PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH



FACULTY OF TECHNOLOGY

THESIS

To obtain the degree of

DOCTOR in: Process engineering

Speciality: Chemical Engineering

Presented by: ABDELLATIF Hayet

Thesis title

Contributions to the Development of Prediction Models for the Toxicity of Ionic Liquids

---

Defended in front of a jury composed of:

Last and first name	Grade	Institution of affiliation	Designation
BENKORTBI Othmane	Professor	University of Medea	President
LAIDI Maamar	Professor	University of Medea	Supervisor
SI-MOUSSA Cherif	Professor	University of Medea	Co-supervisor
KERCHICH Yacine	Professor	ENP Algiers	Examiner 1
KHAOUANE Latifa	professor	University of Medea	Examiner 2
HANINI Salah	Professor	University of Medea	Invited

Academic year : 2025/2026

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE



FACULTE DE TECHNOLOGIE

THESE DE DOCTORAT

Présentée en vue de l'obtention du diplôme de

Docteur en : Génie des Procédés

Specialité : Génie chimique

Présentée par: ABDELLATIF Hayet

Thème

Contribution aux développements des modèles de prédiction de la toxicité des liquides ioniques

---

Présentée devant le jury composé de :

Nom & Prénom	Grade	Institution d'affiliation	Designation
BENKORTBI Othmane	Professeur	Université de Médéa	President
LAIDI Maamar	Professeur	Université de Médéa	Directeur de thèse
SI-MOUSSA Cherif	Professeur	Université de Médéa	Co- Directeur de thèse
KERCHICH Yacine	Professeur	ENP Alger	Examinateur 1
KHAOUANE Latifa	Professeur	Université de Médéa	Examinateur 2
HANINI Salah	Professeur	Université de Médéa	Invité

Année académique : 2025/2026

Acknowledgments

First and foremost, I am deeply grateful to Allah (سُبْحَانَهُ وَتَعَالَى), for His endless mercy, strength, and guidance, which have accompanied me throughout this journey and made it possible to bring this thesis to completion.

I would like to express my sincere and heartfelt thanks to Professor Laidi Mammar and Professor Si-Moussa Chérif, my doctoral supervisors, whose wisdom, patience, and unwavering support were instrumental in the successful realization of this research. Their generous commitment, insightful guidance, and continuous encouragement helped shape both the direction of my work and my academic development. I consider myself privileged to have worked under their mentorship.

I also extend my warm thanks to Professor Abdeltif Amrane, whose valuable suggestions and constructive feedback greatly enriched this work.

I am sincerely thankful to all the members of the jury for the honor they have shown me by accepting to evaluate this thesis and for the time they devoted to reading and assessing my work. Their observations and remarks were of great value.

My gratitude also goes to the faculty members and teachers who have guided me throughout my academic years, transmitting not only knowledge but also critical thinking and scientific rigor.

A special thank you to my family, whose unconditional support, encouragement, and belief in me have been my anchor throughout all the challenges. Their love and sacrifices are the foundation upon which this achievement stands.

Lastly, I am especially grateful to my husband, whose patience, unwavering support, and constant motivation have accompanied me through every stage of this journey. His belief in me, even during the most difficult moments, gave me the strength to persevere and succeed

Finally, I would like to thank every person who, in one way or another, contributed to the completion of this work. Whether through academic support, moral encouragement, or friendship, your presence made this journey meaningful.

Dedication

With immense love and gratitude, I dedicate this work:

*To my dear **Father** and **Mother**, whose sacrifices, unwavering love, and endless support have been the bedrock of my life. Your belief in me has been my greatest motivation.*

*To my dear **husband**, for his constant understanding, patience, and encouragement that saw me through every challenge.*

*To my precious **son, Diaaeddine**, my greatest joy and inspiration, whose smile brightened my hardest days and gave me the strength to keep going.*

*To my beloved **sisters**, **Nawel** and **Meriem**, and their wonderful children, for their warmth, support, and constant presence.*

*To my brothers, **Mohamed** and **Abderaouf**, for their enduring support, boundless affection, and for being my true right-hand men.*

*To my **husband's family**, for their kindness and heartfelt encouragement.*

*To my dear friends, **Widad** and **Imene**, for their invaluable help, sincere encouragement, and constant support throughout the journey of this thesis.*

*To all my other friends, and to all the **teachers of the Faculty of Technology**, who have supported me throughout this process. I will always appreciate all they have done.*

And finally, to all who wish me only the best.

Hayet

Abstract

ملخص

مساهمات في تطوير نماذج التنبؤ بسمية السوائل الأيونية

تعدّ السوائل الأيونية (ILs) فئة جديدة من المذيبات الخضراء المبتكرة التي تحظى باهتمام كبير من الناحية الأكاديمية والصناعية والبيئية، وذلك بفضل خصائصها الفريدة. ونظرًا للتنوع البنيوي الكبير الذي تمتاز به، فإن تقييم سُميّتها يُعدّ خطوة ضرورية وأساسية قبل استخدامها في التطبيقات المختلفة. يهدف هذا العمل إلى تطوير نماذج تنبؤية قوية للعلاقة الكمية بين البنية والسمية (QSTR) للتنبؤ بسُمية السوائل الأيونية، مما يُوفر بدائل حاسوبية فعّالة للتقييمات التجريبية التقليدية.

في المرحلة الأولى من هذا العمل، تم تطوير نماذج آلة ناقلات الدعم (SVM) وتم تحسينها باستخدام ثلاث خوارزميات استمثال ميتاهيوريستية، وهي خوارزمية اليعسوب(DA) ، وخوارزمية اللهب والعثة(MFO) ، وخوارزمية الذئب الرمادي (GWO) تم اختيار الموصِفات الجزيئية باستخدام طريقتين: خوارزمية جينية مدمجة مع الانحدار الخطي المتعدد (GA-MLR) ، وطريقة بوروتا (Boruta) وقد أظهر نموذج DA-SVM أداءً تنبؤيًا متميزًا، حيث حقق قيمة جذر متوسط مربع الخطأ (RMSE) بلغت 0.1787 ومعامل تحديد (R^2) بلغ 0.9742 على مجموعة البيانات العالمية.

في المرحلة الثانية، تم إدماج موصِفات جديدة تُعرف باسم ARKA. مع نماذج تعلم تجميعي متقدمة، وهي XGBoost و Random Forest وقد أظهرت النماذج المبنية بهذه الموصفات أداءً تنبؤيًا محسّنًا مقارنةً بالموصِفات التقليدية. على سبيل المثال، حقق نموذج RF-ARKA قيمة R^2 بلغت 0.9421 وقيمة R بلغت 0.9707 على مجموعة البيانات العالمية.

بشكل عام، تقدم النماذج المطورة في هذا العمل أدوات فعالة وقابلة للتفسير للتنبؤ بسُمية السوائل الأيونية، مما يُسهم في تصميم كيميائي أكثر أمانًا، ويعزز التوجه نحو كيمياء مستدامة وصديقة للبيئة.

الكلمات المفتاحية :السوائل الأيونية، السمية، العلاقة الكمية بين البنية والسمية (QSTR) ، التعلم الآلي، آلة ناقلات الدعم (SVM) ، خوارزمية اليعسوب(DA) ، خوارزمية اللهب والعثة(MFO) ، خوارزمية الذئب الرمادي (GWO)، موصِفاتARKA ، تعزيز التدرج المتطرف (XGBOOST)، الغابة العشوائية (RF)، مجال التطبيق.

Abstract

Contributions to the Development of Prediction Models for the Toxicity of Ionic Liquids

Ionic Liquids (ILs) are a new class of innovative green solvents of significant importance from academic, industrial, and environmental perspectives due to their attractive properties. For designing any process involving ILs, understanding their toxicity is crucial. This work develops robust Quantitative Structure-Toxicity Relationship (QSTR) models to predict the toxicity of ILs, offering efficient in silico alternatives to traditional experimental assessments.

Initially, this study developed Support Vector Machine (SVM) models optimized using three meta-heuristic algorithms: Dragonfly Algorithm (DA), Moth-Flame Optimization (MFO), and Grey Wolf Optimizer (GWO). These models utilized molecular features selected by Genetic Algorithm-Multiple Linear Regression (GA-MLR) and Boruta descriptor selection techniques. Among these, the DA-SVM model consistently demonstrated superior predictive accuracy and robustness. For instance, the DA-SVM model with GA-MLR descriptors achieved a global RMSE of 0.1787 and an R^2 of 0.9742.

Subsequently, a distinct approach explored the employment of novel ARKA (Autoencoded Reduced Kernel-based Architecture) descriptors combined with advanced ensemble learning methods, XGBoost and Random Forest. Models built with these ARKA descriptors consistently outperformed their GA-MLR counterparts, showing significantly enhanced predictive power and generalization. For example, the RF-ARKA model achieved a global R^2 of 0.9421 and R of 0.9707.

Overall, the developed models offer reliable, interpretable, and efficient tools for predicting IL toxicity, supporting safer chemical design and contributing to sustainable chemistry.

Keywords: Ionic Liquids (ILs), Toxicity, Quantitative Structure-Toxicity Relationship (QSTR), Machine Learning, Support Vector Machine (SVM), Dragonfly Algorithm (DA), Moth-Flame Optimization (MFO), Grey Wolf Optimizer (GWO), ARKA Descriptors, eXtreme Gradient Boosting (XGBOOST), Random Forest (RF), Applicability Domain.

Abstract

Résumé

Contribution aux développements des modèles de prédiction de la toxicité des liquides ioniques

Les liquides ioniques (LIs) constituent une nouvelle classe de solvants verts innovants d'une grande importance académique, industrielle et environnementale grâce à leurs propriétés attrayantes. Pour concevoir tout processus impliquant des LIs, la compréhension de leur toxicité est cruciale. Ce travail développe des modèles robustes de Relations Structure-Toxicité Quantitatives (QSTR) pour prédire la toxicité des LIs, offrant des alternatives in silico efficaces aux évaluations expérimentales traditionnelles.

Initialement, cette étude a développé des modèles de Machine à Vecteurs de Support (SVM) optimisés par trois algorithmes métaheuristiques : l'Algorithme de la Libellule (DA), l'Optimisation par la Moth-Flame (MFO) et l'Optimiseur des Loups Gris (GWO). Ces modèles ont utilisé des caractéristiques moléculaires sélectionnées via les techniques GA-MLR (Algorithme Génétique-Régression Linéaire Multiple) et Boruta. Parmi ceux-ci, le modèle DA-SVM a systématiquement démontré une précision et une robustesse prédictives supérieures. Par exemple, le modèle DA-SVM avec les descripteurs GA-MLR a atteint un RMSE global de 0.1787 et un R^2 de 0.9742.

Ensuite, une autre approche a exploré l'utilisation des descripteurs innovants ARKA (Résidus Arithmétiques dans l'Analyse des groupes K), combinés à des algorithmes d'apprentissage ensembliste comme XGBoost et Random Forest. Les modèles développés avec ces descripteurs ARKA ont systématiquement surpassé ceux basés sur GA-MLR, en termes de performance prédictive et de généralisation. Par exemple, le modèle RF-ARKA a atteint un R^2 global de 0.9421 et un R de 0.9707.

Dans l'ensemble, les modèles développés offrent des outils fiables, interprétables et efficaces pour la prédiction de la toxicité des LIs, contribuant ainsi à une conception chimique plus sûre et à une chimie durable.

Mots-clés: Liquides Ioniques, Toxicité, Relations Structure-Toxicité Quantitatives (QSTR), Apprentissage Automatique, Machine à Vecteurs de Support (SVM), Algorithme de la Libellule (DA), Optimisation par la Moth-Flame (MFO), Optimiseur des Loups Gris (GWO), Boruta, Descripteurs ARKA, eXtreme Gradient Boosting (XGBOOST), Random Forest (RF), Domaine d'Application.

List of Contents

List of Contents……..……I
List of Abbreviations III
List of Nomenclatures and symbols.....V
List of Figures. VII
List of Tables.....X
General Introduction 1

Chapter I
Ionic Liquids: Overview and Literature Review

I.1 Introduction.....5
I.2. General Overview of Ionic Liquids (ILs).....5
I.2.1. Definition, History, and Structure of ILs 5
I.2.2. Physicochemical Properties of Ionic Liquids 6
I.2.3. Applications of Ionic Liquids 9
I.3 Toxicity of Ionic Liquids and the Role of Predictive Models 10
I.3.1. Literature Review on Predictive Models for Assessing ILs Toxicity 11
I.4 Conclusion.....27

Chapter II
Overview of modeling with machine and deep learning

II.1. Introduction 28
II.2. Machine and Deep Learning Models.....29
II.3. Types of Machine Learning 31
II.3.1. Supervised Learning (SL).....32
II.3.2. Unsupervised Learning (UL) 32
II.3.3. Reinforcement Learning (RL).....32
II.3.4. Semi-supervised Learning (SeL) 34
II.4. Specific Machine Learning Algorithms Employed.....34
II.4.1. Support Vector Machine (SVM) 34
II.4.2. Ensemble Learning: Random Forest (RF).....36
II.4.3. Ensemble Learning: eXtreme Gradient Boosting (XGBoost).....38
II.5. Optimization of Model Parameters 41
II.5.1. Dragonfly Algorithm (DA).....41
II.5.2. Grey Wolves optimization (GWO) 44
II.5.3. Moth–Flame Optimization (MFO) 45
II.6. Model Evaluation and validation 48
II.7. Addressing Overfitting and Underfitting in Machine Learning 48

List of Contents

II.8. Conclusion 49

Chapter III
Development of prediction models for the toxicity of ionic liquids

III.1 Introduction 51
III.2. Overview of the Methodology 51
III.3 Materials and Methods 52
III.3.1 Experimental Dataset 52
III.3.2. Descriptor Calculation 53
III.3.3. Descriptor Selection 54
III.3.3.1. Genetic Algorithm combined with Multiple Linear Regression (GA-MLR) 54
III.3.3.2.Boruta Algorithm 56
III.3.4. Data Division 58
III.3.5. Meta-heuristic Optimized SVM for QSTR Modeling 58
III.3.6. Statistical Criteria for Evaluation of Model Performance 59
III.4. Results and Discussion 62
III.4.1. Optimized SVM Hyperparameters and Comparisons of Performances 62
III.4.2. Influence of molecular descriptors 76
III.4.3. SHAP Analysis 80
III.4.4. Applicability Domain 83
III.4.4.1.William's Plot 83
III.4.4.2. Insubria Graph 85
III.4.5. External Test 87
III.4.6. Analysis of Activity and Prediction Cliffs using the ARKA Method 90
III.4.6.1. ARKA descriptor computation and dataset description 91
III.4.6.2. Violin plots (ARKA Descriptors vs. Activity) 93
III.4.6.3. $ARKA_1$ vs $ARKA_2$scatter plot(Activity and Prediction Cliffs) 94
III.4.6.4. Descriptor Contribution Analysis to ARKA Descriptors 97
III.4.6.5. Comparative Modeling with XGBoost and Random Forest Using GA-MLR and ARKA Descriptors 97
III.4.6.5.1. Models Development and Hyperparameter Optimization 98
III.4.6.5.2. Comparative Performance Analysis 99
III.4.6.6. Applicability Domain Analysis (William's Plots) 106
III.5. Conclusion 107
General Conclusion 109
Bibliography 112

List of Abbreviations

Abbreviation	Definition
AD	Applicability Domain
AIC	Akaike's Information Criterion
ANN	Artificial Neural Network
ARKA	Arithmetic Residuals in K-groups Analysis
CV	Cross Validation
DA	Dragonfly Algorithm
GA	Genetic Algorithm
GC-COSMO	Group Contribution COSMO
GA-MLR	Genetic Algorithm-Multiple Linear Regression
GWO	Grey Wolf Optimizer
Ils	Ionic Liquids
MAE	Mean Absolute Error
MLR	Multiple Linear Regression
MFO	Moth-Flame Optimization
MSE	Mean Squared Error
MFO-SVM	Moth-Flame Optimization-Support Vector Machine
QSTR	Quantitative Structure–Toxicity Relationship
QSAR	Quantitative Structure–Activity Relationship
RF	Random Forest
RMSE	Root Mean Square Error
SMILES	Simplified Molecular Input Line Entry System
SSE	Sum of Squared Errors
SHAP	SHapley Additive exPlanations

List **of Abbreviations**

SVM	Support Vector Machine
XGBoost	eXtreme Gradient Boostin

List of Nomenclatures and Symbols

Symbol	Name
$\bar{x}$_obs	Average of the observed values
$\bar{x}$_pred	Average of the predicted values
$\bar{y}$_pred	Average of predicted Y
$\bar{r}m^2$	Average of rm
$\bar{y}$_obs	Average of observed Y
h^*	Threshold leverage value
x_i_obs	Observed value for instance i
x_i_pred	Predicted value for instance i
y_i_obs	Observed (experimental) Y value for instance i
y_i_pred	Predicted Y value for instance I, in training, test or validation set
NP	Number of independent variables in each model
$QF1^2$, $QF2^2$, $QF3^2$	Different types of cross-validated correlation coefficients
Rm^2, $R'm^2$	Marginal coefficients of determination
R^2	Coefficient of determination
R_adj	Adjusted correlation coefficient
R_adj^2	Adjusted coefficient of determination
h	Leverage value
Q^2	Cross-validated correlation coefficient
Q^2ccc	Concordance correlation coefficient
R	Correlation coefficient
EC50	Half-maximal effective concentration
$\log EC_{50}$	Logarithm of the half-maximal effective concentration (toxicity value)
α (alpha)	Alpha parameter
β (beta)	Beta parameter
δ (delta)	Delta parameter
ε (epsilon)	Epsilon parameter in SVM

List of Nomenclatures and Symbols

σ (sigma)	Kernel scale parameter in SVM
φ(x)	Kernel function
ω (omega)	Omega parameter
A_f	Accuracy Factor (Statistical metric)
B_f	Bias Factor (Statistical metric)
K	Slope (Statistical metric)
K′	Alternative Slope (Statistical metric)

