL models such as DDPG and TD3 model were built. The RL model iteratively search for optimal laser parameters like time, scanspeed and fluence based on the desired DLS mentioned by the user. The DDPG model showed fast convergence but instability but the TD3 model with the advancement in techniques like twin critic, target policy smoothing and delayed updates. The error by the TD3 model significantly reduced in comparison with DDPG.

Particularly the TD3 model showed promising approach for parameter selection in PLAL, with better stabilization and optimization than DDPG. Despite this, the study lacks the many uncovered areas like the model gets stuck misleading the prediction. The implementation of RL integration with PLAL in real world still faces certain challenges.

0.5.1 Possible future work that can be achieved

:

- Hybrid physics models: Combining of RL with physics based models like energy equations and extrapolation beyond the dataset.
- Parameter improvement: By considering more factors that involves in manufacturing of PLAL can help in better prediction.
- Active learning: By making the model train on the live data regularly, increase the model performance.

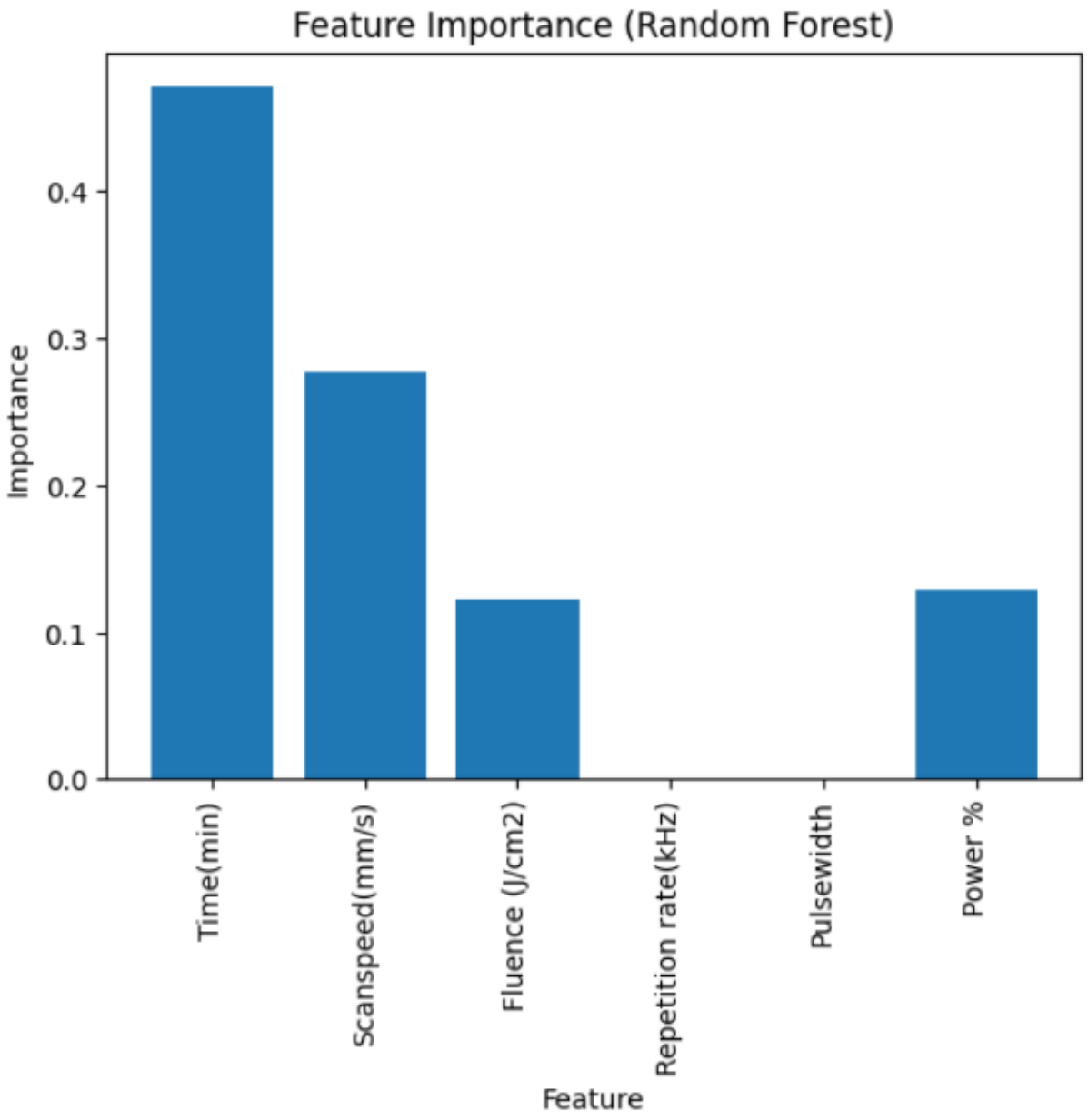

Figure 13: Feature importance based on the random forest regressor

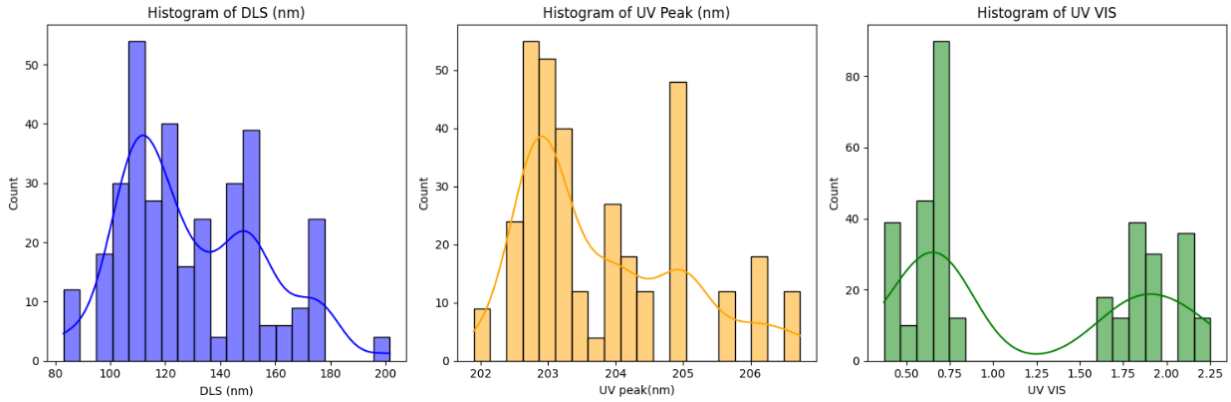

Figure 14: Histogram of new dataset

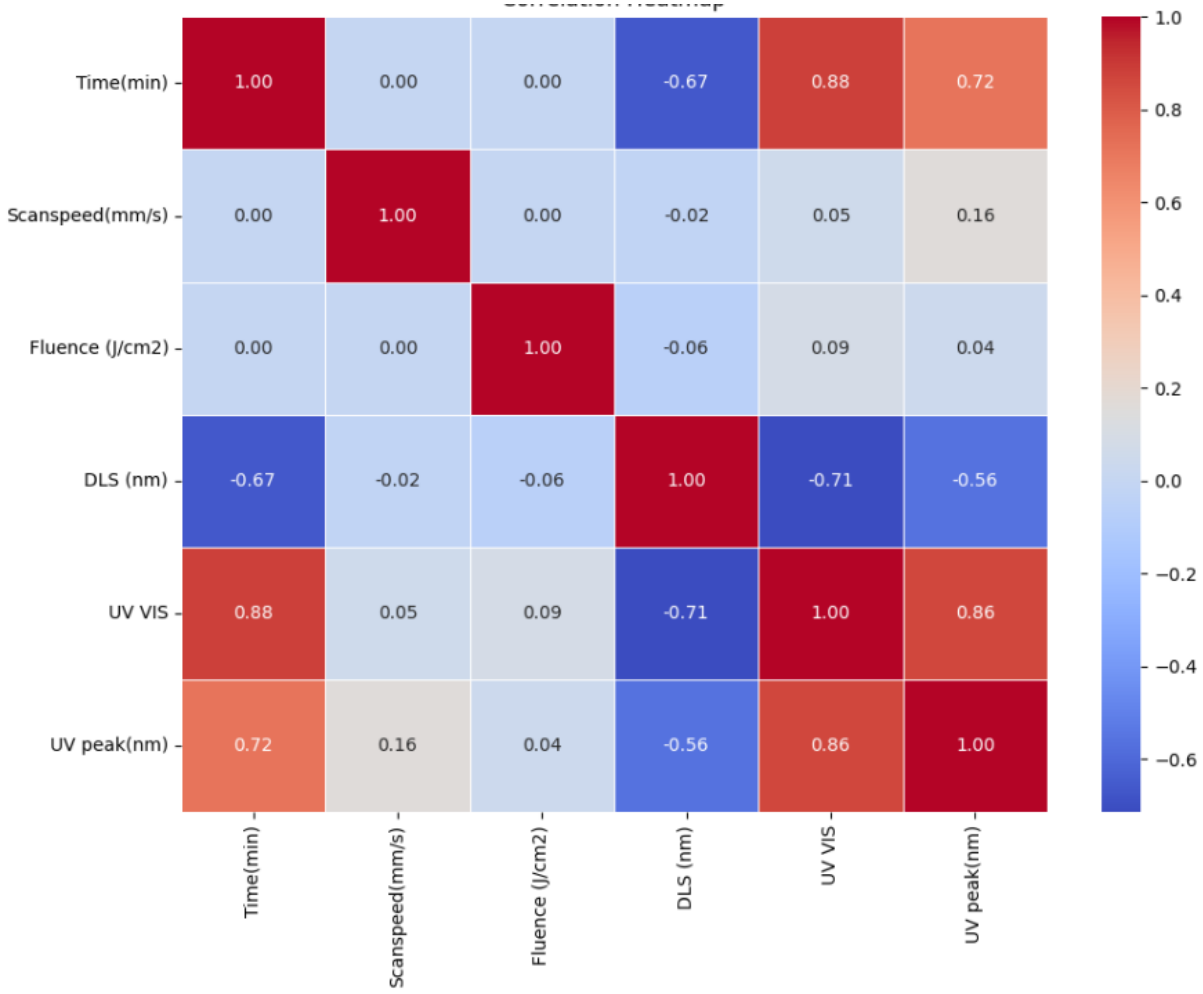

Figure 15: Correlation map of surrogate model dataset

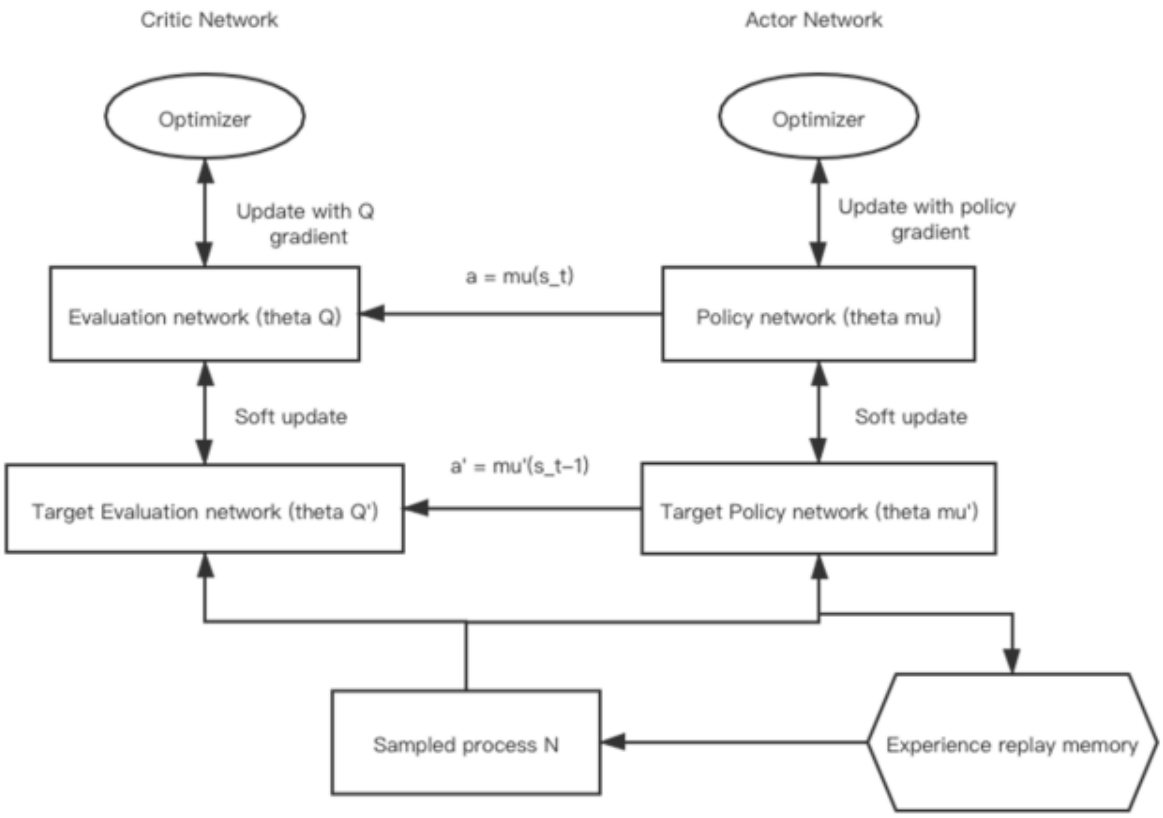

Figure 16: DDPG
Tan (2021)

Bibliography

Abbas, I. K. & Adim, K. A. (2023), 'Synthesis and characterization of magnesium oxide nanoparticles by atmospheric non-thermal plasma jet', *Kuwait Journal of Science* **50**(3), 223–230.

