Prediction of Grain Boundary Pop-in Events using Hybrid Machine Learning and Optimization Strategy

By

Noor Ul Ain

(Registration No: 00000361995)

Department of Materials Engineering

School of Chemical and Materials Engineering

National University of Sciences & Technology (NUST)

Islamabad, Pakistan

(2024)

Prediction of Grain Boundary Pop-in Events using Hybrid Machine Learning and Optimization Strategy

By

Noor Ul Ain

(Registration No: 00000361995)

A thesis submitted to the National University of Sciences and Technology, Islamabad,

in partial fulfillment of the requirements for the degree of

Master of Science in
Materials and Surface Engineering

Supervisor: Dr. -Ing. Farhan Javaid

Co-Supervisor: Dr. Muhammad Nouman Aslam Khan

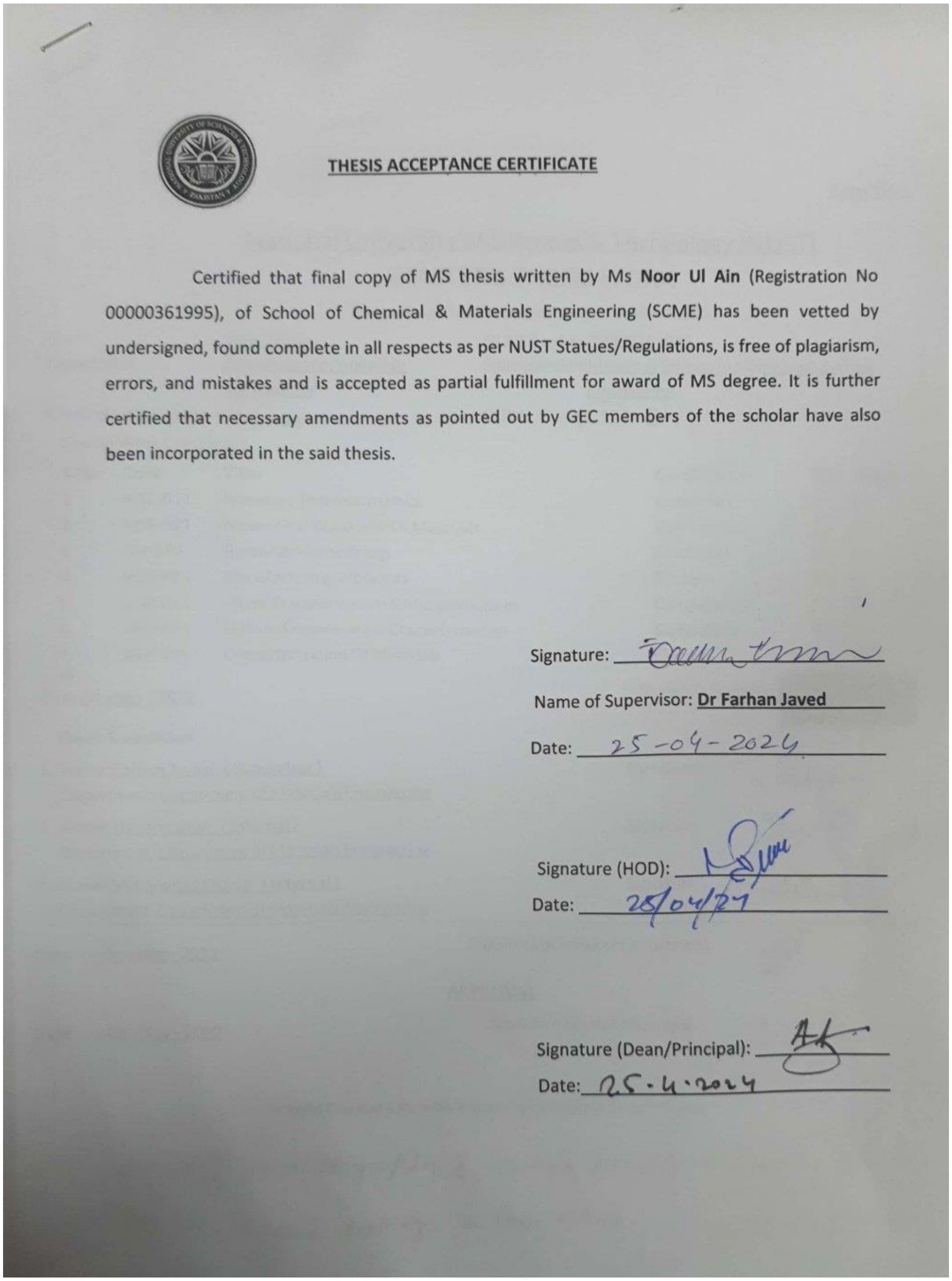
School of Chemical and Materials Engineering

National University of Sciences & Technology (NUST)

Islamabad, Pakistan

(2024)

THESIS ACCEPTANCE CERTIFICATE



THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS thesis written by Ms **Noor Ul Ain** (Registration No 00000361995), of School of Chemical & Materials Engineering (SCME) has been vetted by undersigned, found complete in all respects as per NUST Statues/Regulations, is free of plagiarism, errors, and mistakes and is accepted as partial fulfillment for award of MS degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in the said thesis.

Signature:

Name of Supervisor: **Dr Farhan Javed**

Date: 25-04-2024

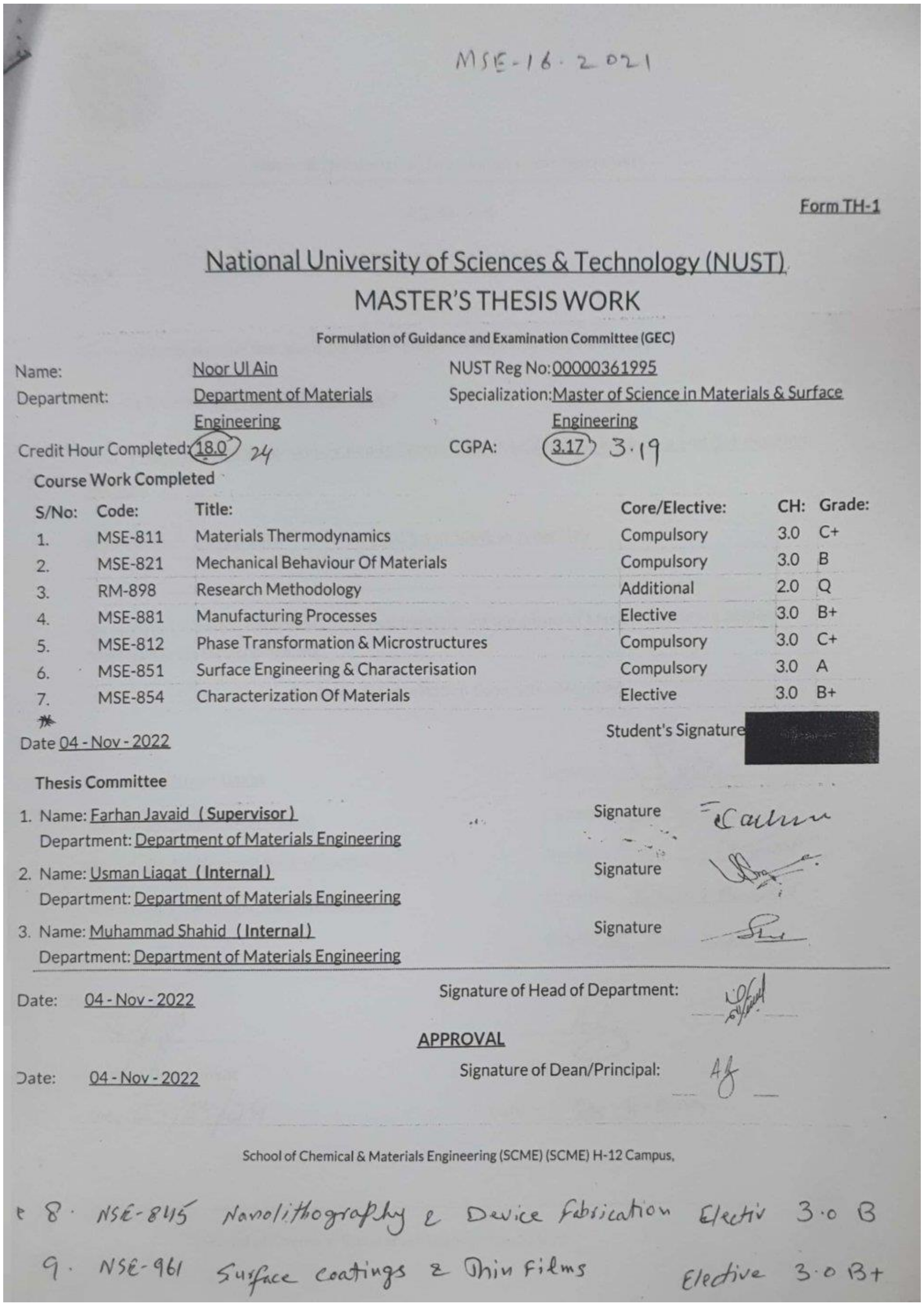
Signature (HOD):

Date: 25/04/24

Signature (Dean/Principal):

Date: 25·4·2024

TH-1



MSE-16-2021

Form TH-1

National University of Sciences & Technology (NUST)

MASTER'S THESIS WORK

Formulation of Guidance and Examination Committee (GEC)

Name: Noor Ul Ain
NUST Reg No: 00000361995
Department: Department of Materials Engineering
Specialization: Master of Science in Materials & Surface Engineering
Credit Hour Completed: ~~18.0~~ 24
CGPA: ~~3.17~~ 3.19

Course Work Completed

S/No:	Code:	Title:	Core/Elective:	CH:	Grade:
1.	MSE-811	Materials Thermodynamics	Compulsory	3.0	C+
2.	MSE-821	Mechanical Behaviour Of Materials	Compulsory	3.0	B
3.	RM-898	Research Methodology	Additional	2.0	Q
4.	MSE-881	Manufacturing Processes	Elective	3.0	B+
5.	MSE-812	Phase Transformation & Microstructures	Compulsory	3.0	C+
6.	MSE-851	Surface Engineering & Characterisation	Compulsory	3.0	A
7.	MSE-854	Characterization Of Materials	Elective	3.0	B+

*

Date 04 - Nov - 2022
Student's Signature [redacted]

Thesis Committee

1. Name: Farhan Javaid **(Supervisor)** Signature
   Department: Department of Materials Engineering
2. Name: Usman Liaqat **(Internal)** Signature
   Department: Department of Materials Engineering
3. Name: Muhammad Shahid **(Internal)** Signature
   Department: Department of Materials Engineering

Date: 04 - Nov - 2022
Signature of Head of Department:

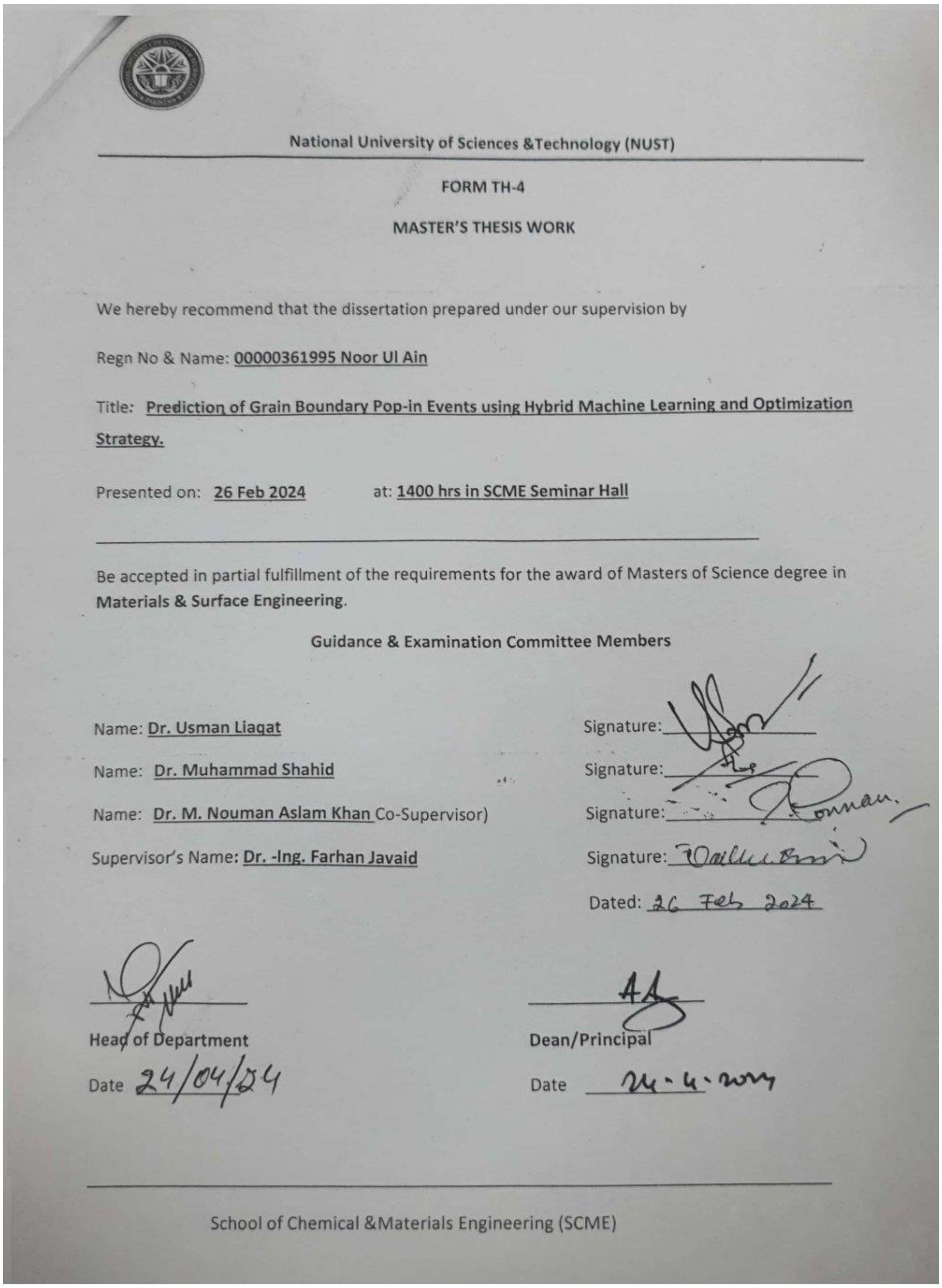
APPROVAL

Date: 04 - Nov - 2022
Signature of Dean/Principal:

School of Chemical & Materials Engineering (SCME) (SCME) H-12 Campus,

* 8. NSE-845 Nanolithography & Device fabrication Electiv 3.0 B
9. NSE-961 Surface coatings & Thin Films Elective 3.0 B+

TH – 4



National University of Sciences &Technology (NUST)

FORM TH-4

MASTER'S THESIS WORK

We hereby recommend that the dissertation prepared under our supervision by

Regn No & Name: **00000361995 Noor Ul Ain**

Title: **Prediction of Grain Boundary Pop-in Events using Hybrid Machine Learning and Optimization Strategy.**

Presented on: **26 Feb 2024** at: **1400 hrs in SCME Seminar Hall**

Be accepted in partial fulfillment of the requirements for the award of Masters of Science degree in **Materials & Surface Engineering**.

Guidance & Examination Committee Members

Name: **Dr. Usman Liaqat** Signature:
Name: **Dr. Muhammad Shahid** Signature:
Name: **Dr. M. Nouman Aslam Khan** Co-Supervisor) Signature:
Supervisor's Name: **Dr. -Ing. Farhan Javaid** Signature:
Dated: 26 Feb 2024

Head of Department
Date 24/04/24

Dean/Principal
Date 24-4-2024

School of Chemical &Materials Engineering (SCME)

AUTHOR'S DECLARATION

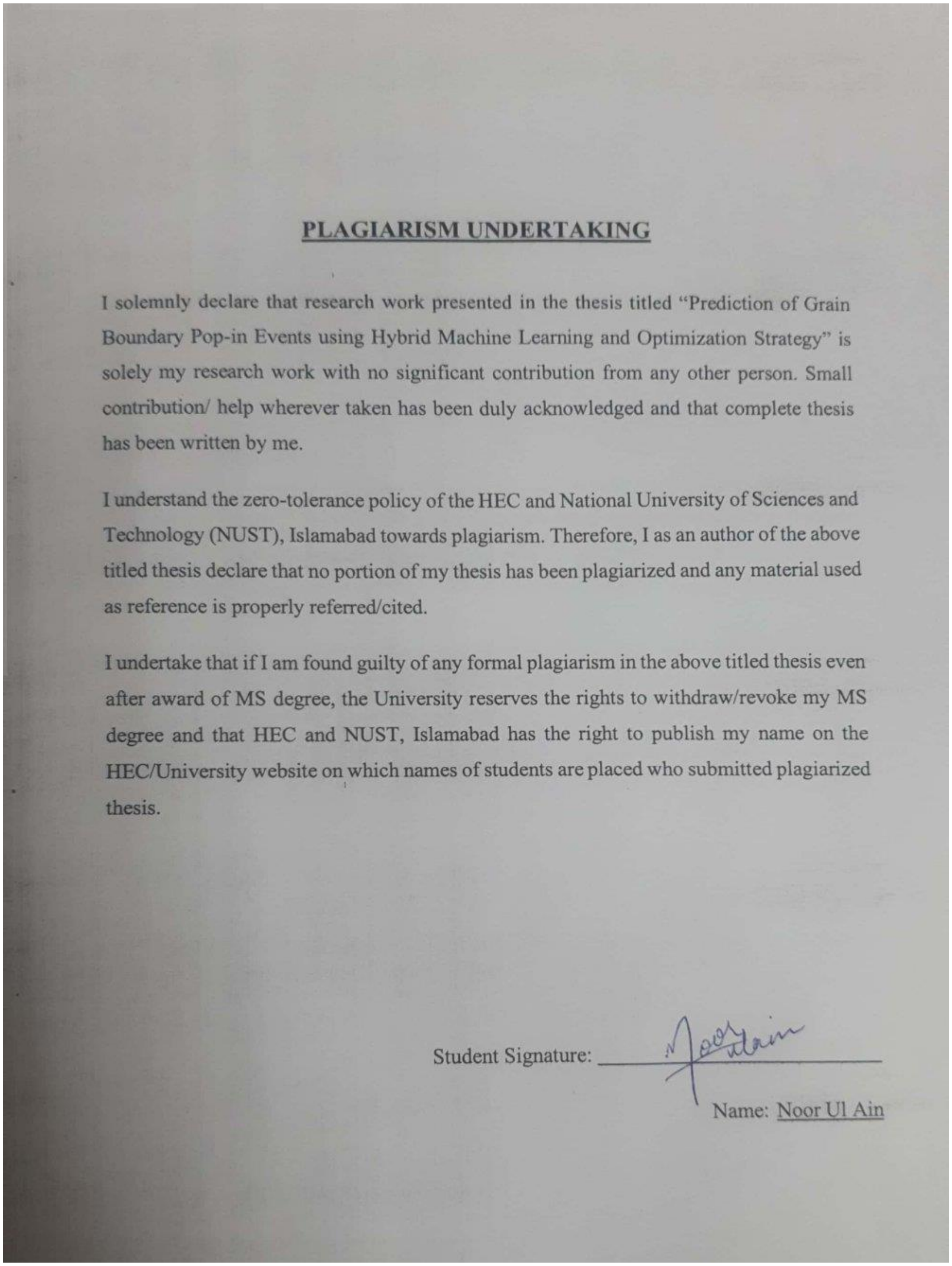
I Noor Ul Ain hereby state that my MS thesis titled "Prediction of Grain Boundary Pop-in Events using Hybrid Machine Learning and Optimization Strategy" is my own work and has not been submitted previously by me for taking any degree from National University of Sciences and Technology, Islamabad or anywhere else in the country/ world.

At any time if my statement is found to be incorrect even after I graduate, the university has the right to withdraw my MS degree.

Name of Student: Noor Ul Ain

Date: 2 February 2024

PLAGIARISM UNDERTAKING



PLAGIARISM UNDERTAKING

I solemnly declare that research work presented in the thesis titled "Prediction of Grain Boundary Pop-in Events using Hybrid Machine Learning and Optimization Strategy" is solely my research work with no significant contribution from any other person. Small contribution/ help wherever taken has been duly acknowledged and that complete thesis has been written by me.

I understand the zero-tolerance policy of the HEC and National University of Sciences and Technology (NUST), Islamabad towards plagiarism. Therefore, I as an author of the above titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred/cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of MS degree, the University reserves the rights to withdraw/revoke my MS degree and that HEC and NUST, Islamabad has the right to publish my name on the HEC/University website on which names of students are placed who submitted plagiarized thesis.

Student Signature: Noor ulain

Name: Noor Ul Ain

DEDICATION

This thesis is dedicated to my Parents for their Love, Support, and Prayers.

ACKNOWLEDGEMENTS

I would like to begin by expressing my gratitude to Almighty **Allah** for bestowing his blessings and guiding me through this effort.

My heartfelt appreciation goes to my supervisor, **Dr. -Ing. Farhan Javaid**, and my co-supervisor, **Dr. Muhammad Nouman Aslam Khan**, for their timely advice, helpfulness, and practical approach throughout this project. It's important to remember that my project was only accomplished thanks to their constant efforts. Despite various barriers and problems throughout my inquiry, they remained accessible and encouraged me to complete the task more successfully.

I would like to express my gratitude to GEC members **Dr. Usman Liaqat** and **Dr. Muhammad Shahid** for their insightful comments and constructive criticism, which greatly contributed to the quality of this work.

I would like to express my heartfelt gratitude to our principal, **Dr. Amir Azam Khan,** and department head, **Dr. Khurram Yaqoob** for their everlasting dedication to educational growth and outstanding leadership. Their help and encouragement have been invaluable throughout this academic journey.

My sincere appreciation also extends to my Parents. Their constant motivation, assistance, and affection served as fundamental components of my resilience. This accomplishment would not have been feasible without their unwavering support, love, and care.