List of Figures

Figure No.	Caption	Page No.
Chapter I Ionic Liquids: Overview and Literature Review		
Figure I.1	Examples of commonly used cations and anions in ionic liquids, along with their abbreviations.	06
Figure I.2	Schematic representation of the main industrial and scientific applications of ionic liquids (ILs)	10
Chapter II Overview of modeling with machine and deep learning		
Figure II.1	Illustration of the hierarchy and interconnections among artificial intelligence (AI), machine learning (ML), deep learning/artificial neural networks (ANN), and generative AI	31
Figure II.2	overview of machine learning (ML) methods	33
Figure II.3	Illustration of the various branches and practical applications of ML.	34
Figure III.4	Overfitting and Underfitting in Machine Learning	49
Chapter III Development of prediction models for the toxicity of ionic liquids		
Figure III.1	The distribution of the various ionic liquid families utilized in this study.	53
Figure III.2	Convergence curve of the three algorithms (using GA_MLR descriptors).	64
Figure III.3	Convergence curve of the three algorithms (using Boruta descriptors).	64
Figure III.4	Scatter plots predicted vs. observed values for the (a) train, (b) test and (c) global sets, for DA_SVM model (GA_MLR).	65
Figure III.5	Scatter plots predicted vs. observed values for the (a) train, (b) test and (c) global sets, for MFO_SVM model (GA_MLR).	66
Figure III.6.	Scatter plots predicted vs. observed values for the (a) train, (b) test and (c) global sets, for GWO_SVM model (GA_MLR).	67
Figure III.7	Scatter plots predicted vs. observed values for the (a) train, (b) test and (c) global sets, for DA_SVM model (Boruta).	68
Figure III.8	Scatter plots predicted vs. observed values for the (a) train, (b) test and (c) global sets, for MFO_SVM model (Boruta).	69

List of Figures

Figure III.9	Scatter plots predicted vs. observed values for the (a) train, (b) test and (c) global sets, for GWO_SVM model (Boruta).	70
Figure III.10	The relative error (%) of the toxicity by the DA_SVM model (GA_MLR descriptors).	75
Figure III.11	The relative error (%) of the toxicity by the DA_SVM model (Boruta descriptors).	75
Figure III.12	Distribution diagrams of relative error (%) of the DA_SVM model.	76
Figure III.13	The results of the relevance factor performed on the DA_SVM model (GA_MLR descriptors).	77
Figure III.14	The results of the relevance factor performed on the DA_SVM model (Boruta Descriptors).	78
Figure III.15	Correlation matrix of the thirteen descriptors selected by GA_MLR.	79
Figure III.16.	Correlation matrix of the seventeen descriptors selected by Boruta.	80
Figure III.17	SHAP interpretability and features importance analysis for detected Log(EC_{50}) DA_SVM model for GA_MLR sele descriptors: summary plots (a) and Bar plot (b).	81
Figure III.18	SHAP interpretability and features importance analysis for t detected Log(EC_{50}) DA_SVM model for Boruta select descriptors: summary plots (a) and Bar plot (b).	82
Figure III.19	William's plot for DA_SVM model (GA_MLR selected descriptors).	84
Figure III.20	William's plot for DA_SVM model (Boruta selected descriptors).	85
Figure III.21	Insubria graph for the DA_SVM model (descriptors selected by GA_MLR).	86
Figure III.22.	Insubria graph for the DA_SVM model (descriptors selected by Boruta).	86
Figure III.23	Scatter Plot of Predicted vs. Observed Values for DA_SVM in External Testing (GA_MLR).	87
Figure III.24	Scatter Plot of Predicted vs. Observed Values for DA_SVM in External Testing (Boruta).	88
Figure III.25	Violin Plot for Comparative analysis of residuals in the external test set.	90
Figure III.26	Process for calculating ARKA descriptors.	92

List of Figures

Figure III.27	Violin Plots of ARKA 1 vs Activity and ARKA 2 vs Activity for the training and test sets.	94
Figure III.28	$ARKA_1$ vs $ARKA_2$ in the training set.	95
Figure III.29	ARKA 1 vs ARKA 2 in the test set.	96
Figure III.30	Contributions of QSTR Descriptors to ARKA 1 (Outer Ring) and ARKA 2 (Inner Ring).	97
Figure III.31	Comparison of Error Metrics (RMSE, MSE and MAE) Across Models for Train, Test, and Global Sets.	101
F igure III.32	Scatter plots predicted vs. observed values for XGBOOST model using ARKA descriptors.	102
Figure III.33	Scatter plots predicted vs. observed values for XGBOOST model using GA_MLR descriptors.	103
Figure III.34	Scatter plots predicted vs. observed values for RF model using ARKA descriptors.	104
Figure III.35	Scatter plots predicted vs. observed values for RF model using GA_MLR descriptors.	105
Figure III.36	Williams Plot for the RF Model Using ARKA Descriptors.	106
Figure III.37	Williams Plot for the XGBOOST Model Using ARKA Descriptors.	107

List of Tables

Table No.	Caption	Page No.
Chapter I **Ionic Liquids: Overview and Literature Review**		
Table I.1	Comparison between organic solvents and ionic liquids (ILs), adapted from the work of Plechkova and Seddon	08
Chapter III **Result and Discussion**		
Table III.1	Characteristics of the descriptors selected by GA_MLR.	55
Table III.2	Characteristics of the descriptors selected by Boruta.	56
Table III.3	Error and Statistical metrics commonly used for model evaluation	60
Table III.4	Optimized SVM hyperparameters for GA-MLR and Boruta descriptors.	63
Table III.5	Different errors of the three SVM models.	71
Table III.6	AIC values of the three SVM model.	71
Table III.7	Comprehensive statistical metrics for SVM models (GA-MLR and Boruta Descriptors).	72
Table III.8	Prediction Errors for the External Validation Set for DA_SVM model.	89
Table III.9	Hyperparameter Value for Random Forest Model.	98
Table III.10	Hyperparameter Value for XGBoost Model.	99
Table III.11	Performance Metrics for XGBoost and Random Forest using GA-MLR and ARKA descriptors.	100

General Introduction

General Introduction

In recent decades, chemical innovation has seen an unprecedented acceleration, leading to the synthesis and application of myriad novel compounds crucial for advancements across diverse industries, including pharmaceuticals, materials science, agriculture, and energy [1]. This progression has been markedly influenced by a paradigm shift from traditional empirical, trial-and-error discovery methods towards more rational and predictive design methodologies, which aim to engineer chemicals with optimized functionalities and inherent safety. Concurrently, a heightened global consciousness regarding environmental protection and human health has underscored the critical need for rigorous hazard and property assessment of these emerging chemical entities. The tenets of "green chemistry," advocating for the design of chemical products and processes that minimize the use and generation of hazardous substances, stand at the forefront of promoting sustainability and reducing environmental burdens [2–4].

However, the inherent trade-off between maximizing chemical utility and minimizing potential environmental or health impacts remains a formidable challenge. Conventional experimental approaches for assessing chemical properties and toxicity are frequently resource-intensive, time-consuming, and often fraught with ethical considerations, particularly due to their reliance on animal testing [5–7]. These inherent limitations significantly impede the efficient screening of vast chemical libraries and the rapid evaluation of novel compounds, thereby creating a substantial bottleneck in the transition from chemical discovery to their safe and widespread application. To address this pressing need, computational chemistry, encompassing disciplines such as Quantitative Structure-Activity Relationships (QSAR) and Quantitative Structure-Property Relationships (QSPR), has emerged as an indispensable in silico tool. These methods offer a highly promising avenue for rapid, cost-effective, and predictive hazard and property assessment, crucially supporting the "benign-by-design" principle for a safer and more sustainable chemical future [8–10].

Among the vast array of modern chemical innovations, Ionic Liquids (ILs) represent a particularly captivating and rapidly expanding class of compounds. Defined fundamentally as salts composed entirely of ions that possess a melting point below 100°C, ILs present a remarkable deviation from the conventional understanding of salts, which are typically characterized by high melting temperatures at ambient conditions [11, 12]. Although their

initial discovery can be traced back to the early 20th century with Paul Walden's seminal identification of ethylammonium nitrate, it was not until the fervent rise of green chemistry principles in the 1990s that ILs truly began to capture significant academic and industrial attention [13, 14]. This resurgence of interest has been vigorously fueled by their array of unique and highly tunable physicochemical properties, which position them as exceptionally promising alternatives to traditional volatile organic solvents [15].

The unparalleled versatility of ILs is primarily derived from their modular nature, allowing for the precise tailoring of their properties. They are typically constituted by large, bulky, and often asymmetric organic cations (e.g., imidazolium, pyridinium, phosphonium and ammonium) in combination with a diverse range of organic or weakly coordinating inorganic anions (e.g., tetrafluoroborate $[BF_4]^-$, hexafluorophosphate $[PF_6]^-$, bis(trifluoromethanesulfonyl)imide $[NTf_2]^-$, acetate) [16–19]. This nearly limitless combinatorial potential of cations and anions enables the fine-tuning of their physical and chemical attributes, a characteristic that has justly earned them the distinction of "designer solvents" [20, 21].

Their unique physicochemical properties, including negligible vapor pressure, high thermal stability, and tunable solvation capabilities, make them attractive alternatives to conventional volatile organic solvents. These properties have led to their designation as “green solvents” with potential applications in catalysis, electrochemistry, separations, and biotechnology, among others [22–24].

Although ionic liquids (ILs) are often described as “green” because of their low volatility, many recent studies have shown that they are not always harmless. In fact, ILs can pose risks to both the environment and human health [25, 26]. Importantly, this toxicity is strongly structure-dependent. For example, researchers have found that ILs with longer alkyl side chains on the cation tend to be more toxic to various organisms such as bacteria, algae, and mammalian cells. A robust understanding of these intricate structure-toxicity relationships is therefore not just beneficial but paramount for the rational design of safer and more sustainable ILs [27–29].

Given the enormous variety of ionic liquids (ILs) that can be theoretically synthesized (estimated at up to 10^{18} possible combinations of cations and anions) [16, 17] it is neither practical nor economically feasible to conduct experimental toxicity testing for each individual compound. Moreover, experimental procedures are often time-consuming,

costly, and may introduce uncertainties. These challenges highlight the urgent need for reliable computational approaches to predict the toxicity of ILs before they are synthesized and applied. As a result, predictive modeling has emerged as an essential alternative. By building accurate and robust models capable of estimating IL properties under different conditions, researchers can accelerate the development of safer and more effective ILs while minimizing the reliance on extensive experimental testing. These models play a critical role in supporting the industrial use of ILs, ensuring both efficiency and environmental safety [30, 31].

In this context, Quantitative Structure–Toxicity Relationship (QSTR) modeling emerges as a powerful and cost-effective alternative to experimental testing. QSTR models aim to establish mathematical relationships between the molecular structure of a compound (represented by a variety of molecular descriptors) and its observed toxicological behavior. These models enable the early-stage screening of large chemical libraries, support regulatory decision-making, and contribute to the principles of green chemistry by significantly reducing the need for animal testing and experimental resource consumption [8, 31–34].

This thesis is meticulously structured to systematically address the development and application of advanced computational methodologies for predicting the toxicity of ionic liquids, contributing to a more sustainable chemical future. It is organized into three main chapters, each focusing on a critical component of the study, from foundational knowledge and literature review, to the theoretical principles of machine learning and optimization algorithms, and finally, to the practical implementation and evaluation of QSTR models for toxicity prediction. This structure ensures a logical progression from conceptual groundwork to methodological execution and result interpretation, providing a comprehensive and cohesive framework for understanding and advancing predictive toxicology of ionic liquids.

Chapter I: Ionic Liquids: Overview and Literature Review: This chapter provides a comprehensive foundational understanding of ionic liquids, detailing their definition, historical background, intricate structural diversity, unique and tunable physicochemical properties, and extensive range of applications across various sectors. Crucially, it initiates a critical and exhaustive literature review focused on existing QSTR/QSAR models for the

toxicity of ILs, highlighting the significant progress achieved, persistent limitations, and current challenges that necessitate further research.

Chapter II: Theoretical Foundations of Machine Learning and Optimization Algorithms: This chapter is dedicated to providing a detailed theoretical exposition of the key machine learning and meta-heuristic optimization algorithms strategically employed throughout this research. It will delve into the fundamental concepts and historical development of Support Vector Machines (SVM), and comprehensively describe the principles and operational mechanisms of the Dragonfly Algorithm (DA), Moth-Flame Optimization (MFO), and Grey Wolf Optimizer (GWO). Furthermore, ensemble learning methods, specifically Random Forest (RF) and eXtreme Gradient Boosting (XGBoost), will be thoroughly explored, thereby providing the essential technical and conceptual context for their sophisticated application in predictive modeling within this thesis.

Chapter III: Development of prediction models for the toxicity of ionic liquids (Result and Discussion): This chapter represents the methodological and empirical core of this thesis, detailing the comprehensive QSTR modeling workflow. It meticulously describes the rigorous data preprocessing pipeline, the advanced methodologies for molecular descriptor generation, and the implementation of two distinct and powerful dimensionality reduction techniques: the Genetic Algorithm combined with Multiple Linear Regression (GA-MLR) and the Boruta algorithm, for the discerning selection of the most relevant molecular features influencing IL toxicity. A significant focus is placed on the precise optimization of Support Vector Machine models utilizing the aforementioned meta-heuristic algorithms. Additionally, the chapter elaborates on a novel approach involving the development and evaluation of QSTR models integrating ARKA (Autoencoded Reduced Kernel-based Architecture) descriptors within state-of-the-art ensemble learning frameworks (XGBoost and Random Forest). A rigorous comparative analysis of all developed models is presented, employing a wide array of statistical metrics, comprehensive visual evaluation tools (such as scatter plots and violin plots), and advanced interpretability methods, including SHAP analysis. Furthermore, robust applicability domain analyses (Williams plots and Insubria graphs) are conducted, alongside the crucial identification of activity and prediction cliffs, providing critical insights into the models' reliability and pinpointing specific areas for future structural optimization of ILs.

Chapter I

Ionic Liquids: Overview and Literature Review

Chapter I
Ionic Liquids: Overview and Literature Review

I.1 Introduction

This chapter provides a comprehensive overview of ionic liquids (ILs), beginning with their definition, historical background, structure, and principal physicochemical properties. Their main fields of application are also discussed, emphasizing their versatility across various scientific and industrial sectors. Furthermore, the issue of IL toxicity is introduced, along with the growing importance of predictive models for assessing their potential risks. In this context, a literature review of research papers published between 2003 and 2024 is presented. This review, which covers most of the published work related to predicting and modeling the toxicity of ionic liquids, focuses on the various QSTR/QSAR models proposed in the literature. Particular attention is given to their methodological approaches, dataset composition, descriptor choices, and statistical performance

I.2. General Overview of Ionic Liquids (ILs)

I.2.1. Definition, History, and Structure of ILs

Ionic liquids (ILs) are defined as salts composed exclusively of cations and anions that remain in the liquid state at temperatures below 100 °C [12, 35, 36]. Their discovery dates back to 1914, when Paul Walden identified ethylammonium nitrate ($[EtNH_3][NO_3]$) as the first room-temperature ionic liquid [12, 13, 35–38]. However, ILs only began to attract real scientific interest in the 1990s with the rise of green chemistry [39, 40], which highlighted their potential as promising alternatives to volatile organic solvents [12, 13, 23, 35, 36, 41].

The chemical structure of ILs generally relies on bulky and asymmetric organic cations, such as imidazolium, pyridinium, phosphonium, and substituted ammonium species, combined with either organic or weakly coordinating inorganic anions. Common inorganic anions include $[BF_4]^-$ (tetrafluoroborate) and $[PF_6]^-$ (hexafluorophosphate), while widely used organic anions include $[NTf_2]^-$ (bis(trifluoromethanesulfonyl)imide) and

acetate. Examples of typical cations and anions, along with their preferred abbreviations, are illustrated in **Figure I.1** [13, 22, 41–43].

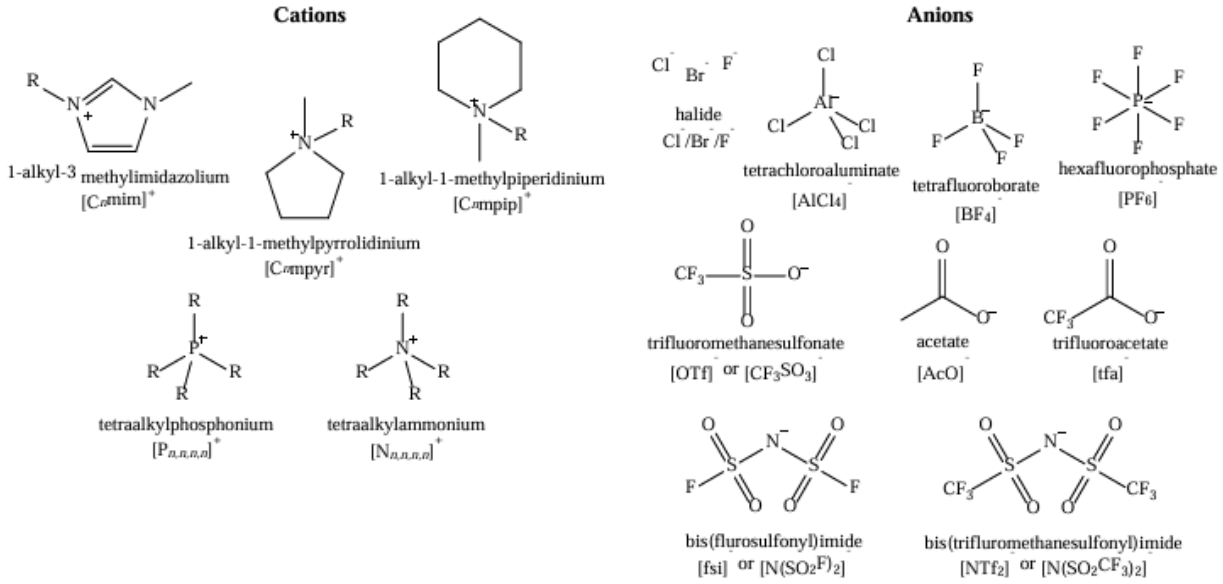


Figure I.1. Examples of commonly used cations and anions in ionic liquids, along with their abbreviations [42].

I.2.2. Physicochemical Properties of Ionic Liquids

Ionic liquids (ILs) have attracted growing scientific interest due to their remarkable physicochemical properties, which arise from the structural diversity of their cations and anions. This diversity offers significant potential for applications in various fields such as catalysis, electrochemistry, and waste treatment. One of the most important characteristics of ILs is their extremely low vapor pressure, often negligible, which minimizes volatilization and reduces the risks of inhalation and atmospheric pollution [13, 23, 44].