Butler, K. T., Davies, D. W., Cartwright, H., Isayev, O. & Walsh, A. (2018), 'Machine learning for molecular and materials science', *Nature* **559**(7715), 547–555.

Costa, D. C., Fernandes, M., Moura, C., Miranda, G., Silva, F., Carvalho, Ó. & Madeira, S. (2025), 'Laser ablation in liquid-assisted synthesis of three types of nanoparticles for enhanced antibacterial applications', *International Journal of Precision Engineering and Manufacturing-Green Technology* pp. 1–19.

Dasari, S. K., Cheddad, A. & Andersson, P. (2019), Random forest surrogate models to support design space exploration in aerospace use-case, *in* 'IFIP International Conference on Artificial Intelligence Applications and Innovations', Springer, pp. 532–544.

del Real Torres, A., Andreiana, D. S., Ojeda Roldán, Á., Hernández Bustos, A. & Acevedo Galicia, L. E. (2022), 'A review of deep reinforcement learning approaches for smart manufacturing in industry 4.0 and 5.0 framework', *Applied Sciences* **12**(23), 12377.

Fazio, E., Gökce, B., De Giacomo, A., Meneghetti, M., Compagnini, G., Tommasini, M., Waag, F., Lucotti, A., Zanchi, C. G., Ossi, P. M. et al. (2020), 'Nanoparticles engineering by pulsed laser ablation in liquids: Concepts and applications', *Nanomaterials* **10**(11), 2317.

Fujimoto, S., Hoof, H. & Meger, D. (2018), Addressing function approximation error in actor-critic methods, *in* 'International conference on machine learning', PMLR, pp. 1587–1596.

Graves, A., Wayne, G. & Danihelka, I. (2014), 'Neural turing machines', *arXiv preprint arXiv:1410.5401* .

Harris, S. B., Biswas, A., Yun, S. J., Roccapriore, K. M., Rouleau, C. M., Puretzky, A. A., Vasudevan, R. K., Geohegan, D. B. & Xiao, K. (2024), 'Autonomous synthesis of thin film materials with pulsed laser deposition enabled by in situ spectroscopy and automation', *Small Methods* **8**(9), 2301763.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D. & Wierstra, D. (2015), 'Continuous control with deep reinforcement learning', *arXiv preprint arXiv:1509.02971* .

Liu, H., Ge, J., Yang, S., Zhang, L., Xue, Y. & Lan, J. (2022), 'Reflection coefficient estimation of femtosecond laser surface processing using support vector regression', *IEEE Photonics Journal* **14**(6), 1–9.

Ma, Y., Zhu, W., Benton, M. G. & Romagnoli, J. (2019), 'Continuous control of a polymerization system with deep reinforcement learning', *Journal of Process Control* **75**, 40–47.

Mareev, E., Garmatina, A., Semenov, T., Asharchuk, N., Rovenko, V. & Dyachkova, I. (2023), Self-adjusting optical systems based on reinforcement learning, *in* 'Photonics', Vol. 10, MDPI, p. 1097.

Miao, R., Bissoli, M., Basagni, A., Marotta, E., Corni, S. & Amendola, V. (2023), 'Data-driven predetermination of cu oxidation state in copper nanoparticles: Application to the synthesis by laser ablation in liquid', *Journal of the American Chemical Society* **145**(47), 25737–25752.

Mobarak, M. H., Mimona, M. A., Islam, M. A., Hossain, N., Zohura, F. T., Imtiaz, I. & Rimon, M. I. H. (2023), 'Scope of machine learning in materials research—a review', *Applied Surface Science Advances* **18**, 100523.

Nyabadza, A. & Brabazon, D. (2025), 'Machine learning-based recommender system for pulsed laser ablation in liquid: Recommendation of optimal processing parameters for targeted nanoparticle size and concentration using cosine similarity and knn models', *Crystals* **15**(7), 662.