Noor Ul Ain

TABLE OF CONTENTS

ACKNOWLEDGEMENTS IX

TABLE OF CONTENTS X

LIST OF TABLES XII

LIST OF FIGURES XIII

LIST OF SYMBOLS, ABBREVIATIONS AND ACRONYMS XV

ABSTRACT XVI

CHAPTER 1: INTRODUCTION 1
1.1 Mechanical Properties of Materials 1
1.2 Slip transfer and dislocation interaction mechanisms in adjacent grains. 3
1.3 First Pop-in 4
1.4 Secondary Pop-ins 5
1.5 Grain Boundary Pop-in 7
1.6 Research problem and hypothesis 13
1.7 Proposed Methodology 13
1.8 Objectives 15

CHAPTER 2: LITERATURE REVIEW 16
2.1 Nanoindentation 16
2.2 Types of Indenters Tips: 18
2.3 Indentation Methods 19
2.4 Oliver-Pharr method 19
2.5 Factors affecting Nanoindentation Data: 21
2.6 Artificial Intelligence (AI) 23
  2.6.1 Machine Learning 25
  2.6.2 ML Models 26
2.7 Overview of Optimization Algorithm 32
  2.7.1 Genetic Algorithm 32
  2.7.2 Particle Swarm Optimization 34

CHAPTER 3: METHODOLOGY 36
3.1 Data Collection: 36
  3.1.1 Overview of Python Plotting 39
  3.1.2 Overview of MATLAB 42
3.2 Data Pre-processing 45

CHAPTER 4: RESULTS AND DISCUSSION 47
4.1 Box plot, Pair plot, Correlation heatmap 47
4.2 Performance Criterion of ML models 50
4.3 Features Selection 51
4.4 GA and PSO based hyperparameters tuning. 52
4.5 Models' performance 55
4.6 Partial dependence plots 55
4.7 Graphical representation between actual and predicted Pop-ins. 56
4.8 Graphical User Interface (GUI) 59

CHAPTER 5: CONCLUSIONS AND RECOMMENDATIONS 61
5.1 Conclusions 61
5.2 Recommendations 62

REFERENCES 63

LIST OF TABLES

Table 3.1: Data Collection Using Published Paper .... …..36
Table 4.1: GA and PSO parameters .... …..51
Table 4.2: GA and PSO based Features Selection .... …..52
Table 4.3: Parameter Ranges and Optimized Values in Selection Process. .... …..53
Table 4.4: Comparison of ML methods using PSO and GA .... …..55

LIST OF FIGURES

Figure 1.1: Mechanical Properties of Materials ... 1
Figure 1.2: Hall-Petch Relation between Strength and Grain Size [7] ... 2
Figure 1.3: Mechanisms involving dislocation interaction and slip transmission in neighboring grains (a), (b), (c), (d) [12] ... 3
Figure 1.4: First Pop-in event Occurrence within the Load-Displacement Curve ... 4
Figure 1.5: Pop-in Occurrence within the Load-Displacement Curve with critical load and critical displacement ... 5
Figure 1.6: Pop-in Events on the LD curve due to cracking in strontium titanate[13] ... 6
Figure 1.7: Pop-ins due to Phase transformation in Ni-Mn-Ga Films[14] ... 6
Figure 1.8: A load–displacement curve displaying the prominence of the grain boundary [16]. ... 7
Figure 1.9: Illustration showing dislocation pile-up and transmission at point [16] ... 8
Figure 1.10: The LD curves showing the pop-in events in interstitial free steel using a Berkovich tip [5] ... 9
Figure 1.11: The distribution of statistics of the c/d ratio and number of GB pop-in events [16] ... 10
Figure 1.12: A scanning electron microscope (SEM) image showing three closely spaced indents created next to a grain boundary [8] ... 10
Figure 1.13: Transmission of screw dislocations directly across a grain boundary[8] ... 11
Figure 1.14: Optical micrograph shows indentations in the vicinity of a GB [8] ... 12
Figure 1.15: The LD curves for three indents shown in figure 1.14 [8] ... 12
Figure 1.16: Schematic of the Workflow for Pop-in Prediction Utilizing Machine Learning ... 14
Figure 2.1: (a) Nano-indenter G200, (b) Cross-sectional View of G200 Nano-indenter [22] ... 17
Figure 2.2: Load-Displacement Curve during Nanoindentation [21] ... 17
Figure 2.3: Different Schematics of indenter tips [26] ... 18
Figure 2.4: The illustration of section of nanoindentation showing various quantities used in analysis [34] ... 21
Figure 2.5: Illustration demonstrating the impact of the initial depth of penetration on the load-displacement data obtained from a depth-sensing indentation test. The first contact load, denoted as Pi, leads to an initial penetration depth, referred to as hi. Depth measurements at loads P need to be adjusted for the influence of hi. [38]. ... 22
Figure 2.6: Figure demonstrating the contrast in contact areas between an ideal conical indenter and a non-ideal indenter[36]. ... 23
Figure 2.7: The illustration of the interconnections of data science, machine learning, artificial intelligence, deep learning, and data mining [43]. ... 24
Figure 2.8: The Machine Learning Workflow ... 26
Figure 2.9: Graphical Representation of Decision Tree [56] ... 27
Figure 2.10: Graphical Representation of SVM Structure [62] ... 29
Figure 2.11: Representation of GPR Structure ... 30
Figure 2.12: Representation of ELT Structure[56] ... 32

Figure 2.13: Flowchart of Genetic Algorithm [75] 33
Figure 2.14: Nature-inspired Technique (PSO) 34
Figure 2.15: Workflow of PSO [79]. 35
Figure 3.1: Berkovich indentations performed near GB in tungsten [4]. 37
Figure 3.2: Load Displacement curves of indent I1 to I4 [4]. 37
Figure 3.3: ECCI images of indent I4 at the surface and various polishing depths [4]. ... 38
Figure 3.4: Data Collection Using Google Collaboratory (Python Programming Language) 39
Figure 3.5: Representation of Box Plots 40
Figure 3.6: Representation of Pair Plots 41
Figure 3.7: Representation of Pearson correlation coefficient 42
Figure 3.8: Representation of Pre-Processing Algorithm. 42
Figure 3.9: Representation of Feature Selection in GA-GPR. 43
Figure 3.10: Representation of Feature Selection in GA-GPR. 44
Figure 3.11: Representation of Feature Selection in GA-GPR. 45
Figure 3.12: Machine Learning Workflow 46
Figure 4.1: (a) Depth (µm), Load (mN), Distance, dh Box Plot representation (b) Distance, dh Box Plot 47
Figure 4.2: Depth (µm), Load (mN), Distance, dh Pair Plot representation 48
Figure 4.3: Depth (µm), Load (mN), Distance, dh Pearson Correlation representation ... 49
Figure 4.4: PDP'S Demonstrating the Influence of Inputs on Pop-in Prediction 56
Figure 4.5: (a) GPR-GA predicted vs actual Pop-in (b) GPR-PSO (predicted Pop-in against actual Pop-in 57
Figure 4.6: (a) ELT- GPR predicted vs actual Pop-in (b) PSO-ELT (predicted Pop-in against actual Pop-in 58
Figure 4.7: Graphical User Interface (GUI) to predict pop-in 59
Figure 4.8: Graphical User Interface (GUI) to predict pop-in (a) indent I5, (b) indent I6. 60

LIST OF SYMBOLS, ABBREVIATIONS AND ACRONYMS

AI	Artificial Intelligence
GB	Grain Boundaries
CSM	Continuous Stiffness Measurement
LD	Load Displacement
ML	Machine Learning
DL	Deep Learning
DT	Decision Tree
SVM	Support Vector Machine
GPR	Gaussian Process Regression
EL	Ensembled Learning
ET	Ensembled Tree
ELT	Ensembled Learning Tree
GA	Genetic Algorithm
R^2	Coefficient of Determination
PSO	Particle Swarm Optimization
PDP	Partial Dependence Plot
GUI	Graphical User Interface

ABSTRACT

Nanoindentation is often utilized to examine the mechanical properties of materials and the interactions between grain boundaries (GBs) and dislocations. During the nano-indentation, Load-Displacement (LD) curves usually display the load or displacement burst, which is known as the “Pop-in”. If the indentations are conducted in close proximity to a GB, in addition to the first pop-in, a second distinctive displacement event could be seen on the LD curve under higher loads, often referred to as the "GB pop-in". The transmission of dislocations to the adjacent grains is one of the causes of these secondary GB pop-in occurrences. The present study introduces a novel strategy that differs from traditional advanced characterization methods by using machine learning techniques to predict GB pop-in events. Genetic algorithm (GA) and particle swarm optimization (PSO)-based machine learning models, including Gaussian process regression (GPR), ensemble learning tree (ELT), support vector machine (SVM), and decision tree (DT), are developed. Model selection is based on coefficient of determination (R^2) value. For GA the GPR, ELT, SVM, and DT, R^2 values were found to be 0.9999, 0.9264, 0.9711 and 0.9811, respectively, whereas for PSO, GPR, ELT, SVM and DT were found to be 0.9999, 0.9976, 0.9611, and 0.9682, respectively. It is evident from the aforementioned R^2 value that the GPR shows a value close to 1 as compared to the other three models, hence showing the best performance. Partial dependence plot (PDP) analysis underscores the significance of load and displacement parameters for precise prediction. Lastly, a user-friendly graphical interface (GUI) is meticulously designed based on the GA-GPR model. The integration of these novel methods enhances both their accuracy of predictions and the researchers' ability to detect the GB pop-in events, resulting in improving the fields of predictive modeling and materials science.

Keywords:

Nanoindentation, Machine Learning, Genetic Algorithm, Particle Swarm Optimization, Pop-in, Grain Boundaries

CHAPTER 1: INTRODUCTION

1.1 Mechanical Properties of Materials

The mechanical behavior of materials refers to how the material responds to external forces or loads [1]. The response of a material to applied forces is determined by its mechanical properties, which include strength, rigidity, ductility, hardness, toughness, resistance to fatigue and creep, and elasticity [2] as depicted in figure 1.1. These properties are critical to evaluate via impact, tensile, and hardness evaluations to determine a material's suitability for use in industries such as construction, aerospace, and automotive [3].

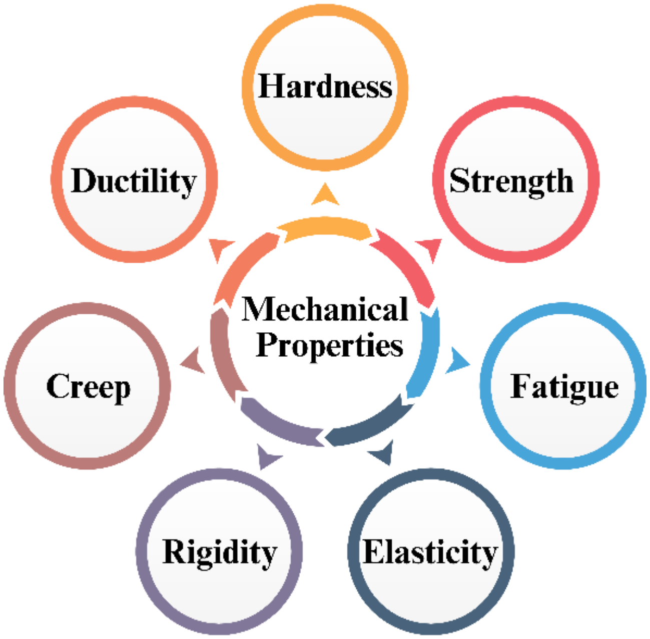


Figure 1.1: Mechanical Properties of Materials

Mechanical characteristics at the nanoscale exhibit unique behaviors in contrast to those that exist at the macroscopic level [4]. The grain boundaries impede the movement of dislocations, thereby increasing the strength of the material. Furthermore, ductility can be enhanced through the regulation of dislocation motion, which facilitates plastic deformation [5]. The restricted particle size in nanocrystalline materials induces modifications in dislocation behavior, thereby generating unique deformation mechanisms that may impact the material's overall mechanical response [6].

The Hall-Petch relationship in materials science states that there is a direct correlation between the decrease in grain size of a metal and the rise in its yield strength [7].

$$\sigma_y = \sigma_\theta + K.d^{1/2}$$

The term "σ_y" represents the yield strength of the material. The symbol "σ_θ" is a constant that signifies the friction stress, which is the intrinsic opposition to the movement of dislocations, the Hall-Petch constant, denoted as "k", quantifies the phenomenon of grain boundary strengthening and d represents the mean size of the grains [8]. This relationship tells us about the average grain size. More grain boundaries in a material with smaller grain sizes limit the movement of dislocations and increase the material's strength. However, at very small grain sizes, other variables such as grain boundary sliding or changes in the underlying deformation process might change the material's characteristics, thus this link may not last forever.

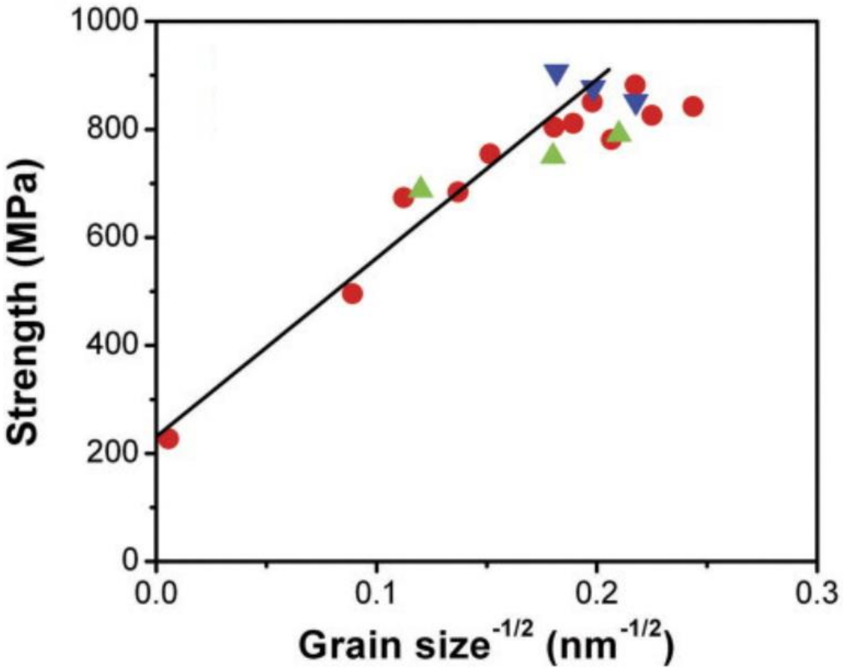


Figure 1.2: Hall-Petch Relation between Strength and Grain Size **[7]**

The relationship between grain size and material strength is perfectly proportional; furthermore, a larger grain size corresponds to a lower strength of the material; conversely, a smaller grain size corresponds to a higher strength of the material as shown in figure 1.2.

1.2 Slip transfer and dislocation interaction mechanisms in adjacent grains.

A critical aspect in understanding material deformation is the correlation between slip transfer and dislocation interaction at grain boundaries [9]. The compatibility of dislocations across grain boundaries is critical for slip transfer, which involves the movement of dislocations. The behaviors of dislocations—transmitting, reflecting, or bowing out at boundaries—have an impact on the deformation of materials [10]. Slip transfer is dependent on the alignment of slip systems and dislocation interactions, which can either facilitate or impede movement. The mechanical response of polycrystalline materials is substantially shaped by this interaction, which in turn affects their deformation mechanisms and material properties [11].

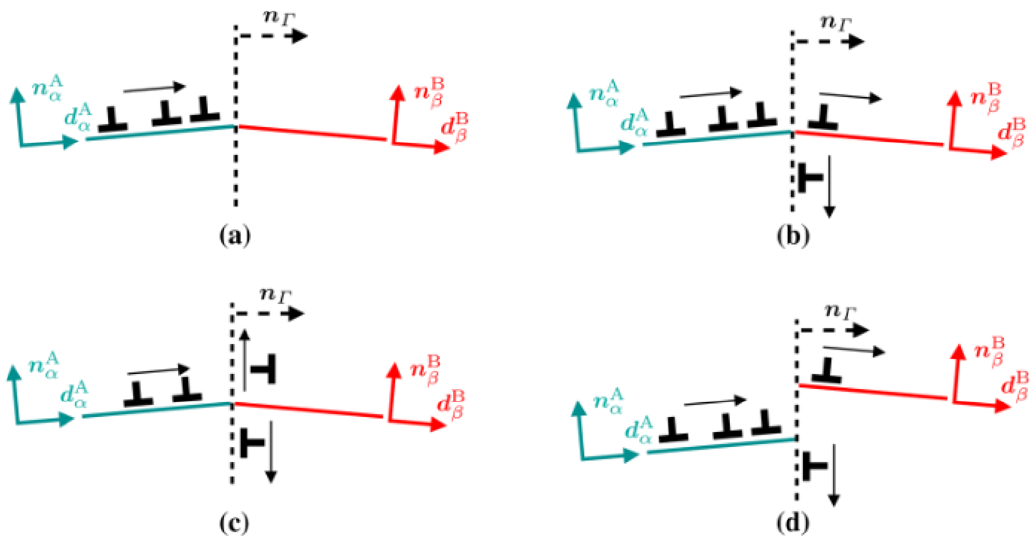


Figure 1.3: Mechanisms involving dislocation interaction and slip transmission in neighboring grains (a), (b), (c), (d) **[12]**

Furthermore, figure 1.3 depicts several modes of slip transmission and dislocation contact between grains A and B, which are separated by a grain boundary with normal n.

The orientations of slip plane normal are denoted as $n_A{}^a$, $n_B{}^b$, and the orientations of slip directions as $d_A{}^a$, $d_B{}^b$. There are N slip systems inside grain A, labelled from a=1 to N, and N slip systems within grain B, labelled from b=1 to N. The illustration presents many situations: At the grain boundary, dislocations may initially accumulate (Fig. 1.3a). As a result, dislocation sources may become active, which would cause dislocations to be emitted from grain B as well as from the grain boundary itself (Fig. 1.3b). On the other hand, dislocations might separate into the grain boundary without first releasing dislocations into grain B (Fig. 1.3c). Nonetheless, grain B may experience a re-emission of resolved dislocations in the future (Fig. 1.3d) [12].

1.3 First Pop-in

In nanoindentation testing, pop-in events represent a sudden burst in the load displacement curve, while keeping the load constant. These events provide information about the material’s behavior, which changes based on alterations in the material or its characteristics. Pop-in events in the load-displacement curve represent the material's response at different applied loads. Several factors contribute to these pop-in occurrences. As schematized in figure 1.4, the initial pop-in occurrence generally arises from the transition from elastic to plastic transition.

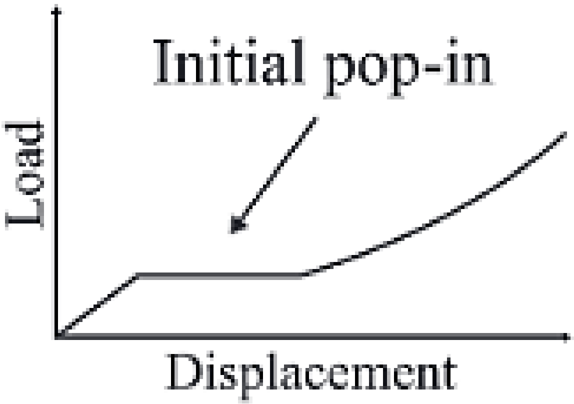


Figure 1.4: First Pop-in event Occurrence within the Load-Displacement Curve

When the nanoindentation experiment is load-controlled, a horizontal plateau can be observed on the load-displacement curve. This plateau arises when a pop-in event takes place at the critical load and critical displacement as depicted in figure 1.5.

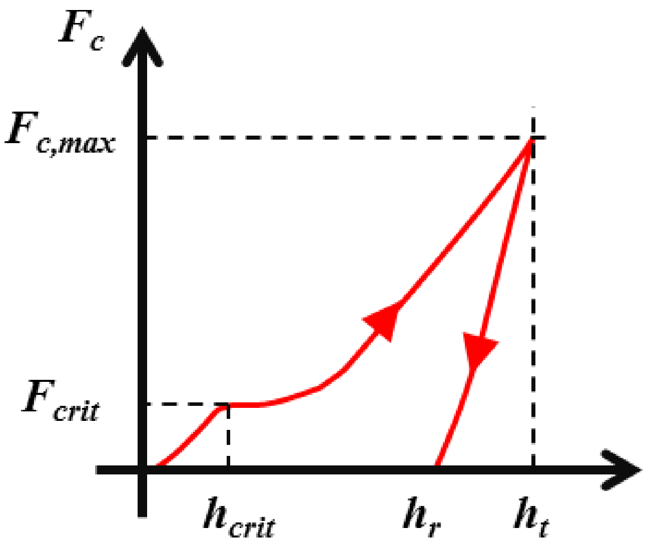


Figure 1.5: Pop-in Occurrence within the Load-Displacement Curve with critical load and critical displacement

1.4 Secondary Pop-ins

Secondary or Multiple pop-in events can be caused by various factors such as phase transformations, cracking, and the transmission of dislocations from one grain to an adjacent grain. Figure 1.6 visually represents the occurrence of multiple pop-in events due to cracking in the strontium titanate single crystal.

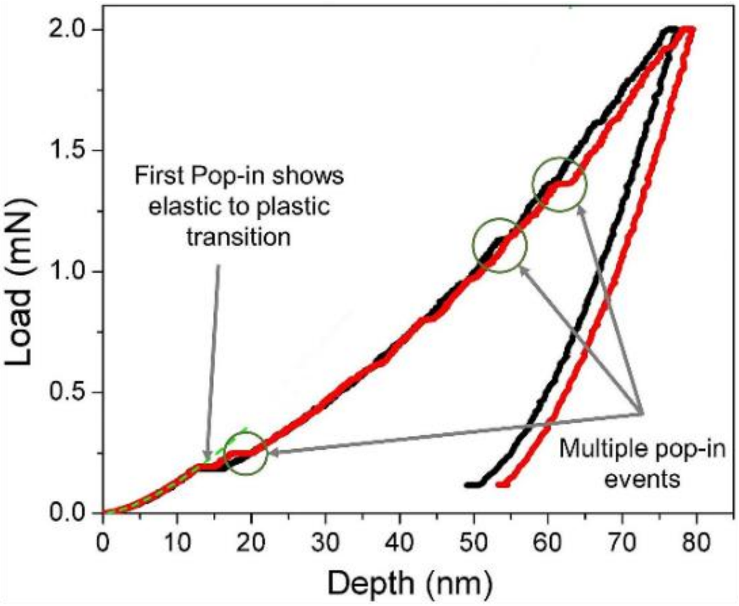


Figure 1.6: Pop-in Events on the LD curve due to cracking in strontium titanate[13]

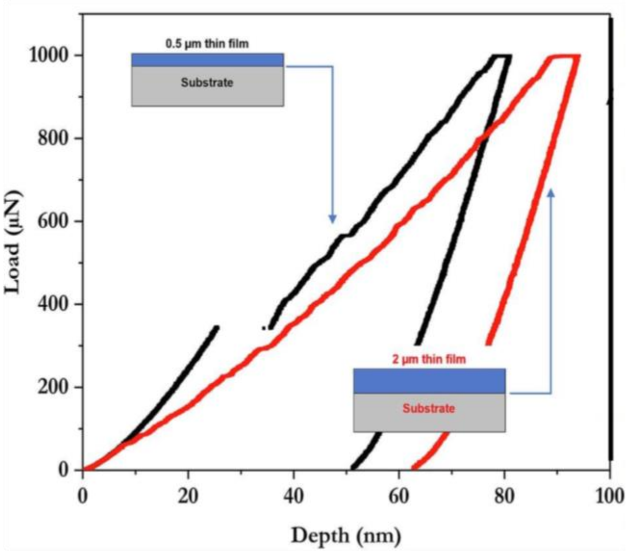


Figure 1.7: Pop-ins due to Phase transformation in Ni-Mn-Ga Films**[14]**

Figure 1.7 shows the multiple pop-ins due to phase transformation in Ni-Mn-Ga films [14]. These events serve as significant sources of information regarding the mechanical characteristics of materials, including hardness, fracture toughness, and incipient plasticity. The transmission of dislocations from one grain to an adjacent grain which is Grain Boundary Pop-in (GB) details are given in proceeding section.

1.5 Grain Boundary Pop-in

During nanoindentation testing, grain boundary pop-ins are observed on the load-displacement graph, identified as displacement bursts as shown in figure 1.8. [15]

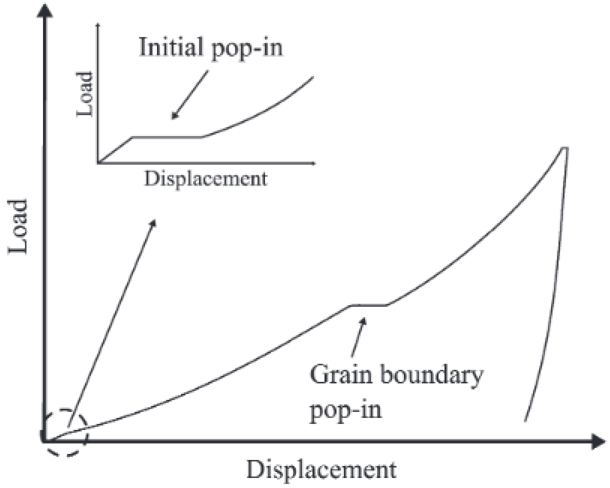


Figure 1.8: A load–displacement curve displaying the prominence of the grain boundary **[16].**

Dislocations exhibit a progressive displacement within the grains in response to applied stress; however, their motion is impeded at the GB until a critical stress is achieved. Subsequently, strain and stress undergo an abrupt increase, because of which a multitude of dislocations abruptly overcome the boundary barrier and achieve a constrained yet rapid traversal across the grain boundary as shown in figure 1.9.

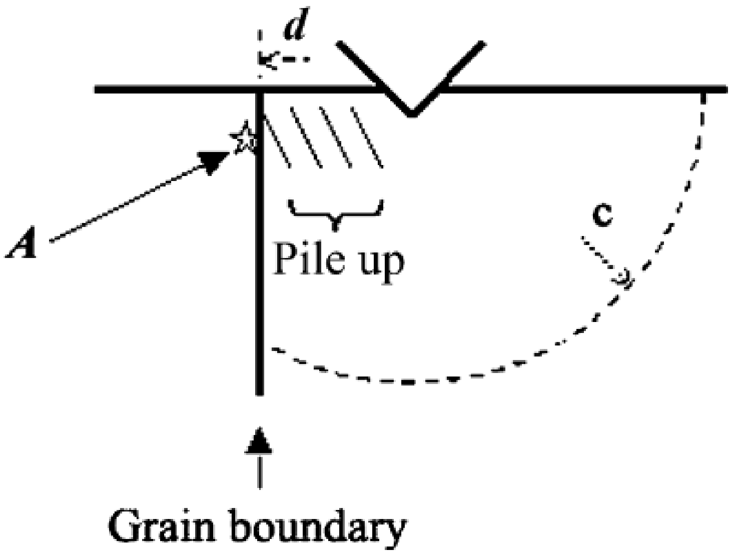


Figure 1.9: Illustration showing dislocation pile-up and transmission at point **[16]**

Kalidindi et al. [5], reported that sharp indenters are exceptionally efficient in inducing extremely localized plastic deformation. As a result, when fully annealed, well-prepared materials were tested by indentation using sharp tips, the elastic to plastic transition as explained earlier is called Pop-in or dislocation burst. When these tests were performed, the dislocation activation sources and occurrence of Pop-ins were very difficult, and the values of load and displacement reported are greater. However, when these tests are performed near the (GB) with the use of indenters having sharp tip, the pop-ins can occur on small loads and depth as depicted in figure 1.10 compared to values reported for load and displacement in grain interior [5].

