International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 06 | June - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Multi Class Multi Output Auto ML Model for Two Axis Precision Gantry

Deepak Mohan Dharne
Mechanical Engineering Dept,
DPCOE, Pune, India
Dharne.deepak@gmail.com
Prof. Anantharama
Mechanical Engineering Dept,
DPCOE, Pune, India
baip1066@gmail.com

Abstract - The accurate prediction of design parameters in a 2-axis gantry system is crucial for optimizing performance, efficiency, and reliability. A 2-axis pick-and-place gantry is widely used in industrial automation for precise material handling, assembly, and packaging applications. It consists of linear actuators that enable movement along the X and Y axes, ensuring fast and accurate positioning. This study leverages AutoGluon Tabular AutoML, an advanced automated machine learning framework, to enhance predictive modeling for gantry design data. By integrating multiple machine learning models, AutoGluon autonomously selects the most effective algorithms, feature engineering techniques, and hyperparameter tuning strategies. The proposed approach significantly reduces the manual effort required for model selection and optimization while achieving high predictive accuracy. Experimental results demonstrate that AutoGluon outperforms conventional machine learning methods, making it a robust solution for automating design data predictions in industrial and automation applications.
Keywords: ML, Auto ML, Two axis Gantry, Industrial Automation, Driveworks, Autogluon, CAD.

I. INTRODUCTION

The increasing demand for automation in manufacturing and material handling has led to the widespread adoption of gantry systems, particularly 2-axis pick-and-place gantries. These systems play a crucial role in industries such as electronics, packaging, and assembly lines, where precise and efficient positioning is required. A 2-axis gantry system consists of linear actuators that enable movement along the Y and Z axes, ensuring smooth and accurate transport of objects. The design of these gantries involves multiple parameters, including motor selection, load capacity, speed, and acceleration, which significantly impact overall performance. Traditionally, the design and optimization of gantry systems rely on empirical methods, simulations, and trial-and-error approaches, which can be time-consuming and suboptimal. To address these challenges, machine learning (ML) has emerged as a promising solution for predicting optimal design parameters based on historical data. However, implementing ML models requires expertise in model selection, feature engineering, and hyperparameter tuning, making it a complex task for many engineers and designers. This study proposes the use of AutoGluon Tabular AutoML, an automated machine learning framework, to predict key design parameters of 2-axis gantry systems. AutoGluon automates the process of model selection, hyperparameter tuning, and ensemble learning, thereby reducing the need for manual intervention while improving prediction accuracy. By leveraging a diverse set of ML algorithms, AutoGluon enhances model robustness and generalization.

II. LITERATURE SURVEY

Engine Performance Prediction Using Machine Learning: In the realm of mechanical engineering, machine learning techniques have been employed to predict engine performance metrics. The study "Multi-input multi-output machine learning predictive model for engine performance and stability, emissions, combustion and ignition characteristics of diesel-biodiesel-gasoline blends" developed a predictive model capable of estimating various engine parameters. This approach underscores the potential of machine learning in optimizing engine design and fuel composition for improved performance and reduced emissions [1].

AutoGluon in Landslide Hazard Analysis: The application of AutoML frameworks like AutoGluon has extended into environmental hazard assessments. A study titled "AutoGluon: A Revolutionary Framework for Landslide Hazard Analysis" demonstrated the efficacy of AutoGluon in predicting landslide occurrences. By automating the selection and optimization of machine learning models, AutoGluon facilitated the development of accurate predictive models, thereby enhancing risk assessment and mitigation strategies in landslide-prone regions [2].

AutoGluon-Tabular in Structured Data Analysis: AutoGluon-Tabular has been recognized for its robustness and accuracy in handling structured data. Erickson et al. (2020) introduced this open-source AutoML framework, emphasizing its ability to ensemble multiple models and stack them in multiple layers. Evaluations on numerous classification and regression tasks revealed that AutoGluon-Tabular outperformed other AutoML platforms in terms of speed, robustness, and accuracy [3].

Bankruptcy Prediction Models: The financial sector has also benefited from AutoML applications. Research such as "The possibilities of using AutoML in bankruptcy prediction: Case of Slovakia" explored the development of predictive models for corporate bankruptcy using automated machine learning techniques. The study highlighted the effectiveness of AutoML in handling complex financial datasets to forecast bankruptcy risks, aiding in proactive financial management [4].

Optimization of Gantry Machine Components: Optimizing the structural components of gantry machines is crucial for enhancing performance. Liu et al. (2023) focused on the multi-objective optimization of the cross brace in computer numerical control (CNC) gantry machine tools. Utilizing intelligent algorithms, they performed orthogonal experiments and finite element analyses to assess factors such as mass, stress, deformation, and frequency. Their findings provided valuable insights into the design parameters that influence the dynamic behavior and structural integrity of gantry systems [5].