Nyabadza, A., Kane, J., Vázquez, M., Sreenilayam, S. & Brabazon, D. (2021), 'Multi-material production of 4d shape memory polymer composites'.

Nyabadza, A., McCarthy, É., Vázquez, M. & Brabazon, D. (2024), 'Post-fabrication adjustment of metalloid mg–c-graphene nanoparticles via pulsed laser ablation for paper electronics and process optimisation', *Materials & Design* **240**, 112869.

Nyabadza, A., Shan, C., Murphy, R., Vazquez, M. & Brabazon, D. (2023), 'Laser-synthesised magnesium nanoparticles for amino acid and enzyme immobilisation', *OpenNano* **11**, 100133.

Nyabadza, A., Vazquez, M., Coyle, S., Fitzpatrick, B. & Brabazon, D. (2021), 'Magnesium nanoparticle synthesis from powders via pulsed laser ablation in liquid for nanocolloid production', *Applied Sciences* **11**(22), 10974.

Phuoc, T. X., Howard, B. H., Martello, D. V., Soong, Y. & Chyu, M. K. (2008), 'Synthesis of mg (oh) 2, mgo, and mg nanoparticles using laser ablation of magnesium in water and solvents', *Optics and lasers in Engineering* **46**(11), 829–834.

Saedi, S., Blissett, S., Raji, H., Hesabizadeh, T., Osterlin, B. & Guisbiers, G. (2023), 'Enhanced elasticity in magnesium nanoparticle reinforced acrylic elastomer', *Polymer Engineering & Science* **63**(10), 3223–3230.

Saha, P., Datta, M. K., Velikokhatnyi, O. I., Manivannan, A., Alman, D. & Kumta, P. N. (2014), 'Rechargeable magnesium battery: Current status and key challenges for the future', *Progress in Materials Science* **66**, 1–86.

Sasanapuri, S. K., Dhanya, C. & Gosain, A. (2025), 'A surrogate machine learning model using random forests for real-time flood inundation simulations', *Environmental Modelling & Software* **188**, 106439.

Schimkowitsch, B. (2024), Control of pulsed lasers with high repetition rate using reinforcement learning, PhD thesis, Technische Universität Wien.

Shen, X. (2024), Comparison of ddpg and td3 algorithms in a walker2d scenario, *in* '2023 International Conference on Data Science, Advanced Algorithm and Intelligent Computing (DAI 2023)', Atlantis Press, pp. 148–155.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D. & Riedmiller, M. (2014), Deterministic policy gradient algorithms, *in* 'International conference on machine learning', Pmlr, pp. 387–395.

Suresh, J., Yuvakkumar, R., Sundrarajan, M. & Hong, S. I. (2014), 'Green synthesis of magnesium oxide nanoparticles', *Advanced Materials Research* **952**, 141–144.

Tan, H. (2021), Reinforcement learning with deep deterministic policy gradient, *in* '2021 International conference on artificial intelligence, big data and algorithms (CAIBDA)', IEEE, pp. 82–85.

Tani, S. & Kobayashi, Y. (2022), 'Ultrafast laser ablation simulator using deep neural networks', *Scientific reports* **12**(1), 5837.

Tsai, C.-C. & Yiu, T.-H. (2023), 'Investigation of laser ablation quality based on data science and machine learning xgboost classifier', *Applied Sciences* **14**(1), 326.

Wang, C., Zhang, Z., Jing, X., Yang, Z. & Xu, W. (2022), 'Optimization of multistage femtosecond laser drilling process using machine learning coupled with molecular dynamics', *Optics & Laser Technology* **156**, 108442.

Yifei, Y. & Lakshminarayanan, S. (2022), Multi-agent reinforcement learning system for multiloop control of chemical processes, *in* '2022 IEEE International Symposium on Advanced Control of Industrial Processes (AdCONIP)', IEEE, pp. 48–53.

Youshia, J., Ali, M. E. & Lamprecht, A. (2017), 'Artificial neural network based particle size prediction of polymeric nanoparticles', *European Journal of Pharmaceutics and Biopharmaceutics* **119**, 333–342.

Zheng, Y., Blake, C., Mravac, L., Zhang, F., Chen, Y. & Yang, S. (2024), 'A machine learning approach capturing hidden parameters in autonomous thin-film deposition', *arXiv preprint arXiv:2411.18721* .

Zhou, Z., Li, X. & Zare, R. N. (2017), 'Optimizing chemical reactions with deep reinforcement learning', *ACS central science* **3**(12), 1337–1344.