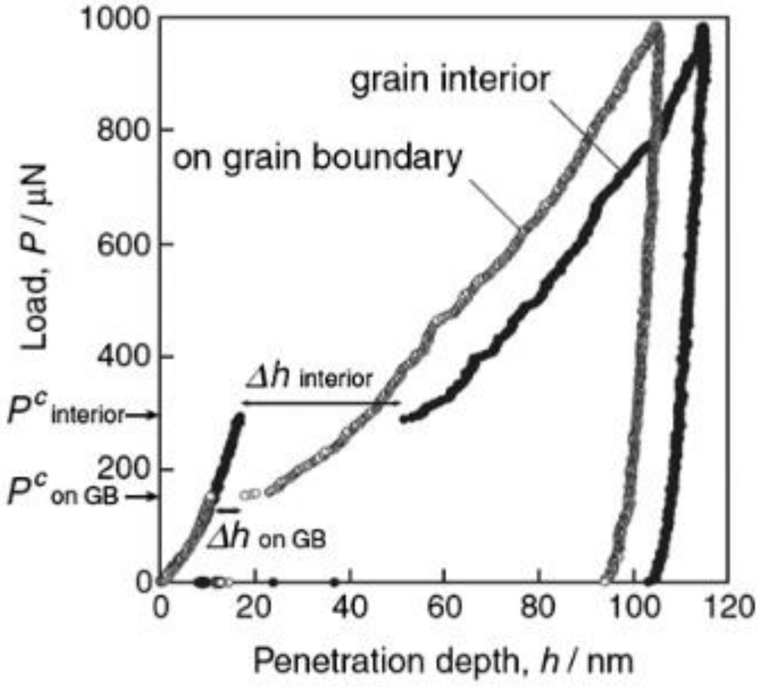


Figure 1.10: The LD curves showing the pop-in events in interstitial free steel using a Berkovich tip **[5]**

Wang et al. **[16],** performed the nanoindentation testing in niobium and proposed a c/d criteria for the occurrence of the GB pop-in events, where c is the radius of elasto-plastic boundary and d is the distance of indenter tip from the GB. They find a narrow range (between 1.5 to 5) of c/d ratio for different GBs (figure 1.11)**[16]**.

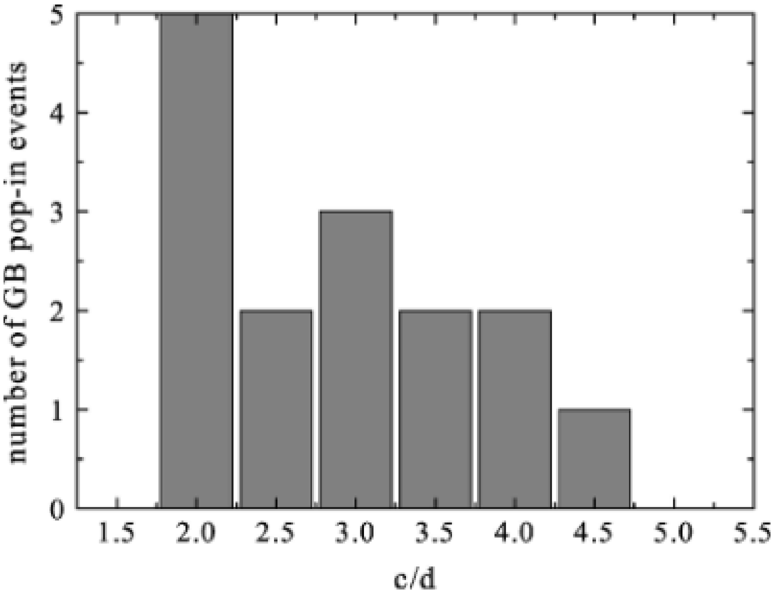


Figure 1.11: The distribution of statistics of the c/d ratio and number of GB pop-in events **[16]**

Figure 1.12 shows the exemplary image of the SEM indents from the same work [16], which shows that out of three indents, the GB pop-in was observed only in indent 1. It is also evident from the image that indent 1 was closer to GB as compared to indent 2 and 3, thus no slip transmission was observed.

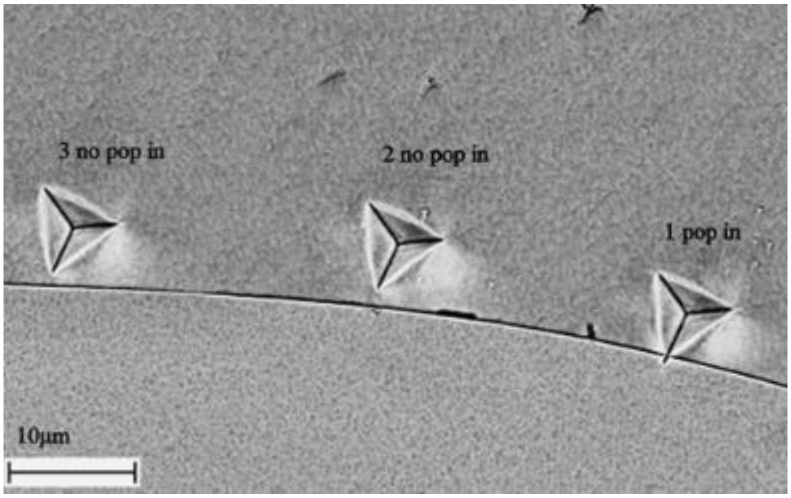


Figure 1.12: A scanning electron microscope (SEM) image showing three closely spaced indents created next to a grain boundary **[8]**

Britton et al. [8], conducted nanoindentation experiments in iron 0.01 weight percentage carbon and found the GB pop-in events, which were also associated with the transmission of dislocation in the adjacent grain (figure 1.13). They also studied the c/d ratio for the various GBs and observed a narrower range [8].

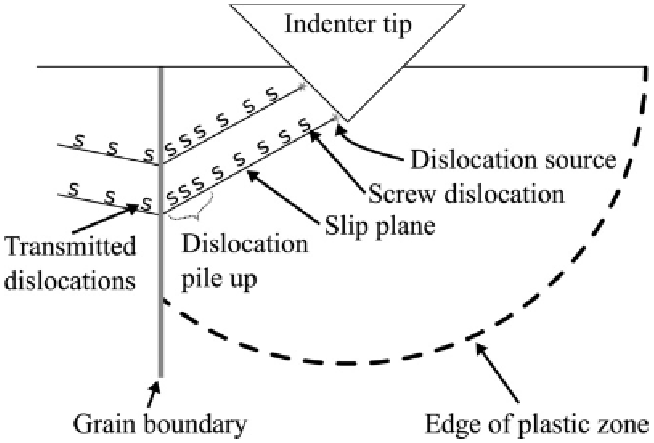


Figure 1.13: Transmission of screw dislocations directly across a grain boundary**[8]**

Figure 1.14 shows the three indentations performed at various distances from a GB. Britton et al. observed the GB pop-in for indent 2 and 3 (figure 1.15) and reported that c/d =1.2, critical value was between 1.5 to 5 [8].

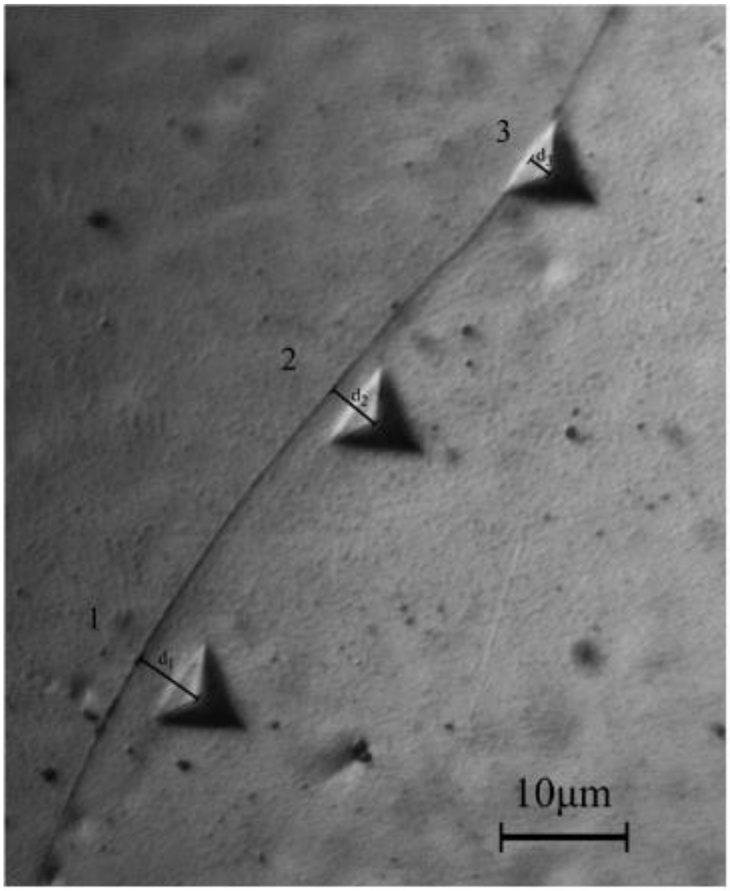


Figure 1.14: Optical micrograph shows indentations in the vicinity of a GB **[8].**

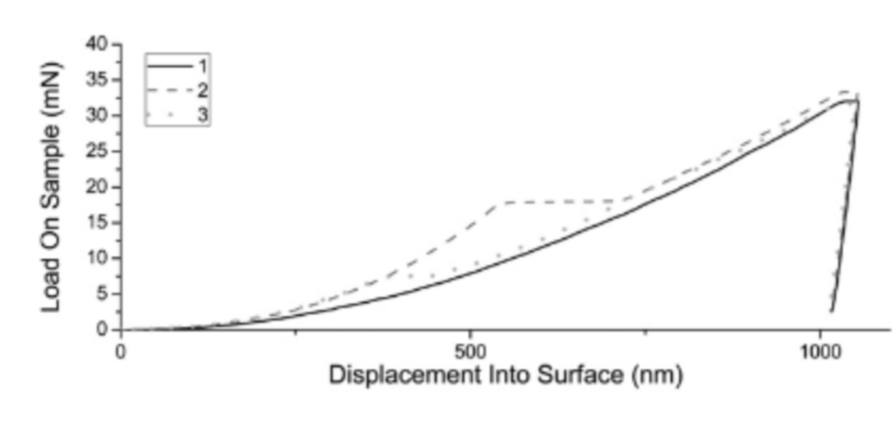


Figure 1.15: The LD curves for three indents shown in figure 1.14 **[8]**

1.6 Research problem and hypothesis

In the different literature, [5], [8], [16], GB pop-ins were not always likely to occur when indentation was performed near the GB. One potential hypothesis regarding the GB pop-in phenomenon suggests that as the indenter tip penetrates, dislocations nucleate in the plastic zone and move towards the GB, eventually forming a pile-up of dislocations. Once a critical shear stress is reached on the slip system of the neighboring grain, the dislocations are expected to transfer across the grain, resulting in the GB pop-in event observed on the LD curve. The absorption of dislocations at the GB and their subsequent re-emission from the GB are suggested to be the underlying process responsible for the grain boundary pop-in occurrences, but to observe this kind of phenomenon, a comprehensive tool is required. Due to their high cost, nanoindentation devices are not widely available. It takes expertise and training to carry out and interpret nanoindentation tests correctly. To address this issue, an artificial intelligence (AI)-based machine learning (ML) approach is used to predict the occurrence of the GB pop-in phenomenon using nanoindentation data.

1.7 Proposed Methodology

Factors such as applied stress, indenter geometry, and misorientation significantly contribute to the formation of multiple GB pop-ins[17]. Due to the influence of these factors, the GB pop-ins were not observed on the load-displacement curves, even for similar GB conditions [4]. Therefore, in the present work, machine learning (ML) will be used to predict the grain boundary pop-in events. For this purpose, four machine learning (ML) models, were developed, which include Ensemble Learning Tree (ELT), Gaussian Process Regression (GPR), Support Vector Machine (SVM), and Decision Tree (DT), to forecast the GB pop-in by using nanoindentation data. A comprehensive visual representation of building ML models with the help of nanoindentation data for the Pop-in prediction is shown in figure 1.16. These machine learning models essentially use depth (μm) and load (mN) data as input parameters [4]. Data-driven modelling approaches were used to create models that were optimal for process prediction and validation. Using nanoindentation data, the next step was developing machine learning models based on genetic algorithms (GA) and particle

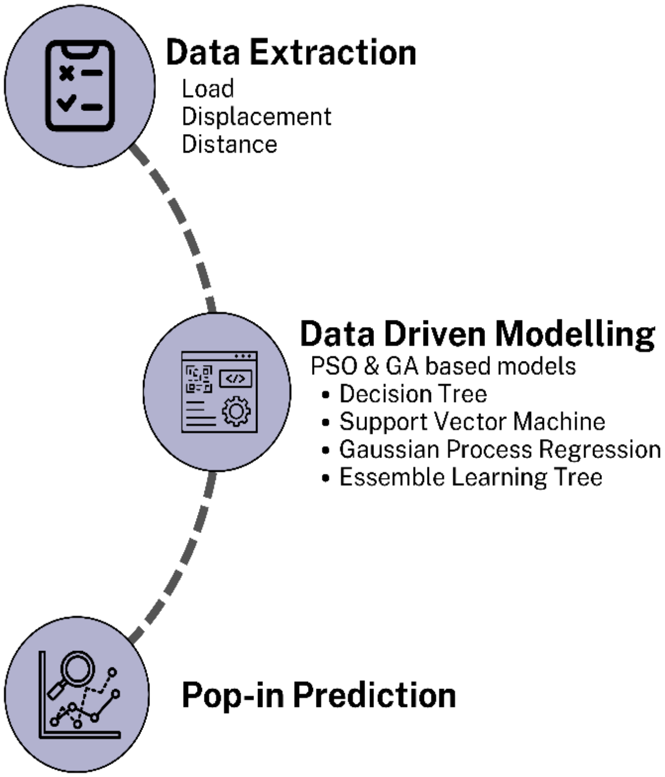


Figure 1.16: Schematic of the Workflow for Pop-in Prediction Utilizing Machine Learning

swarm optimization (PSO) to optimize parameters that accurately predict and assess Pop-in occurrences. This integrated strategy's combining of ML models with GA and PSO modification has a lot of potential for using ML models in experimental Pop-in prediction research. In the end, MATLAB was used to develop a graphical user interface that combines machine learning techniques with ideal features to enable Pop-in formation predictions using nanoindentation. Researchers will find this study convenient as it reduces the time necessary to use resources efficiently by using a trained machine learning model based on nanoindentation data.

1.8 Objectives

- Development of Machine Learning models (SVM, GPR, ELT, DT) to predict the GB pop-in event.
- Optimization of ML models’ hyperparameters using PSO and GA.
- Development of Graphical User Interface (GUI).

CHAPTER 2: LITERATURE REVIEW

2.1 Nanoindentation

Nanoindentation is a characterization technique to measure the different mechanical properties of materials at both the small and bulk scales [18] which include hardness, yield stress, complex modulus, strain rate sensitivity, fracture, toughness etc. A typical nano indenter G200 and its cross-sectional representation are depicted in Figure 2.1. The G200 nano indenter comprises various components, including an indenter, an optical microscope, and a workbench or sample stage. The indenter and optical microscope are interconnected through a capacitance gauge, set at a specific angled distance.

Figure 2.1 (b) presents a cross-sectional view of the G200 nanoindenter. In the experimental setup, a sample is positioned on the workbench, and the load is applied to the sample. Following the experimentation, the indenter's impression is meticulously analyzed. The microscope is carefully positioned over the indented region for detailed examination and assessment. This is significant in disciplines such as microelectronics, where comprehension of the efficiency and durability of thin layers is essential [19]. In addition to these characteristics, it is also possible to determine the viscoelasticity, creeping, phase transition, dislocation movement, strain hardening effect, and residual stress [18]. It studies the effects of surface modifications including chemical doping and ion implantation. Additionally, it has several uses in the biomedical sector, such as the study of dental materials and bone [20]. The nano indenters are also capable of precisely recording small loads and movements in the form of load vs. displacement data, as shown in figure 2.2 [21].

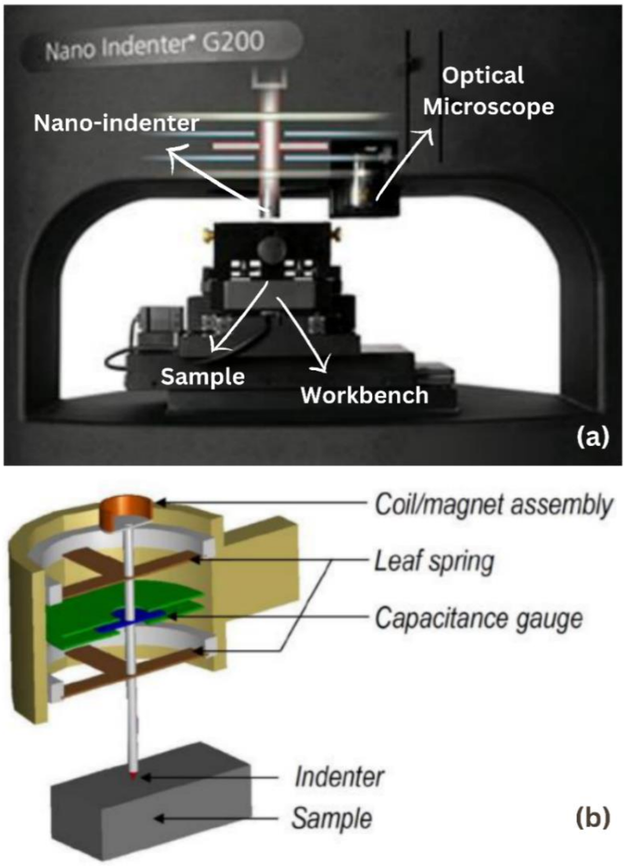


Figure 2.1: (a) Nano-indenter G200, (b) Cross-sectional View of G200 Nano-indenter **[22]**

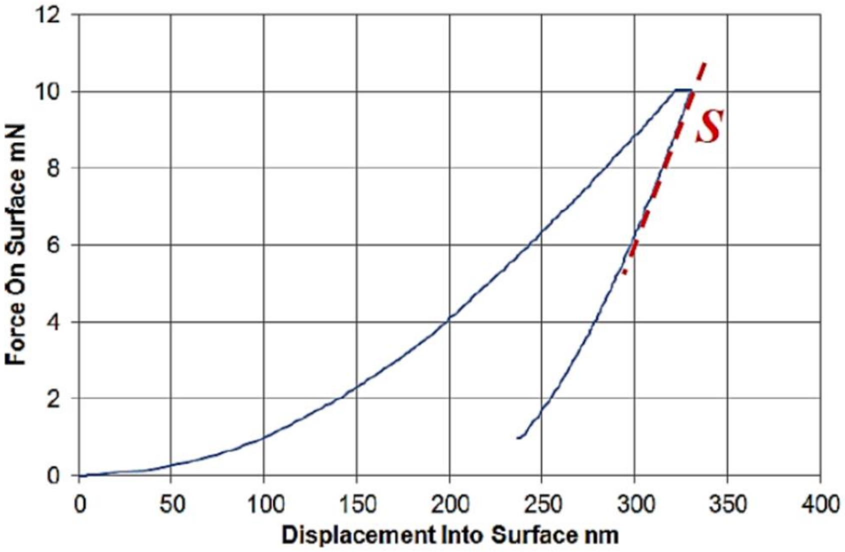


Figure 2.2: Load-Displacement Curve during Nanoindentation **[21]**

2.2 Types of Indenters Tips:

A range of indenters are used in nanoindentation, although wedge-, spherical-, and pyramid-shaped indenters are the most common types. The cubic-angled indenter, triangle pyramidal Berkovich indenter, rectangle pyramidal Vickers indenter, and Knoop indenter are examples of common pyramidal indenters [23]. Every indenter has unique uses and constraints of its own. For example, the Berkovich indenter's well-defined geometry and three-sided pyramidal form serve as an appropriate tool for measuring hardness and elastic modulus [24]. As a result of its conical form, the conical indenter is useful in circumstances where materials display non-linear behavior. The spherical indenter has very little initial contact stress since it is made in the form of sphero-cones to make installation easier [24]. They are thus suitable for testing flexible materials and modelling contact damage that might happen during operation. However, achieving the higher quality of submicron-scale spherical diamond indenters is a major challenge to their broader use. Furthermore, cube-edge indenters are utilized for the micro-indentation of soft materials. They feature a pyramidal form with a minutely split corner [25]. The different indenter tips are illustrated in figure 2.3.

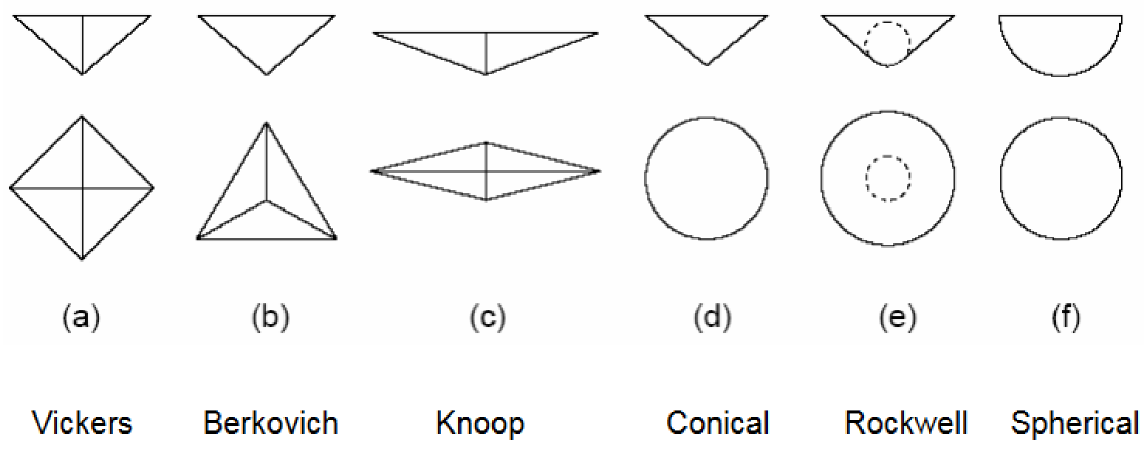


Figure 2.3: Different Schematics of indenter tips **[26]**

2.3 Indentation Methods

Nanoindentation, an experimental technique that investigates mechanical properties at the nanoscale, employs a variety of indentation methodologies to precisely evaluate material properties. There are two modes of operation for the nano-indenter: quasi-static loading and continuous stiffness measurement (CSM) [27]. The quasi-static loading method is a useful tool for determining the hardness and elastic modulus of materials under static conditions. It also sheds light on how materials behave when subjected to gradual and consistent force application. This critical phase is reached by substances that are consistently under sustained stress. The CSM mode from MTS Systems Corporation offers an additional technique for dynamic nanoindentation. The indenter receives a low-amplitude signal as it passes through the material. This method enables the simultaneous assessment of hardness and elastic modulus at various depths in a single indentation [23]. For examining gradient changes in the mechanical properties of coatings, thin films, or layered materials, this function is very useful. Moreover, the CSM mode is especially helpful for studying the viscoelastic properties of materials that, when deformed, display both elastic and viscous properties. Because CSM is dynamic, studying how these materials respond to changing pressures over time may help researchers understand how these materials perform under real-world conditions. The CSM mode may also adjust the strain rate indefinitely, allowing for consistent evaluation of materials under comparable strain rate settings and research into strain rate sensitivity in sectors such as aerospace and automotive. Furthermore, the CSM mode makes it easier to adjust the area function of the indenter, which is essential for accurately determining material properties [28]. Overall, the nano-indenter provides extensive material characterization capabilities, with the quasi-static loading mode giving critical data on material reactions under constant load and the CSM mode providing dynamic testing capabilities [29].