Another notable property is their excellent thermal stability. Some ionic liquids can resist decomposition at temperatures reaching or exceeding 300 °C. This stability largely depends on the ion structure. Fluorinated anions such as $[PF_6]^-$ and $[NTf_2]^-$, along with bulky and symmetrical cations, generally improve thermal resistance. However, less stable ILs may begin to degrade at temperatures around 200 °C, depending on their chemical composition [41, 43, 45]. ILs also demonstrates good ionic conductivity due to their entirely ionic composition, which facilitates charge transport. Conductivity varies according to the nature of the cations and anions. Systems containing imidazolium cations

and weakly coordinating anions, such as $[Tf_2N]^-$, tend to have higher ion mobility. For example, ILs like [EMIM][Tf_2N] and [BMIM][$FeCl_4$] have shown high conductivities, making them promising candidates for electrochemical applications such as batteries and fuel cells[45]. Furthermore, ionic liquids possess a wide electrochemical window, which can reach 5 to 6V. This is significantly higher than the 1.23V limit of water, and is a crucial advantage for applications in batteries, super capacitors, and photovoltaic cells [12, 45, 46].

Among their various properties, ionic liquids also exhibit low flammability, which enhances their safety, particularly in sensitive or high-temperature environments [42, 45]. In addition, their high polarity improves their ability to dissolve various types of compounds. This property is particularly important for applications in chemical extraction and reactions [13, 43, 45]. Moreover, the polarity of ILs can be adjusted by modifying the ion structures, allowing for the design of hydrophilic (water-soluble) or hydrophobic (organic-soluble) liquids depending on the intended use [13, 43, 45].

Finally, all of these properties can be tuned through the molecular design of the cations and anions. For example, increasing the alkyl chain length of the cation raises the viscosity and hydrophobicity, while replacing halogenated anions with fluorinated ones can influence polarity, thermal stability, and hygroscopicity [12, 13, 23, 43, 46]. This flexibility gives ILs the reputation of "designer solvents," tailored for specific functions.

To better highlight the main similarities and differences between conventional organic solvents and ionic liquids (ILs), a comparative analysis was proposed by Plechkova and Seddon (2008) [12], providing a clear overview of the advantages and limitations of ILs compared to traditional solvents. To further emphasize these contrasts, Table I.1 presents a summary comparing the physicochemical, environmental, and functional properties of the two classes of solvents [12, 42].

Table I.1. Comparison between organic solvents and ionic liquids (ILs), adapted from the work of Plechkova and Seddon [12]. The value ranges were updated based on data from the NIST ILThermo database at 25 °C and under atmospheric pressure [12, 42].

Property	**Organic solvents**	**Ionic liquids**
Number of solvents	>1000	$>10^6$
Applicability	Single function : in a specific process	Multifunction: The same IL can serve as a solvent, catalyst, electrolyte…
Cost	Generally cheap	2 to100 times more expensive than organic solvents
Recyclability/Toxicity	Controlled by REACH	Often not well known
Vapour pressure	Measurable - Solvents have vapour pressure> limit used in the classification of volatile organic compounds (VOCs)	Negligible vapour pressure under normal conditions (especially for aprotic ILs)
Flammability	Usually flammable	Usually nonflammable
Tuneability	Limited range of solvents available	Virtually unlimited range means ‘‘designer solvents’’
Chirality	Rare	Common and tuneable
Catalytic ability	Rare	Common and tuneable
Viscosity **(mPa·s)**	0.2-100	20-97 000
Density **($g \cdot cm^{-3}$)**	0.6–1.7	0.8-3.3
Refractive Index	1.3-1.6	1.3-2.2
Thermal conductivity (**$W.m^{-1}.K^{-1}$**)	0.1-0.6	0.1-0.3
Electrochemical window (V)	**Limited range (generally < 3V)**	**Sometimes up to 5–6 V**
Electrical conductivity ($mS.cm^{-1}$)	Usually insulator	Up to 120

I.2.3. Applications of Ionic Liquids

Ionic liquids (ILs), known for their high structural versatility and exceptional physicochemical properties, are increasingly attracting attention across various industrial and scientific fields [12][13]. Their flexibility comes from the ability to adjust their ionic structure (by selecting different cations and anions), which allows the design of liquids tailored to meet specific requirements. A visual summary of the main application sectors is presented in Figure I.II below, highlighting the wide range of potential uses for these compounds.

In extraction processes, for example, certain ILs have proven effective in recovering heavy metals or pharmaceutical compounds, while also reducing the emission of volatile organic compounds (VOCs) [13, 41]. In the area of catalysis, ILs play an important role in reactions such as alkylation, hydrogenation, and multi-step transformations, thanks to their ability to dissolve and stabilize metal complexes [36][43]. From an electrochemical perspective, their high ionic conductivity and wide electrochemical window make them especially suitable for batteries, super capacitors, and energy storage systems [45][23]. Additionally, the use of IL-based stationary phases in chemical separations and mass spectrometry improves selectivity and sensitivity in certain analytical methods [42, 43]. Furthermore, ILs are the focus of advanced research in the field of carbon dioxide capture, offering an alternative to traditional solvents. It has been shown that their ability to dissolve CO_2 strongly depends on the nature of the anions, with some anions significantly improving gas solubility [47]. Moreover, the development of functionalized ILs, designed to interact specifically with CO_2 molecules, opens the door to more efficient and regenerable systems for carbon capture processes [48].



Figure I.2. Schematic representation of the main industrial and scientific applications of ionic liquids (ILs), adapted from [49].

I.3 Toxicity of Ionic Liquids and the Role of Predictive Models

Although ionic liquids (ILs) are widely promoted for their low volatility and favorable physicochemical properties, growing evidence suggests that they may still pose significant risks to both environmental and human health [50, 51]. One important factor is the cation's structure, particularly the alkyl side-chain length, as longer chains are often associated with increased toxicity in bacteria, algae, and mammalian cells [52, 53] Moreover, the anion strongly affects membrane permeability and hydrophobicity, thus influencing IL bioaccumulation and toxicity levels [25, 26]. For instance, Zhao et al. [51] observed that imidazolium or pyridinium based ILs exhibit pronounced toxic effects on certain aquatic

organisms, with EC_{50} values inversely related to side-chain length [51, 52]. EC_{50} (half maximal effective concentration) expressed in μmol/L and refers to the concentration of a substance required to produce a response halfway between the baseline and maximum after a defined exposure time, and is commonly used as a standard indicator of toxicity [54].

To better understand these toxicity patterns, recent research has increasingly focused on QSTR/QSAR (Quantitative Structure–Toxicity Relationships / Quantitative Structure–Activity Relationships) models. These models provide systematic frameworks to predict and assess the harmful effects of ILs [55, 56]. They represent a valuable tool for identifying structural features associated with toxicity and offer a promising approach toward designing safer and more environmentally friendly ILs [55][57]. In this context, various studies have been published proposing different QSTR/QSAR models to assess IL toxicity. Consequently, a critical review of research articles published between 2003 and 2024 has been undertaken to examine their methodological strategies, model designs, dataset compositions, and statistical performances. This analysis aims to provide deeper insights into the progress, limitations, and current challenges in predicting the toxicity of ionic liquids.

I.3.1. Literature Review on Predictive Models for Assessing ILs Toxicity

Jastorff et al. (2003) [50] conducted one of the earliest systematic investigations into the potential environmental hazards associated with ionic liquids (ILs), particularly focusing on their toxicity and biodegradability. The study evaluated a series of imidazolium- and pyridinium-based ILs, using ecotoxicological assays on *Vibrio fischeri*, *Daphnia magna*, and *Scenedesmus subspicatus*. The findings demonstrated that ILs containing longer alkyl side chains (such as butyl or hexyl) exhibited higher toxicity toward aquatic organisms, in some cases rivaling or exceeding that of conventional organic solvents. Moreover, the biodegradability of the ILs was found to be very limited, especially for those containing fluorinated anions like $[PF_6]^-$ or $[BF_4]^-$. These results challenged the prevalent notion of ILs as inherently "green" solvents. Although the study did not involve predictive modeling techniques, it highlighted a clear structure–toxicity relationship and introduced the concept of “benign-by-design” ILs, arguing that rational

structural modifications could significantly reduce toxicity. This paper laid a foundational understanding for later computational QSAR/QSTR approaches aimed at predicting and minimizing IL-related environmental risks.

Couling et al. (2005) [58] carried out a rigorous and methodologically balanced study to examine the structural factors influencing the toxicity of ionic liquids (ILs) toward aquatic species, specifically *Vibrio fischeri* and *Daphnia magna*, through quantitative structure–property relationship (QSPR) modeling. The authors combined previously published data with novel experimental measurements, creating two curated datasets: 25 ILs for *V. fischeri* (16 in training, 9 in prediction) and 17 ILs for *D. magna* (12 in training, 5 in prediction). Descriptors capturing electronic, topological, spatial, thermodynamic, and structural properties were computed separately for cations and anions. These were scaled to a [0, 1] interval before being processed through Genetic Function Approximation (GFA), which excluded highly correlated variables ($R^2 > 0.8$) to minimize redundancy. In the final QSPR models, four descriptors were selected for Vibrio fischeri and three for Daphnia magna, striking a balance between simplicity and predictive reliability. The resulting QSPR models demonstrated strong statistical performance, with R^2 values ranging from 0.78 to 0.88 across different training and prediction sets ($R^2 = 0.887$ for the *V. fischeri* training set, $R^2 = 0.782$ for the full combined model, and $R^2 = 0.862$ for the *D. magna* training set), highlighting their applicability in predicting IL toxicity. The authors demonstrated the relevance and importance of descriptor independence and model interpretability. The study is particularly valuable for introducing a transparent descriptors-based strategy to predict ILs toxicity, offering early evidence that QSPR tools can support safer IL design.

Luis et al. (2007) [59] proposed a novel QSAR approach based on group contribution methods (GCM) to predict the toxicity(log EC 50) of ionic liquids (ILs) toward *Vibrio fischeri*. . The dataset included 43 ILs. The range of $\log_{10}(EC_{50})$ values extended from –0.18 to 4.76, ensuring chemical diversity and a wide toxicity spectrum. Multilinear regression modeling was performed using Polymath software, and the resulting model demonstrated high statistical performance, with $R^2 = 0.925$ and adjusted $R^2 = 0.907$, and a low root mean square deviation (RMSD) of 0.0097. Furthermore, 88% of the predictions had an absolute error below 0.10, highlighting the model's precision. The results also

emphasized that pyridinium and pyrrolidinium cations, along with longer alkyl chains, tend to increase toxicity, whereas imidazolium structures and short-chain substitutions were generally less harmful. The model accurately predicted both the most and least toxic ILs in the dataset, illustrating its robustness.

Ranke et al. (2007) [60] explored the relationship between the chemical structure of ionic liquid (IL) cations and their cytotoxic effects in vitro. Their approach relied on an experimental method combining chromatographic analysis with toxicological assays to create a predictive model. A total of 74 ILs were investigated, encompassing a diverse set of cation classes including imidazolium, pyridinium, pyrrolidinium, quinolinium, and quaternary ammonium and phosphonium types. To minimize external variability, only small and non-reactive anions such as chloride, bromide, tetrafluoroborate, and hexafluorophosphate were included. Cytotoxicity was measured using *rat leukemia IPC-81 cells* in a photometric assay. As the main descriptor, the authors used a chromatographically derived lipophilicity parameter to represent each cation's behavior. A regression model was developed to correlate this descriptor with the observed toxicity values. Statistically, the model demonstrated moderate predictive performance, the full dataset produced a coefficient of determination (R^2) of 0.78 with a residual standard error of 0.47, while exclusion of structurally divergent ILs improved the model to $R^2 = 0.86$ and reduced the error to 0.35. These results supported the reliability of the approach and reinforced the role of cation structure in influencing IL toxicity.

García-Lorenzo et al. (2008) [61] presented a quantitative study to investigate the cytotoxicity of a series of imidazolium-based ionic liquids (ILs) using *human epithelial Caco-2 cells*, with the aim of developing a QSAR model capable of predicting toxicity based on molecular structure. Their methodology combined experimental in vitro testing with a computational modeling framework, applying the Topological Sub-Structural Molecular Design (TOPS-MODE) approach. This method calculates descriptors called spectral moments derived from molecular graphs, where atomic and bond-level properties are encoded mathematically to capture relevant physicochemical characteristics such as hydrophobicity, polarizability, and electronic distribution. The dataset included 15 ILs, and descriptor selection was performed using a genetic algorithm to prioritize parsimony and minimize overfitting. The final model, developed through multiple linear regression,

included three spectral moment descriptors and exhibited excellent internal performance, with R^2 = 0.989, and a cross-validated Q^2 of 0.983. This study illustrates the utility of graph-based molecular descriptors for modeling biological effects and supports the application of interpretable QSAR approaches in the rational design of less toxic ionic liquids.

Torrecilla et al. (2009) [62] developed computational models to predict the toxicity of 153 ionic liquids (ILs) against *IPC-81 leukemia rat cells* and *acetylcholinesterase (AChE)* using principal component analysis (PCA), multiple linear regression (MLR), and artificial neural networks (ANNs). By reducing 46 structural descriptors to 12 principal components (retaining > 94% variance for cations and 88% for anions), the study achieved efficient dimensionality reduction. The multilayer perceptron (MLP) neural network emerged as the top-performing model, demonstrating exceptional predictive accuracy (R^2 > 0.98) and low error rates (MPE < 3.8%) for both toxicity endpoints—significantly outperforming MLR (R^2 = 0.867, MPE = 10.6%) and radial basis networks (R^2 = 0.861, MPE = 11.1%). The model's reliability was further verified through external validation (R^2 > 0.94, MPE < 9%).

Luis et al. (2010) [55] constructed a Quantitative Structure Activity Relationship (QSAR) model to predict the ecotoxicity of ionic liquids (ILs) toward *Vibrio fischeri*. The model was based on a group contribution methodology combined with multiple linear regression (MLR), and it was implemented using Polymath 5.0 software. The study analyzed a dataset of 96 ILs, incorporating diverse structural features across 9 cation families and 17 anions. The final model included 15 descriptors corresponding to different structural components. Statistical evaluation demonstrated a coefficient of determination (R^2) of 0.924 and an adjusted R^2 of 0.911, along with a Low Root Mean Square Deviation (RMSD) of 0.0064. Additionally, 89% of the predicted values deviated by less than 0.10 units from experimental observations. The authors addressed by using all available data points without splitting the data into training and validation sets, which allowed for comprehensive utilization of the available information. Overall, the model demonstrated satisfactory predictive ability based on the presented dataset.

Ismail Hossain et al. (2011) [63] introduced a quantitative structure activity relationship (QSAR) model designed to estimate the ecotoxicity of various ionic liquids (ILs) toward *Daphnia magna*, using a hybrid methodology that combined multiple linear

regression (MLR) with polynomial fitting, implemented using MATLAB software (version 7.8.0.347). The model was built on a dataset consisting of 64 ILs, collected from eight different literature sources, encompassing diverse cationic structures (such as, imidazolium, pyridinium, ammonium and phosphonium) paired with six types of anions. Moreover, the authors used all available data points for model development without splitting the dataset into training and validation sets. While this approach maximized data usage, it potentially compromised the model's ability to generalize to new compounds, as no external validation was performed. In terms of descriptors, the model included 37 input variables derived from 25 initial structural fragments. Nevertheless, the relatively high number of descriptors compared to the limited dataset size raised concerns about the risk of overfitting. Statistically, the model achieved a high coefficient of determination (R^2 = 0.974) and a low standard error (0.028), suggesting strong internal performance.

Fatemi and Izadiyan (2011) [53] developed two robust QSTR models to predict the toxicity of 227 ionic liquids (ILs) based on their molecular structure. The biological endpoint used was the logarithm of EC_{50} values $\log_{10}(EC_{50})$ measured on *IPC-81 leukemia rat cells*, a mammalian in vitro system commonly applied for chemical toxicity screening. The dataset encompassed a structurally diverse range of ILs obtained from the UFT/Merck Ionic Liquids Biological Effects Database, including 94 imidazolium, 53 pyridinium, 23 pyrrolidinium, 22 ammonium, 15 piperidinium, 10 morpholinium, 5 phosphonium, and 5 quinolinium cations, in combination with 25 different anions. Molecular structures of the IL cations were drawn and optimized using ChemSketch and HyperChem, and a comprehensive set of molecular descriptors was calculated via Dragon software. After eliminating redundant and highly correlated variables, 120 descriptors were retained for model development. For the linear model, a genetic algorithm (GA) was used in combination with multiple linear regression (MLR) to identify five key structural descriptors—mainly from the cation—related to symmetry, lipophilicity, and molecular topology. A nonlinear model was also developed using a multilayer perceptron neural network (MLP NN) trained with the Levenberg–Marquardt algorithm, which yielded higher predictive accuracy. The MLR model achieved R^2 = 0.91 with RMSE = 0.30, while the MLP NN reached R^2 = 0.98 on the external test set. The authors emphasized that IL toxicity is significantly influenced by cation symmetry, presence of electron-donating substituents, and anion hydrolysis sensitivity. Internal and external validations were

conducted to confirm the robustness of the models, and the predictive domain was clearly defined. These models not only clarify how structural features impact toxicity, but also offer useful tools for guiding the in silico design of safer ionic liquids, aimed at reducing experimental testing in environmental and toxicological assessments.

Yan et al. (2012) [64] introduced a quantitative structure–activity relationship (QSAR) model to predict the toxicity of 173 ionic liquids (ILs) on IPC-81 leukemia rat cells using a novel class of topological indexes (TIs). The study introduced a generalized TI calculation approach based on atom-level characteristics and structural matrices, accounting for both cation and anion properties, as well as their interactions. The resulting model, built with Multiple Linear Regression (MLR), achieved solid statistical performance with an R^2 of 0.938 and an Average Absolute Error (AAE) of 0.226, supporting its reliability in estimating $\log_{10}(EC_{50})$ values. Validation was conducted using leave-one-out cross-validation and external test sets, confirming consistent predictive capability across diverse IL chemical families. Compared to earlier MLR-based approaches,, this approach demonstrated improved reliability while maintaining mathematical simplicitye. Moreover, the research emphasizes the importance of selecting relevant topological indexes, as the study found that not all calculated TIs contributed significantly to the predictive power of the model. By focusing on the most informative TIs, the authors were able to streamline the model, enhancing its efficiency and interpretability. This aspect is particularly beneficial for researchers looking to apply QSAR methodologies in their work, as it underscores the need for careful feature selection in predictive modeling.