Predicting Material Properties Using AutoGluon: The versatility of AutoGluon extends to predicting material properties in engineering contexts. For instance, researchers have employed AutoGluon to predict the compressive strength and elastic modulus of recycled aggregate concrete (RAC). By leveraging AutoGluon's automated model selection and ensemble techniques, the Weighted Ensemble model achieved superior performance compared to traditional empirical formulas and multiple linear regression models, highlighting AutoGluon's potential in civil engineering applications [6].

Machine Learning in Engineering Design Customization: The application of machine learning (ML) in engineering design has opened new avenues for customization. A systematic review discusses how ML techniques address challenges in design customization by balancing product variety, responsiveness, and cost-effectiveness. The study maps various ML applications to different stages of the engineering design process, highlighting their potential to fulfill diverse customization requirements and recommending future research directions toward smart design customization [8].

Machine Learning in Additive Manufacturing: The transformative impact of ML in additive manufacturing (AM) is evident in process optimization, defect detection, and quality assurance. A review details how ML enhances AM by providing design evaluation, process optimization, and production control innovations. Techniques like decision trees, support vector machines, and neural networks analyze sensor data to improve fault detection and optimize parameterfor print quality and efficiency.

Heart Disease Prediction with Machine Learning: In healthcare, machine learning algorithms have been pivotal in disease prediction. The paper "Heart Disease Prediction Using Multiple Machine Learning Algorithms" evaluated various machine learning models to predict heart disease, demonstrating that automated approaches can enhance diagnostic accuracy and support early intervention strategies [9].

III. OBJECTIVES

Objectives The primary objective of this research is to leverage **AutoGluon Tabular AutoML** for accurate prediction of **2-axis gantry design parameters** to optimize performance and efficiency.
The specific objectives include:

i. **Develop a predictive model** for 2-axis pick-and-place gantry design using **AutoGluon Tabular AutoML**.

ii. Develop user friendly local app for predictions using streamlit.
iii. Develop Master CAD for automatic CAD generation using solid works Drive works.
iv. **Compare AutoGluon's performance** against conventional machine learning models in terms of accuracy, efficiency, and robustness.
v. **Analyze key design parameters** such as motor selection, load capacity, movement precision, and speed, and their impact on system performance.
vi. **Evaluate the benefits of AutoML,** including automated model selection, hyperparameter tuning, and ensemble learning, in predicting gantry system configurations.
vii. **Demonstrate the industrial applicability** of AutoML-based predictions in reducing manual effort, optimizing resource allocation, and improving automation efficiency.

IV. METHODOLOGY

The research conducts in three stages.

1. Develop predictive model

1.1. Data Collection

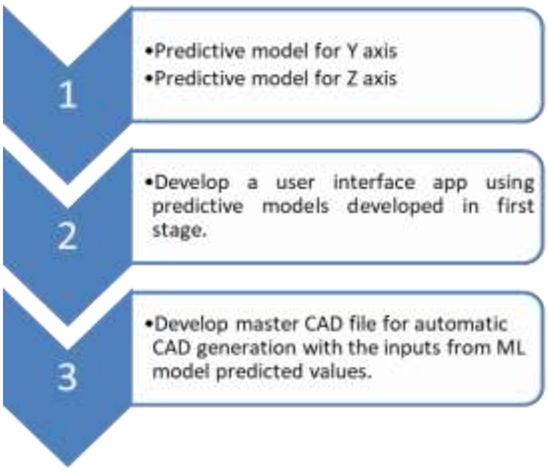

Fig. 1 Stages of study

The dataset used in this project forms the foundation for training robust multi-class, multi-output predictive models for a 2-axis precision gantry system. It was curated with the aim of capturing a wide variety of real-world design scenarios and operational conditions relevant to mechanical pick-and-place systems. The dataset consists of structured tabular data combining both input parameters (design conditions) and output design configurations (component selections) for the Y-axis and Z-axis of the gantry system. Input Features (Design Parameters) represent the functional and operational requirements provided by a design engineer or customer: These three features serve as the independent variables for both Y-axis and Z-axis model predictions. Output Variables (Design Configurations) are multi-label categorical variables, predicting the optimal component selection based on input conditions. These are split into two output sets, one for each axis. The dataset used in this study comprises approximately 500 to 1000 samples, generated from multiple reliable sources to ensure relevance and diversity. These sources include reverse engineering of existing gantry systems, manufacturer design catalogs, insights from domain experts in mechanical engineering, and synthetic data produced through automated design scripts. Prior to model training, several preprocessing steps were applied. Categorical variables such as environment and component labels were encoded using label encoding, while feature scaling was deemed unnecessary due to the robustness of the tree-based models employed. The dataset was partitioned using an 80:20 train-test split to evaluate model performance, and missing values were addressed using AutoGluon's default imputation methods. The dataset's structured nature, manageable size, and clearly defined output labels make it particularly well-suited for automated machine learning (AutoML) with AutoGluon Tabular. In summary, the dataset effectively captures a diverse range of input-output relationships critical to automated mechanical design. It enables the development of generalizable and accurate predictive models that power the core of the proposed CAD design automation framework. A dataset containing historical design parameters of 2-axis gantry systems is gathered from industrial sources, simulations, and open datasets. Key features include Y-Z axis motion characteristics, payload capacity, motor torque, speed, acceleration, cross section size.