2.4 Oliver-Pharr method

The Oliver-Pharr method is used to determine the hardness and elastic modulus during the nanoindentation testing[30]. Researchers derive vital information regarding a material's resistance to deformation and its intrinsic elasticity by evaluating this data using specialized equations that take into account parameters such as indenter shape and material behavior [31].

This standardized process has a broad range of applications across many materials such as metals, ceramics, and thin films. The precision and standardized procedures of this tool make it a great instrument for comparing findings across different investigations [32]. Nevertheless, the use of specialized equipment and experience is essential, and there are certain constraints to consider, such as the need for meticulous surface preparation and the reliance on assumptions about material behavior. The Oliver-Pharr approach continues to be a very effective method for uncovering the complex nature of material mechanics at the micro and nano levels [33] as depicted in figure 2.4

The hardness is extracted out by using following equation.

$$\boldsymbol{H} = \frac{\textbf{Load (P)}}{\textbf{Area(A)}}$$

Whereas $\boldsymbol{A} = \mathbf{24.56}\ \boldsymbol{h_c^2}$ when the depth of indentation is greater than 2 microns

$A = 24.56\ h_c^2 + C1h_c^1 + C2h_c^{1/2} + C3h_c^{1/4} + \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots + C8h_c^{1/128}$, when There is less than 2 microns of indentation depth, while here c^1 to c^8 are constants,

This method is known as "**Area Function Calibration**".

Contact depth h_c is measured by using the following Equation.

$$\boldsymbol{h_c = h_{max} - \varepsilon \frac{P_{max}}{S}}$$

whereas ε = 0.75 for Berkovich Geometry.

Here S is called Sneddon's relation, and it can be extracted by following equation.

$$\boldsymbol{E_r = \frac{\sqrt{\pi}}{2} \frac{S}{\sqrt{A}}}$$

Now Reduced Modulus E_r can be extracted using the following formula.

$$\frac{1}{E_r} = \frac{1-\nu_s^2}{E_s} + \frac{1-\nu_i^2}{E_i}$$

Er denotes the elastic deformation that occurs in both the sample and the indenter tip.

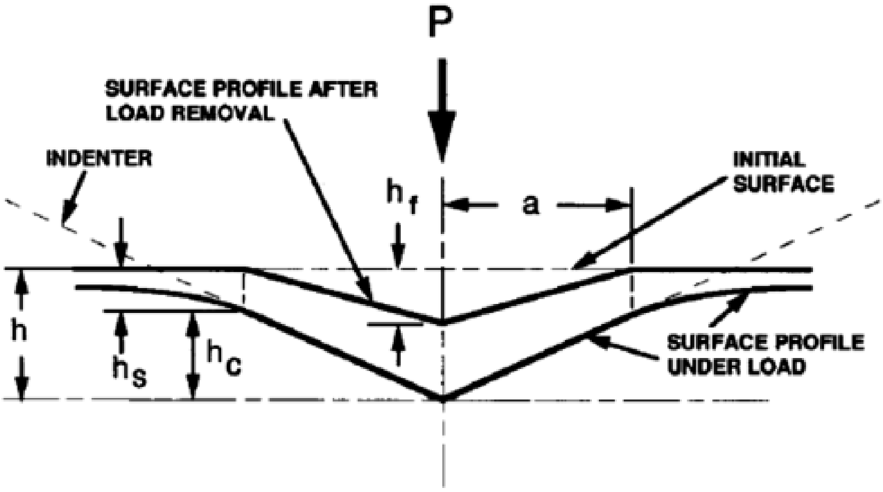

P
SURFACE PROFILE AFTER LOAD REMOVAL
INDENTER
h_f
a
INITIAL SURFACE
h
h_s
h_c
SURFACE PROFILE UNDER LOAD

Figure 2.4: The illustration of section of nanoindentation showing various quantities used in analysis **[34]**

2.5 Factors affecting Nanoindentation Data:

Analyzing the nanoindentation data is not as simple as it seems. A complex domain of variables impacts the results, perhaps leading researchers into misleading directions [35].

The material itself plays a starring role. The complex arrangement of grains, limits, and defects inside its microstructure determines its response to the force applied by the indenter[24]. The measured hardness, modulus, and occurrence of pop-ins are all influenced by initial penetration depth, Poisson’s ratio, thermal drift, piling-up, and surface roughness, grain size, residual stress, size effect, indenter geometry, and the presence of impurities [36]. The surface state, regardless of its pristine or oxidized, and whether it is smooth or rough, may impact the outcomes due to variables such as friction and adhesion. For materials that showcase different properties depending on direction, the chosen indentation direction can significantly impact the data obtained[37]. Figure 2.5 depicted the effect of effect of initial penetration depth on load-displacement data for a depth-sensing indentation test.

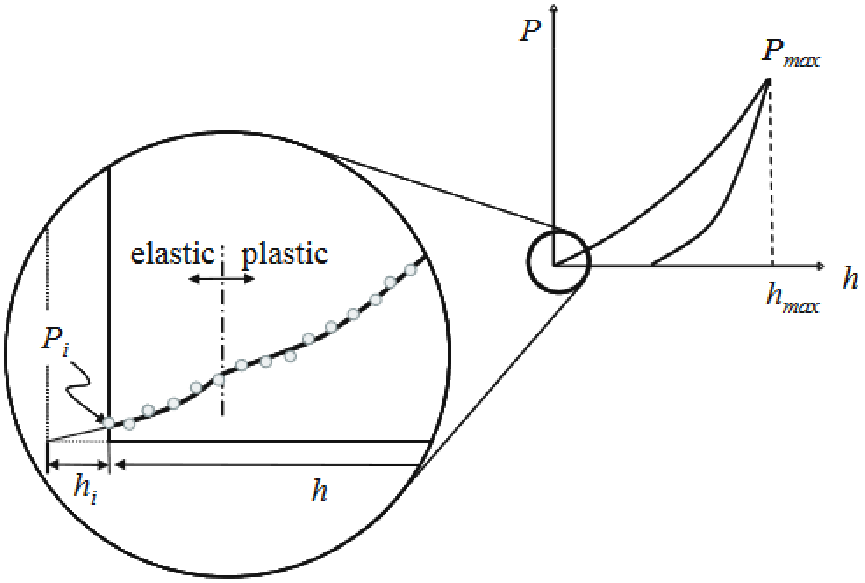


Figure 2.5: Illustration demonstrating the impact of the initial depth of penetration on the load-displacement data obtained from a depth-sensing indentation test. The first contact load, denoted as Pi, leads to an initial penetration depth, referred to as hi. Depth measurements at loads P need to be adjusted for the influence of hi. [38].

The area of contact in nanoindentation is essential for precise determination of material parameters such as hardness and elastic modulus. Beyond numerical parameters other parameters that affect[39]. The material next to the indentation point may pile up (pile-up) or sink inwards (sink-in). The real contact area is distorted by these effects, hence accurate measurement is essential. Pile-up might result in an underestimating of the contact area, while sink-in can lead to an overestimation, depending on the material and loading circumstances[40] as shown in figure 2.6.

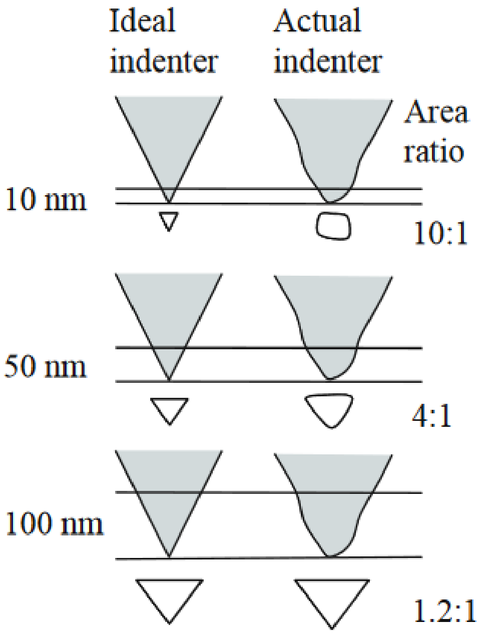


Figure 2.6: Figure demonstrating the contrast in contact areas between an ideal conical indenter and a non-ideal indenter**[36].**

2.6 Artificial Intelligence (AI)

AI pertains to the advancement of algorithmic tools that exhibit intelligent behavior and simulate human intelligence [41]. To achieve optimal efficiency, the discipline places emphasis on three fundamental abilities: learning, reasoning, and self-correction. AI may pertain to computer programmes that are either trained through machine learning or are explicitly programmed [42].

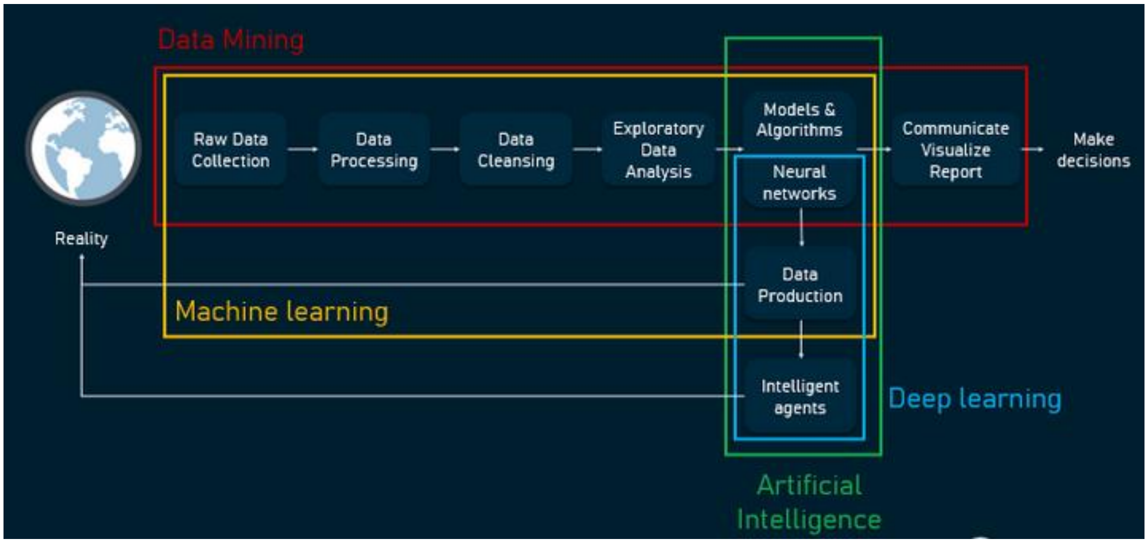


Figure 2.7: The illustration of the interconnections of data science, machine learning, artificial intelligence, deep learning, and data mining **[43].**

ML is a subfield of AI that focuses on the creation of models that enable machines to gradually enhance their performance. Deep Learning, a significant machine learning methodology, employs multilayered neural networks that demonstrate remarkable efficacy in both areas of speech and image recognition [44]. In contrast to machine vision, which interprets visual data, Natural Language Processing enables computers to comprehend and generate human language. AI and its subfields are represented in the figure 2.7 [45]. Artificial intelligence (AI) has applications in a wide range of domains, including robotics, natural language generation, and expert systems. The fundamental objective of general AI is to more closely resemble human intelligence, whereas narrow AI is purposefully designed to carry out particular responsibilities [46]. Its significance extends across industries, including finance and healthcare, and is crucial for the advancement of intelligent systems and automation. Our data is based on numerical values, and ML is particularly well-suited for handling numerical data, often excels in tasks that involve quantitative information. Therefore, it is better to use the ML approach to develop a model for the prediction of Pop-in phenomenon. Moreover, ML is a purposeful endeavor rooted in the acknowledgment of its capacity for revolutionization. It acts as a catalyst to facilitate the disclosure of valuable insights, the automation of intricate processes, and the improvement of decision-making capabilities in a data-rich environment. Gaining knowledge in the field of machine learning equips individuals with the essential abilities to proficiently adjustments through the complexities of data analytics, forecast forthcoming trends, and extract functional insights [47]. A detailed overview of machine learning is addressed in the subsequent section.

2.6.1 Machine Learning

Machine learning (ML) is an artificial intelligence (AI) subfield that focuses on the construction of algorithms and statistical models. Operating without explicit programming, these models empower computers to independently acquire knowledge and produce predictions or decisions [43]. ML techniques are utilized for data interpretation and analysis, pattern discovery, and data-driven decision-making [48]. Here’s an overview of ML and its methods. It operates on the principle that computers can learn from data and improve their performance over time [49]. The key components of ML are as follow:

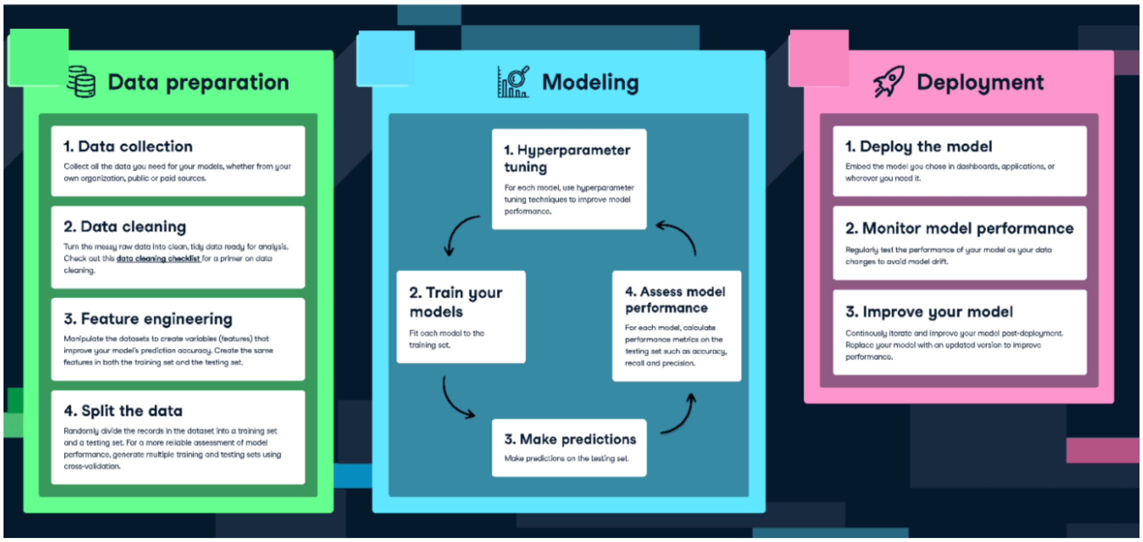
Data: ML algorithms require data to self-train and generate predictions by recognizing patterns. Relevant and high-quality data are essential for effective machine learning.

Features: Predictions are generated by the ML algorithm using features, which are columns or specific characteristics of the input variables in the data. Feature engineering or selection is a crucial phase in ML.

Model: The primary objective of a model is to provide a mathematical representation of the relationships that exist between the objective variable and the data features [50]. By developing the model through training on historical data, it gains the ability to produce predictions on new, unobserved data.

Training: ML models alter their internal parameters to minimize prediction errors as they learn from historical data during the training phase.

Testing and Evaluation: After training, the model is tested on new data to assess its performance and accuracy. Various evaluation metrics are used, depending on the problem type (e.g., classification, regression) [51].



Data preparation

1. Data collection
Collect all the data you need for your models, whether from your own organization, public or paid sources.

2. Data cleaning
Turn the messy raw data into clean, tidy data ready for analysis. Check out this data cleaning checklist for a primer on data cleaning.

3. Feature engineering
Manipulate the datasets to create variables (features) that improve your model's prediction accuracy. Create the same features in both the training set and the testing set.

4. Split the data
Randomly divide the records in the dataset into a training set and a testing set. For a more reliable assessment of model performance, generate multiple training and testing sets using cross-validation.

Modeling

1. Hyperparameter tuning
For each model, use hyperparameter tuning techniques to improve model performance.

2. Train your models
Fit each model to the training set.

3. Make predictions
Make predictions on the testing set.

4. Assess model performance
For each model, calculate performance metrics on the testing set such as accuracy, recall and precision.

Deployment

1. Deploy the model
Embed the model you chose in dashboards, applications, or wherever you need it.

2. Monitor model performance
Regularly test the performance of your model as your data changes to avoid model drift.

3. Improve your model
Continously iterate and improve your model post-deployment. Replace your model with an updated version to improve performance.

Figure 2.8: The Machine Learning Workflow

2.6.2 ML Models

ML comprises an extensive array of methodologies and strategies. The method selection is determined by the nature of the data available and the issues at hand. A total of four ML models SVM, GPR, ELT, and DT were used to train and forecast the Pop-ins using load vs displacement data in MATLAB. Detailed descriptions of each of these models are presented in the subsequent sections:

2.6.2.1 Decision Tree

In discriminant analysis, decision trees are regarded as a basic method for information discovery. The system's enhanced standing is attributed to its efficiency, promptness of operation, and simple installation. Using this method, a large configuration of nodes and branches forms a structure like a tree. The process of categorizing individuals into groups is accomplished by employing function-specific mechanisms for classification or regression via internal nodes that possess outgoing edges. Furthermore, values may be computed or maintained at the terminal nodes or leaves, contingent on the model type [52]. Regression trees are fundamentally represented by a constant value at the root level, which is typically the target attribute's mean value. To further develop this concept, model trees replace these constants with linear or nonlinear regression functions.

To make a prediction, a model advances from the root to a leaf, where the attributes of the instance direct the path at each internal node. The prediction is subsequently produced by the leaf's regression model [53]. The decision tree's functionality is predicated on the algorithm, which is constructed using the samples provided. This algorithm uses the minimization of a fitness function to identify the optimal tree. In situations where classes are not predefined, independent variables are utilized to apply a regression model to the objective variable. The dataset is partitioned into distinct positions for each variable. The algorithm assesses the discrepancy between the values reported and those predicted for each division, employing the fitness function as its foundation. The procedure proceeds recursively, selecting the division point that minimizes the error, as mentioned earlier [54]. In various domains such as finance, healthcare, and machine learning, decision trees are widely utilized due to their efficacy in handling intricate datasets and producing comprehensible outcomes. Figure 2.9 presents a graphical representation of the comprehensive operational process of the DT algorithm, which demonstrates its capacity to streamline decision-making processes and deliver satisfactory predictions [55].

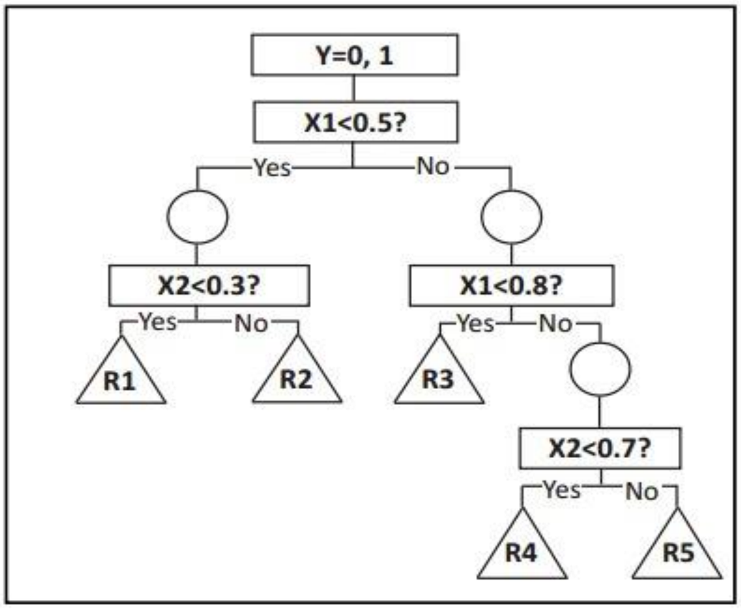


Figure 2.9: Graphical Representation of Decision Tree **[56]**

2.6.2.2 Support Vector Machine

Support Vector Machines (SVM) is a supervised machine learning method that excels at both classification and regression [57]. As a further development of the SVM algorithm, it regulates the connection between an output variable and one or more predictors. By posing an optimization problem, the SVR algorithm gains understanding of a regression function that establishes a correlation between observed response values and input variables [58]. Support Vector Regression (SVR) is widely recognized for its unique capability of effectively managing high-dimensional data by precisely balancing model complexity and prediction accuracy [59]. To preserve this equilibrium, a number of foundational principles derived from SVM and modified for regression are implemented. (a) The separating hyperplane; (b) The kernel function; (c) The maximum-margin hyperplane; and (d) The soft margin. A critical concept in SVM, the separating hyperplane establishes a boundary for differentiating classes in classification tasks. As illustrated in Figure 2.10, the maximum-margin hyperplane improves the robustness of classification by maximizing the distance between the boundary and the nearest data points belonging to different classes. By incorporating imperfections present in real-world data and striking a balance between margin width and misclassification levels, the soft margin approach enhances the generalizability of support vector machines (SVMs). Furthermore, support vector machines (SVM) and support vector regions (SVR) rely significantly on the kernel function, an essential instrument that facilitates linear solutions to nonlinear issues by expanding the input variables to a higher dimensional space. As opposed to SVM, which optimizes class margins, this approach utilizes a hyperplane to determine the majority of data points fall within a designated distance (ε-tube) so as to minimize error. While support vectors play a crucial role in establishing the hyperplane’s position, points that fail to adhere to this tolerance are penalized [60]. SVR is an effective predictive analytics tool that can extract valuable insights from large, complex datasets in numerous industries, including environmental modeling, healthcare, and finance, due to its methodology [61].

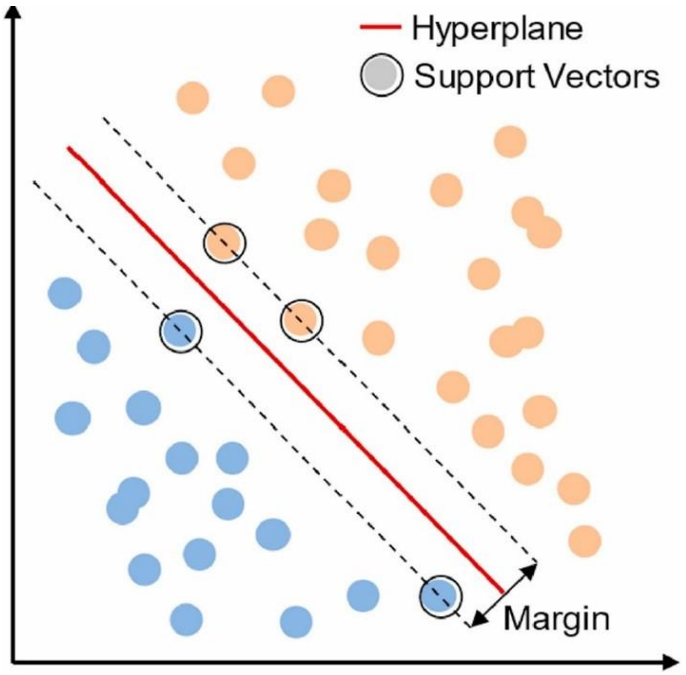


Figure 2.10: Graphical Representation of SVM Structure **[62]**

2.6.2.3 Gaussian Process Regression

Figure 2.11 illustrates how GPR, an exceptionally versatile supervised learning approach, can be applied to both classification tasks requiring discrete outputs and regression tasks requiring continuous outputs. Its practicality extends to a wide array of scenarios, encompassing the examination of datasets and incorporation into more intricate problem-solving processes [63]. The Gaussian process, which is a collection of random variables of any finite number that share a jointly consistent Gaussian distribution, serves as the foundation for GPR. By employing this statistical framework, imperfect predictions of continuous values via GPR are possible [64]. GPR is an exceptional technique due to the fact that it is neither parametric nor linear. Connecting data elements in complex, high-dimensional spaces is facilitated by this. GPR is built upon the principles of Bayesian probability theory, which enables the smooth amalgamation of observed data and

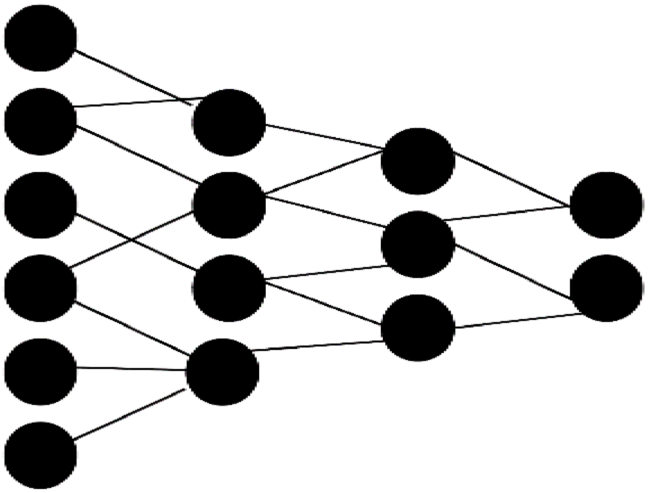


Figure 2.11: Representation of GPR Structure

pre-existing knowledge for the purpose of generating predictions. GPR exhibits strong compatibility with alternative Bayesian framework regression techniques, notably Kernel Ridge Regression (KRR) and radial basis function-based linear regression. This serves as a demonstration of the framework’s capability to adjust and endure in the face of diverse predictive modeling scenarios [65]. In the realm of neural networks, the Bayesian approach employs a prior distribution over weights that bears resemblance to a distribution over functions. A posterior is generated for prediction functions through the integration of a noise model with this initial assumption. Nevertheless, practical implementations frequently require approximations as a result of the complex characteristics inherent in the distribution of this function within neural networks [66]. The Bayesian framework of GPR oversees uncertainty and model complexity in a manner distinct from that of methods such as KRR. A comprehensive understanding of these differentiations is crucial for the selection of the most suitable tool for specific ML tasks, especially those that require precise interpolation and quantification of uncertainty [67].