Cho et al. (2013) [65] presented predictive models for ionic liquid toxicity using COSMO-RS derived descriptors based on Linear Free Energy Relationship (LFER) theory, applying a rigorous dataset partitioning strategy: IPC-81 leukemia cells (97 training and 94 test), Vibrio fischeri (69 training and 34 test), and Scenedesmus vacuolatus (45 data points). By employing descriptors such as McGowan volume (V) and hydrogen-bonding basicity (B), calculated from sigma profiles and moments, the authors successfully correlated molecular interactions with observed toxic effects. The models demonstrated strong predictive performance, with R^2 values ranging from 0.762 to 0.838 for IPC-81 and similarly satisfactory results for Vibrio fischeri. However, the prediction for S. vacuolatus was less robust, with an R^2 of 0.776 and a standard error (SE) of 0.825, likely due to the

absence of dataset division for this endpoint. This work significantly advanced computational toxicology for ILs while demonstrating how proper dataset partitioning is crucial for model validation. Although the study provided a valuable framework for future research, it also highlighted the challenges associated with limited data availability for certain biological systems.

Das and Roy (2014) [66] aimed to develop Quantitative Structure-Toxicity Relationship (QSTR) models to predict the ecotoxicity of ionic liquids (ILs) towards the green algae *Scenedesmus vacuolatus* by constructing both classification and regression models. Their methodology incorporated the use of Extended Topochemical Atom (ETA) descriptors, supplemented by non-ETA topological and atom-type fragment descriptors. Classification modeling was conducted via Linear Discriminant Analysis (LDA), while regression modeling employed Multiple Linear Regression (MLR) and Genetic Function Approximation (GFA). The study utilized a dataset of 60 ILs for classification and 49 ILs for regression, selected from the UFT/Merck biological effects database, the dataset was split into training sets (70%) and test sets (30%). Descriptor selection followed a structured approach, focusing on the mechanistic relevance of features such as cation alkyl chain length and anion characteristics like double-bonded oxygen atoms. Statistically, the models achieved strong internal performance, with the regression model reaching an R^2 of 0.883 and a cross-validated Q^2 of 0.829, and classification models demonstrating reasonable discrimination through metrics like Wilk's lambda (Wilk's λ=0.468) and ROC curve analysis. External validation criteria such as the Golbraikh-Tropsha tests were satisfied. Overall, while the work offers a structured and statistically validated approach to ILs toxicity prediction.

Zhao et al. (2014) [32] performed a robust and methodologically sound study aimed at predicting the toxicity of ionic liquids (ILs) through quantitative structure–activity relationship (QSAR) modeling. Recognizing the increasing industrial use of ILs and the growing concerns regarding their environmental impact, the authors sought to address the challenges of large-scale toxicity assessment by combining experimental data with computational approaches. A curated dataset of 100 ILs was used, with toxicity measured against the *IPC-81 leukemia rat cell line* and expressed in $\log_{10}(EC_{50})$ format to allow for better comparison and analysis. The dataset was divided into a training set (80 compounds)

and a test set (20 compounds) to validate the models. Initially, over 200 molecular descriptors (covering constitutional, topological, electrostatic, and geometrical properties) were calculated using the CODESSA software. These were refined to a subset of 91 statistically relevant descriptors through the heuristic method (HM), which formed the basis for developing two QSAR models: a multiple linear regression (MLR) model and a support vector machine (SVM) model. Comparative analysis revealed that the nonlinear SVM model performed better, achieving an R^2 of 0.959 and RMSE of 0.179 on the training set, and an R^2 of 0.958 with RMSE of 0.234 on the test set. These results notably exceeded the performance of the MLR model, highlighting the superior ability of SVM to capture complex structure-toxicity relationships. Overall, this study makes a valuable contribution to predictive toxicology by not only presenting validated models but also emphasizing the crucial role of descriptor selection in ensuring model reliability and interpretability. In addition, it offers an important step toward the design of ILs that balance functional performance with improved environmental safety.

Ma et al. (2015) [67] aimed to develop predictive models to assess the ecotoxicity of ionic liquids (ILs) towards *Vibrio fischeri*. Their approach combined two methods: a linear model constructed using Genetic Function Approximation (GFA) and a nonlinear model based on Least Squares Support Vector Machine (LSSVM). The dataset used in this study included 69 ILs with ecotoxicity data expressed as logEC50 values, retrieved from the UFT/Merck Ionic Liquids Biological Effects Database. It was randomly divided into a training set of 52 ILs and an external test set of 17 ILs. The authors generated a large number of molecular descriptors (more than 2000 for both cations and anions), calculated after 3D molecular optimization using Dragon software. After preliminary screening to remove constant or highly correlated descriptors, five of the most relevant descriptors were selected using GFA. Standard statistical metrics were then used to evaluate model performance. The GFA model achieved an R^2 of 0.893 on the training set and 0.903 on the test set, with RMSE values of 0.402 and 0.388, respectively. The LSSVM model slightly improved upon these results, with an R^2 of 0.910 on the training set and 0.933 on the test set, and corresponding RMSE values of 0.371 and 0.317. Although both models performed well, the nonlinear model demonstrated slightly better predictive performance.

Yan et al. (2015) [57] focused on developing a Quantitative Structure-Activity Relationship (QSAR) model to estimate the toxicity of ionic liquids (ILs) toward the marine bacterium *Vibrio fischeri*, using a topological approach grounded in molecular graph theory. Their methodology centered on using novel topological indices (TIs), previously developed by their research group, as molecular descriptors. The approach involved collecting toxicity data, calculating these specific TIs based on molecular structure characteristics (such as atom types, connectivity and electronegativity), and establishing a predictive relationship using Multiple Linear Regression (MLR). The dataset included 157 ILs, comprising 74 cation types and 22 anions, with toxicity values $\log_{10}(EC_{50})$ collected from 13 publications prior to 2014. To evaluate the model's performance and generalizability, the authors implemented a data split strategy, assigning 125 ILs to the training set and 32 to the test set. The final model exhibited strong statistical indicators, with an R^2 of 0.908 for the full dataset, Q^2 of 0.858 from cross-validation, and an external validation R^2 of 0.890 on the test set. Overall, the study offered a structurally diverse and statistically robust approach.

Ben Ghanem et al. (2017) [44] investigated the use of both linear and nonlinear QSAR modeling approaches to predict the toxicity of ionic liquids (ILs) toward the marine bacterium *Vibrio fischeri*. Their core principle was to utilize σ-profile descriptors, derived from COSMO-RS theory, as molecular representations for the ionic liquids, aiming to establish a link between the structural characteristics encoded by these descriptors and the observed toxicity. Two modeling techniques were applied: Multiple Linear Regression (MLR) and a Multi-Layer Perceptron (MLP) neural network.The dataset compiled consisted of 110 ILs with experimental $\log EC_{50}$ toxicity values collected from various literature sources. The training and test sets were split in an 80:20. In terms of predictive performance, the MLR model achieved a high coefficient of determination ($R^2 = 0.906$) and a cross-validated Q^2 of 0.907, with a mean squared error of 0.371, indicating good internal consistency. The MLP model, using the same descriptors, outperformed MLR by achieving R^2 values between 0.961 and 0.979 across the training, validation, and test sets, along with substantially lower RMSE values (as low as 0.123). These results highlight the superior ability of neural networks to capture nonlinear relationships in toxicity data. The analysis confirmed the relevance of COSMO-RS descriptors and confirmed the advantage of nonlinear modeling techniques in predicting ILs toxicity.

Sosnowska et al. (2017) [68] investigated the comparative performance of global and local Quantitative Structure-Activity Relationship (QSAR) models for predicting the toxicity of ionic liquids (ILs) toward the *IPC-81 rat leukemia* cell line. Multiple linear regression (MLR) was employed as the modeling algorithm. The methodology involved constructing six local QSAR models, each specific to a major IL cation class (ammonium, imidazolium, morpholinium, piperidinium, pyridinium, and pyrrolidinium), in addition to a single global model encompassing all available IL data. The dataset comprised 304 ILs sourced from the literature, including 104 imidazolium, 67 pyridinium, and 35 ammonium-based compounds, among others. For both global and local models, datasets were divided into training sets (70–75%) and external validation sets (25–30%). Molecular descriptors were calculated using Dragon software, focusing on five categories considered relevant to IL toxicity: WHIM descriptors, ring descriptors, functional group counts, topological indices, and constitutional descriptors. Descriptor selection for each model was performed using a Genetic Algorithm (GA) implemented in QSARINS software. The local models typically included three to four descriptors each, while the global model used five. From a statistical standpoint, the global model achieved a coefficient of determination (R^2) of 0.77, a cross-validated Q^2 of 0.76, and a root mean square error of prediction (RMSEP) of 0.43. Local models generally exhibited better internal fits within their respective chemical families-for instance, the ammonium-based model reached an R^2 of 0.90 and the imidazolium model 0.84. Model robustness was further assessed through Y-randomization tests and applicability domain analysis using Williams plots. In conclusion, while local models offered improved mechanistic specificity, the global model demonstrated comparable predictive capability with greater efficiency, suggesting its practical utility for general screening tasks.

Cao et al. (2018) [69] introduced a modeling framework to predict the toxicity of ionic liquids (ILs) against the *IPC-81 leukemia rat* cell line by comparing three approaches: Multiple Linear Regression (MLR), Support Vector Machine (SVM), and Extreme Learning Machine (ELM), all using a common set of descriptors derived from quantum chemical calculations. Specifically, the study employed descriptors based on electrostatic potential surface area (S_{EP}) and charge distribution area ($S_{\sigma\text{-profile}}$). S_{EP} descriptors represent the surface area corresponding to different ranges of electrostatic potential values around the ion, calculated using Multiwfn software, and $S_{\sigma\text{-profile}}$ descriptors represent the surface

area associated with different ranges of surface screening charge density, obtained from COSMO calculations and processed with MATLAB software. The dataset comprised 119 ILs with experimental $\log_{10}(EC_{50})$ values sourced from the literature, randomly split into a training set of 99 and a test set of 20 compounds. All three QSAR models demonstrated robust predictive capability, but comparison based on the total root mean square error (RMSE) clearly favored the Extreme Learning Machine (ELM), which achieved the lowest RMSE of 0.157, compared to 0.189 for the Support Vector Machine (SVM) and 0.239 for the Multiple Linear Regression (MLR) model. This performance indicates that ELM was the most accurate model overall. Consistently, ELM also delivered the highest coefficient of determination ($R^2 = 0.969$), followed by SVM ($R^2 = 0.949$) and MLR ($R^2 = 0.930$). The external predictive correlation (Q^2) further supported this result, with ELM reaching 0.940, while SVM and MLR followed with values of 0.874 and 0.849, respectively. The study concluded that the combination of quantum chemistry descriptors and machine learning algorithms, particularly ELM, provided a robust and accurate approach for predicting IL toxicity.

Yan et al. (2019) [70] established a novel approach for developing a Quantitative Structure–Toxicity Relationship (QSTR) model to predict the eco-toxicity of ionic liquids (ILs) towards *the IPC-81 leukemia rat cell line*. The central principle guiding this work was to address a limitation observed in earlier QSTR studies, which often neglected the interaction between the cation and anion of ionic liquids when calculating molecular descriptors. To overcome this, a new set of descriptors, termed "norm indexes," was proposed and calculated from constructed atomic distribution matrices. These descriptors captured atomic spatial arrangements, electronegativity differences, atomic weights, and electronic shell distributions, thereby offering a more comprehensive representation of IL molecular characteristics. Specifically, the authors employed 21 distinct atomic distribution matrices from which 24 norm-based descriptors were generated and statistically evaluated for relevance. A Multiple Linear Regression (MLR) model was subsequently constructed. The modeling relied on a dataset comprising 173 ILs sourced from a previous study by Ranke et al. (2007), which was randomly divided into a training set of 138 compounds and a test set of 35 compounds. The developed model demonstrated

strong statistical performance, achieving an R^2 of 0.954, a root mean square error (RMSE) of 0.241, and a Q^2 of 0.935, indicating strong predictive power and stability.

Zhu et al. (2019) [71] explored the prediction of ionic liquids (ILs) toxicity against *acetylcholinesterase (AChE)* by constructing two QSAR models: Multiple Linear Regression (MLR), and Extreme Learning Machine (ELM). Both models were built using 11 descriptors, 8 from electrostatic potential surface area (S_{EP}) and 3 from screening charge density distribution area (Sσ), calculated through COSMO-RS and quantum chemistry techniques. The dataset comprised 160 ILs, split into an 80% training set (128 compounds) and a 20% test set (32 compounds). The statistical evaluation showed that the ELM model outperformed the MLR model, with higher R^2 values (0.964 vs. 0.917) and lower errors, as reflected by RMSE (0.096 vs. 0.145) and AARD% (3.29% vs. 6.35%). These results suggest that ELM has a superior ability to capture nonlinear relationships associated with ILs toxicity.

Wu et al. (2020) [72] established an interpretable QSTR model to predict the toxicity of ionic liquids (ILs) against the leukemia rat cell line IPC-81. Their method focused on simple 2D structural features closely related to the chemical structure of ILs. The model was built using multiple linear regression (MLR) and relied on a set of 33 descriptors chosen based on empirical knowledge and literature. The dataset used in this study included 304 ILs, which was split into a training set of 228 compounds and a test set of 76. Additionally, 11 ILs from external sources were used for independent validation. The model demonstrated satisfactory performance. On the training set (with 10-fold cross-validation), it achieved an R of 0.88 and an RMSE of 0.51, while the independent test set showed an R of 0.90 and an RMSE of 0.47. External validation also confirmed its robustness, with an R of 0.94 and an RMSE of 0.27. This work highlights the balance between simplicity and precision in QSTR modeling.

Kang et al. (2020) [73] aimed to develop a predictive model to estimate the ecotoxicity of ionic liquids (ILs) on *Vibrio fischeri*. They employed machine learning techniques, particularly the Extreme Learning Machine (ELM), using electrostatic potential surface area (SEP) descriptors obtained through quantum calculations for both cations and anions. To identify the most important variables, a stepwise multiple linear regression (MLR) approach was used. The dataset employed in the study consisted of 142 ILs, randomly

divided into a training set of 113 compounds (80%) and a test set of 29 compounds (20%). In terms of performance, the ELM model achieved strong statistical results. The training set produced an R^2 of 0.9308 and an RMSE of 0.3387, while the test set showed an R^2 of 0.9035 with an RMSE of 0.2718. When the entire dataset was evaluated, the model maintained similar accuracy, confirming its ability to capture the relationship between molecular structure and toxicity.

Peng and Picchioni (2020) [74] designed a QSAR modeling approach to predict the toxicity of ionic liquids (ILs) toward the *IPC-81 leukemia rat cell line*, using a more efficient method based on descriptors calculated through the Group Contribution COSMO (GC-COSMO) technique. To generate these descriptors, they applied two segmentation methods to the σ-profiles: one divided the charge densities into wider regions, while the other used a finer scale, together creating a large number of possible descriptors, and employing the Enhanced Replacement Method (ERM) to select the optimal subset, which selected 7 and 9 statistically significant descriptors for the respective models. The authors developed predictive models using both Multiple Linear Regression (MLR), and Multilayer Perceptron (MLP) neural networks. A dataset of 127 ILs was compiled for model development, split into training (93 compounds), test (19 compounds), and validation (15 compounds) sets. The results showed that the MLP model performed better, achieving an R^2 of 0.975 for the training set and 0.938 for the test set, with external validation also confirming strong predictive ability. These results indicate that the developed QSAR model, utilizing GC-COSMO derived descriptors, provides a satisfactory prediction of the toxicity of the studied ionic liquids against the *IPC-81 cell line*.

Wang et al. (2021) [30] utilized machine learning (ML) methodologies to predict the toxicity of ionic liquids (ILs) towards the leukemia rat cell line (IPC-81). The investigation incorporated two ML algorithms, specifically feedforward neural networks (FNN) and support vector machines (SVM). The research compiled a dataset encompassing 355 ILs, which was randomly divided into a training set consisting of 284 ILs (comprising 80% of the data) and a test set containing the remaining 71 ILs (constituting 20%). The authors characterized the ionic liquids by employing molecular descriptors derived from their SMILES representations. Each IL was regarded as a singular molecular entity, and its

SMILES representation was processed utilizing the RDKit cheminformatics toolkit to generate relevant features [25]. The study implemented a feature extraction algorithm that yielded 42 structural descriptors. These descriptors facilitated the translation of the chemical structure of each IL into a machine-readable numerical format, thereby enabling the models to analyze and correlate them with toxicity values. In the comparative analysis of the two models, SVM consistently demonstrated superior performance relative to FNN, attaining elevated determination coefficients (R^2 of 0.9289 for training and 0.9202 for testing) and diminished errors (RMSE of 0.2787 for training and 0.3204 for testing). Although FNN exhibited a commendable level of performance, it was marginally less precise, particularly concerning the test set, (with an R^2 of 0.9227 on the training set and 0.8917 on the test set) accompanied by corresponding RMSE values of 0.2906 for the training set and 0.3732 for the test set. The authors concluded that both ML models, with particular emphasis on SVM, provided dependable predictions for IL toxicity against IPC-81 and could be valuable tools for computational screening and design of safer ILs.

Chipofya et al. (2022) [31] designed a probabilistic deep learning framework for predicting the toxicity of ionic liquids. In this method, cheminformatics tools were integrated with an innovative machine learning concept, Deep Kernel Learning (DKL), that merged representations from neural networks with Gaussian processes in a probabilitic form. Of equal importance to improving accuracy was the consideration that their modeling approach would enable a better handling of uncertainty in predictions of toxicity. The study was based on a dataset of 155 ionic liquids with toxicity data reported in the literature against the *IPC-81*, these were subsequently divided into 140 compounds used for training and validation and 15 compounds reserved for testing. The descriptors used in this study were derived from two sources. Ten basic molecular descriptors, which included atom counts and molecular weight, were generated using RDKit software, rest came from Mol2vec software. This resulted in a total of 310 features used as input for the model. The model achieved an RMSE of 0.228 and an R^2 of 0.943 on the entirety of the dataset. Furthermore, on the training and validation subsets, the model exhibited robust performance, achieving an R^2 of 0.940 and an RMSE of 0.233.