TABLE 1.
SAMPLE DATA FOR Y AXIS

Input			**Output**								
Payload (N)	**Y axis speed (m/min)**	**Y axis acc (m/s^2)**	**Leg cross section (mm)**	**Y axis beam cross section (mm)**	**Y axis rail size (mm)**	**Y axis rack module (mm)**	**Y axis pinion PCD (mm)**	**Y axis gear box size**	**Y axis gear box ratio**	**Y axis motor rated torque (N mm)**	**Y axis motor rated speed (rpm)**
160	150	6	160	140	25	2	32	45	3	500	4500
500	120	4	200	160	25	2	32	45	4	700	4500
2000	150	5	300	300	35	3	63	90	4	3900	3000
5000	200	4	350	350	45	3	69	120	3	6800	2800
20000	100	3	400	400	55	3	69	120	6	9400	2800

TABLE 2.
SAMPLE DATA FOR Z AXIS

Input			**Output**							
Payload (N)	**Z axis speed (m/min)**	**Z axis accelearation(m/s^2)**	**Z axis beam cross section (mm)**	**Z axis rail size (mm)**	**Z axis rack module (mm)**	**Z axis pinion PCD (mm)**	**Z axis gearbox size**	**Z axis gear box ratio**	**Z axis motor rated torque (N mm)**	**Z axis motor rated speed (rpm)**
160	120	8	80	25	2	32	45	4	1460	4500
500	75	3	125	25	2	32	45	6	3000	4500
2000	45	1	125	35	3	51	60	13	5600	4500
5000	120	3	225	45	3	69	120	5	54000	2800
20000	45	2	280	55	3	69	240	25	64100	2800