2.6.2.4 Ensemble Learning Tree

ELT methods are characterized by the integration of a substantial number of learners in a particular way so as to improve the accuracy of predictions [68]. Tree-based machine learning models are distinguished from earlier, obfuscator-oriented algorithms by virtue of their simplicity and transparency [69]. These models demonstrate exceptional efficacy in addressing linear as well as nonlinear problems. The approach being evaluated is founded upon two fundamental model classifications: meta-learners and base learners. It is the responsibility of base learners to predict the posterior class probabilities of a given sample. On the other hand, the meta-learner determines the final class designation by combining these predictions as visualized in figure 2.12. The utilization of a dual-model approach facilitates the processing of more intricate data, leading to the generation of more accurate predictions. One distinguishing feature of this methodology is its comprehensive implementation of multi-objective optimization throughout the model development stage, with an emphasis on model selection and combination. By employing this approach, ensemble complexity can be optimized while accuracy is minimized, precision maintained, and interpretability ensured. An additional hill-climbing algorithm is incorporated into the ensemble method in order to generate a stable collection of diagnostic principles. By selecting rules and prioritizing them strategically, the aforementioned method guarantees that the ensemble will produce satisfactory outcomes and provide insightful observations. A significant advancement in the field of machine learning (ML) is this ensemble learning approach, which strategically employs base and meta-learners in addition to multi-objective optimization. It is particularly advantageous in the development of diagnostic instruments that possess a high degree of comprehensibility [70].

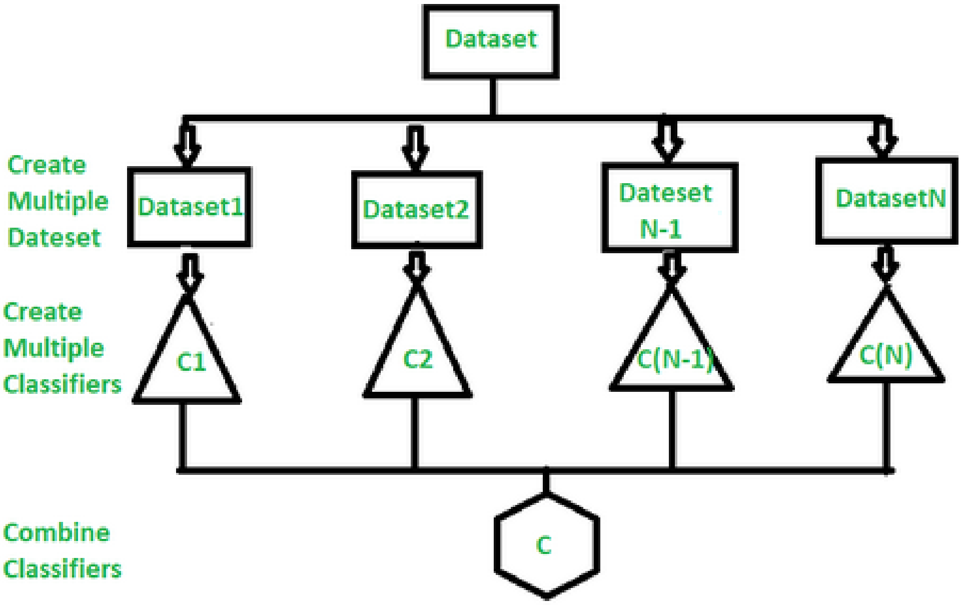


Figure 2.12: Representation of ELT Structure**[56]**

2.7 Overview of Optimization Algorithm

A comparative analysis was carried out to identify significant features and optimize the hyperparameters with the help of two distinct optimization algorithms. The Genetic algorithm comes first, followed by Particle Swarm Optimization.

2.7.1 Genetic Algorithm

The process of generating computational algorithms is a methodology that is influenced by natural evolution. It determines the optimal solution to a given problem from a set of potential solutions. This approach entails the conceptualization of each conceivable resolution as a “chromosome” and assess its quality by employing a specialized metric known as the “fitness function” [71]. GA utilizes this fitness function to quantify the effectiveness of each proposed solution. The algorithm evaluates these solutions iteratively and terminates execution when a sufficient solution is identified [72].

It fulfills the pre-established criteria. Should the intended solution remain unattainable, GA will generate alternative solution candidates via selection, crossover, and mutation; these processes constitute the subsequent generation of possibilities as a whole [73]. Throughout this iterative process, solutions that demonstrate outstanding performance are retained, while those that fail to meet the criteria for precision are discarded. As this process goes on, solutions that work well are kept, and the others are discarded. This way, over time, GA helps in finding ever-improving solutions to that problem. The key is to use these operations wisely to improve the solutions until the best one is found. Figure 2.13 visually describe the whole process effectively [74].

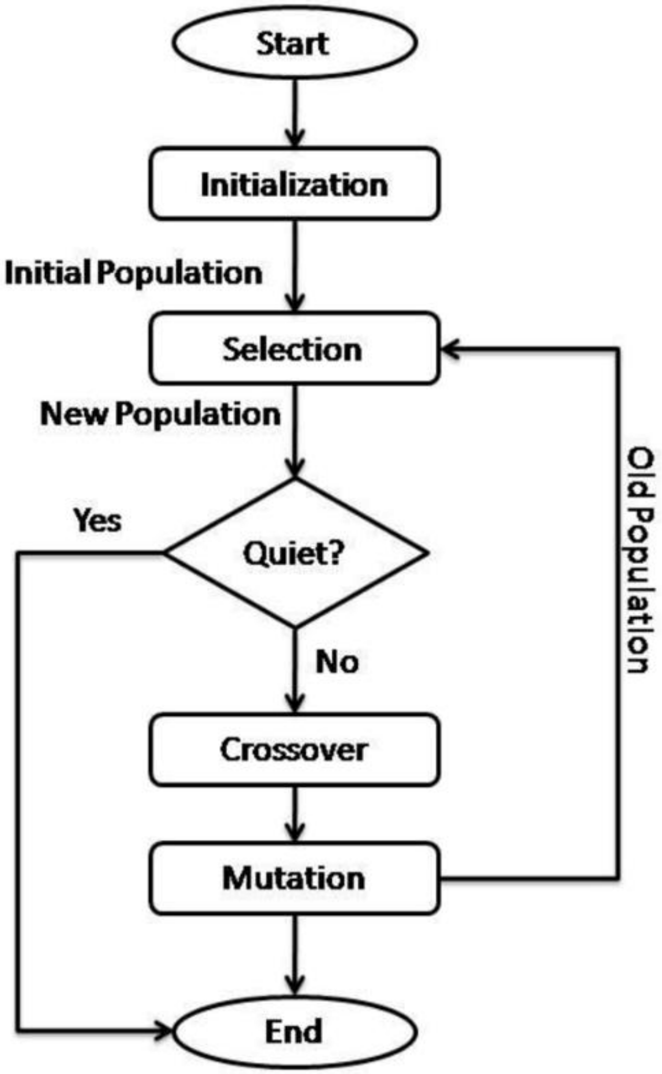


Figure 2.13: Flowchart of Genetic Algorithm **[75]**

2.7.2 Particle Swarm Optimization

As illustrated in Figure 2.14, PSO is an advanced optimization technique distinguished by its stochastic, self-adaptive, and population-based characteristics [76]. In this methodology, particles are employed to represent potential solutions within a designated search space. An individual particle determines its function value and optimal position by considering both its current velocity and the optimal positions of its neighbors.

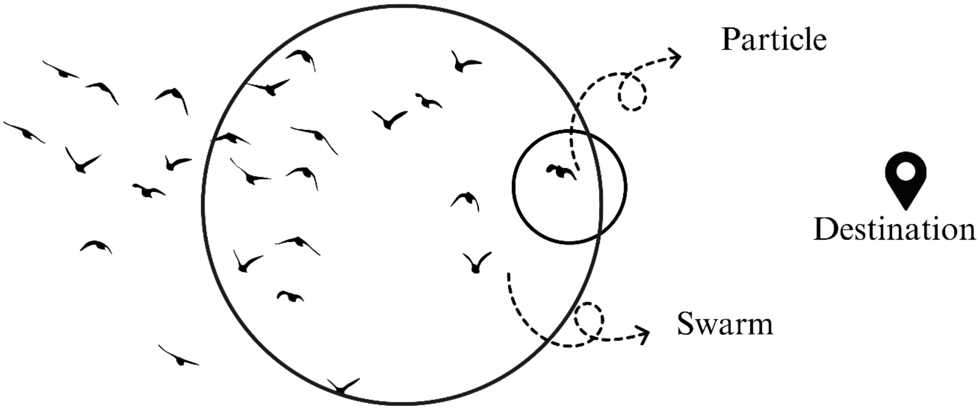


Figure 2.14: Nature-inspired Technique (PSO)

This evaluation provides guidance to the particles as they modify their velocities and positions, ensuring that they approach the most efficient solutions. Constraints regarding the whereabouts, velocities, and neighboring interactions of particles can be established through the implementation of periodic modifications [77]. PSO functions are based on the underlying principle of traversing extensive solution spaces with the objective of identifying optimal values for diverse system characteristics, while simultaneously considering cost constraints [78]. This methodology is extensively utilized in a variety of scientific disciplines to address problems that demand optimal resolutions while efficiently managing costs. Each solution in PSO is comparable to an aerial particle or bird that traverses the search space. These particles collectively ascertain the most advantageous positions. Particles are associated with position and velocity vectors in this multivariate space. Understanding the correlation between the position and motion pattern of a particle is critical, as this correlation establishes the boundaries that particles are capable of traversing. By employing this methodology, intricate motion patterns have been elucidated and our understanding of optimization challenges across numerous scientific fields has been enhanced. It is crucial to recognize that the versatile and cooperative characteristics of PSO among particles make it an exceptional tool for addressing practical issues, ensuring successful resolutions while accommodating the complexities inherent in numerous scientific inquiries. Figure 2.15 illustrates the comprehensive visual representation of the PSO algorithm's operation [79].

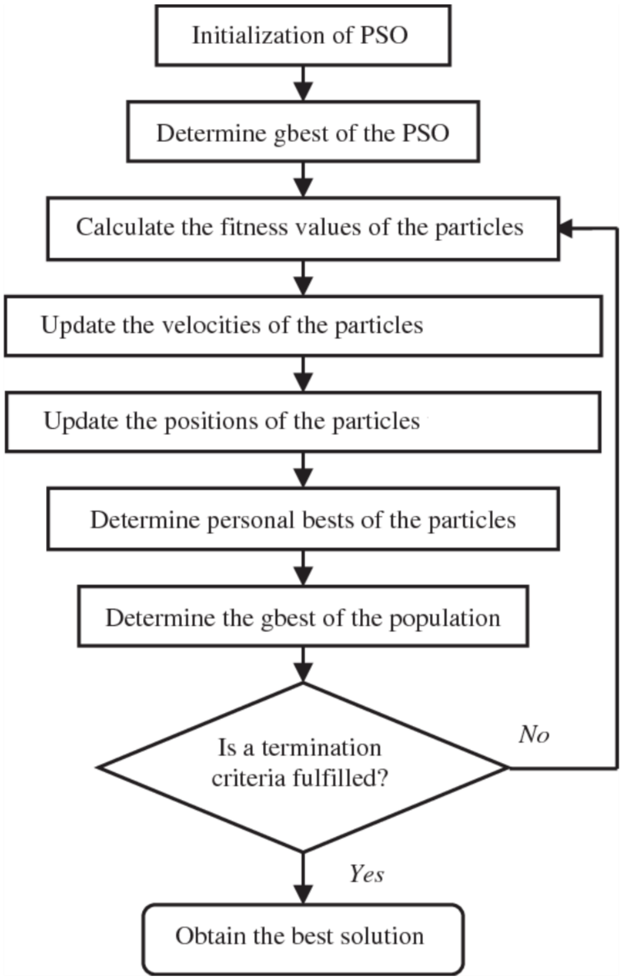

Initialization of PSO
Determine gbest of the PSO
Calculate the fitness values of the particles
Update the velocities of the particles
Update the positions of the particles
Determine personal bests of the particles
Determine the gbest of the population
Is a termination criteria fulfilled?
No
Yes
Obtain the best solution

Figure 2.15: Workflow of PSO **[79].**

CHAPTER 3: METHODOLOGY

3.1 Data Collection:

The load displacement data of different indents of our previously published work was used for the present ML based study [4]. The rationale for choosing this specific article stems from its reliability and authenticity, which are attributed to its use of real-time analysis. Four indent's data were utilized I1, I2, I3, I4 and the I1 indent was fully loaded, and its data points are 544 whereas I2, I3, I4 were stopped immediately after GB pop-in, and their data points were 380, 356, 363 respectively as depicted in Table 3.1.

Table 3.1: Data Collection Using Published Paper

Indents	Total data Points
I_1	544
I_2	380
I_3	356
I_4	363

These data points were used for simulation purposes, and these were obtained through nanoindentation testing. I1, I2, I3 and I4 are also named as the first set of indentation as depicted in figure 3.1 and their Load Displacement curves are shown in figure 3.2.

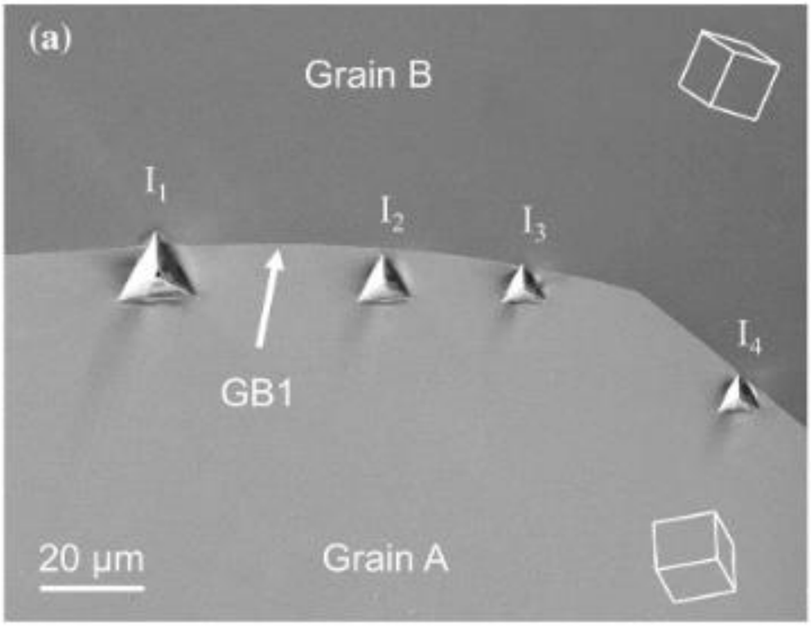


Figure 3.1: Berkovich indentations performed near GB in tungsten **[4].**

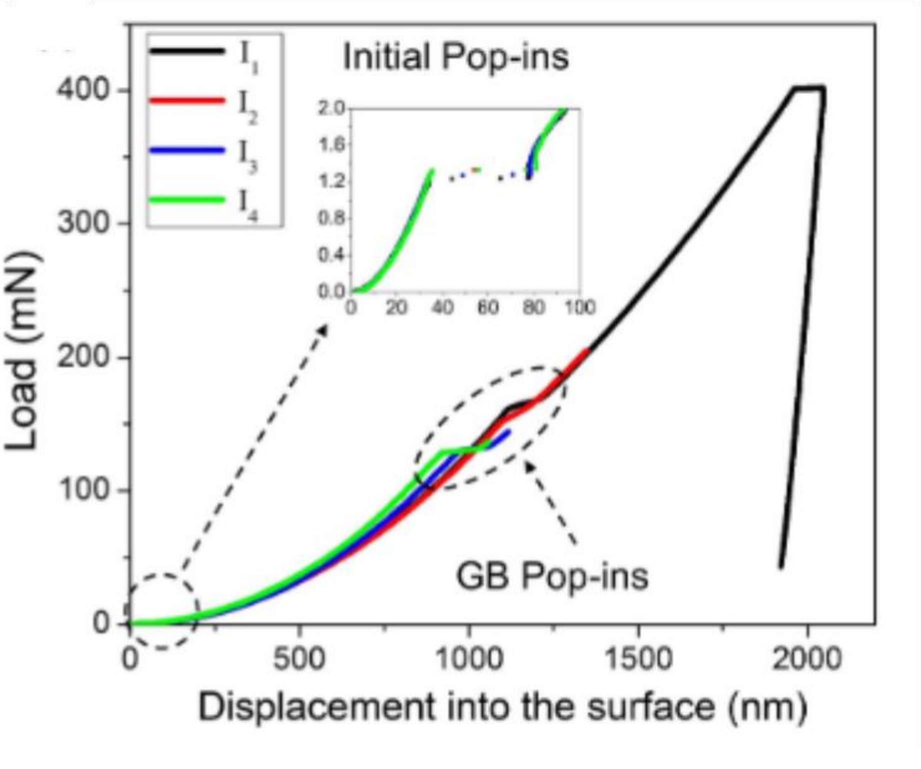


Figure 3.2: Load Displacement curves of indent I1 to I4 **[4].**

The dislocation structures and hardening Figure 3.3 during grain boundary (GB) pop-ins were studied using electron channeling contrast imaging (ECCI). For studying of dislocations under the surface, sequential polishing was done. The second set indentations I5, I6, I7 were also used to check the precision of prepared models and GPR model also showed the best result as compared to the other three models.

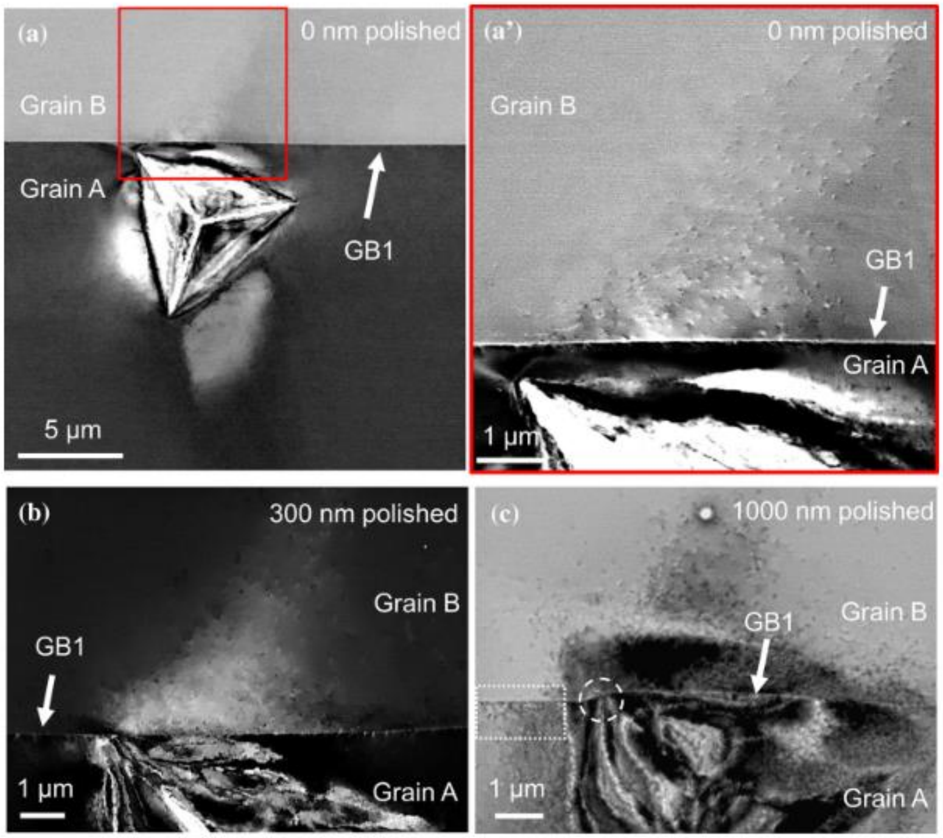
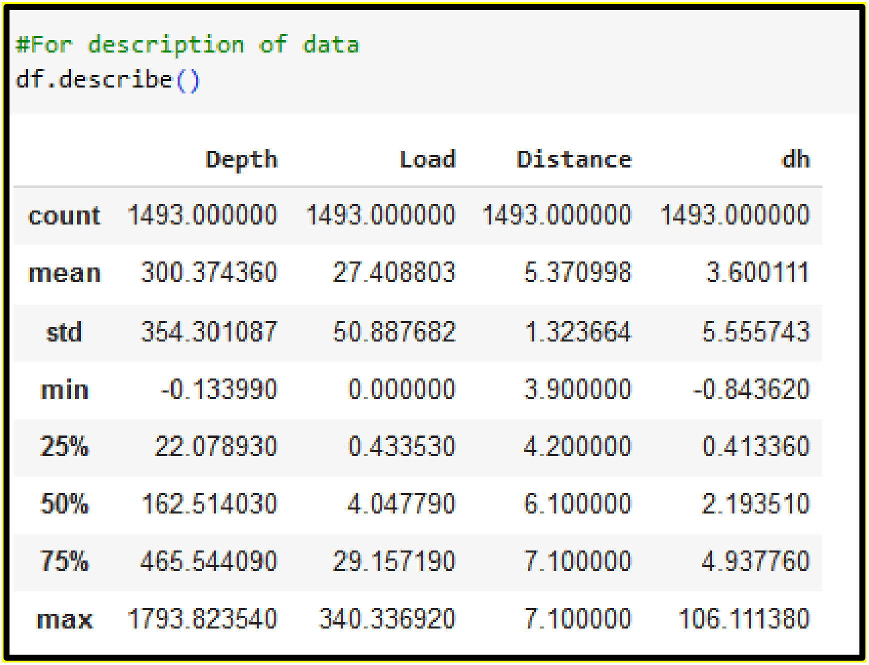


Figure 3.3: ECCI images of indent I4 at the surface and various polishing depths **[4].**

These data points were categorized using Python by importing the data file in CSV format, as seen in Figure 3.4. A variety of techniques were used on the given data to get data insights in the form of box plots, pair plots, and Pearson correlation.



```
#For description of data
df.describe()
```

```
             Depth         Load     Distance           dh
count  1493.000000  1493.000000  1493.000000  1493.000000
mean    300.374360    27.408803     5.370998     3.600111
std     354.301087    50.887682     1.323664     5.555743
min      -0.133990     0.000000     3.900000    -0.843620
25%      22.078930     0.433530     4.200000     0.413360
50%     162.514030     4.047790     6.100000     2.193510
75%     465.544090    29.157190     7.100000     4.937760
max    1793.823540   340.336920     7.100000   106.111380
```

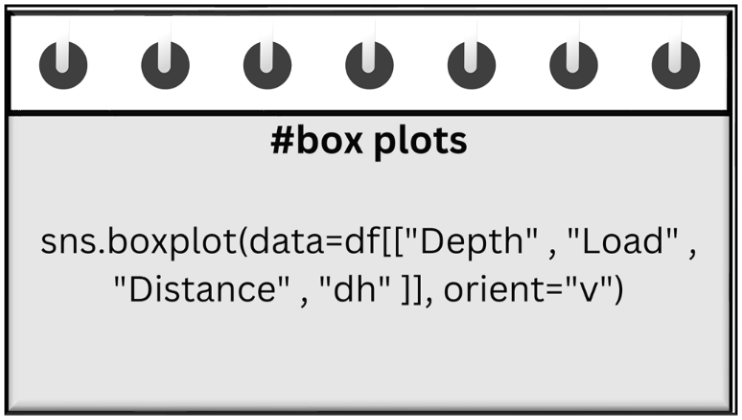
Figure 3.4: Data Collection Using Google Collaboratory (Python Programming Language)

3.1.1 Overview of Python Plotting

Python charts have a major impact on data exploration and analysis because they can transform unstructured information into relevant tales. Python plots make it easier to see correlations, trends, and patterns in data by using visualization capabilities; they give a dynamic lens through which information is brought to life. This feature not only increases our understanding of complicated information, but also allows us to communicate results compellingly and understandably. Various plots, such as box plots, pair plots, and Pearson correlation, can be obtained from the Python language [80].