Tabaaza et al. (2023) [75] explored the use of machine learning to predict the cytotoxicity of ionic liquids (ILs). They implemented a detailed approach using five

supervised machine learning models: decision tree, random forest, extra tree regression, gradient boosting, and extreme gradient boosting (XGBoost). The main objective of their work was to combine in silico descriptors, specifically VolSurf principal properties, with supervised algorithms to develop predictive quantitative structure–property relationship (QSPR) models. The choice of descriptors was based on earlier research by Paternò et al. [76], and those calculated with VolSurf+. To simplify the molecular information, the authors used principal component analysis (PCA) to reduce the various molecular interaction parameters to nine principal properties (PPs): five describing the cationic structure and four describing the anionic structure. These PPs reflect important molecular properties such as solubility, hydrophobicity, size, shape, flexibility, and hydrogen bonding ability. The study was based on a dataset of cytotoxicity values for 243 ionic liquids tested against *the IPC-81 leukemia rat cell line*, with toxicity expressed as the logarithm of the half-maximal effective concentration ($logEC_{50}$). This dataset was divided into a training set (85%, 206 compounds) and a test set (15%, 37 compounds) using a holdout cross-validation approach to ensure proper model development and assessment. In terms of performance, tree-based models, and especially XGBoost, delivered the best predictive results. Although all models showed good accuracy on the training data, XGBoost performed particularly well on the test set, reaching an R^2 of 0.79 and an RMSE of 0.45, along with the lowest Akaike Information Criterion (AIC) of 46.50. These findings demonstrated its strong ability to generalize to new data and confirmed that this machine learning approach offers a solid framework for predicting the toxicity of ionic liquids.

Danush and Dutta (2023) [77] aimed to create predictive mathematical models for the toxicity (log_{EC50}) of ionic liquids (ILs) against *the IPC-81 leukemia rat cell line*, employing multiple linear regression (MLR), elastic net regularized MLR (MLR-EN), and artificial neural networks (ANN). The study utilized a dataset of 355 ILs, with (log_{EC50}) as the target variable and adopted an 80:20 train-test split repeated 10 times for validation. To generate input variables for the models, the authors used a group contribution method (GCM), transforming the molecular structures of ILs into numerical features. A total of 51 molecular descriptors were selected, and feature extraction was performed using RDKit software, while MATLAB software was used for model building and statistical analysis. Among the models tested, MLR-EN consistently outperformed the others, balancing

prediction accuracy and model simplicity. This model achieved a root mean square error (RMSE) of 0.4676 and a mean absolute error (MAE) of 0.3674 across the entire dataset. Furthermore, Akaike Information Criterion (AIC) analysis supported its superiority, with MLR-EN displaying the lowest AIC value (−418.4), indicating a better trade-off between fit and complexity compared to ANN (416.7) and standard MLR (−391.8). In terms of predictive power, MLR-EN again showed clear advantages. It reached an average R^2 of 0.7971 on the full dataset, surpassing both ANN (0.7671) and standard MLR (0.7809), The study concluded that the MLR approach incorporating Elastic Net regularization provided the most reliable and accurate framework among those tested for predicting the toxicity of ILs using GCM-based descriptors.

Fan et al. (2023) [78] developed a deep learning model to predict the toxicity of ionic liquids (ILs) towards the *IPC-81 leukemia rat cell line*. They built a Deep Convolutional Neural Network (DCNN) and used two complementary types of molecular descriptors: MACCS fingerprints and sigma profiles. The MACCS fingerprints provided information about the structure of the ILs, while the sigma profiles, derived from quantum chemical calculations, described their electronic properties. In total, 218 descriptors were used for each IL, including 167 MACCS fingerprints and 51 sigma profile values. To optimize the DCNN's structure and learning settings, the authors applied a combination of Bayesian optimization and local search methods. This helped them achieve a good balance between prediction performance and computational efficiency. To improve the reliability of the model, they also used 10-fold cross-validation during the training process. The dataset included experimental $\log_{10}(EC_{50})$ toxicity values for 155 ILs, which were split into 80% for training and 20% for testing. The results showed strong predictive ability, with an R^2 of 0.972 for the training set and 0.965 for the test set. The model also had low mean squared error (MSE) values of 0.025 and 0.027 for training and testing, respectively. Based on these results, the authors concluded that the DCNN, using the combined MACCS and sigma profile descriptors, offered a reliable and accurate approach for predicting the toxicity of ILs against the IPC-81 cell line.

Fan et al. (2024) [79] employed machine learning models to predict the toxicity of ionic liquids (ILs) towards the bacterium *Vibrio fischeri*. To achieve this, they used two ensemble learning methods: Random Forest (RF) and eXtreme Gradient Boosting

(XGBoost). These models were built using a carefully selected set of molecular descriptors derived from sigma profiles. The sigma profiles were calculated through COSMO-SAC following quantum chemical optimization. In total, 10 descriptors were used. The dataset employed in this study consisted of 145 ionic liquids, collected from 17 different studies, and was divided into a training set (80%) and a test set (20%). The results demonstrated that the XGBoost model achieved better predictive performance than the Random Forest model. On the test set, the RF model obtained an R^2 of 0.933, with a mean absolute error (MAE) of 0.195, mean squared error (MSE) of 0.09, root mean squared error (RMSE) of 0.3, and an average absolute relative deviation (AARD) of 12.355%. In comparison, the XGBoost model performed better across all metrics, with an R^2 of 0.957, a lower MAE of 0.179, MSE of 0.057, RMSE of 0.24, and an AARD of 9.681%. These findings indicate that XGBoost provided more accurate and reliable predictions.

I.4 Conclusion

This chapter has highlighted the diversity of ionic liquids (ILs), whose unique structures and tunable physicochemical properties offer wide potential across many scientific and industrial fields. However, their possible toxicity remains a growing concern that requires careful attention. The increasing research efforts to predict IL toxicity reflect the need to better understand and control these risks. Through the review of studies published in the past years, important progress has been identified in the development of QSTR/QSAR models, even though challenges still exist, particularly in ensuring their applicability to a wide range of ILs. Overall, this chapter has not only provided an overview of IL fundamentals and their applications but also underlined the importance of predictive models in evaluating toxicity. This sets a solid foundation for the next stages of this thesis, which will further explore and contribute to the advancement of reliable tools for assessing and minimizing the potential hazards associated with ionic liquids.

Chapter II
Overview of modeling with machine and deep learning

Chapter II:
Overview of modeling with machine and deep learning

II.1. Introduction

Predictive modeling has become a cornerstone of modern chemical and toxicological research, providing efficient and cost-effective alternatives to traditional experimental methods. By identifying and quantifying the relationship between chemical structure and biological activity, such models enable the estimation of toxicity for new or untested compounds, which is particularly valuable in regulatory, environmental, and pharmaceutical contexts. Machine learning (ML) and deep learning (DL) approaches have significantly advanced this field due to their ability to capture complex, non-linear relationships without requiring explicit assumptions about data distribution. ML techniques such as Support Vector Machines (SVM) can extract meaningful patterns from structured data like molecular descriptors and are known for their robustness in high-dimensional spaces and effectiveness in regression tasks. SVMs operate by finding an optimal hyperplane that maximizes the margin between data points, making them highly suitable for structured chemical data. while DL models, particularly neural networks, offer the additional capability of hierarchical feature learning, which is especially useful when dealing with large, high-dimensional datasets. These techniques are increasingly applied in the development of Quantitative Structure-Toxicity Relationship (QSTR) models to predict toxicological endpoints. In doing so, they reduce reliance on in vivo testing and support safer and more sustainable chemical design. The motivation behind this study stems from the need to accurately predict the toxicity of ionic liquids (ILs), which, despite their promising industrial applications, raise significant environmental and safety concerns. By combining ML methods with optimization algorithms, this research aims to enhance predictive accuracy and model generalizability, offering a robust framework for evaluating chemical toxicity in the absence of experimental data[80].

The increasing reliance on machine learning (ML) and deep learning (DL) in toxicology stems from the urgent need to assess chemical safety more efficiently and ethically. Traditional toxicity testing methods, such as animal experiments and in vitro assays, are often time-consuming, expensive, and constrained by ethical concerns. As the volume of chemical data grows and regulatory demands tighten, predictive modeling offers a powerful alternative by leveraging computational approaches to forecast toxic

effects based on chemical structure and biological activity. ML and DL techniques enable the extraction of complex, nonlinear patterns from vast datasets, improving the accuracy and scalability of toxicity predictions. This shift not only accelerates risk assessment but also supports the development of safer chemicals and pharmaceutical compounds[81].

When comparing traditional machine learning (ML) approaches to deep learning (DL) in the context of chemical toxicity prediction, several key differences emerge. Traditional ML methods, such as random forests, support vector machines, and gradient boosting, often rely heavily on manual feature engineering, where domain knowledge is used to craft molecular descriptors or fingerprints that represent chemical structures. While this can yield interpretable models with relatively small datasets, it also limits the ability to capture complex, nonlinear relationships. In contrast, DL models, particularly neural networks like convolutional and graph-based architectures, can automatically learn hierarchical representations directly from raw data, such as molecular graphs or SMILES strings. However, this power comes with higher data demands and reduced interpretability, making DL more suitable for large-scale datasets but challenging to validate and explain. Ultimately, the choice between ML and DL depends on the specific application, data availability, and the need for transparency in model prediction [82].

II.2. Machine and Deep Learning Models

Machine learning (ML) has emerged as a powerful tool in computational chemistry and toxicology, offering data-driven approaches for predicting chemical properties such as toxicity, solubility, and bioactivity. Among the widely used ML methods, Support Vector Machines (SVM) are particularly effective for regression and classification tasks involving high-dimensional data, making them well-suited for modeling relationships between molecular descriptors and toxicity endpoints. Random Forests (RF), an ensemble learning technique based on decision trees, are valued for their robustness against overfitting and ability to handle non-linear relationships and feature interactions. Another commonly employed algorithm is k-Nearest Neighbors (k-NN), which makes predictions based on similarity measures but may struggle with high-dimensional datasets. Gradient Boosting Machines (GBM) and XGBoost have also gained popularity due to their ability to build highly accurate models by iteratively correcting errors from previous learners. These methods rely heavily on well-curated input features, such as molecular descriptors, and often require tuning of hyperparameters to achieve optimal performance. By leveraging

these techniques, researchers can build reliable Quantitative Structure-Toxicity Relationship (QSTR) models that support rapid and efficient assessment of chemical safety[83].

Deep learning (DL), a subfield of machine learning, has shown remarkable potential in the field of chemical and toxicological modeling due to its ability to automatically learn complex, non-linear relationships from large datasets. One of the most fundamental deep learning architectures is the Feedforward Neural Network (FNN), which consists of input, hidden, and output layers where information flows in a single direction. FNNs are commonly used in toxicity prediction tasks involving molecular descriptors or fingerprints. More advanced architectures like Convolutional Neural Networks (CNNs), originally developed for image recognition, have been adapted for chemical data by representing molecules as 2D grids or graphs, enabling the model to learn spatial features. Additionally, Recurrent Neural Networks (RNNs) and their improved variants like Long Short-Term Memory (LSTM) networks have been applied to sequential chemical data, such as SMILES strings, to model temporal or order-dependent features. Recent developments include Graph Neural Networks (GNNs), which are particularly well-suited for chemical modeling as they can directly process molecular graph structures, capturing the relationships between atoms and bonds more naturally than traditional descriptors. These deep learning methods typically require large datasets and substantial computational resources, but they offer powerful capabilities for extracting high-level features and improving prediction accuracy in chemical property modeling, including toxicity assessment[83].

ML models, rely on structured input features and often require manual feature engineering. They work well with smaller datasets and offer better interpretability. In contrast, DL models, automatically learn features from raw or minimally processed data. DL excels at handling complex, high-dimensional data but typically requires large datasets and more computational power. While ML is faster to train and easier to interpret, DL offers superior performance in capturing deep patternsespecially in tasks involving unstructured data like images or molecular graphs.

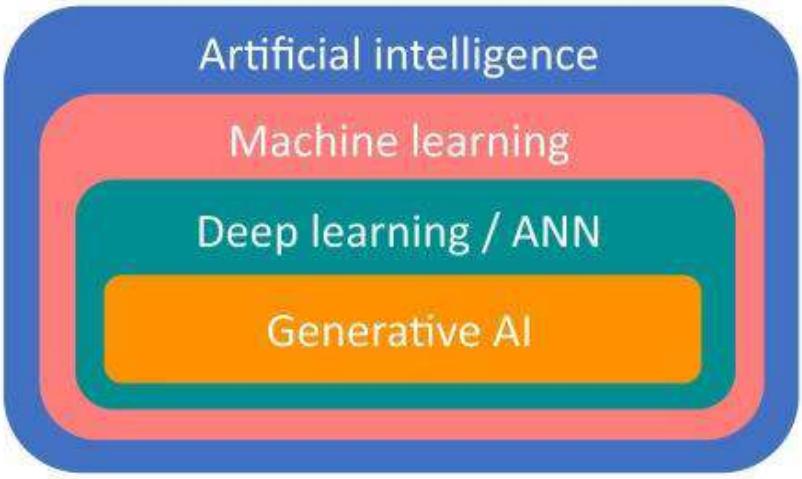


Figure II.1. Illustration of the hierarchy and interconnections among artificial intelligence (AI), machine learning (ML), deep learning/artificial neural networks (ANN), and generative AI [83].

II.3. Types of Machine Learning

Machine learning (ML) techniques are generally grouped into three main categories: supervised learning, unsupervised learning, and reinforcement learning (as shown in Fig. 1). Supervised learning is a task-oriented method where models learn to map inputs to outputs using labeled data, it referred to as ground truthduring training. In contrast, unsupervised learning involves data-driven techniques that rely on unlabeled data to uncover hidden patterns within datasets. Reinforcement learning differs significantly from both, as it is not defined by the presence or absence of labels. Instead, it is based on an agent (e.g., a Go player) interacting with an environment (e.g., a game board), aiming to learn optimal behaviors. While both supervised and unsupervised methods typically optimize a loss or objective function to assess performance, reinforcement learning aims to maximize cumulative rewards over time.

Another category, known as semi-supervised learning, falls between supervised and unsupervised learning by incorporating both labeled and unlabeled data, it's usually with a much larger portion of unlabeled dataduring training. In the field of mechanical material design, supervised learning is currently the most widely adopted due to the higher accuracy and maturity of its tools compared to those from other categories. Given the fast-paced advancements in ML, the methods discussed here represent a selective overview of those considered applicable to mechanical material design. Finally, it's worth noting that

the most basic ML techniques, those that do not involve deep, multilayer architecturesare referred to as classical machine learning algorithms[84].

II.3.1. Supervised Learning (SL)

Supervised Learning is a foundational method in machine learning where the model is trained on a labeled dataset, meaning that each input data point is paired with a correct output or label. The primary goal is for the model to learn the relationship or function that maps inputs to outputs. As training progresses, the model adjusts its parameters to minimize the error between its predictions and the actual labels. This method excels in scenarios where historical data with known outcomes is available, such as in email spam classification, fraud detection, and medical diagnosis. Its effectiveness, however, heavily relies on the quality and quantity of labeled data, which can be expensive and time-consuming to obtain[84].

II.3.2. Unsupervised Learning (UL)

Unsupervised Learning operates without labeled data, aiming instead to discover hidden patterns, structures, or relationships within the data. The algorithms try to group similar data points (clustering), reduce the dimensionality of data (dimensionality reduction), or find associations between features. For instance, clustering algorithms like K-Means can identify distinct customer segments in marketing, while principal component analysis (PCA) can simplify complex datasets for visualization or preprocessing. Because it does not require labeled data, unsupervised learning is ideal for exploratory data analysis, anomaly detection, and data compression. However, interpreting the results can be more challenging since there is no explicit output to validate against[85].

II.3.3. Reinforcement Learning (RL)

Reinforcement Learning is a type of machine learning where an agent learns to make decisions by interacting with an environment. The agent performs actions and receives feedback in the form of rewards or penalties, allowing it to learn a policy, a strategy for choosing actions that maximize cumulative rewards over time. Unlike supervised learning, RL does not require labeled input/output pairs but instead relies on delayed feedback from the environment. It is particularly effective for tasks involving sequential decision-making and dynamic environments, such as game playing (e.g., AlphaGo), robotics, and autonomous driving. Reinforcement learning often employs

algorithms like Q-learning and policy gradients, and may incorporate deep learning in the form of deep reinforcement learning for handling complex, high-dimensional problems.

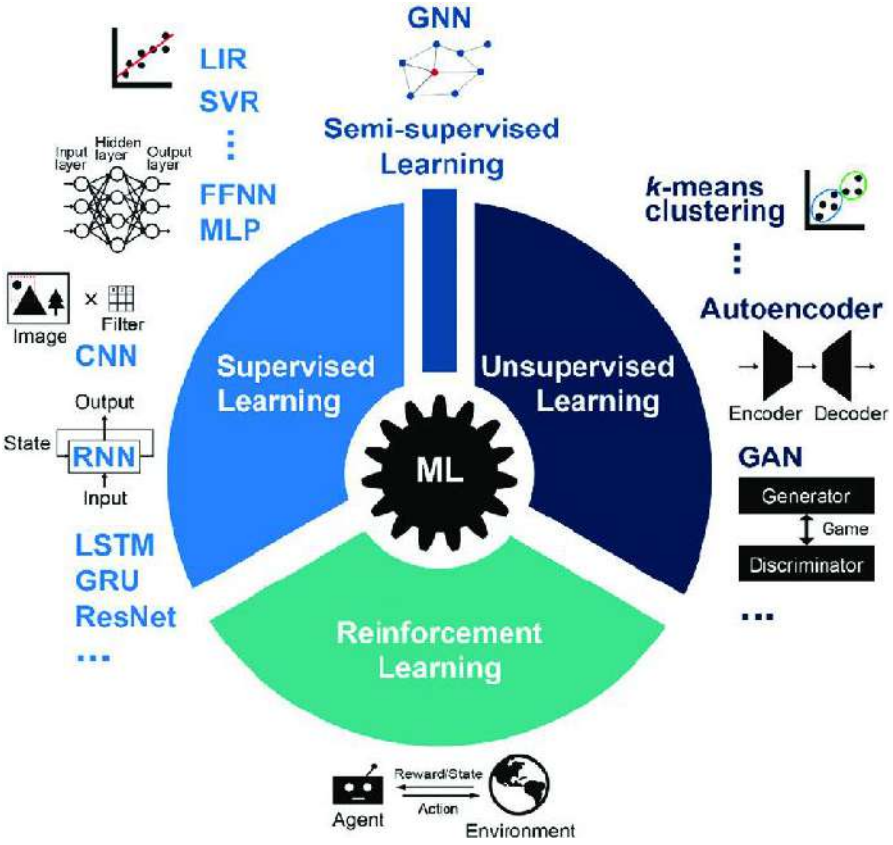


Figure II.2. overview of machine learning (ML) methods [86].