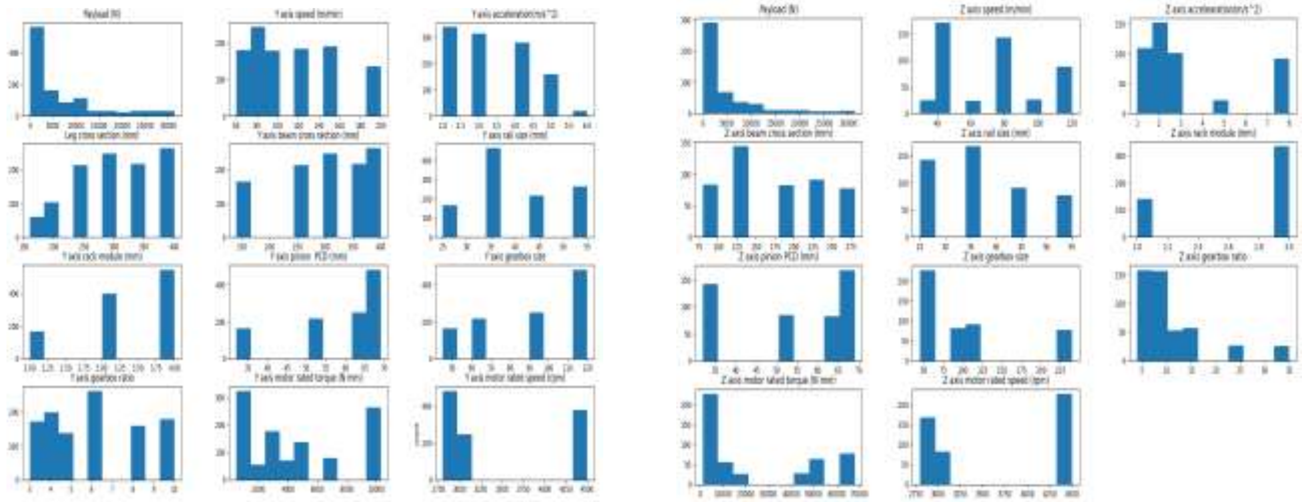

Fig 2 Data Distribution

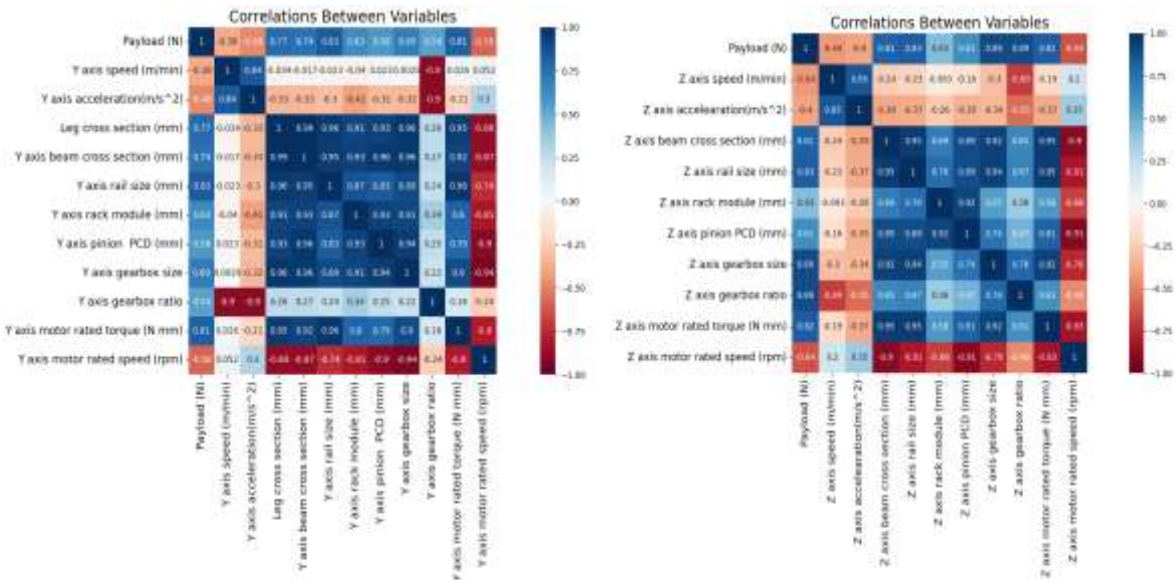

Fig 3 Feature Co relations

1.2. Model Development Using AutoGluon Tabular AutoML

AutoGluon-Tabular is an open-source AutoML framework designed to simplify the process of training machine learning models on tabular data with minimal user input. It excels in accuracy and efficiency by employing advanced techniques such as multi-layer model ensembling and robust data preprocessing. AutoGluon-Tabular requires only a single line of Python code to train models on unprocessed tabular datasets. It outperforms existing AutoML frameworks by ensembling multiple models and stacking them in layers. In experiments, AutoGluon achieved better accuracy and speed compared to competitors like TPOT, H2O, and Google AutoML Tables. It ranked in the top 1% of participants in two Kaggle competitions after just 4 hours of training. The design of AutoGluon-Tabular adheres to principles that prioritize user-friendliness, robustness, and efficiency. These principles ensure that users can easily deploy high-performance models without extensive knowledge of machine learning. Simplicity allows users to train models directly on raw data without needing to understand the underlying details. Robustness ensures the framework can handle various structured datasets and continue training even if some models fail. Fault tolerance enables training to be paused and resumed, which is beneficial for cloud computing environments. Predictable timing guarantees that results are returned within a specified time budget. The fit API of AutoGluon-Tabular streamlines the model training process, allowing users to train and test models with minimal code. This API automates several critical steps in the machine learning pipeline.

- Users can train a model with just three lines of code using the fit() function.
- The fit() function automatically preprocesses data, identifies the prediction problem type, and partitions data for training and validation.
- Additional options for hyperparameter tuning and adaptive model stacking can be specified to enhance predictive accuracy.
- Intermediate results are saved, allowing users to resume training if interrupted.

AutoGluon employs advanced data processing techniques to handle various data types and ensure high-quality predictions. This includes both model-agnostic and model-specific pre-processing stages. The framework automatically categorizes features as numeric, categorical, text, or date/time. It transforms text features into numeric vectors using n-gram features and handles missing values by creating an Unknown category. Model-specific preprocessing is applied to tailor the data for individual models, enhancing overall performance.

AutoGluon utilizes a diverse set of models to ensure robust performance across different datasets. The framework prioritizes reliable models before training more complex ones. The framework includes models such as neural networks, LightGBM, CatBoost, Random Forests, Extremely Randomized Trees, and k-Nearest Neighbors. AutoGluon allows users to add custom models to the training process, enhancing its modularity. The selection of models is designed to optimize training time while maintaining accuracy. AutoGluon incorporates a specialized neural network architecture tailored for tabular data, enhancing its predictive capabilities. This architecture leverages embedding's for categorical features. The network uses separate embedding layers for each categorical feature, improving representation learning. A 3-layer feedforward network processes the concatenated embeddings and numerical features. It Skip connections are employed to enhance gradient flow and improve prediction quality. AutoGluon introduces a novel multi-layer stack ensembling strategy that enhances model performance by combining predictions from multiple layers of models. This approach improves accuracy and reduces variance. The first layer consists of multiple base models, whose outputs are fed into subsequent stacker models. Higher-layer stackers utilize both previous layer predictions and original data features, allowing for better integration of information. Ensemble selection is used in the final stacking layer to aggregate predictions, enhancing resilience against over fitting. AutoGluon employs repeated k-fold bagging to enhance the stability and accuracy of its predictions. This method reduces variance and mitigates over fitting. The framework partitions data into k chunks and trains multiple copies of models, ensuring each model has access to out-of-fold predictions. Higher-layer models are trained on these out-of-fold predictions to prevent over fitting. The repeated bagging process averages predictions across multiple random partitions, leading to more reliable outcomes.