3.1.1.1 Box Plots

Box plots in Python, also known as box-and-whisker plots, are very effective graphical representations used to demonstrate critical statistical information about the distribution of a given dataset. Box plots provide essential insights on outliers, skewness, symmetry, variation, and central tendency. The following algorithm in figure 3.5 is required to attain the box plots based on provided data sets.

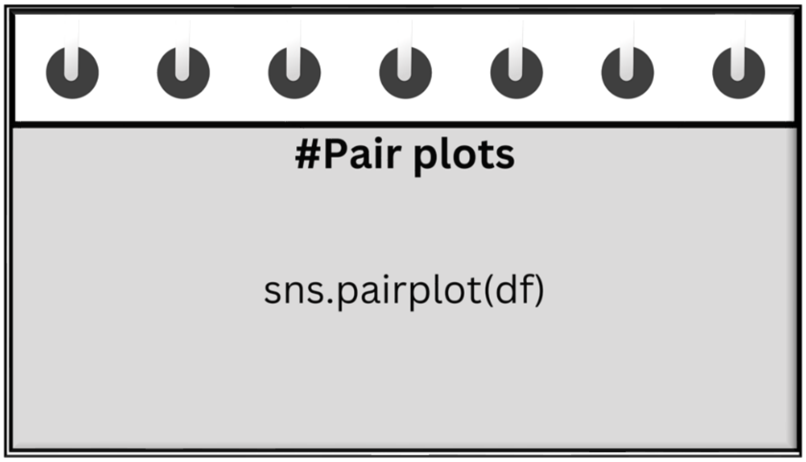


```
#box plots

sns.boxplot(data=df[["Depth" , "Load" ,
"Distance" , "dh" ]], orient="v")
```

Figure 3.5: Representation of Box Plots

3.1.1.2 Pair Plots:

Pair plots, which are often constructed in Python using tools like Seaborn, allow you to visually investigate the connections between variables in a dataset. The histograms depict the distribution of each variable independently, allowing us to better comprehend their distinct properties. Scatter plots depict bivariate connections, making it easy to discover correlations, clusters, and probable outliers. These graphics help you understand how variables interact and how their distributions are formed. They ease the examination of datasets and provide direction for future research. The process for generating these plots is illustrated in figure 3.6.



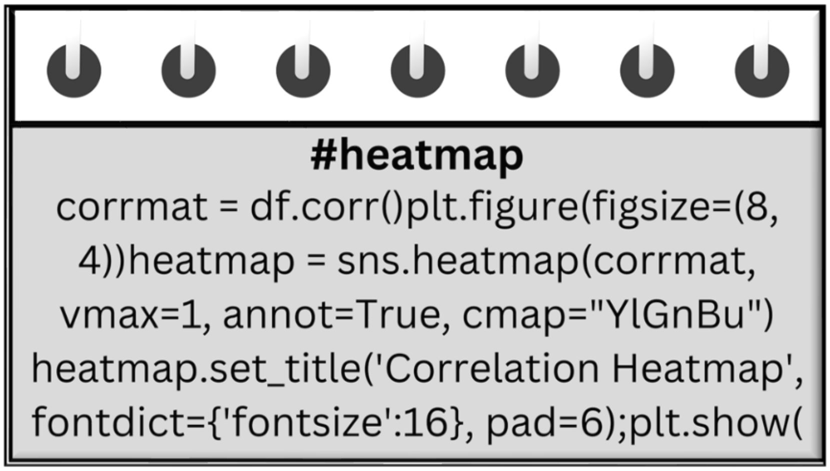
```
#Pair plots

sns.pairplot(df)
```

Figure 3.6: Representation of Pair Plots

3.1.1.3 Pearson Correlation Coefficient

The Pearson correlation coefficient, abbreviated as "r," determines the direction and strength of a linear connection between dependent and independent variables [81]. A correlation of +1 represents an ideal positive linear connection on a scale of -1 to +1, while a correlation of -1 suggests an ideal negative linear link, and a correlation of 0 indicates the lack of any linear association. The Pearson correlation technique, represented in figure 3.7, is used to analyze the level of linkage between variables. It provides useful insights into the correspondence between changes in one variable and another.

It is critical to understand that correlation does not always indicate a link, and the measure's validity assumes of a linear relationship between the variables under analysis; additionally, it may be associated with data outliers.



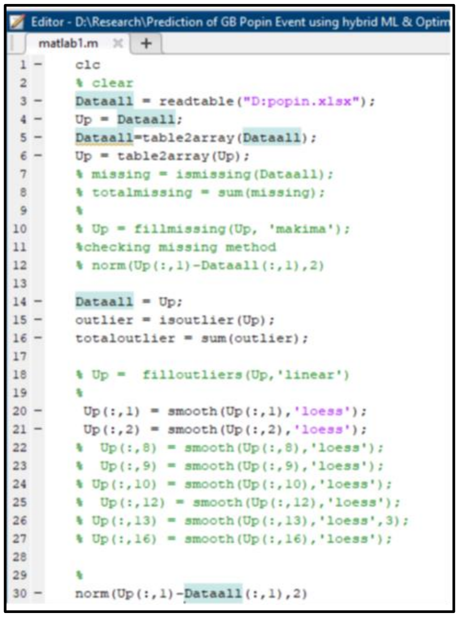
```
#heatmap
corrmat = df.corr()plt.figure(figsize=(8,
4))heatmap = sns.heatmap(corrmat,
vmax=1, annot=True, cmap="YlGnBu")
heatmap.set_title('Correlation Heatmap',
fontdict={'fontsize':16}, pad=6);plt.show(
```

Figure 3.7: Representation of Pearson correlation coefficient

3.1.2 Overview of MATLAB

3.1.2.1 Pre-Processing Algorithm

Figure 3.8 depicts algorithm loads data from a specified directory and then conducts several tasks, including detecting and eliminating outliers, smoothing the data, and normalizing it.

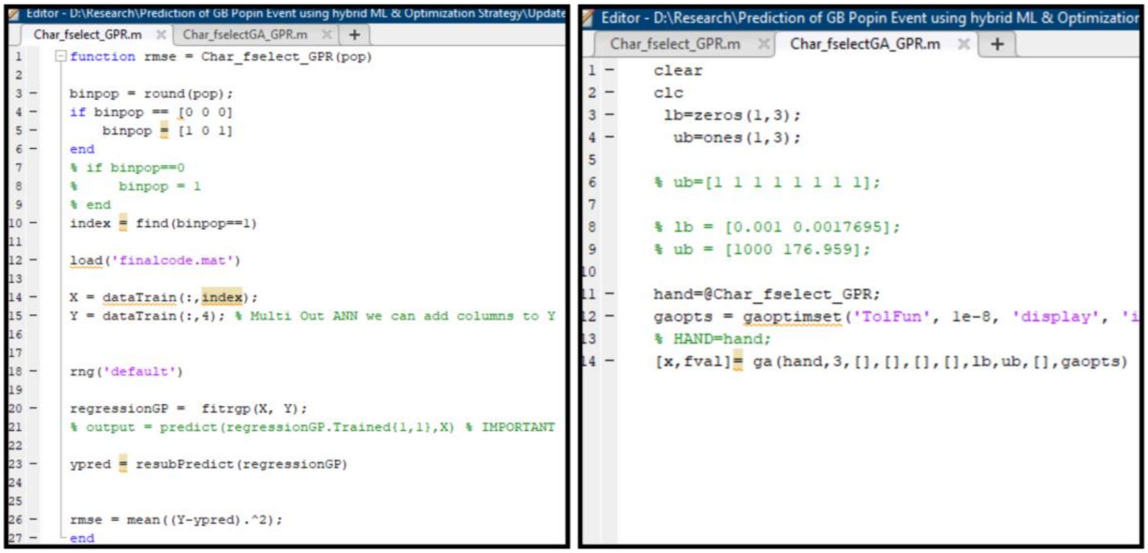

Editor - D:\Research\Prediction of GB Popin Event using hybrid ML & Optim
matlab1.m
+

```
clc
% clear
Dataall = readtable("D:popin.xlsx");
Up = Dataall;
Dataall=table2array(Dataall);
Up = table2array(Up);
% missing = ismissing(Dataall);
% totalmissing = sum(missing);
%
% Up = fillmissing(Up, 'makima');
%checking missing method
% norm(Up(:,1)-Dataall(:,1),2)

Dataall = Up;
outlier = isoutlier(Up);
totaloutlier = sum(outlier);

% Up =  filloutliers(Up,'linear')
%
 Up(:,1) = smooth(Up(:,1),'loess');
 Up(:,2) = smooth(Up(:,2),'loess');
%  Up(:,8) = smooth(Up(:,8),'loess');
%  Up(:,9) = smooth(Up(:,9),'loess');
% Up(:,10) = smooth(Up(:,10),'loess');
%  Up(:,12) = smooth(Up(:,12),'loess');
% Up(:,13) = smooth(Up(:,13),'loess',3);
% Up(:,16) = smooth(Up(:,16),'loess');


%
norm(Up(:,1)-Dataall(:,1),2)
```

Figure 3.8: Representation of Pre-Processing Algorithm.

3.1.2.2 Features Selection using GA and PSO

Feature selection is a method used to choose the most ideally suited collection of features for developing an effective recognition model [82]. The use of genetic algorithm (GA) and particle swarm optimization (PSO) in Feature Selection is significant. Figure 3.9 depicts the algorithm used for feature selection for GA-GPR.


Editor - D:\Research\Prediction of GB Popin Event using hybrid ML & Optimization Strategy\Update
Char_fselect_GPR.m
Char_fselectGA_GPR.m
+
Editor - D:\Research\Prediction of GB Popin Event using hybrid ML & Optimizatior
Char_fselect_GPR.m
Char_fselectGA_GPR.m
+

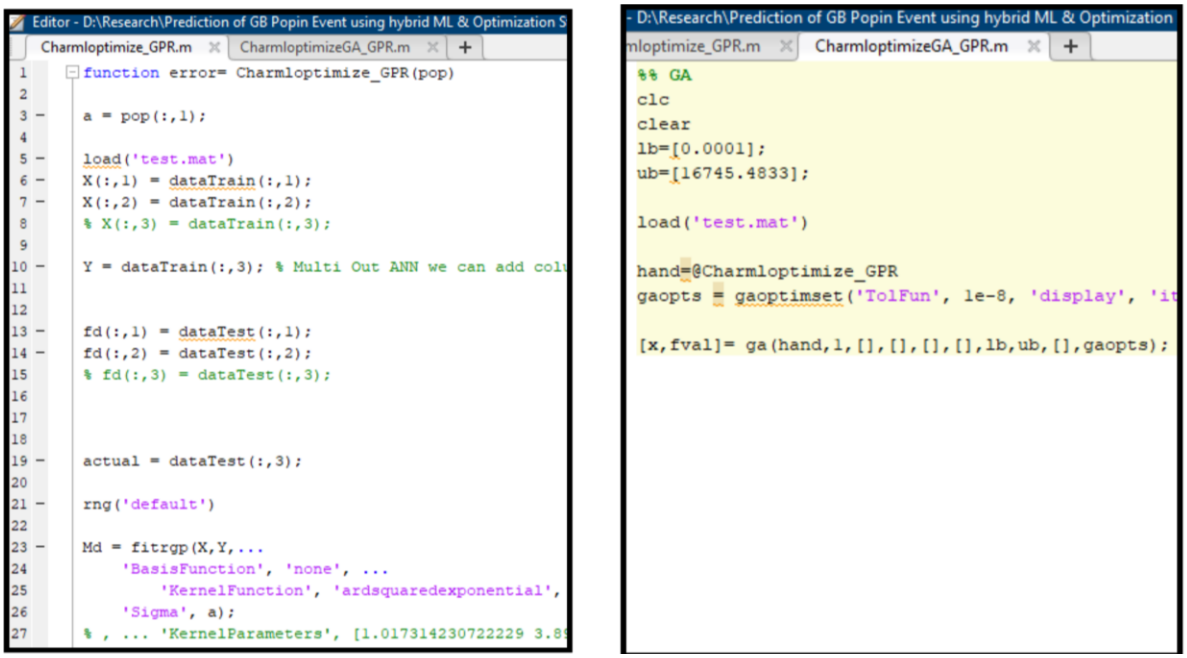
```
function rmse = Char_fselect_GPR(pop)

binpop = round(pop);
if binpop == [0 0 0]
    binpop = [1 0 1]
end
% if binpop==0
%     binpop = 1
% end
index = find(binpop==1)

load('finalcode.mat')

X = dataTrain(:,index);
Y = dataTrain(:,4); % Multi Out ANN we can add columns to Y


rng('default')

regressionGP =  fitrgp(X, Y);
% output = predict(regressionGP.Trained{1,1},X) % IMPORTANT

ypred = resubPredict(regressionGP)


rmse = mean((Y-ypred).^2);
end
```

```
clear
clc
 lb=zeros(1,3);
  ub=ones(1,3);

% ub=[1 1 1 1 1 1 1 1];

% lb = [0.001 0.0017695];
% ub = [1000 176.959];

hand=@Char_fselect_GPR;
gaopts = gaoptimset('TolFun', 1e-8, 'display', 'i
% HAND=hand;
[x,fval]= ga(hand,3,[],[],[],[],lb,ub,[],gaopts)
```

Figure 3.9: Representation of Feature Selection in GA-GPR.

3.1.2.3 Hyperparameters tunning using GA and PSO

The optimization or tuning of hyper-parameters involves selecting a suitable range of hyper-parameters for a learning algorithm. A hyper-parameter is a parameter that regulates the learning process by controlling its value[83]. Figure 3.10 depicts the algorithm used for Hyperparameters tunning for GA-GPR.


Editor - D:\Research\Prediction of GB Popin Event using hybrid ML & Optimization S
Charmloptimize_GPR.m
CharmloptimizeGA_GPR.m
- D:\Research\Prediction of GB Popin Event using hybrid ML & Optimization
mloptimize_GPR.m
CharmloptimizeGA_GPR.m

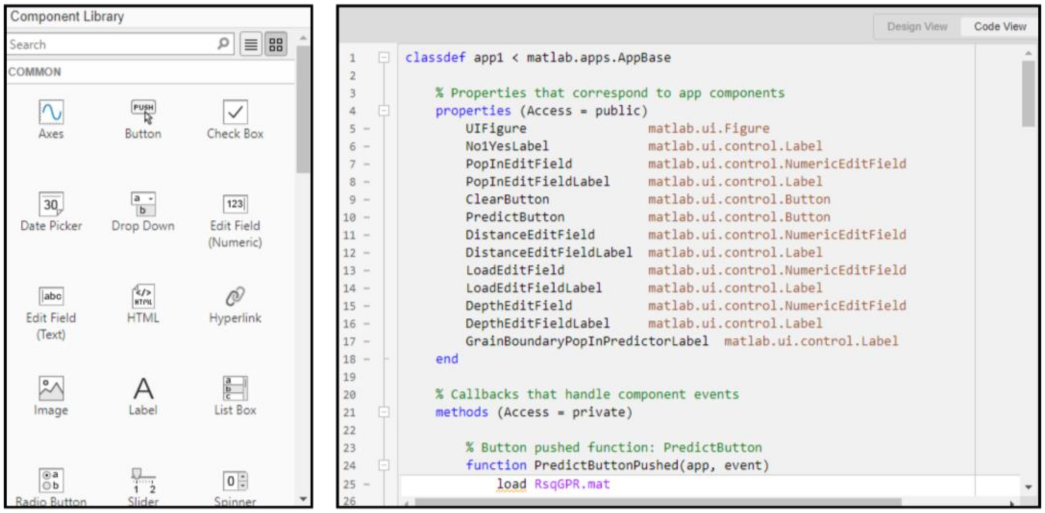
```
function error= Charmloptimize_GPR(pop)

a = pop(:,1);

load('test.mat')
X(:,1) = dataTrain(:,1);
X(:,2) = dataTrain(:,2);
% X(:,3) = dataTrain(:,3);

Y = dataTrain(:,3); % Multi Out ANN we can add col


fd(:,1) = dataTest(:,1);
fd(:,2) = dataTest(:,2);
% fd(:,3) = dataTest(:,3);



actual = dataTest(:,3);

rng('default')

Md = fitrgp(X,Y,...
    'BasisFunction', 'none', ...
        'KernelFunction', 'ardsquaredexponential',
    'Sigma', a);
% , ... 'KernelParameters', [1.01731423072229 3.8
```

```
%% GA
clc
clear
lb=[0.0001];
ub=[16745.4833];

load('test.mat')

hand=@Charmloptimize_GPR
gaopts = gaoptimset('TolFun', 1e-8, 'display', 'it

[x,fval]= ga(hand,1,[],[],[],[],lb,ub,[],gaopts);
```

Figure 3.10: Representation of Feature Selection in GA-GPR.

3.1.2.4 Graphical User Interface (GUI)

A graphical user interface (GUI) is a user interface that enables people to interact with electronic devices using visual representations as indicators [84]. Graphical user interfaces have emerged as the prevailing approach to user-centered design in software application programming. They enable users to operate computers and electronic devices intuitively by directly manipulating graphical icons, including buttons, scroll bars, windows, tabs, menus, cursors, and the mouse pointing device. Modern graphical user interfaces often include touchscreen and voice-command functionalities[85]. Figure 3.11 represents the component library for developing GUI in MATLAB and also the algorithm used for developing the GUI.



```
classdef app1 < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure                 matlab.ui.Figure
        No1YesLabel              matlab.ui.control.Label
        PopInEditField           matlab.ui.control.NumericEditField
        PopInEditFieldLabel      matlab.ui.control.Label
        ClearButton              matlab.ui.control.Button
        PredictButton            matlab.ui.control.Button
        DistanceEditField        matlab.ui.control.NumericEditField
        DistanceEditFieldLabel   matlab.ui.control.Label
        LoadEditField            matlab.ui.control.NumericEditField
        LoadEditFieldLabel       matlab.ui.control.Label
        DepthEditField           matlab.ui.control.NumericEditField
        DepthEditFieldLabel      matlab.ui.control.Label
        GrainBoundaryPopInPredictorLabel  matlab.ui.control.Label
    end

    % Callbacks that handle component events
    methods (Access = private)

        % Button pushed function: PredictButton
        function PredictButtonPushed(app, event)
            load RsqGPR.mat
```

Figure 3.11: Representation of Feature Selection in GA-GPR.

Different ML models were created by combining (1) GA and (2) PSO for Pop-in Prediction. Operational characteristics include load, depth, and Distance. Prior to being imported from an Excel sheet into MATLAB R2021, the data was properly categorized in Python and the relevance of each characteristic with the independent variables was determined. ML models were used with optimization methods such as GA and PSO to anticipate pop-ins and select features. In addition, a machine learning model was developed utilizing the data set to forecast the commencement of Pop-ins by using nanoindentation data under hypothetical settings. In this study, a GPR-GA-based model was selected over all other models to predict. Therefore, the workspace of the GA-based GPR model was used for building the GUI that fulfills the main objective of this study.

3.2 Data Pre-processing

After successfully classifying and visualizing the retrieved data using Python, it is assured that it is suitable for the next ML stages. To begin, all data points have been standardized to preserve logical values and avoid any attribute from having too much impact. To ensure forecast accuracy, care was taken to exclude any unusual data points. The dataset was then divided into two parts: 80% for training our models and 20% for accuracy testing, to ensure fairness via random selection. Missing data was handled thoughtfully, which included filling up gaps or eliminating undesirable areas. ML models, including SVM, GPR, DT, and ELT, were trained and assessed using metrics like coefficient of determination (R^2) using specialized applications. We fine-tuned our models using advanced approaches such as GA and PSO, which significantly improved their prediction capabilities. Our forecasts were quite accurate, as shown by high R^2 values. This rigorous technique not only assured correctness, but also improved the overall performance of our models. Figure 3.12 depicts the complete procedure, from data collection to pop-in prediction.

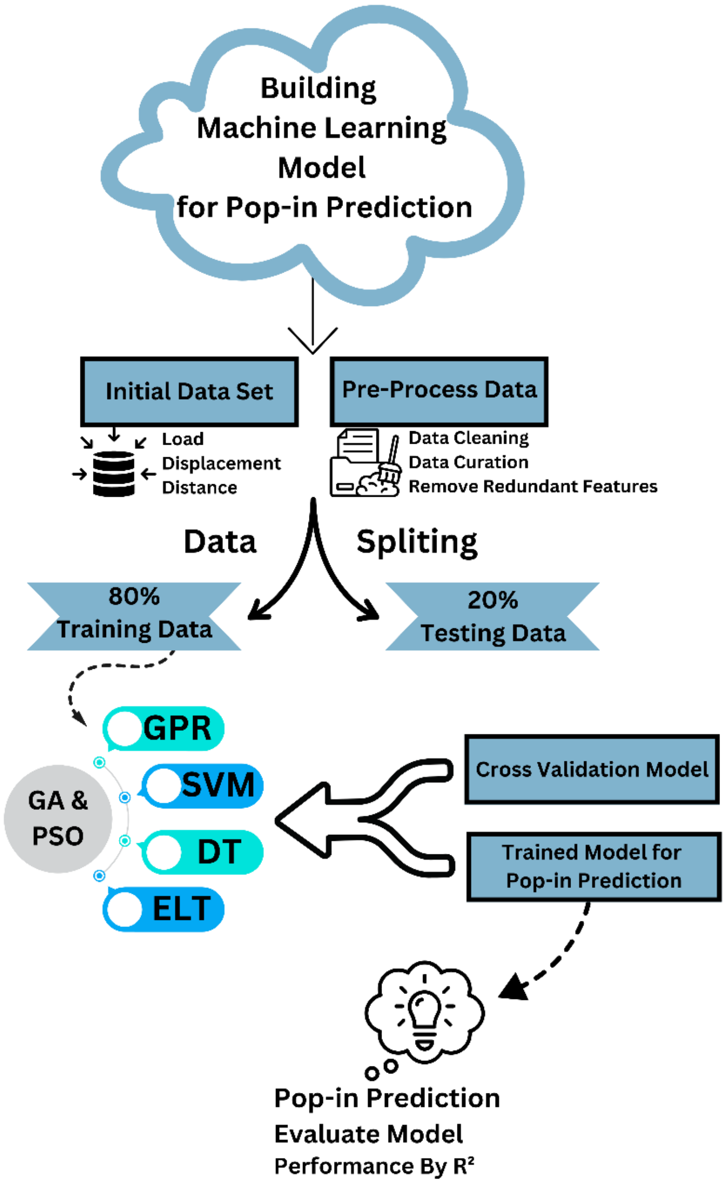


Figure 3.12: Machine Learning Workflow

CHAPTER 4: RESULTS AND DISCUSSION

4.1 Box plot, Pair plot, Correlation heatmap

A rigorous classification approach applying the Python programming language in the Google Colab environment is carried out for the categorization of a comprehensive exploratory data analysis employs a range of visualization approaches to get a thorough understanding of the dataset's complicated properties. Furthermore, box graphs, as seen in figure 4.1, were very useful for graphically describing the distribution of data across various categories or attributes. These plots helped identify median points, likely outliers, skewness, clusters, and a dispersion in the load, depth, distance, and dh data sets, improving our comprehension of the dataset's complexities.