Figure 2 presents an overview of machine learning (ML) methods, which are primarily divided into three main categories: supervised learning, unsupervised learning, and reinforcement learning. In supervised learning, models such as linear regression (LIR), support vector regression (SVR), feedforward neural networks (FFNNs), multilayer perceptrons (MLPs), convolutional neural networks (CNNs), and recurrent neural networks (RNNs) are commonly employed. Unsupervised learning, on the other hand, often utilizes techniques like k-means clustering, autoencoders, and generative adversarial networks (GANs). Reinforcement learning is characterized by a continuous feedback loop between an agent and its environment. The key distinction between supervised and unsupervised learning lies in the presence or absence of labeled training data. Bridging these two is semi-supervised learning, which incorporates a small portion of labeled data alongside a larger set of unlabeled data for training. It's important to note that some ML models are versatile and not confined to just one category. For example, graph neural networks (GNNs), though frequently applied in semi-supervised learning, are also effective in supervised and unsupervised tasks involving graph-based data[85].

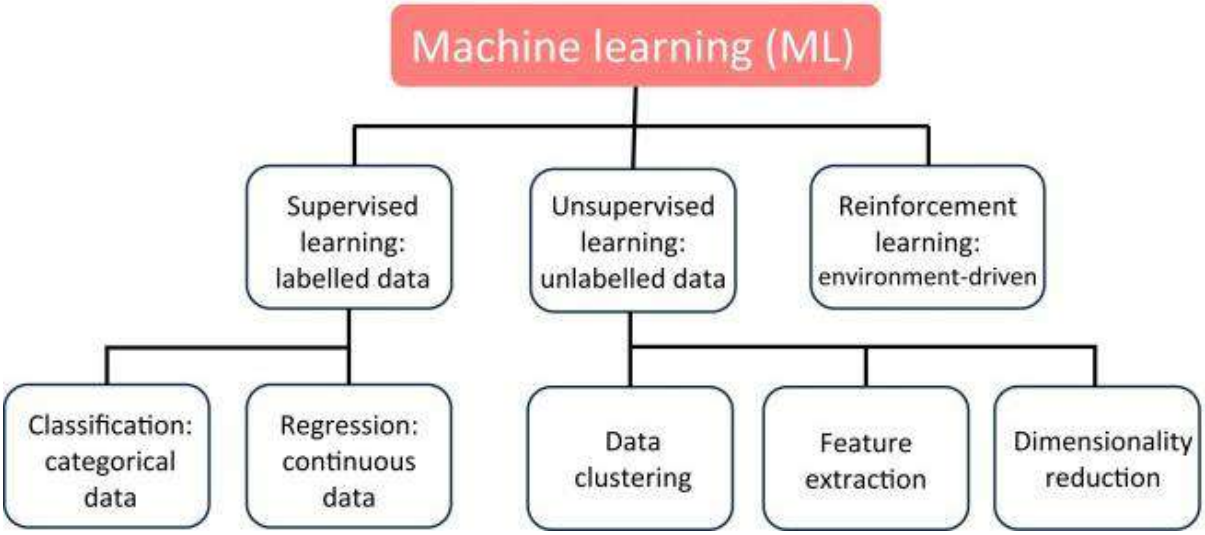


Figure II.3. Illustration of the various branches and practical applications of ML [87].

II.3.4. Semi-supervised Learning (SeL)

Semi-supervised Learning bridges the gap between supervised and unsupervised learning by leveraging a small amount of labeled data with a large amount of unlabeled data. This approach aims to enhance learning accuracy and generalization by using the structure of the unlabeled data to guide the training process. It is especially powerful in situations where acquiring labeled data is costly, such as in medical imaging or speech recognition. Algorithms in this category use techniques like self-training, co-training, and graph-based methods to propagate label information to the unlabeled examples. Semi-supervised learning can significantly reduce the need for manual labeling while still achieving performance close to fully supervised methods[88].

II.4. Specific Machine Learning Algorithms Employed

This section provides a detailed theoretical background for the specific machine learning algorithms utilized in this thesis, including Support Vector Machines (SVM) and ensemble methods like Random Forest (RF) and eXtreme Gradient Boosting (XGBOOST).

II.4.1. Support Vector Machine (SVM)

Support Vector Machine (SVM) is a novel machine learning technique introduced by Vapnik [89] in 1995 [90] it is based on the statistical learning theory and the structural risk minimization principle to overcome the drawbacks of the traditional Artificial Neural Network (ANN) [89, 91]. Due to its robustness and accuracy, SVM proves to be a dependable algorithm suitable for both classification and regression tasks [92].

SVM regression hypothetically minimizes predictable errors in the learning process and decreases the problem of over-fitting encountered in machine learning. This often results in superior performance of SVM compared to alternative learning methods [92].

For a given data set: $\{(x_i, y_i)_i = 1, 2, . . ., n\}$, where x_i and y_i are the input and output sets, respectively, n is the overall number of the dataset . SVM predicts the target values by utilizing the regression function as follows [89, 91, 93]:

$$F(x) = \omega.\varphi(x) + b \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ (II.1)$$

Where φ (x) is the kernel function, w is the weight vector, b is the bias term That can be computed by minimizing the cost function, as given below. The coefficients w and b were computed by minimizing the risk function, as given below [91, 94]:

$$R\,(F) = \frac{1}{2}\|\omega^2\| + C\frac{1}{n}\sum_{i=1}^{n} L_\varepsilon\,(y_{i\,,}F_i) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (II.2)$$

Where:

$$L_\varepsilon(y_{i\,,}F_i) = \begin{cases} 0 & if\ \ |y_i - F_i| \le \varepsilon \\ |y_i - F_i| - \varepsilon & otherwise \end{cases}$$

The penalty parameter C is used to identify the trade-off between function intricacy and losses, the parameter ε is the loss function which represents the range between the actual values and the regression function, $L_\varepsilon(y_i, F_i)$ is called the loss function, where y_i is the true or observed value, and F_i is the predicted or estimated value.

The equation (2) can be changed to the following constrained form [89, 91, 93]:

$$\text{minimize } \frac{1}{2}\|\omega\|^2 + c\sum_{i=1}^{k}(\xi_i + \xi_i^*) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ (II.3)$$

$$Subject\ to: \begin{cases} y_i - (\omega\,.\varphi(x_i) + b) \le \varepsilon + \xi_i^* \\ (\omega.\varphi(x_{\ i}) + b) - y_i \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0, \qquad i = 1,2, \ldots .\, n \end{cases}$$

Where ε and ξ^* are two positive slack variables.

The optimization problem in (4) can be solved more easily in its dual formulation by using Lagrangian multipliers, this is a common technique in convex optimization. Expressing this problem in its dual formulation, it can be written as follows [91, 93, 94]:

$$Maximize \left\{-\frac{1}{2} + \sum_{i,j}^{n}(\alpha_i , \alpha_i^*)(\alpha_j , \alpha_j^*).\varphi(x_j)) - \varepsilon \sum_{i=1}^{n}(\alpha_i , \alpha_i^*) + \sum_{i=1}^{n} y_i(\alpha_i , \alpha_i^*)\right\} \text{..........} \quad (II.4)$$

$$Subject\ to \begin{cases} \sum_{i=1}^{n}(\alpha_i - \alpha_i^*) = 0 \\ 0 \le \alpha_i \le C \quad i = 1, 2, 3, \dots\dots\dots, n \\ 0 \le \alpha_i \le C \quad i = 1, 2, 3, \dots\dots\dots, n \end{cases}$$

The SVM function can be acquired by solving the dual maximization problem in (4), where α_i and α^*_i represent nonlinear Lagrangian multipliers.

$$F(x, \alpha_{i,}\alpha_i^*) = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)(\varphi(x_i) . \varphi(x_j)) + b \text{................} \quad (II.5)$$

The mapping function is represented by the vector inner product ($\varphi(x_i)$.($\varphi(x_j)$)), and it can be replaced by a kernel function K(x_i, x_j) as illustrated in the equation (6)

$$F(x, \alpha_{i,}\alpha_i^*) = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)\, k(x_i, x_j) + b \text{................} \quad (II.6)$$

II.4.2. Ensemble Learning: Random Forest (RF)

The Random Forest algorithm was developed by Leo Breiman and Adele Cutler and introduced to the scientific community in the early 2000s (specifically, Breiman's seminal paper on Random Forests was published in 2001) [95]. It built upon earlier work on decision trees and ensemble methods, particularly "bagging" (Bootstrap Aggregating), also introduced by Breiman in 1996 [96]. Random Forest quickly gained popularity due to its strong predictive performance, ease of use, and robustness to various data characteristics.

Random Forest is a versatile, non-linear ensemble learning method predominantly used for classification and regression tasks. Its fundamental principle involves constructing a large number of individual decision trees during the training phase and combining their outputs [95] [97]The base learner in a Random Forest is typically a decision tree, which partitions the data space into a set of rectangular regions. For regression, the prediction within each region is the average of the target values of the training observations falling into that region [98].

Two key randomization techniques contribute significantly to the power and robustness of Random Forest:

1. **Bagging (Bootstrap Aggregating):**

Instead of training a single decision tree on the entire dataset, Random Forest creates multiple bootstrapped samples from the original training data. Bootstrapping involves randomly drawing samples with replacement from the original dataset. Each bootstrapped sample is typically the same size as the original dataset, meaning some original samples may appear multiple times in a subset, while others may not appear at all (these are called "out-of-bag" (OOB) samples and can be used for internal validation, providing an unbiased estimate of the generalization error) [96].

A separate decision tree is then trained on each of these bootstrapped subsets. This parallel training helps to reduce variance and improve the stability of the model.

2. **Random Feature Subspace (Feature Randomness):**

When growing each individual decision tree, at every split node, only a random subset of the available features is considered as candidates for the best split. This means that even if one feature is very strong, other features will also get a chance to be selected in different trees, significantly decorrelating the trees and preventing any single feature from dominating the entire forest [95]. This "random subspace method" was also explored in earlier work by Tin Kam Ho [99].

For regression tasks, the typical number of features to consider at each split, often denoted as m_try, is often **P/3** (where **P** is the total number of features) [99]**.**

For classification, it is commonly $\boldsymbol{sqrt(P)}$ or $\boldsymbol{log2(P)+1}$

Where:

m_try: This refers to the number of features (or variables) that Random Forest randomly selects and considers at each individual split point within any given decision tree. Instead of looking at all possible features to find the best way to divide the data, it only looks at a random subset of them at each step. This randomness helps make the individual trees in the forest different from each other, which reduces overfitting and improves the overall model's performance.

P: This represents the total number of features (or variables) available.

Sqrt(P): It means that at each split, Random Forest will randomly consider about the square root of the total number of features.

log2(P)+1: This is another alternative rule of thumb for classification, though less frequently cited than sqrt(P). It means that mtry would be calculated as the base-2 logarithm of the total number of features, plus one.

Once all K decision trees are grown, the final prediction from the Random Forest is obtained by combining the predictions of these individual trees:

For regression problems**,** the final prediction H(x) from the Random Forest is the average of predictions from all individual trees [100]:

$$H(x) = \frac{1}{K}\sum_{k=1}^{K} h_k\,(x) \qquad \text{(II.7)}$$

Where:

H(x) is the final ensemble prediction,

K is the number of decision trees,

$h_k(x)$ is the prediction of the k-th tree.

For classification problems, the final prediction is determined by a majority vote among the individual trees.

II.4.3. Ensemble Learning: eXtreme Gradient Boosting (XGBoost)

eXtreme Gradient Boosting (XGBoost) is an optimized, distributed, and highly flexible implementation of the Gradient Boosting Machine (GBM) algorithm. Gradient Boosting itself was formalized by Jerome H. Friedman in the early 2000s [100], building upon earlier boosting concepts like AdaBoost. XGBoost was developed by Tianqi Chen and introduced around 2014, quickly becoming a leading algorithm in machine learning competitions due to its exceptional performance, speed, and parallelizability [101]. It represents a significant improvement over earlier GBM implementations by incorporating sophisticated system- and algorithm-level optimizations.

XGBoost is a powerful and highly efficient ensemble learning method that builds a strong predictive model by combining multiple "weak" prediction models, typically decision trees (specifically, regression trees). It operates on the principle of gradient boosting, where models are added sequentially to the ensemble. Each new model attempts to correct the errors made by the previous ones, focusing on minimizing a defined loss function [102]

1. **Additive Training and Objective Function:**

XGBoost builds its model in an **additive manner**. It starts with an initial prediction (e.g., the mean of the target variable for regression), and then iteratively adds new trees to minimize the residuals (errors) of the previous step.

The prediction at iteration ttt is given by[102, 103]

$$\hat{y}_i^t = \hat{y}_i^{(t-1)} + f_t(x_i) \qquad \text{(II.8)}$$

Where:

$\hat{y}_i^t$: Prediction for sample i at iteration t.

$f_t(x_i)$: The prediction from the new decision tree added at iteration t.

The objective function at step t is[102][104]:

$$Obj^t = \sum_i^n l.(yi, \hat{y}_i^{(t-1)} + f_t(x_i) + \Omega(f_t)\,. \qquad \text{(II.9)}$$

Where:

l: The loss function (e.g., squared error).

y_i: is the actual target value.

$\hat{y}_i^{(t-1)}$: is the prediction from the previous t−1 trees.

$\Omega(f_t)$: Regularization term to control the complexity of f_t.

2. Second-Order Taylor Expansion

To optimize this objective function efficiently, XGBoost uses a **second-order Taylor expansion** of the loss function around the current prediction $\hat{y}_i^{(t-1)}$ This approximation allows for a faster and more accurate optimization of the objective [103].

$$Obj^t \approx \sum_i^n \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) . \qquad \text{(II.10)}$$

Where:

$g_i = \frac{\partial l(y_i, \hat{y}_i)}{\partial \hat{y}_i}$: First-order gradient. (II.11)

$h_i = \frac{\partial l(y_i, \hat{y}_i)^2}{\partial \hat{y}_i^2}$: Second-order gradient (Hessian). (II.12)

3. Regularization:

XGBoost includes explicit regularization terms (L1 and L2 regularization, also known as Lasso and Ridge regularization) in its objective function. This penalizes complex models and encourages simpler trees, preventing overfitting and improving generalization. For a tree f_t with T leaves and leaf weights w_j [102]:

$$\Omega(f_t) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{T} \omega_j^2 \qquad \text{(II.13)}$$

Where:

T: Number of leaves in the tree.

wj : Weight of leaf j.

γ: Penalty for the number of leaves.

λ : L2 regularization term on leaf weights.

4. Tree Construction and Splitting Criteria:

XGBoost constructs each tree by greedily finding the best splits. The "gain" of a split is a key metric, derived from the objective function. It quantifies how much the split improves the model's performance after considering regularization.

A split gain can be approximated as:

$$Gain = \frac{1}{2}\left[\frac{\left(\sum_{i\in I_L} g_i\right)^2}{\sum_{i\in I_L} h_i+\lambda} + \frac{\left(\sum_{i\in I_R} g_i\right)^2}{\sum_{i\in I_R} h_i+\lambda} - \frac{(\sum_{i\in I} g_i)^2}{\sum_{i\in I} h_i+\lambda}\right] - \gamma \qquad \text{(III.14)}$$

Where:

I_L,I_R : Instances in left and right child nodes.

I: Instances in parent node.

II.5. Optimization of Model Parameters

Optimizing model parameters is a crucial step in enhancing machine learning model performance, as hyperparameter tuning directly influences accuracy, generalization, and overall efficiency. This plan focuses on the strategic application of various optimization techniques, including grid search, random search, and advanced metaheuristic algorithms. Specifically, algorithms such as Dragonfly (DA), Grey Wolf (GWO), and Moth-Flame Optimization (MFO) are employed to explore the hyperparameter space more effectively than traditional methods. These algorithms simulate natural or swarm-based behaviors to intelligently navigate complex search landscapes. The optimization process is tightly integrated with model training, allowing for iterative refinement based on performance metrics. However, selecting the optimal parameters presents challenges such as high computational cost, the curse of dimensionality, and the risk of overfitting. To address these, the plan incorporates techniques like early stopping, cross-validation, and adaptive search strategies to balance exploration and exploitation, ensuring robust and efficient parameter selection [105].

Proper optimization not only helps prevent underfitting or overfitting but also ensures that the model is well-suited to the complexity of chemical toxicity data, ultimately enhancing predictive accuracy and robustness.

II.5.1. Dragonfly Algorithm (DA)

The Dragonfly Algorithm (DA) developed by Mirjalili in 2016 [92], is a recent optimization algorithm use to solve a wide variety of optimization problems [106]. It was influenced by the hunting technique known as static swarm (feeding) and migration

strategies of dragonflies, which are referred to as the dynamic swarm (migratory) in nature [107].

The static behavior of a dragonfly is similar to the exploitation phase in meta-heuristics, During this phase, dragonflies form small groups and fly back and forth over a small area to search for food [106] and hunt other flying preys such as butterflies and mosquitoes . Local movements and abrupt changes in the flying path are the main characteristics of a static swarm. On the other hand, the dynamic behavior is akin to the exploration phase [108, 109], In this phase a massive number of dragonflies make the swarm and migrate in one direction over long distances and distract from enemies (while evading enemies) [106, 108].

According to Reynolds [108] The dragonflies swarming behavior is characterized by five operators [110]. These behaviors are mathematically modeled as follows [111]:

Separation

Separation is a strategy that dragonflies utilized to prevent the collision of individuals with their neighbors in the static swarm [91, 107], It is mathematically represented as:

$$s_i = -\sum_{j=1}^{N} X - X_j \qquad \text{(II.15)}$$

Where X is the position of the current individual, X_j shows the position j_{th} neighboring individual, and N is the number of neighboring individuals.

Alignment

Alignment ensures that the velocity of each individual dragonfly matches that of others within the same group. it is represented as follows [91, 110]:

$$A_i = \frac{\sum_{j=1}^{N} V_j}{N} \qquad \text{(II.16)}$$

Where V_j represents neighbor the velocity of j_{th} neighboring individual.

Cohesion

The cohesion refers to the tendency of individuals towards the centre of the swarm's group. It can be defined as [91, 106, 112]:

$$C_i = \frac{\sum_{j=1}^{N} X_j}{N} - X \qquad \text{(II.17)}$$

Attraction to food

As the primary goal of dragonfly swarms is survival, the principle of attraction to food dictates that all individuals tend to move towards the food sources

$$F_i = X^+ - X \qquad \text{(II.18)}$$

Where X^+ shows the position of the food source.