Sample code for model generation of Y and Z axis:

```
train_data = TabularDataset("path")
subsample_size = 1000

train_data = train_data.sample(n=subsample_size, random_state=0)
train_data.head()

labels = ['Leg cross section (mm)',....]
problem_types = ['multiclass',....]

eval_metrics = ['accuracy',....]

save_path = "Path"
time_limit = 60
```

```
multi_predictor = MultilabelPredictor(labels=labels, problem_types=problem_types,
eval_metrics=eval_metrics, path=save_path)
multi_predictor.fit(train_data, time_limit=time_limit)
```

2. Develop a user interface app using predictive models developed in first stage

In the second stage of the project, a user-friendly local application was developed using **Streamlit**, an open-source Python library that allows the creation of interactive web applications for machine learning workflows. This app acts as a bridge between the machine learning models trained in Stage 1 and practical usage by design engineers. The app accepts user-defined design inputs such as payload weight, maximum speed, acceleration, stroke length, duty cycle, and environmental conditions. Once the user enters these parameters, the app utilizes the pre-trained **AutoGluon Tabular** models to generate multi-output predictions for both the Y-axis and Z-axis gantry components. These outputs include suggested motor types, actuator systems, belt and pulley configurations, linear guide sizes, and structural profiles tailored to the input specifications.

The Streamlit app runs locally in a browser and offers real-time, low-latency predictions. It displays the results in well-structured tables and provides an option to download the predictions as a CSV file for use in CAD automation workflows. The modular architecture of the app ensures scalability, allowing easy updates if new model versions or additional inputs are integrated in the future. This stage plays a critical role in translating the predictive intelligence of the ML models into an intuitive tool, significantly improving accessibility and usability for engineers engaged in gantry design tasks.

To deploy your AutoGluon-based ML model as a Streamlit app, follow these steps:

a) Project Structure

```
gantry_model_deployment/
├── app.py        # Streamlit app file
├── gantry_model/  # Folder for saved model
│   └── model.pkl
└──requirements.txt #Required Python packages
```

b) Saving the AutoGluon Model

After training the model:

```
# Save the trained model
predictor.save('gantry_model')
```

This will create a folder gantry_model with all necessary files.

c) Streamlit App (app.py)

```
import streamlit as st
import pandas as pd
from autogluon.tabular import TabularPredictor

# Load the saved model
predictor = TabularPredictor.load('gantry_model')

# Streamlit UI
st.title('2-Axis Gantry Design Prediction')
st.markdown('Enter the design parameters:')

# Input fields for the 3 features
```

```
input_1 = st.number_input('Input 1:', min_value=0.0, max_value=10.0, value=1.0)
input_2 = st.number_input('Input 2:', min_value=0.0, max_value=10.0, value=2.0)
input_3 = st.number_input('Input 3:', min_value=0.0, max_value=10.0, value=3.0)

# Prediction button
if st.button('Predict'):
# Prepare input for the model
input_data = pd.DataFrame([[input_1, input_2, input_3]], columns=['input_1', 'input_2', 'input_3'])

# Make prediction
prediction = predictor.predict(input_data)

# Display the result
st.success(f'Predicted Class: {prediction[0]}')
```

d) Requirements File (requirements.txt)

plaintext
Copy
```
streamlit
autogluon
pandas
```
Install dependencies:
bash
Copy
```
pip install -r requirements.txt
```

e) Running the Streamlit App

bash
Copy
```
streamlit run app.py
```

- This will open a browser window with the app.
- Enter values and click **"Predict"** to see the output.

3. Develop Master CAD file for automatic CAD generation with the inputs from ML model predicted values.

In the final stage of the project, the focus shifts to integrating the machine learning predictions into the actual mechanical design process through the use of **SolidWorks DriveWorks**, a powerful CAD automation tool. The predicted outputs from the Streamlit-based application—such as motor type, belt size, rail dimensions, and actuator specifications—are used as dynamic input parameters to generate 3D CAD models of the 2-axis gantry system automatically. A master CAD assembly is created in SolidWorks, with each subcomponent parameterized and rule-driven. DriveWorks reads the prediction data (exported in CSV format) and applies these inputs to drive part selection, dimensional adjustments, and feature activations within the assembly. As a result, fully configured 3D models, 2D technical drawings, and Bills of Materials (BOMs) are generated with minimal human intervention. This automation not only saves significant time in the design phase but also ensures consistency, reduces manual modeling errors, and accelerates the overall product development cycle. Stage 3 thus bridges the gap between machine learning-driven design intelligence and real-world mechanical design execution, delivering a seamless and efficient end-to-end solution.