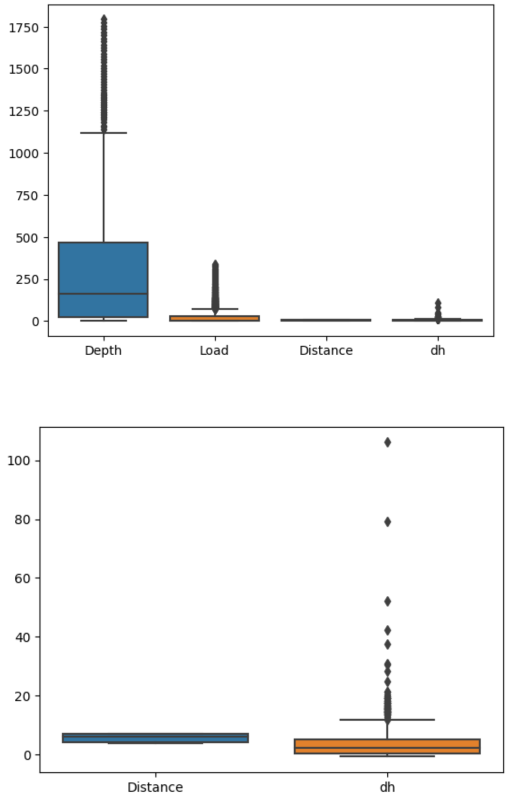


Figure 4.1: (a) Depth (µm), Load (mN), Distance, dh Box Plot representation (b) Distance, dh Box Plot

The use of pair graphs, as shown in figure 4.2, was essential to thorough analysis because plots provide a graphic representation of data dispersion, recognizable patterns, histograms for individual variables, and an overall understanding of the interconnections between numerous factors.

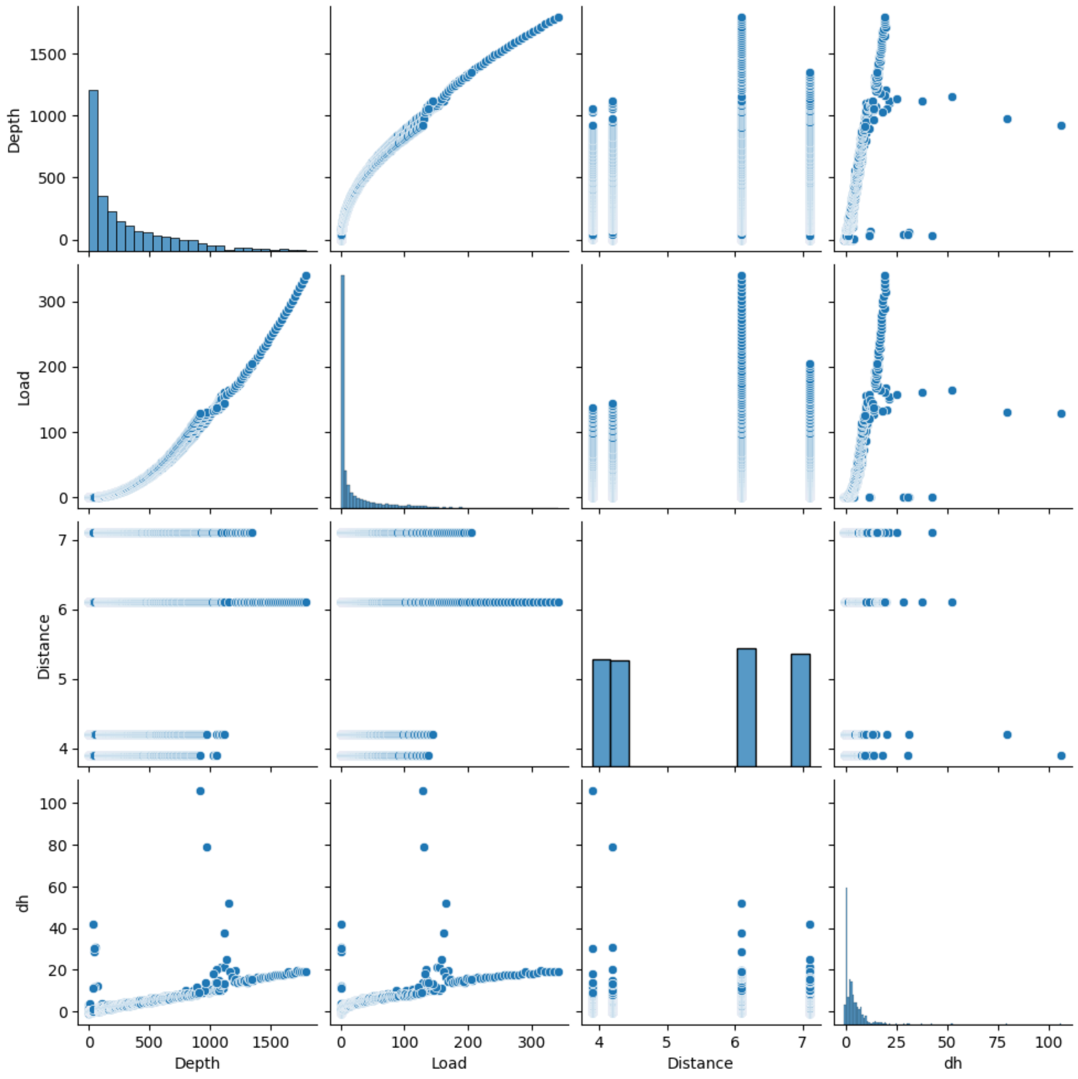


Figure 4.2: Depth (µm), Load (mN), Distance, dh Pair Plot representation

It is observed that the component's depth and load have a strong relationship with each other, and that depth and distance also have a significant relationship with each other. By carefully examining these diagrams, we may be able to identify potential connections, clusters, and correlations among the data points. The linear correlations between the variables in the data were then determined using Pearson correlation analysis, which provides the correlation coefficient and important details about the connections between variables such as load, depth, and distance, dh measures the strength and direction of linear correlations. It identifies significant connections through this technique, which may be used to direct future modeling or research efforts. It is observed that the component's depth and load have a strong relationship with each other, and depth and distance also have significant relationship with each other illustrated in figure 4.3.

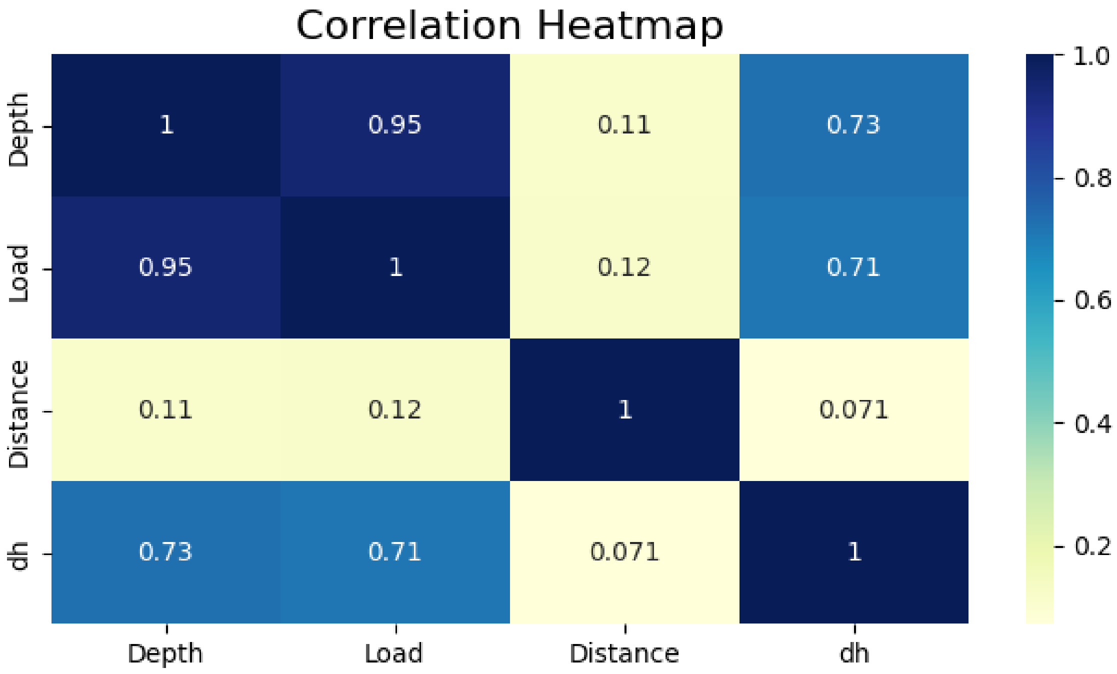


Figure 4.3: Depth (µm), Load (mN), Distance, dh Pearson Correlation representation

Sorting the data into groups, examining it using box plots and pair plots, and ultimately taking a close look at the associations between variables using the Pearson correlation coefficient were all steps in the data analysis process in Google Colab with Python. Together, these initiatives sought to improve our comprehension of the dataset's properties and open the possibility for improved decision-making in tasks using data in the future.

4.2 Performance Criterion of ML models

One of the most important measures used to evaluate the performance of different models, particularly in regression tasks, is the coefficient of determination (R^2). R^2 indicates how much of the variability in the output variable can be accounted for by the independent variables. A greater correlation between the variables and a more precise model fit are indicated by a higher R^2 value [86]. The R^2 measure is often used to assess how well a model's characteristics account for data variability. It is common practice to use metric since a model with a high R^2 value strikes a balance between explaining variation and producing accurate predictions. However, choosing between these possibilities should be based on the specific goals of the study, considering whether to emphasize the model's sufficiency or the accuracy of the predictions, or possibly both, depending on the situation [87]. Four machine learning models, such as SVM, GPR, DT, and ELT, were used in this investigation. MATLAB R2021b's regression toolbox was used to train these models. The development of prediction skills for detecting Pop-ins was the aim of the training. To assess the prediction precision of these machine learning models, the graphs that show the relationship between actual and observed Pop-in events were examined. The ML models used a mix of hyperparameter optimization and GA and PSO-based feature selection methods to precisely forecast the pop-ins. This was accomplished by using a training dataset, which made up 80% of the whole dataset, which included over 1157 data points, and a testing dataset, which made up the remaining 20% which included over 258 data points.

As shown in Table 4.1, the features obtained by using the GA and PSO were used to train the machine learning models. In addition, the process of feature selection and hyperparameter tuning resulted in the establishment of ideal hyperparameters that were used to train the machine learning models. The data points obtained from the location of horizontal straight line along the line $x = y$ for each of the Pop-ins, indicating a high level of prediction accuracy. However, it is important to recognize that machine learning models often show greater predictive accuracy on training datasets than on testing datasets.

Table 4.1: GA and PSO parameters

Algorithm Parameters			
Genetic Algorithm	**Value**	**Particle Swarm Optimization**	**Value**
Number of generations	100	Max. Iterations	100
Crossover	Scattered	Max. Velocity	-1 to 1
Crossover Probability	0.8	Inertia weight	0.4 to 0.9
Elite Count	3.95	Cognitive Coefficient	2.0
Population Size	50	Swarm Size	25
Population Type	Bitstring	Social Coefficient	2.0
Mutation	Uniform		
Mutation Probability	0.1		

4.3 Features Selection

In certain cases, while building and training a model, the number of input parameters might be large. However, not all these characteristics contribute significantly to the result. In such instances, designing a user interface with many input parameters is impracticable and overwhelming. To solve this issue, feature selection emerges as a critical technique for finding and prioritizing the input characteristics that really impact output predictions. GA and PSO are especially helpful in this respect, since they help to remove duplicate or less impacting characteristics from the input dataset. Essentially, the GA examines the relevance of each input parameter and the effectiveness of it in delivering meaningful outcomes. For example, in our investigation, both GA and PSO consistently chose load depth and distance as critical input parameters for all models, as shown in Table 4.2. This conclusion was made based on the importance of these elements in properly anticipating the outcome. This type of approach is very useful when dealing with a large number of input parameters, as it improves model performance and allows for the design of a user-friendly interface.

Table 4.2: GA and PSO based Features Selection

GA and PSO based features selection	
Models	Selected Features
SVM	Load, Depth, Distance
GPR	Load, Depth, Distance
DT	Load, Depth, Distance
ELT	Load, Depth, Distance

4.4 GA and PSO based hyperparameters tuning.

Utilizing the regression model toolbox, tuning parameters for many machine learning models were found. The tuning process consisted of a 5-fold cross-validation using standardized data. Table 4.3 provides parameters on the chosen hyperparameters, their respective ranges, and the optimal values. Using the GA and PSO techniques, the hyperparameters for the DT, SVM, GPR, and ELT models were adjusted and optimized.

In this work, GA and PSO were used to optimize the SVM model with certain parameters. The box constraint of 0.0368, the kernel scale of 1.7066, the Gaussian kernel function, and the epsilon value of 0.4354 were the optimal parameters for GA. The box constraint of 973.7567, the kernel function of Gaussian, the kernel scale of 3.4755, and the epsilon value of 1.0007 were the settings used in PSO. The hyperparameters of the ensemble model were adjusted with the aid of PSO and GA.

Table 4.3: Parameter Ranges and Optimized Values in Selection Process.

Models	**Hyperparameters**	**Ranges**	**Selected Range**	
			GA	**PSO**
SVM	Box Constraint	0.001-1000	0.0368	973.7567
	Kernel Scale	0.001-1000	1.7066	3.4755
	Kernel Function	Gaussian, Linear, Quadratic, Cubic	Linear	Gaussian
	Epsilon	0.00311-311.415	0.4354	1.0007
GPR	Sigma	0.0001-58.5453	45.3625	53.9366

	Kernel Function	Non/iso-tropic Exponential, Non/iso-tropic Matern 3/2, Non/iso-tropic Matern 5/2 Non/iso-tropic Rational Quadratic, Non/iso-tropic Squared Exponential,		Squared Exponential
	Basic Function	Constant, Zero, Linear		
	Kernel Scale	1.794-1793.95	324.950	320.9508
DT	Leaf Size	1-597	67	50
ELT	No. of Learners	10-500	10	409.2146
	Learning Rates	0.001-1	0.1	0.7579
	Methods	Bag, LSBoost	Bag	LSBoost

4.5 Models' performance

The models (GPR, DT, SVM, and ELT) were chosen using a feature selection strategy based on GA and PSO approaches, and they all generated effective predictions using the input variables load versus displacement. Table 4.4 illustrates a comparison of these models, indicating that overall GPR exhibited the R^2 value of 0.9999, which is close to one and shows the best performance as compared to the other three models. For GA the GPR, ELT, SVM and DT R^2 values were found to be 0.9999, 0.9264, 0.9711 and 0.9811, whereas for PSO, GPR, ELT, SVM and DT were found to be 0.9999, 0.9976, 0.9611, 0.9682.

Table 4.4: Comparison of ML methods using PSO and GA

Models	**Training R^2**		**Testing R^2**	
	GA	**PSO**	**GA**	**PSO**
ELT	0.9607	0.9999	0.9264	0.9976
GPR	0.9858	0.9858	0.9999	0.9999
SVM	0.9867	0.9767	0.9711	0.9611
DT	0.9197	0.8456	0.9811	0.9682

4.6 Partial dependence plots

Partial Dependence Plots (PDPs) in the figure 4.4, are used in machine learning to show the relationship visually between predictor variables and the expected result [88]. Distinct non-linear correlations with the goal variable 'Y' are shown by the two PDPs that are provided, one for 'Load' and the other for ' Depth'. The 'Load' PDP has complicated effects on 'Y', with a steep early increase, a plateau, and a continuing decrease. On the other hand, the connection between 'Y' and the 'Depth' PDP is smoother. Y rises to a peak and then gently declines, indicating a different but equally significant effect on the result. These charts emphasize the complex dynamics at work in predictive models, which also emphasize how important it is to understand feature interactions to enhance and interpret models.

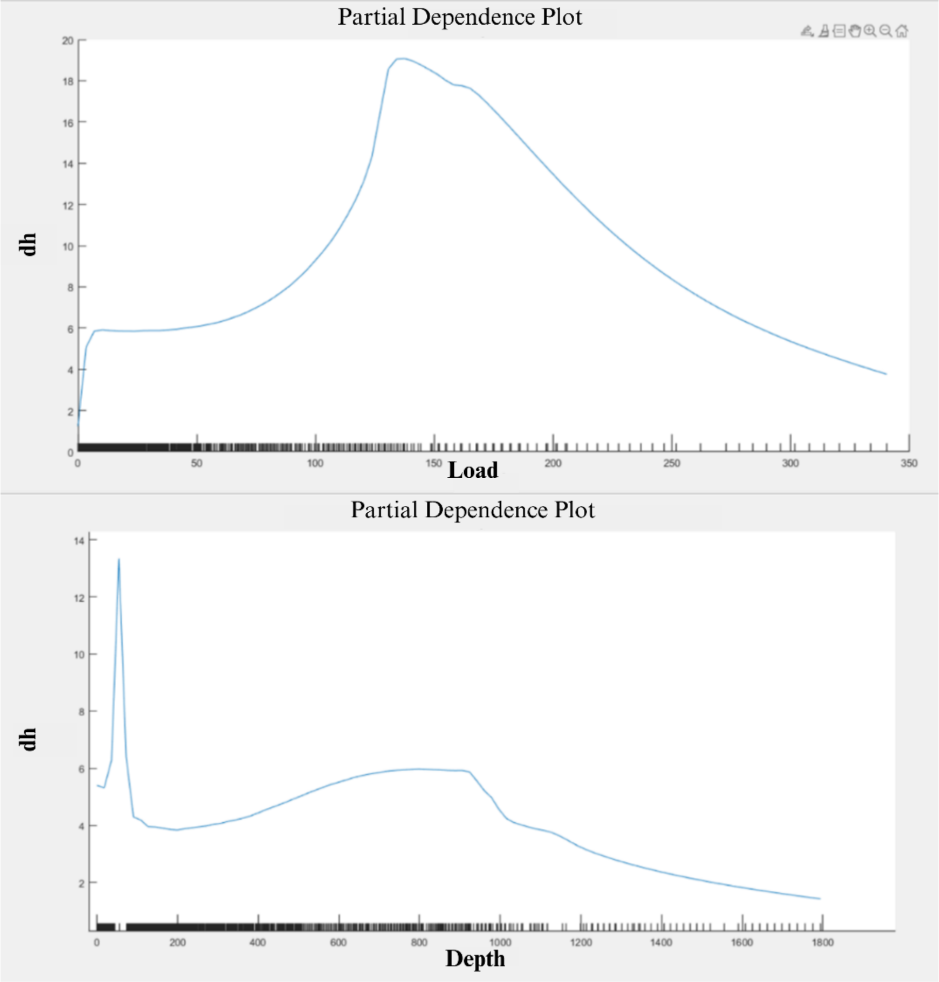


Figure 4.4: PDP’S Demonstrating the Influence of Inputs on Pop-in Prediction

4.7 Graphical representation between actual and predicted Pop-ins.

The graphical representations presented in below figure 4.5 offer a comprehensive overview of both the actual and predicted performance of Gaussian Process Regression (GPR) within the framework of Genetic Algorithm (GA) and Particle Swarm

Optimization (PSO). Additionally, a clear observation emerges, indicating minimal deviation between the predicted and actual outcomes of pop-in events. This visual analysis contributes to a significance understanding of the effectiveness of GPR in conjunction with GA and PSO methodologies, showcasing its capability to accurately forecast pop-in events while maintaining a close alignment with observed results.

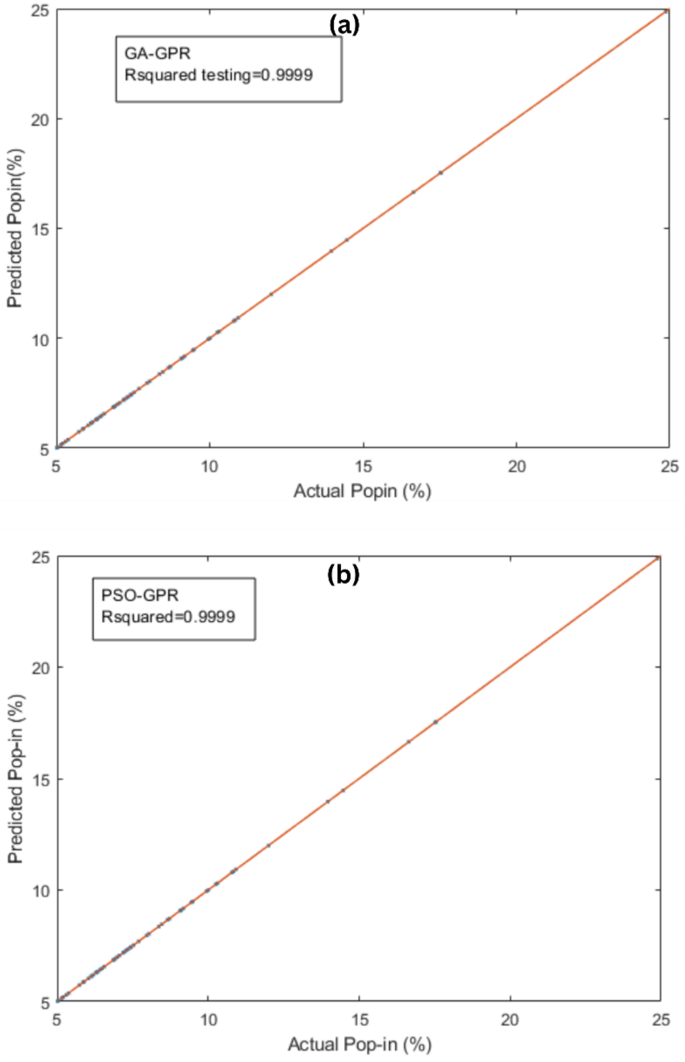


Figure 4.5: (a) GPR-GA predicted vs actual Pop-in (b) GPR-PSO (predicted Pop-in against actual Pop-in

The graphical representations presented in figure 4.6 below offer a comprehensive overview of both the actual and predicted performance of Ensembled Learning Tree (ELT) within the framework of Genetic Algorithm (GA) and Particle Swarm

Optimization (PSO). Additionally, a clear observation emerges, indicating maximal deviation between the predicted and actual outcomes of pop-in events. This visual analysis contributes to a significant understanding of the effectiveness of ELT in conjunction with GA and PSO methodologies.

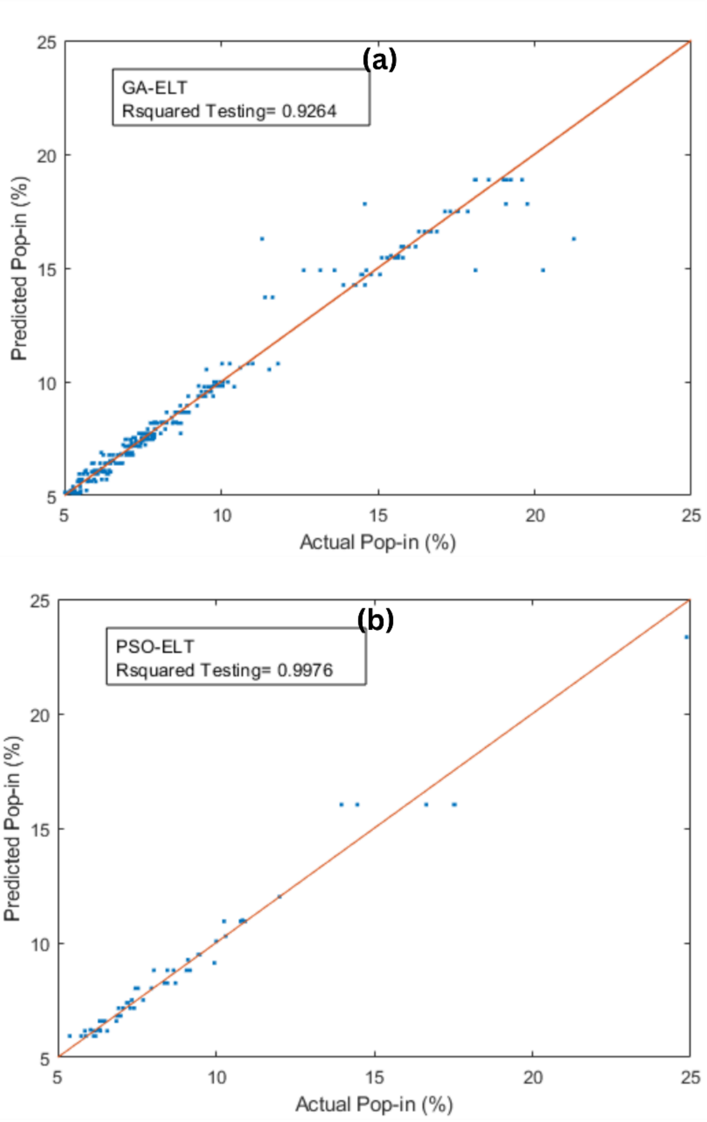
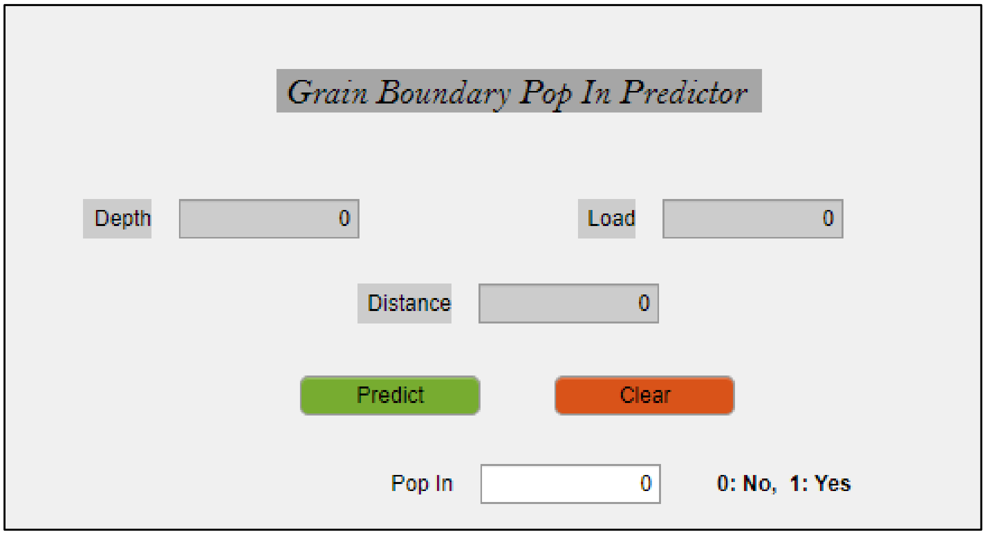


Figure 4.6: (a) ELT- GPR predicted vs actual Pop-in (b) PSO-ELT (predicted Pop-in against actual Pop-in

4.8 Graphical User Interface (GUI)

A Graphical User Interface (GUI) utilizes symbols, graphical icons, and user-friendly applications to facilitate user interaction with electronic equipment. The GUI developed in this research, allowed researchers to put input data of depth, load and distance. The model prediction function used by GUI in this study to predict Pop-in events is GA-GPR. MATLAB 2021b was used to develop GUI. Figure 4.7 depicts the picture of the GUI.