Distraction from enemies

The distraction from enemies happens when all dragonflies move far away from the enemy sources to survive. This step is formulated as [109]

$$E_i = X^- - X \qquad \text{(II.19)}$$

Where X^- shows the position of the enemy.

To update the position of dragonflies for testing another weight solution and have another fitness value [91]. two vectors are calculated: the step vector (DX) and the position vector (X) [111], The step vector shows the direction of the movement of the dragonflies and can be defined according to equation (12). Position vectors are determined using equation (21) [112]

$$\Delta X_i = (sS_i + aA_i + cC_i + fF_i + eE_i) + w\Delta X_t \qquad \text{(II.20)}$$

$$X_{t+1} = X_t + \Delta X_{t+1} \qquad \text{(II.21)}$$

$$e = 0.1 - i * \left(\frac{0.1}{\frac{I}{2}}\right) \qquad \text{(II.21)}$$

$$w = 0.9 - i * \left(\frac{0.9-0.4}{I}\right) \qquad \text{(II.22)}$$

Where s, a, c, f, e, and w are the weights of their correspondent element, e is calculated using equation (14) where i is the current iteration and I is the number of iterations, where s, a, and c are three different random numbers between 0 and 2e, f is a random number between 0 and 2, and w is calculated in equation (22).

II.5.2. Grey Wolves optimization (GWO)

Mirjalili et al. introduced the GWO algorithm in 2014 [113] which is a population-based swarm intelligence algorithm that mimics the social hierarchy and the group hunting behavior of wolves. GWO has been widely used to solve various optimization problems due to its simplicity, requirement of few control parameters, and ease of implementation.

In nature, Grey Wolves form groups of 5–12 individuals, divided into four types based on their roles during hunting: the alpha (α) wolf leads the group, followed by the beta (β) wolves as second-in-command, then the subordinate delta (δ) wolves, those three are consider as the best solutions to lead [114] , and finally the lowest-ranking omega (ω) wolves [89].

The hunting process of wolves consists of three main steps. Firstly, they approach the prey through tracking and chasing. Then, the gray wolf pack encircles the prey and harasses it until it stops moving. Finally, they attack the prey [113].

The GWO algorithm uses the following mathematical technique of hunting [114–118]:

Encircling

The encircling behavior of grey wolves can be mathematically modeled as follows:

$$\vec{D} = \left|\vec{C} \times \overrightarrow{X_P}(t) - \vec{X}(t)\right| \qquad \text{(II.23)}$$

$$\vec{X}\,(t+1) = \overrightarrow{X_P}\,(t) - \vec{A} \times \vec{D} \qquad \text{(II.24)}$$

Where t indicates the current iteration, $\vec{A}$ and $\vec{C}$ are coefficient vectors, $\overrightarrow{X_P}$ is the position vector of the prey, $\vec{X}$ indicates the position vector of a grey wolf, and $\vec{D}$ is computed vector employed to denote a new position of the grey wolf. $\vec{A}$ and $\vec{C}$ can be computed using the formulas below:

$$\vec{A} = 2\vec{a}.\overrightarrow{r_1} - \vec{a} \qquad \text{(II.25)}$$

$$\vec{C} = 2.\overrightarrow{r_2} \qquad \text{(II.26)}$$

Where components of $\vec{a}$ are linearly decreased from 2 to 0 over the course of iterations and $\overrightarrow{r_1}$, $\overrightarrow{r_2}$ are random vectors in [0, 1].

Hunting

For mathematical modeling of wolves' hunting behavior, it is assumed that alpha (α), beta (β), and delta (δ) possess superior knowledge of prey location. Consequently, the remaining wolves (ω) are compelled to follow them. The following equations (27-29) depict this hunting behavior:

$$\vec{D}_\alpha = \left|\vec{C}_1 . \vec{X}_\alpha - \vec{X}\right|, \vec{D}_\beta = \left|\vec{C}_2 . \vec{X}_\beta - \vec{X}\right|, \vec{D}_\delta = \left|\vec{C}_3 . \vec{X}_\delta - \vec{X}\right| \quad \text{(II.27)}$$

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 . \left(\vec{D}_\alpha\right), \ \vec{X}_2 = \vec{X}_\beta - \vec{A}_2 . \left(\vec{D}_\beta\right), \overline{\vec{X}_3} = \vec{X}_\delta - \vec{A}_3 . \left(\vec{D}_\delta\right) \quad \text{(II.28)}$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad \text{(II.29)}$$

Where C_1, C_2, and C_3 are calculated by equation (26),X_α, X_β, and X_δ are the first three best solutions at iteration t, A_1, A_2, and A_3 are calculated as in equation (25), and D_α, D_β, and D_δ are defined as equation (27).

Attacking:

The hunting process is terminated when the prey stops moving, and wolves start an attack. This can be done mathematically by the value of $\vec{a}$ that controls the tradeoff between exploitation and exploration. The parameter $\vec{a}$ is linearly updated in each iteration to range from 2 to 0 according to the equation (30).

$$\vec{a} = 2 - t . \frac{2}{Max_{iter}} \quad \text{(II.30)}$$

Where t is the iteration number and Max_{iter} is the total number of iteration allowed for the optimization.

II.5.3. Moth–Flame Optimization (MFO)

The moth–flame optimization (MFO) is a prominent bio-inspired metaheuristic algorithm inspired by the moths' spiral movement around the light source at night [119]. Moths are a group of insects [120] which resemble butterflies in their life cycle (i.e., moth consists of two-level life: larva and adult, where it is converted to moth by cocoons) in moths' life, their special navigation method at night is the most interesting fact, it is named as transverse orientation [121, 122].

At night, moths maintain a constant angle relative to the moon while flying [122] and this method is a very effective mechanism for travelling long distances in a straight line

especially when the light source is far away. However, when the light source is near, the moths fly spirally around it and eventually converge towards it. Consequently, when there is a best solution, each moth around a flame will update itself. In this way, the better position of the moth will never be lost [120–122].

The mathematical model of MFO is based on moths and flames. Flames are the candidate solutions, whereas the moth positions are the problem variables. Moths positions can be represented by the matrix in (31)

$$M=\begin{bmatrix} m_{1,1} & m_{1,2} & \dots & m_{1,d} \\ m_{2,1} & m_{2,2} & \dots & m_{2,d} \\ \dots & \dots & \dots & \dots \\ m_{n,1} & m_{n,2} & \dots & m_{n,d} \end{bmatrix} \quad \text{(II.31)}$$

Where n is the number of moths and d is the number of variables (dimension).

For all the moths, we also assume that there is an array for storing the corresponding fitness values as follows:

$$OM=\begin{bmatrix} OM_1 \\ OM_2 \\ . \\ . \\ OM_n \end{bmatrix} \quad \text{(II32)}$$

Where, OM is the value of the objective function for each of the moths

There is also a set of flames indicating the best solution that every moth has found so far. Flames have a similar matrix representation as follows:

The following matrix shows the flames followed by their fitness function vector

$$F=\begin{bmatrix} F_{1,1} & F_{1,2} & \dots & F_{1,d} \\ F_{2,1} & F_{2,2} & \dots & F_{2,d} \\ \dots & \dots & \dots & \dots \\ F_{n,1} & F_{n,2} & \dots & F_{n,d} \end{bmatrix} \quad \text{(II.33)}$$

$$OF=\begin{bmatrix} OF_1 \\ OF_2 \\ . \\ . \\ OF_n \end{bmatrix} \quad \text{(II.34)}$$

During the iteration process, the update strategy for the variables in the two arrays is different. In the Moth-Flame Optimization algorithm, moths search for the best solution, which is flame. Moths move toward their own flame, and if they find a better solution, they update their location to that flame. This helps the algorithm to find the best solution

To mathematically model how each moth moves towards a flame, the mechanism for updating each moth's position relative to the flame is expressed using the following equation [121, 123]:

$$M_i = S(M_i, F_j) \quad \text{(II.35)}$$

Where M_i indicates the i_{th} moth, F_j indicates the j_{th} flame, and S is the spiral function.

When utilizing a logarithmic spiral as a mechanism, three conditions must be met, as follows[121, 123–125]:

1- The spiral's initial point should start from the moth's position.
2- The spiral's final point should be the position of the flame.
3- The fluctuation range of the spiral should not exceed the search space.

$$S(M_i, F_j) = D_i . e^{bt} . \cos(2\pi t) + F_j \quad \text{(II.36)}$$

Where D_i indicates the distance of the i_{th} moth for the j_{th} flame, b is a constant for defining the shape of the logarithmic spiral, and t is a random number in [-1, 1].

Di is calculated using the following equation.

$$D_i = |F_j - M_i| \quad \text{(II.37)}$$

The number of flames decreases in the course of iterations based on Equation (38):

$$flame\ no = round(N - l * \frac{N-1}{T}) \quad \text{(II.38)}$$

Where *l* is the current iteration number, N is the initial maximum number of flames set, and T represents the maximum number of flames set, and T represents the maximum number of iterations set.

II.6. Model Evaluation and validation

The plan for model evaluation and validation is designed to ensure that the developed models are both accurate and generalizable. Key evaluation metrics such as the correlation coefficient, R^2 (coefficient of determination), Root Mean Squared Error (RMSE), and Mean Squared Error (MSE) are used to quantitatively assess model performance. Internal validation is conducted using techniques like k-fold cross-validation to ensure robustness and minimize bias during training. To further validate generalizability, external validation is performed on unseen datasets, providing a realistic measure of how well the model performs in practical scenarios. Comparative analysis between different models and the optimization techniques applied helps identify the most effective approach. Additionally, strategies to detect and prevent overfitting, such as monitoring validation performance, regularization, and model complexity control, are discussed to ensure reliability and adaptability across diverse data environments[126]

II.7. Addressing Overfitting and Underfitting in Machine Learning

Overfitting and underfitting are two critical challenges in machine learning that affect a model's ability to generalize to new data. Overfitting occurs when a model learns the training data too precisely, capturing noise and minor details that do not represent the true data distribution. This leads to excellent performance on training data but poor accuracy on unseen data. In contrast, underfitting happens when a model is too simplistic to capture the underlying patterns in the data, resulting in poor performance on both training and test datasets. Avoiding these issues requires finding the right balance between model complexity and training data. Techniques to mitigate overfitting include cross-validation, regularization (such as L1 or L2), dropout in neural networks, and early stopping during training. To prevent underfitting, more expressive models may be used, such as increasing model depth or adding relevant features, along with ensuring the data is properly preprocessed. Additionally, using an adequate and diverse dataset, performing feature selection, and carefully tuning hyperparameters through methods like grid search or metaheuristic optimization can help achieve a well-balanced, generalizable model[127].

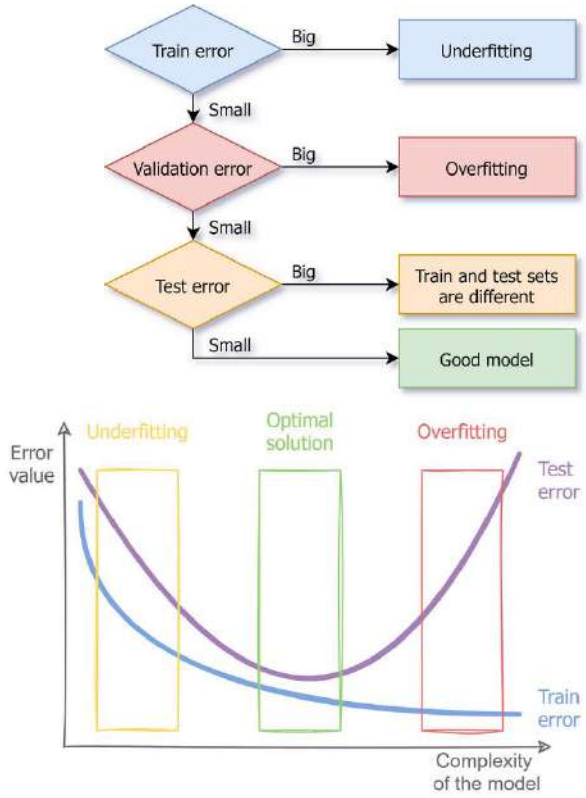


Figure III.4. Overfitting and Underfitting in Machine Learning [128].

Underfitting occurs when the model is too simplistic to capture the data's patterns, resulting in both high training error and high validation/test error. Overfitting, on the other hand, happens when the model is too complex and memorizes the training data, leading to very low training error but high validation/test error.

An ideal model strikes a balance, its training error is low (though not as low as in overfitting), and its validation/test error is also low. When validation and test errors are similar and relatively small, it indicates that the model generalizes well and that the training, validation, and test datasets likely share the same distribution. However, if there's a significant gap between validation and test errors, it may suggest data distribution mismatch or improper data splitting. In such cases, collecting more representative data and ensuring correct data partitioning becomes essential.

II.8. Conclusion

In conclusion, this chapter has provided a comprehensive theoretical foundation for the machine learning and optimization algorithms employed in this thesis. It highlighted the principles of Support Vector Machines, the ensemble power of Random Forest and eXtreme Gradient Boosting, and the capabilities of meta-heuristic optimization algorithms such as Dragonfly Algorithm, Grey Wolf Optimizer, and Moth-Flame Optimization.

Understanding these methods is crucial for appreciating their application in predictive modeling, particularly for complex tasks like chemical toxicity assessment. The chapter also discussed essential concepts of model evaluation, validation, and strategies to address common challenges like overfitting and underfitting. This theoretical framework underpins the methodological approaches and experimental design presented in subsequent chapters, providing the necessary tools to develop robust and reliable Quantitative Structure-Toxicity Relationship models for ionic liquids.

Chapter III

Development of prediction models for the toxicity of ionic liquids

Chapter III
Development of prediction models for the toxicity of ionic liquids

III.1 Introduction

This chapter presents the comprehensive results of the QSTR modeling developed for predicting the toxicity of ionic liquids. The modeling workflow integrates rigorous data preprocessing, molecular descriptor computation, dimensionality reduction, and machine learning algorithm optimization. Two distinct descriptor selection methods, Genetic Algorithm combined with Multiple Linear Regression (GA-MLR) and the Boruta algorithm, were employed to identify the most relevant molecular features. These selected descriptors were subsequently used to develop predictive models based on Support Vector Machine (SVM), optimized through three metaheuristic algorithms: the Dragonfly Algorithm, Grey Wolf Optimizer, and Moth-Flame Optimization.

In a second modeling stage, a different descriptor reduction approach, ARKA (Arithmetic Residuals in K-groups Analysis), was applied to the same dataset to produce an alternative descriptor set. The resulting ARKA descriptors were then used to train two additional machine learning models based on ensemble methods: XGBoost and Random Forest.

Each model's performance was assessed through standard error and statistical metrics as well as visual evaluation tools such as scatter plots, violin plots, and SHAP analysis. The objective of this chapter is to compare and interpret the results obtained from the different modeling strategies, highlight the advantages of each descriptor set, and determine the most effective predictive model for the toxicity of ionic liquids.

III.2. Overview of the Methodology

The methodology for this study, it begins with collect and preparing a dataset of ionic liquids with known toxicity values. Molecular descriptors are then calculated to represent the chemical structures numerically. A key part of this work involves comparing and selecting the most relevant descriptors using different methods, specifically Genetic Algorithm (GA) with Multiple Linear Regression (MLR) and the Boruta algorithm. The dataset is then split for model training and testing. Predictive QSTR models are built using

selected algorithms, with the Support Vector Machine (SVM) model being optimized through the use of metaheuristic optimization algorithms such as Dragonfly Algorithm (DA), Moth-Flame Optimization (MFO), and Gray Wolf Optimizer (GWO). Finally, these models are rigorously validated using appropriate statistical metrics to assess their performance and reliability.

III.3 Materials and Methods

III.3.1 Experimental Dataset Collection

In the development of reliable QSTR model, the quality and diversity of the experimental dataset play a crucial role. A representative and carefully compiled dataset is essential to ensure accurate learning.

In this study, a total of 318 ionic liquids (ILs) were collected from various previously published literature sources. Specifically, 304 of these ionic liquids were used to construct the training and test sets, while the remaining 14 ionic liquids were reserved for external validation set to assess the models' robustness and predictive capacity on unseen data.

The specific toxicity endpoint investigated is the logarithm of the half-maximal effective concentration ($\log_{10}(EC_{50})$). This toxicity was measured using the *leukemia rat cell line (IPC-81)*, a common biological target in ionic liquid toxicity studies. The toxicity values within this dataset range from -0.24 to 4.9.

The dataset exhibits significant structural diversity, covering ten different IL families, these include 126 imidazolium (Im), 61 pyridinium (Py), 35 ammonium (N), 33 morpholinium (Morph), 28 pyrrolidinium (Pyr), 17 piperidinium (Pip), 9 aminium, 5 phosphonium (P), 5 quinolinium (Qui), and 2 sulfonium. This distribution provides significant structural variety for modeling and ensures broad chemical coverage, which supports the development of models capable of capturing structure–toxicity relationships across diverse IL families.The distribution of these various ionic liquid families is illustrated in **Figure III.1**.

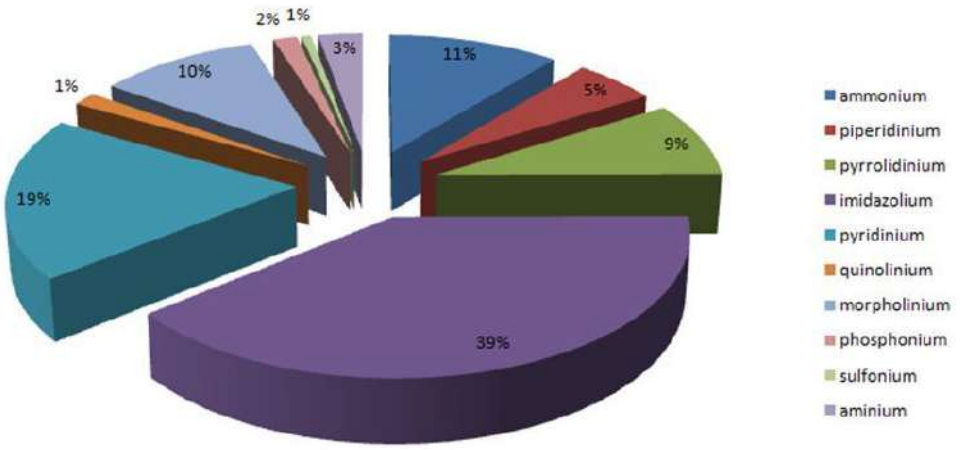


Figure III.1. The distribution of the various ionic liquid families utilized in this study.