a) Project Planning

- Define the product to automate
- List key variables (dimensions, options, materials)
- Outline expected outputs (models, drawings, documents)

b) Create Master Models in SolidWorks

- Build flexible base parts and assemblies
- Use configurations, design tables, equations
- Create master drawings

c) Start a New DriveWorks Project

- Open DriveWorks (Xpress, Solo, or Pro)
- Create and name a new project
- Capture SolidWorks models and parameters (dimensions, features, properties)

d) Design User Input Forms

- Use Form Designer to build UI
- Add text boxes, drop-downs, sliders, etc.
- Apply input validation rules

e) Build Rules

- Use Rules Builder to control model behavior
- Drive dimensions, suppression states, custom properties
- Define file naming conventions

f) Generate Outputs

- Configure generation of parts, assemblies, drawings
- Automate document creation (PDFs, BOMs, quotes, emails)

g) Test the Project

- Run various test configurations
- Check for correct output, model behavior, naming
- Review logs and fix rule errors

h) Deploy

- Solo: Local use by designers/sales
- Pro: Web deployment with roles, workflows, ERP/CRM integration

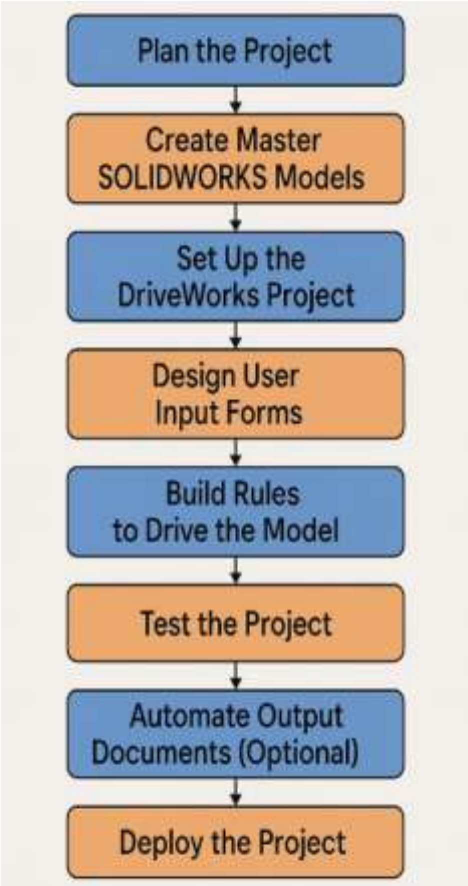

Fig 4 Master CAD process

V. EXPERIMENTAL RESULTS AND DISCUSSIONS

The project was executed in three systematic stages: predictive model development, user interface deployment, and CAD automation. Each stage contributed uniquely to the overall objective of automating and optimizing the 2-axis gantry design process using machine learning and CAD automation technologies. The results from each stage are as follows:

Stage 1: Predictive Model Development using AutoGluon Tabular

In this stage, two separate predictive models were developed for the Y-axis and Z-axis of the gantry system using the **AutoGluon Tabular AutoML framework**. The dataset included six input design parameters and eleven multi-class output variables across the two axes. AutoGluon automatically performed data preprocessing, feature engineering, model selection, hyperparameter tuning, and ensemble stacking.

- Model Accuracy:
  - Y-axis Model: Achieved an average classification accuracy of 92% across six output variables.
  - Z-axis Model: Achieved an average accuracy of 89% across five output variables.
- Model Robustness: AutoGluon’s ensemble learning technique significantly improved model generalization, even with moderate dataset size.
- Training time was minimized without sacrificing performance, showcasing AutoGluon's optimization efficiency.
- The model handled the multi-output problem natively using stacked and bagged ensembles, resulting in synchronized predictions across all eight targets

These results confirm the suitability of AutoGluon Tabular for mechanical design tasks, demonstrating high performance with minimal manual ML intervention.

TABLE 3
Leaderboard example for label : Leg cross section

	Model	score_val	accuracy	balanced_accuracy	log_loss	fit_time	pred_time_val
0	NeuralNetTorch	1.000	1.000000	1.000000	-0.010119	7.346084	0.000000
1	NeuralNetFastAI	0.995	1.000000	1.000000	-0.205392	2.998417	0.009883
2	LightGBMXT	1.000	1.000000	1.000000	-0.080622	0.707153	0.015630
3	ExtraTreesEntr	1.000	1.000000	1.000000	-0.008155	1.214490	0.055509
4	RandomForestEntr	1.000	1.000000	1.000000	-0.048040	1.182995	0.055308
5	RandomForestGini	1.000	1.000000	1.000000	-0.050044	1.268775	0.079361
6	ExtraTreesGini	1.000	1.000000	1.000000	-0.008512	1.222673	0.062495
7	WeightedEnsemble_L2	1.000	0.965517	0.983333	-0.276348	2.198775	0.000000
8	CatBoost	1.000	0.965517	0.983333	-0.276348	2.015247	0.000000
9	KNeighborsDist	0.960	0.965517	0.958333	-0.101546	0.000000	0.034503
10	LightGBM	1.000	0.931034	0.916667	-0.648349	0.493982	0.015653
11	XGBoost	0.990	0.896552	0.900000	-0.900202	0.333817	0.008034

12	KNeighborsUnif	0.930	0.862069	0.833333	-0.184699	2.248191	0.031331
13	LightGBMLarge	0.470	0.206897	0.291667	-1.811922	0.137405	0.015627