Grain Boundary Pop In Predictor

Depth 0

Load 0

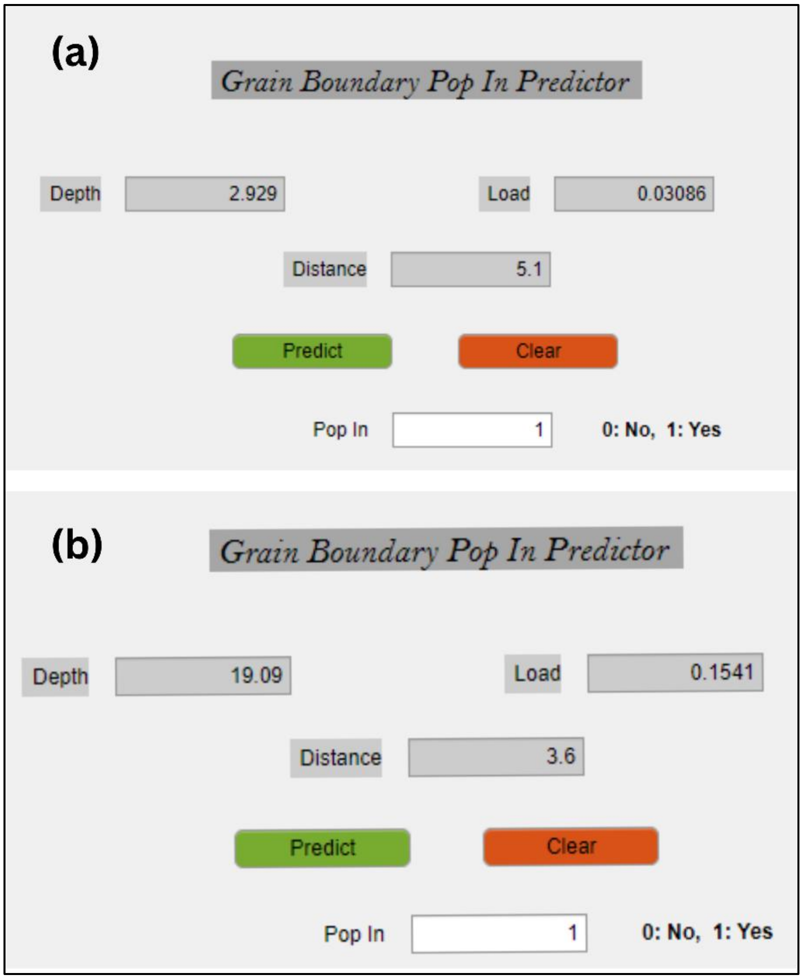
Distance 0

Pop In 0

0: No, 1: Yes

Figure 4.7: Graphical User Interface (GUI) to predict pop-in

In figure 4.8, the prediction of Pop-in events was checked by putting values of depth, load and distance of indents figure 4.8 (a) indent I5 and figure 32 (b) indent I6. The GUI was predicting the probability of pop-in events.


(a)
Grain Boundary Pop In Predictor
Predict
Clear

Depth 2.929
Load 0.03086
Distance 5.1
Pop In 1
0: No, 1: Yes

(b)
Grain Boundary Pop In Predictor
Predict
Clear

Depth 19.09
Load 0.1541
Distance 3.6
Pop In 1
0: No, 1: Yes

Figure 4.8: Graphical User Interface (GUI) to predict pop-in (a) indent I5, (b) indent I6.

CHAPTER 5: CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

In the present work, four machine learning models, Gaussian process regression (GPR), ensemble learning tree (ELT), support vector machine (SVM), and decision tree (DT), were developed to predict the Pop-in events. These models' criteria were assessed by Genetic algorithm (GA) and particle swarm optimization (PSO) in terms of the co-efficient of determination, also known as R^2. If it is close to 1, the ML model will show the best optimization. For GA, the GPR, ELT, SVM, and DT R^2 values were found to be 0.9999, 0.9264, 0.9711, and 0.9811, whereas for PSO, the GPR, ELT, SVM, and DT were found to be 0.9999, 0.9976, 0.9611, and 0.9682. It is evident from the aforementioned R^2 value that the GPR shows a value close to 1 as compared to the other three models, hence showing the best performance. Additionally, the partial dependency plot analysis showed that the proportion of dh in the output is significantly influenced by the optimum parameters selected using the GA-based approach. Lastly, a user-friendly graphical interface (GUI) is designed based on the GA-GPR model. The prediction of Pop-in events was checked by putting values of depth, load, and distance of indents, and the GUI predicted the GB Pop-in events, thus confirming the accuracy of the developed model.

5.2 Recommendations

For Future these recommendations can be used to predict Grain boundary Pop-ins events

- **Additional Experimental:** The current work is done using load, depth, dh. Model should be trained for G.B orientation and G.B character.
- **Uncertainty quantification:** Quantify the uncertainty associated with the model predictions. This can provide valuable insights into the limitations of the model and guide future research directions.
- **Feature importance analysis**: Investigate which features have the most significant impact on the model's predictions. This can help understand the underlying mechanisms of grain boundary pop-in events and identify critical material properties.

REFERENCES

[1] S. S. Murugan, "Mechanical properties of materials: definition, testing and application," *Int. J. Mod. Stud. Mech. Eng,* vol. 6, no. 2, pp. 28-38, 2020.

[2] W. Soboyejo, *Mechanical properties of engineered materials*. CRC press, 2002.

[3] T. Tański, A. Dobrzańska-Danikiewicz, K. Labisz, and W. Matysiak, "Long-term development perspectives of selected groups of engineering materials used in the automotive industry," *Archives of Metallurgy and Materials,* vol. 59, 2014.

[4] F. Javaid, Y. Xu, and K. Durst, "Local analysis on dislocation structure and hardening during grain boundary pop-ins in tungsten," *Journal of Materials Science,* vol. 55, no. 22, pp. 9597-9607, 2020.

[5] S. R. Kalidindi and S. J. Vachhani, "Mechanical characterization of grain boundaries using nanoindentation," *Current Opinion in Solid State and Materials Science,* vol. 18, no. 4, pp. 196-204, 2014.

[6] E. N. Hahn and M. A. Meyers, "Grain-size dependent mechanical behavior of nanocrystalline metals," *Materials Science and Engineering: A,* vol. 646, pp. 101-134, 2015.

[7] H. van swygenhoven and J. Weertman, "Deformation in nanocrystalline metals," *Materials Today - MATER TODAY,* vol. 9, pp. 24-31, 05/01 2006.

[8] T. Britton, D. Randman, and A. Wilkinson, "Nanoindentation study of slip transfer phenomenon at grain boundaries," *Journal of Materials Research,* vol. 24, no. 3, pp. 607-615, 2009.

[9] T. Bieler, P. Eisenlohr, C. Zhang, H. Phukan, and M. Crimp, "Grain boundaries and interfaces in slip transfer," *Current Opinion in Solid State and Materials Science,* vol. 18, no. 4, pp. 212-226, 2014.

[10] H. Pan, Y. He, and X. Zhang, "Interactions between dislocations and boundaries during deformation," *Materials,* vol. 14, no. 4, p. 1012, 2021.

[11] M. A. Meyers, A. Mishra, and D. J. Benson, "Mechanical properties of nanocrystalline materials," *Progress in materials science,* vol. 51, no. 4, pp. 427-556, 2006.

[12] E. Bayerschen, A. McBride, B. Reddy, and T. Böhlke, "Review on slip transmission criteria in experiments and crystal plasticity models," *Journal of materials science,* vol. 51, pp. 2243-2258, 2016.

[13] F. Javaid, E. Bruder, and K. Durst, "Indentation size effect and dislocation structure evolution in (001) oriented SrTiO3 Berkovich indentations: HR-EBSD and etch-pit analysis," *Acta Materialia,* vol. 139, pp. 1-10, 2017.

[14] A. Fareed *et al.*, "Constrained incipient phase transformation in Ni-Mn-Ga films: A small-scale design challenge," *Materials & Design,* vol. 233, p. 112259, 2023.

[15] F. Javaid, H. Pouriayevali, and K. Durst, "Dislocation–grain boundary interactions: Recent advances on the underlying mechanisms studied via nanoindentation testing," *Journal of Materials Research,* pp. 1-13, 2021.

[16] M. Wang and A. Ngan, "Indentation strain burst phenomenon induced by grain boundaries in niobium," *Journal of materials research,* vol. 19, pp. 2478-2486, 2004.

[17] S. Lu *et al.*, "Grain boundary effect on nanoindentation: A multiscale discrete dislocation dynamics model," *Journal of the Mechanics and Physics of Solids,* vol. 126, pp. 117-135, 2019.

[18] Y. I. Golovin, "Nanoindentation and mechanical properties of materials at submicro-and nanoscale levels: Recent results and achievements," *Physics of the Solid State,* vol. 63, pp. 1-41, 2021.

[19] A. Tiwari and S. Natarajan, "Applied Nanoindentation in Advanced Materials."

[20] A. C. Fischer-Cripps and A. C. Fischer-Cripps, "Scaling Relationships in Nanoindentation," *Nanoindentation,* pp. 119-123, 2011.

[21] S. Hainsworth, H. Chandler, and T. Page, "Analysis of nanoindentation load-displacement loading curves," *Journal of Materials Research,* vol. 11, pp. 1987-1995, 1996.

[22] A. C. Fischer-Cripps and A. C. Fischer-Cripps, "Nanoindentation test standards," *Nanoindentation,* pp. 159-177, 2004.

[23] H. Wang, L. Zhu, B. Xu, H. Wang, L. Zhu, and B. Xu, "Principle and methods of nanoindentation test," *Residual Stresses and Nanoindentation Testing of Films and Coatings,* pp. 21-36, 2018.

[24] A. C. Fischer-Cripps and D. Nicholson, "Nanoindentation. Mechanical engineering series," *Appl. Mech. Rev.,* vol. 57, no. 2, pp. B12-B12, 2004.

[25] T. Chudoba, P. Schwaller, R. Rabe, J.-M. Breguet, and J. Michler, "Comparison of nanoindentation results obtained with Berkovich and cube-corner indenters," *Philosophical Magazine,* vol. 86, no. 33-35, pp. 5265-5283, 2006.

[26] J. Marteau, M. Bigerelle, S. Bouvier, and A. Iost, "Reflection on the measurement and use of the topography of the indentation imprint," *Scanning: The Journal of Scanning Microscopies,* vol. 36, no. 1, pp. 115-126, 2014.

[27] Y. Gao, "Beer and diaper," ed: Tsinghua University Press, Beijing, 2008.

[28] X. Li and B. Bhushan, "A review of nanoindentation continuous stiffness measurement technique and its applications," *Materials characterization,* vol. 48, no. 1, pp. 11-36, 2002.

[29] G. Genta and K. A. S. Rojas, "Dynamic Indentation Testing: theory and practice."

[30] W. C. Oliver and G. M. Pharr, "An improved technique for determining hardness and elastic modulus using load and displacement sensing indentation experiments," *Journal of materials research,* vol. 7, no. 6, pp. 1564-1583, 1992.

[31] G. Pharr, W. C. Oliver, and F. Brotzen, "On the generality of the relationship among contact stiffness, contact area, and elastic modulus during indentation," *Journal of materials research,* vol. 7, pp. 613-617, 1992.

[32] E. Herbert, W. Oliver, and G. Pharr, "Nanoindentation and the dynamic characterization of viscoelastic solids," *Journal of physics D: applied physics,* vol. 41, no. 7, p. 074021, 2008.

[33] W. C. Oliver and G. M. Pharr, "Measurement of hardness and elastic modulus by instrumented indentation: Advances in understanding and refinements to methodology," *Journal of materials research,* vol. 19, no. 1, pp. 3-20, 2004.

[34] M. Tehrani and M. Al-Haik, "Magnetically enhanced mechanical and creep properties of a structural epoxy," *International Journal of Materials and Structural Integrity,* vol. 3, no. 2-3, pp. 147-160, 2009.

[35] M. L. Oyen and R. F. Cook, "A practical guide for analysis of nanoindentation data," *Journal of the mechanical behavior of biomedical materials,* vol. 2, no. 4, pp. 396-407, 2009.

[36] A. C. Fischer-Cripps, "Factors Affecting Nanoindentation Test Data," in *Nanoindentation*New York, NY: Springer New York, 2002, pp. 61-82.

[37] C. A. Schuh, "Nanoindentation studies of materials," *Materials today,* vol. 9, no. 5, pp. 32-40, 2006.

[38] H. Huang, C. Shi, H. Zhao, and L. Zhang, "Influence of friction on the residual morphology, the penetration load and the residual stress distribution of a Zr-based bulk metallic glass," *AIP Advances,* vol. 3, no. 4, 2013.

[39] Y. Choi, H.-S. Lee, and D. Kwon, "Analysis of sharp-tip-indentation load–depth curve for contact area determination taking into account pile-up and sink-in effects," *Journal of materials research,* vol. 19, no. 11, pp. 3307-3315, 2004.

[40] A. Elmustafa, "Pile-up/sink-in of rate-sensitive nanoindentation creeping solids," *Modelling and Simulation in Materials Science and Engineering,* vol. 15, no. 7, p. 823, 2007.

[41] J. Korteling, G. C. van de Boer-Visschedijk, R. A. Blankendaal, R. C. Boonekamp, and A. R. Eikelboom, "Human-versus artificial intelligence," *Frontiers in artificial intelligence,* vol. 4, p. 622364, 2021.

[42] S. Das, A. Dey, A. Pal, and N. Roy, "Applications of artificial intelligence in machine learning: review and prospect," *International Journal of Computer Applications,* vol. 115, no. 9, 2015.

[43] A. Smola, "Introduction to machine learning," ed, 2008.

[44] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh, and K. Shaalan, "Speech recognition using deep neural networks: A systematic review," *IEEE access,* vol. 7, pp. 19143-19165, 2019.

[45] E. S. Puchi-Cabrera, E. Rossi, G. Sansonetti, M. Sebastiani, and E. Bemporad, "Machine learning aided nanoindentation: A review of the current state and future perspectives," *Current Opinion in Solid State and Materials Science,* vol. 27, no. 4, p. 101091, 2023.

[46] A. Kaplan and M. Haenlein, "Siri, Siri, in my hand: Who’s the fairest in the land? On the interpretations, illustrations, and implications of artificial intelligence," *Business horizons,* vol. 62, no. 1, pp. 15-25, 2019.

[47] J. Wei *et al.*, "Machine learning in materials science," *InfoMat,* vol. 1, no. 3, pp. 338-358, 2019.

[48] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning,* vol. 20, pp. 273-297, 1995.

[49] Y. Baştanlar and M. Özuysal, "Introduction to machine learning," *miRNomics: MicroRNA biology and computational analysis,* pp. 105-128, 2014.

[50] A. B. Arrieta *et al.*, "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information fusion,* vol. 58, pp. 82-115, 2020.

[51] R.-C. Chen, C. Dewi, S.-W. Huang, and R. E. Caraka, "Selecting critical features for data classification based on machine learning methods," *Journal of Big Data,* vol. 7, no. 1, p. 52, 2020.

[52] W. Y. Loh, "Classification and regression trees," *Wiley interdisciplinary reviews: data mining and knowledge discovery,* vol. 1, no. 1, pp. 14-23, 2011.

[53] E. Pekel, "Estimation of soil moisture using decision tree regression," *Theoretical and Applied Climatology,* vol. 139, no. 3-4, pp. 1111-1119, 2020.

[54] D. Malerba, F. Esposito, M. Ceci, and A. Appice, "Top-down induction of model trees with regression and splitting nodes," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 26, no. 5, pp. 612-625, 2004.

[55] S. S. Rathore and S. Kumar, "A decision tree regression based approach for the number of software faults prediction," *ACM SIGSOFT Software Engineering Notes*, vol. 41, no. 1, pp. 1-6, 2016.

[56] D. Che, Q. Liu, K. Rasheed, and X. Tao, "Decision tree and ensemble learning algorithms with their applications in bioinformatics," *Software tools and algorithms for biological systems,* pp. 191-199, 2011.

[57] H. Drucker, C. J. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," *Advances in neural information processing systems,* vol. 9, 1996.

[58] Z. U. Haq, H. Ullah, M. N. A. Khan, S. R. Naqvi, A. Ahad, and N. A. S. Amin, "Comparative study of machine learning methods integrated with genetic algorithm and particle swarm optimization for bio-char yield prediction," *Bioresource Technology,* vol. 363, p. 128008, 2022.

[59] Y. Liu, L. Wang, and K. Gu, "A support vector regression (SVR)-based method for dynamic load identification using heterogeneous responses under interval uncertainties," *Applied Soft Computing,* vol. 110, p. 107599, 2021.

[60] N. Ancona, "Classification properties of support vector machines for regression," *Technical Report, RIIESI/CNR-Nr. 02/99,* 1999.

[61] A. Ben-Hur, C. S. Ong, S. Sonnenburg, B. Schölkopf, and G. Rätsch, "Support vector machines and kernels for computational biology," *PLoS computational biology,* vol. 4, no. 10, p. e1000173, 2008.

[62] B. Mahesh, "Machine learning algorithms-a review," *International Journal of Science and Research (IJSR).[Internet],* vol. 9, no. 1, pp. 381-386, 2020.

[63] C. K. Williams, "Prediction with Gaussian processes: From linear regression to linear prediction and beyond," in *Learning in graphical models*: Springer, 1998, pp. 599-621.

[64] M. Abe *et al.*, "Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics): Preface," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),* vol. 3960, pp. V-VI, 2006.

[65] A. Banerjee, D. B. Dunson, and S. T. Tokdar, "Efficient Gaussian process regression for large datasets," *Biometrika,* vol. 100, no. 1, pp. 75-89, 2013.

[66] K. Kim, D. Lee, and I. Essa, "Gaussian process regression flow for analysis of motion trajectories," in *2011 International Conference on Computer Vision*, 2011, pp. 1164-1171: IEEE.

[67] V. Nemani *et al.*, "Uncertainty quantification in machine learning for engineering design and health prognostics: A tutorial," *Mechanical Systems and Signal Processing,* vol. 205, p. 110796, 2023.

[68] T. G. Dietterich, "Ensemble learning," *The handbook of brain theory and neural networks,* vol. 2, no. 1, pp. 110-125, 2002.

[69] A. Blanco-Justicia and J. Domingo-Ferrer, "Machine learning explainability through comprehensible decision trees," in *Machine Learning and Knowledge Extraction: Third IFIP TC 5, TC 12, WG 8.4, WG 8.9, WG 12.9 International Cross-Domain Conference, CD-MAKE 2019, Canterbury, UK, August 26–29, 2019, Proceedings 3*, 2019, pp. 15-26: Springer.

[70] Z. U. Haq, H. Ullah, M. N. A. Khan, S. R. Naqvi, and M. Ahsan, "Hydrogen production optimization from sewage sludge supercritical gasification process using machine learning methods integrated with genetic algorithm," *Chemical Engineering Research and Design*, vol. 184, pp. 614-626, 2022.

[71] S. Mirjalili and S. Mirjalili, "Genetic algorithm," *Evolutionary Algorithms and Neural Networks: Theory and Applications,* pp. 43-55, 2019.

[72] A. Konak, D. W. Coit, and A. E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial," *Reliability engineering & system safety,* vol. 91, no. 9, pp. 992-1007, 2006.

[73] A. Lambora, K. Gupta, and K. Chopra, "Genetic algorithm-A literature review," in *2019 international conference on machine learning, big data, cloud and parallel computing (COMITCon)*, 2019, pp. 380-384: IEEE.

[74] J. Shapiro, "Genetic algorithms in machine learning," in *Advanced Course on Artificial Intelligence*: Springer, 1999, pp. 146-168.

[75] A. Dastanpour and R. M. S. Ibrahim, "Using Genetic Algorithm to Supporting Artificial Neural Network for Intrusion Detection System," 2014.

[76] K. R. Harrison, A. P. Engelbrecht, and B. M. Ombuki-Berman, "Self-adaptive particle swarm optimization: a review and analysis of convergence," *Swarm Intelligence*, vol. 12, pp. 187-226, 2018.

[77] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, 1995, vol. 4, pp. 1942-1948: IEEE.

[78] E. H. Houssein, A. G. Gad, K. Hussain, and P. N. Suganthan, "Major advances in particle swarm optimization: theory, analysis, and application," *Swarm and Evolutionary Computation,* vol. 63, p. 100868, 2021.

[79] C. Menos-Aikateriniadis, I. Lamprinos, and P. S. Georgilakis, "Particle swarm optimization in residential demand-side management: A review on scheduling and

control algorithms for demand response provision," *Energies,* vol. 15, no. 6, p. 2211, 2022.

[80] M. F. Sanner, "Python: a programming language for software integration and development," *J Mol Graph Model,* vol. 17, no. 1, pp. 57-61, 1999.

[81] P. Schober, C. Boer, and L. A. Schwarte, "Correlation coefficients: appropriate use and interpretation," *Anesthesia & analgesia,* vol. 126, no. 5, pp. 1763-1768, 2018.

[82] M. Ghosh, R. Guha, I. Alam, P. Lohariwal, D. Jalan, and R. Sarkar, "Binary genetic swarm optimization: A combination of GA and PSO for feature selection," *Journal of Intelligent Systems,* vol. 29, no. 1, pp. 1598-1610, 2019.

[83] T. Yu and H. Zhu, "Hyper-parameter optimization: A review of algorithms and applications," *arXiv preprint arXiv:2003.05689,* 2020.

[84] B. J. Jansen, "The graphical user interface," *ACM SIGCHI Bulletin,* vol. 30, no. 2, pp. 22-26, 1998.

[85] G. Aberg and J. Chang, "Applying Cognitive Science Research in Graphical User Interface (GUI)," *Umea Institute of Design,* pp. 23-28, 2005.

[86] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," *PeerJ Computer Science,* vol. 7, p. e623, 2021.

[87] W. Wang and Y. Lu, "Analysis of the mean absolute error (MAE) and the root mean square error (RMSE) in assessing rounding model," in *IOP conference series: materials science and engineering,* 2018, vol. 324, p. 012049: IOP Publishing.

[88] B. M. Greenwell, "pdp: An R package for constructing partial dependence plots," *R J.,* vol. 9, no. 1, p. 421, 2017.