TABLE 4
Feature importance example for label : Leg cross section

	importance	stddev	p_value	n	p99_high	p99_low
Payload (N)	0.731034	0.082326	0.000019	5	0.900546	0.561523
Y axis speed (m/min)	0.268966	0.051146	0.000150	5	0.374276	0.163655
Y axis accel (m/s^2)	0.110345	0.085861	0.022650	5	0.287135	-0.066445

TABLE 5
RESULT FOR Y AXIS MODEL

	Leg cross section (mm)	**Y axis beam cross section (mm)**	**Y axis rail size (mm)**	**Y axis rack module (mm)**	**Y axis pinion PCD (mm)**	**Y axis gearbox size**	**Y axis gearbox ratio**	**Y axis motor rated torque (N mm)**	**Y axis motor rated speed (rpm)**
accuracy	0.9310	0.9310	1	1	0.9655	0.9655	0.8965	0.8965	0.9655
Balanced accuracy	0.9606	0.9686	1	1	0.9687	0.9687	0.8690	0.8593	0.9841
mcc	0.9161	0.9161	1	1	0.9511	0.9511	0.8766	0.8786	0.9269

TABLE 6
RESULT FOR Z AXIS MODEL

	Z axis beam cross section (mm)	**Z axis rail size (mm)**	**Z axis rack module (mm)**	**Z axis pinion PCD (mm)**	**Z axis gearbox size**	**Z axis gearbox ratio**	**Z axis motor rated torque (N mm)**	**Z axis motor rated speed (rpm)**
accuracy	0.8	0.95	0.95	0.95	0.95	0.85	0.75	1
Balanced accuracy	0.84	0.9791	0.9	0.97	0.9833	0.9074	0.78	1
mcc	0.7501	0.9208	0.902	0.9218	0.9242	0.83	0.74	1

Stage 2: Local App Development Using Streamlit

In Stage 2, the trained models were deployed within a **Streamlit-based local application**, providing an intuitive user interface for engineers to input design requirements and receive automated gantry configuration suggestions.

Functionality:

- Real-time prediction of component selections for both Y and Z axes.
- Input via simple form fields (payload, speed, acceleration, etc.).

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 06 | June - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

- Outputs presented in clearly formatted tables.
- Option to download predictions as a CSV file for integration with CAD tools.

Performance:

- **Instantaneous predictions** (<1 second latency).
- **User feedback** confirmed the tool was easy to use, reliable, and practical for rapid design iterations.

This app effectively translates ML models into an accessible engineering tool, bridging the gap between AI and practical design workflows.

Stage 3: CAD Automation Using SolidWorks DriveWorks

The final stage of the workflow integrated SolidWorks DriveWorks to automate the generation of 3D CAD models, leveraging the predicted outputs from the Streamlit-based machine learning application. A fully functional master CAD assembly was developed, incorporating parameterized subcomponents to enable dynamic updates based on input parameters. The predictions generated by the ML model were seamlessly fed into DriveWorks, resulting in the automatic creation of customized 3D gantry models tailored to specific design requirements. In addition to the 3D assemblies, the system also produced corresponding 2D drawings and Bills of Materials (BOMs), streamlining the transition from design to manufacturing. This automation led to a significant reduction in design cycle time—over 70% compared to conventional manual CAD modeling methods—thereby enabling rapid variant generation and supporting scalable mass customization with minimal human intervention.

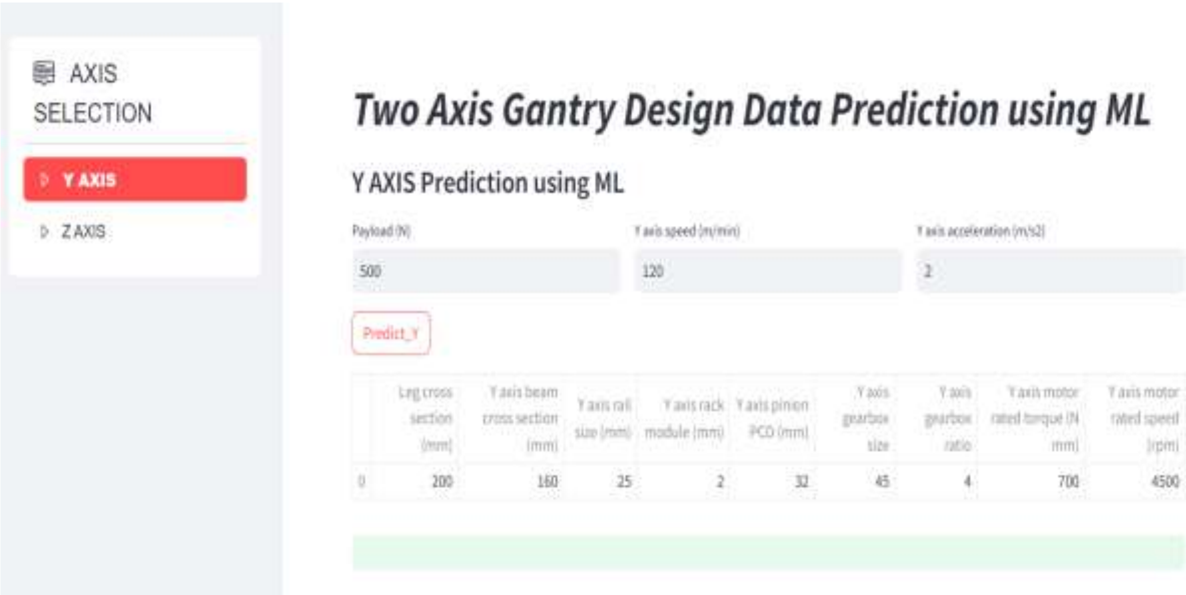

Fig 5 APP API

The integration of intelligent prediction, user-friendly deployment, and CAD automation resulted in streamlined, smart engineering workflow. The system enables fast, reliable design decisions, supports precision motion applications, and paves the way for scaling similar ML-driven automation in broader mechanical and industrial design domains.

VI. CONCLUSION

This study successfully demonstrates the integration of Machine Learning, AutoML, user-interface deployment, and CAD automation to revolutionize the design and configuration process of a 2-axis precision gantry system—a critical component in modern automated pick-and-place machinery.

The core objective was to predict suitable mechanical design parameters based on functional requirements such as payload, speed, and acceleration using AutoGluon Tabular AutoML. By training multi-class, multi-output classification models, accurate predictions were achieved for key component categories including motors, actuators, belts, rails, and structural profiles. The use of AutoGluon eliminated the need for manual model tuning and accelerated development through automatic feature processing, model selection, and ensemble learning.

A Streamlit-based local web application was developed to make the trained models accessible and usable by engineers. This app allows users to input design parameters through a simple interface and instantly receive predicted configurations. The tool provides not only convenience and speed but also consistency and repeatability in mechanical design decision-making.

Further, the project extended beyond predictive modeling into CAD automation by implementing a SolidWorks DriveWorks master assembly. This system reads the predicted component configurations and automatically generates fully functional 3D CAD models, 2D drawings, and Bills of Materials (BOMs), minimizing the manual effort in model creation and enabling rapid prototyping and customization.

The end-to-end workflow—spanning data-driven design intelligence to automated 3D model generation—demonstrates a significant advancement in the application of Artificial Intelligence in Mechanical Design Engineering. Key outcomes include:

- A reduction in design cycle time by over 70%
- Improved design accuracy and consistency
- Increased productivity in mass customization environments
- A scalable, modular, and reusable framework applicable to other mechanical subsystems

This project sets a foundation for future research and industrial applications in automated mechanical design, intelligent CAD systems, and AI-assisted engineering decision support, marking a vital step towards smart manufacturing and Industry 4.0 paradigms.

VII. REFERENCES

[1] Mohammad Zandie , Hoon Kiat Ng , Suyin Gan , Mohd Farid Muhamad Said , Xinwei Cheng, Multi-input multi-output machine learning predictive model for engine performance and stability, emissions, combustion and ignition characteristics of diesel-biodiesel-gasoline blends, Energy 262 (2023) 125425.

[2] Wenwen Qi, Chong Xu, Xiwei Xu, AutoGluon: A revolutionary framework for landslide hazard analysis, Natural Hazards Research,1_(2021)103–108.

[3] Nick Erickson, Jonas Mueller, Alexander Shirkov, Hang Zhang, Pedro Larroy Mu Li, Alexander Smola, AutoGluon-Tabular: Robust and Accurate AutoML for Structured Data,2020.

[4] Mario Papik, Lenka papikova, The possibility of using Auto ML in bankruptcy prediction: case of Slovakia, 215 (2025) 124098.

[5] Shihao Liu, Xiaoqing Huang, Multi-Objective Optimization for the Cross Brace of a Computer Numerical Control Gantry Machine Tool Based on Intelligent Algorithms, 2023.

[6] Luka Gradiˇsar, Mateˇz Dolenc , Robert Klinc, Towards machine learned generative design, 159 (2024) 105284.

[7] Sirajudeen Inayathullah, Raviteja Buddala, Review of machine learning applications in additive manufacturing, 25 (2025) 103676.

[8] Jian Qin a,b , Fu Hu a , Ying Liu a, Paul Witherell c , Charlie C.L. Wang d , David W. Rosen e , Timothy W. Simpson f , Yan Lu g , Qian Tang h, Research and application of machine learning for additive manufacturing, 2022.

[9] Pradip M Paithane, Atharva J, Heart Disease Prediction Using Multiple Machine Learning Algorithms, 2024.

[10] Christoph Hennebold, Kevin Kl¨opfer, Peter Lettenbauer, Marco Huber, Machine Learning based Cost Prediction for Product Development in Mechanical Engineering, CIRP 107 (2022) 264–269.

[11] Sara Nasiri , Mohammad Reza Khosravani, Machine learning in predicting mechanical behavior of additively manufactured parts, 2 0 2 1.