III.3.2. Descriptor Calculation

Molecular descriptors are essential as they translate chemical structures into numerical values suitable for computational analysis. This makes them quite important in building QSTR models, as they essentially act as numerical descriptions of a molecule's structure, serving as the input data for models [129]. As defined by Todeschini and Consonni, a molecular descriptor is "the final result of a logic and mathematical procedure which transforms chemical information encoded within a symbolic representation of a molecule into a useful number or the result of some standardized experiment" [130].

To prepare our ionic liquids for this process, their structures were initially represented using the SMILES notation (Simplified Molecular Input Line Entry System). SMILES strings are a convenient way to represent molecular structures as a sequence of characters that computers can easily handle [75]. The SMILES specifications for the 318 ionic liquids were obtained from PubChem. These were subsequently converted into Mol format, a standard chemical file format commonly used for calculating molecular descriptors.

The calculation of these descriptors was carried out using the AlvaDesc software version 2.0.0 (https://www.alvascience.com/alvadesc). This software generated a large initial set of 5666 molecular descriptors for each ionic liquid, to make this large dataset more manageable and relevant for modeling, a series of initial variable reduction methods were applied. The first simple step was to remove descriptors that had zero values across all samples. Next, invariant descriptors were manually removed – those where all values were constant or nearly identical, offering no variability to inform the model. To further refine the set, any descriptors where 75% of the samples shared identical values were

removed, as well as any descriptors with a relative standard deviation less than 0.05. Finally, descriptors with a high absolute Pearson correlation coefficient (greater than 0.95) were also removed [131]. After applying these filtering steps, a more concise dataset containing 187 descriptors was obtained.

III.3.3. Descriptor Selection

Following the initial filtering steps, a more focused set of 187 molecular descriptors remained. However, using all of these descriptors for model building can still lead to issues like over-fitting and reduced model interpretability. Therefore, selecting the most relevant descriptors that are strongly correlated with toxicity is a crucial step in developing robust QSTR models. In this study, two distinct descriptor selection methodologies were employed and compared to identify optimal subsets of features:

III.3.3.1. Genetic Algorithm combined with Multiple Linear Regression (GA-MLR)

The Genetic Algorithm (GA) combined with Multiple Linear Regression (MLR) is a widely used and powerful approach for descriptor selection in quantitative structure-toxicity relationship (QSTR) modeling[67, 132, 133]. A Genetic Algorithm is a search heuristic inspired by the process of natural selection and biological evolution[132, 133]. It is commonly applied to find optimal solutions in complex search spaces by mimicking evolutionary operators such as selection, crossover (recombination), and mutation[132, 134–136]. GAs are particularly well-suited for feature selection in QSAR applications due to their ability to perform a directed random search in large descriptor spaces and investigate many possible solutions simultaneously, which helps to avoid getting stuck in local optima, a limitation of some other selection methods[129, 132, 133, 135, 137]. Multiple Linear Regression (MLR), on the other hand, is a fundamental statistical technique used to model the linear relationship between a dependent variable, such as toxicity, and one or more independent variables, represented by molecular descriptors[138–140]. MLR aims to find the best-fitting linear equation that describes this relationship, often providing an interpretable model[141]. When these two methods are combined, the GA iteratively searches for subsets of descriptors.For each subset identified by the GA, an MLR model is constructed, and its predictive performance (often assessed using statistical metrics like R^2 or RMSE) serves as the fitness function for the GA, guiding the search towards subsets that yield better predictive models[132, 133, 142].

In this specific application, the Genetic Algorithm method was utilized to select the smallest, but most significant set of descriptors from the filtered pool of 187 that could predict the target property with an acceptable error. The algorithm took as input a training set file, which constituted 80% of the database and had an inter-correlation threshold of 0.9. The selection of the appropriate number of descriptors was systematically carried out over an interval ranging from 5 to 20 descriptors, resulting in the formation of 15 distinct combinations. For each of these 15 descriptor combinations, an MLR model was trained, and its statistical performance was evaluated. The combination of descriptors that yielded the best statistical performance was identified as the optimal set for this method. Furthermore, to ensure the reliability of the selected descriptors and assess potential multicollinearity, their Variance Inflation Factors (VIF) was calculated. All VIF values were found to be below 3.0614, affirming both the statistical significance of the model and the reasonable independence of the selected descriptors. The specific set of 13 chosen descriptors was: S2K, SIC2, B09[C-C], GATS6p, LOC, B06[C-O], Eta_sh_y, MATS5i, piID, H-052, F01[C-N], P_VSA_ppp_N, and MATS3m. Additional details regarding the characterization of these selected descriptors are available in Table III.1.

Table III.1: Characteristics of the descriptors selected by GA_MLR.

Descriptors name	Type	Description	VIF
S2K	Topological indices	2-path Kier alpha-modified shape index	1.7893
SIC2	Information indices	Structural Information Content index (neighborhood symmetry of 2-order)	2.8609
B09[C-C]	2D Atom Pairs	Presence/absence of C - C at topological distance 9	2.5506
GATS6p	2D autocorrelations	Geary autocorrelation of lag 6 weighted by polarizbility	2.2255
LOC	Topological indices	lopping centric index	3.0614
B06[C-O]	2D Atom Pairs	Presence/absence of C - O at topological distance 6	1.4091
Eta_sh_y	ETA indices	eta y shape index	2.3739
MATS5i	2D autocorrelations	Moran autocorrelation of lag 5 weighted by ionization potential	1.3406
PiID	Walk and path counts	conventional bond order ID number	2.5623
H-052	Atom-centered fragments	H attached to C0(sp3) with 1X attached to next C	1.5442

F01[C-N]	2D Atom Pairs	Frequency of C - N at topological distance 1	1.2540
P_VSA_ppp_N	P_VSA-like descriptors	P_VSA-like on potential pharmacophore points, N – negative	1.4087
MATS3m	2D autocorrelations	Moran autocorrelation of lag 3 weighted by mass	2.3845

III.3.3.2.Boruta Algorithm

The second descriptor selection method employed was the Boruta algorithm. Boruta is an all-relevant feature selection wrapper algorithm built around the Random Forest classification or regression algorithm [143][144]. The algorithm's core idea involves creating shuffled copies of the original features, referred to as 'shadow' features, by permuting the values within each. A Random Forest model is then trained on the dataset augmented with these shadow features. The importance of each original descriptor is then compared to the maximum importance observed among the shadow features. Descriptors are considered 'confirmed' as relevant if their importance is significantly higher than that of the best shadow feature. This iterative process helps to robustly identify features that carry significant information for prediction, accounting for complex interactions and reducing the misleading impact of random fluctuations [144, 145].

In this study, the Boruta algorithm was applied to the initial dataset of 5666 molecular descriptors generated by AlvaDesc to identify those features statistically confirmed as relevant for predicting the toxicity of ionic liquids. This process resulted in the selection of 17 descriptors. The specific set of these 17 descriptors, along with their characteristics and VIF values, is detailed below in Table III.2**.**

Table III.2: Characteristics of the descriptors selected by Boruta.

Descriptorname	Type	Description	VIF
MaxDD	Topological indices	Maximum vertex degree of the molecule	3.6856
ChiA_D	Connectivity indices	Average valence connectivity index (order 0)	2.4342
MATS2e	2D autocorrelations	Moran autocorrelation of lag 2 weighted by electronegativity	4.0213
MATS5e	2D autocorrelations	Moran autocorrelation of lag 5 weighted by electronegativity	4.7127

GATS5e	2D autocorrelations	Geary autocorrelation of lag 5 weighted by electronegativity	3.8136
JGI7	Topological charge indices	Mean topological charge index of order 7	1.2086
SpMax2_Bh(p)	Burdeneigenvalues	Largest absolute eigenvalue of Burden matrix weighted by polarizability (order 2)	4.7320
SpMax1_Bh(i)	Burdeneigenvalues	Largest absolute eigenvalue of Burden matrix weighted by ionization potential (1)	2.2978
SpMin2_Bh(p)	Burdeneigenvalues	Smallest absolute eigenvalue of Burden matrix weighted by polarizability (2)	4.9491
P_VSA_charge_1	P_VSA-like descriptors	P_VSA region on atomic partial charges, range 1	5.1342
P_VSA_charge_8	P_VSA-like descriptors	P_VSA region on atomic partial charges, range 8	2.6461
Eig02_EA(ri)	Eigenvaluedescriptors	Eigenvalue 02 from edge adjacency matrix weighted by resonance integrals	5.8399
minssCH2	Atom-centered fragments	Minimum number of –CH_2– fragment occurrences	1.9251
MaxdO	Atom-type E-state indices	Maximum E-state for double-bonded oxygen	3.8253
CATS2D_06_PL	Pharmacophore descriptors	Pharmacophore pair (positive-lipophilic) at topological distance 6	2.6456
B09[C–C]	2D Atom Pairs	Presence/absence of C–C at topological distance 9	2.7398
ESOL	Physicochemicalproperty	Estimatedaqueoussolubility	5.7158

While most VIF values for the Boruta-selected descriptors are below 5, three descriptors, P_VSA_charge_1 (VIF = 5.1342), Eig02_EA(ri) (VIF = 5.8399), and ESOL (VIF = 5.7158), exhibit values slightly exceeding this common threshold. However, considering Boruta's robust methodology in identifying truly relevant features and its inherent ability to mitigate multicollinearity concerns during the selection process, these VIF values are deemed acceptable for the purpose of this study, as they remain well below levels that would typically indicate severe multicollinearity (e.g., VIF > 10). The identified

descriptors were then considered for subsequent model development and comparison with the set obtained from the GA-MLR method.

III.3.4. Data Division

To ensure the effective evaluation of the predictive models, a total of 318 ionic liquids (ILs) were utilized in this study. Of these, 304 ILs were used for model development, while 14 ILs were reserved for external validation. A standard cross-validation strategy known as the holdout method was employed to divide the dataset. Specifically, 80% of the data (244 ILs) was randomly assigned to the training set using the dividerand function in MATLAB® R2020b. This training set was used to construct and calibrate the predictive models. The remaining 20% (60 ILs) formed the test set, which was used to evaluate the generalization performance of the models. This same data split was applied consistently across all three metaheuristic optimization frameworks employed in this study: Dragonfly Algorithm (DA), Grey Wolf Optimizer (GWO), and Moth-Flame Optimization (MFO). The remaining 14 ionic liquids from the original 318 were set aside as a completely independent external validation set to further assess the robustness and predictive accuracy on completely unseen data.

III.3.5. Meta-heuristic Optimized SVM for QSTR Modeling

Support Vector Machine (SVM) is a powerful supervised learning technique frequently employed to solve nonlinear regressionand classification problems in cheminformatics and QSAR/QSTR modeling[91, 146]. A key aspect of applying SVM effectively is determining the best hyperparameter values for a given dataset to ensure high accuracy with minimal errors. Inappropriate parameter values can significantly impact model performance, potentially leading to issues such as overfitting or underfitting[147].

Traditionally, methods like grid search and gradient descent have been commonly used to find optimal SVM parameters. However, these traditional approaches can have drawbacks, such as the risk of getting stuck in local minima, being computationally intensive, and requiring significant time, particularly when dealing with multiple hyperparameters and large search spaces. To address these limitations and improve the efficiency and effectiveness of finding optimal parameters, meta-heuristic optimization algorithms have been developed and applied[148, 149].

The aim of the present work was to apply three distinct meta-heuristic algorithms – the Dragonfly Algorithm (DA) [149], the Grey Wolf Optimizer (GWO)[150], and Moth-Flame Optimization (MFO) [121], to find the optimal values for three key SVM parameters: the box constraint (C), epsilon (ϵ), and the kernel scale (σ). The Gaussian kernel function was specifically chosen for the SVM model due to its flexibility in modeling complex, non-linear relationships.

The analysis and optimization process were conducted using the ***fitrsvm*** function in MATLAB® R2020b. For optimizing the SVM hyperparameters within each meta-heuristic framework, a cross-validation approach known as the "holdout" method was employed, with a split ratio of 0.2. This involves splitting the dataset into 80% for training and 20% for testing.

Initially, each meta-heuristic algorithm (DA, GWO, and MFO) generated random combinations of the three SVM hyperparameters (C, ϵ, and σ) within their specified ranges. These hyperparameter combinations were then iteratively optimized through several iterations. During each iteration, the steps began with the division of the data into temporary training and validation sets and continued up until the development and evaluation of the SVM model using the current hyperparameter combination. The performance of the SVM model was evaluated by calculating the Average Root-Mean-Squared Error (RMSE) on the temporary validation set. The meta-heuristic algorithm used this RMSE value as a fitness measure to guide the search for better hyperparameter combinations. This process was repeated for 200 trials for each meta-heuristic algorithm to ensure a thorough search of the parameter space and robustness in finding the optimal parameters. The minimum Average RMSE obtained across these 200 trials corresponded to the resultant optimal hybrid SVM model for each respective meta-heuristic algorithm. By associating the SVM model with the selected meta-heuristic algorithms (DA, GWO, and MFO), the following hybrid QSTR models were built: DA-SVM, GWO-SVM, and MFO-SVM.

III.3.6. Statistical Criteria for Evaluation of Model Performance

The evaluation of predictive model performance (commonly referred to as validation) is a fundamental step in the development of reliable QSTR models[151][152]. This process is essential to confirm the robustness, generalizability, and predictive accuracy of the

developed models, especially when applied to new, unseen data. Without proper validation, there is a risk of overfitting, which can lead to misleading or unreliable predictions[153, 154].

Metrics play a crucial role in this evaluation, providing quantitative measures of model accuracy and precision, they serve also to compare the performance of different models or variable inputs across various validation stages[153, 155].

The validation process typically involves three levels: internal validation (performed on the training data), external validation (conducted on a test or independent dataset), and overall evaluation on the full dataset. These multiple levels of validation offer a more comprehensive understanding of model behavior, stability, and predictive strength, for a comprehensive evaluation of the QSTR models developed in this study, the key statistical metrics and errors used are summarized and presented in Table III.2[93, 152, 156–159].

TableIII.3: Error and Statistical metrics commonly used for model evaluation.

Error and Statistical metrics	The formula		Acceptability criteria
Root Mean Squared Error (RMSE)	$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(Y_i^{exp} - Y_i^{cal})^2}{N}}$	(III.1)	Lower values indicate better model performance
Mean Squared Error (MSE)	$MSE = \frac{1}{N}\sum_{i=1}^{N}(Y_i^{exp} - Y_i^{cal})^2$	(III.2)	Lower values indicate better model performance
Mean Absolute Error (MAE)	$MAE = \frac{\sum_{i=1}^{N}\lvert Y_i^{exp} - Y_i^{cal}\rvert}{N}$	(III.3)	Lower values indicate better model performance
Predictive Statistic (Q^2_{F1})	$Q^2_{F1} = 1 - \frac{\sum_{i=1}^{Nout}(Y_i^{exp} - Y_i^{cal})^2}{\sum_{i=1}^{Nout}(Y_i^{exp} - \bar{Y}_{TR})^2}$	(III.4)	closer to 1
Predictive Statistic (Q^2_{F2})	$Q^2_{F2} = 1 - \frac{\sum_{i=1}^{Nout}(Y_i^{exp} - Y_i^{cal})^2}{\sum_{i=1}^{Nout}(Y_i^{exp} - \bar{Y}_{OUT})^2}$	(III.5)	closer to 1
Predictive Statistic (Q^2_{F3})	$Q^2_{F3} = 1 - \frac{\sum_{i=1}^{Nout}(Y_i^{exp} - Y_i^{cal})^2 / N_{out}}{\sum_{i=1}^{N_{TR}}(Y_i^{exp} - Y_{TR})^2 / N_{TR}}$	(III.6)	closer to 1

Concordance Correlation Coefficient Q^2_{ccc}	$Q^2_{ccc=} \frac{2.\sum_{i=1}^{Nout}(Y_i - \bar{Y}_{exp}).(\hat{Y}_{i/i} - \bar{y}_{cal})}{\sum_{i=1}^{Nout}(Y_i - \bar{Y}_{exp})^2 + \sum_{i=1}^{Nout}(\hat{Y}_{i/i} - \bar{Y}_{cal})^2 + N_{out}.(\bar{Y}_{exp} - \bar{Y}_{cal})^2}$	(III.7)	closer to 1
Coefficient of Determination (R^2)	$R^2 = 1 - \frac{\sum_{i=1}^{N}(Y_i^{exp} - Y_i^{cal})^2}{\sum_{i=1}^{N}(Y_i^{exp} - \bar{Y}^{exp})^2}$	(III.8)	closer to 1
Correlation Coefficient (R)	$R = \frac{\sum_{i=1}^{N}(Y_i^{exp} - \bar{Y}^{exp}).(Y_i^{cal} - \bar{Y}^{cal})}{\sqrt{\sum_{i=1}^{N}(Y_i^{exp} - \bar{Y}^{exp})^2}.\sqrt{\sum_{i=1}^{N}(Y_i^{cal} - \bar{Y}^{cal})^2}}$	(III.9)	closer to 1
Slope (K)	$K = \frac{\sum_{i=1}^{N}(Y_i^{exp}.Y_i^{cal})}{\sum_{i=1}^{N}(Y_i^{cal})^2}$	(III.10)	closer to 1
Alternative Slope (K')	$K' = \frac{\sum_{i=1}^{N}(Y_i^{exp}.Y_i^{cal})}{\sum_{i=1}^{N}(Y_i^{exp})^2}$	(III.11)	closer to 1
Adjusted Coefficient of Determination (R^2_0)	$R_0^2 = 1 - \frac{\sum_{i=1}^{N}(Y_i^{exp} - KY_i^{cal})^2}{\sum_{i=1}^{N}(Y_i^{exp} - \bar{Y}_i^{cal})^2}$	(III.12)	closer to 1
Adjusted Coefficient of Determination with Alternative Slope(R'^2_0)	$R_0'^2 = 1 - \frac{\sum_{i=1}^{N}(Y_i^{exp} - K'Y_i^{cal})^2}{\sum_{i=1}^{N}(Y_i^{exp} - \bar{Y}_i^{cal})^2}$	(III.13)	closer to 1
Model Efficiency Metric (m)	$m = \frac{(R^2 - R_0^2)}{R^2}$	(III.14)	<0.1
Model Efficiency Metric (n)	$n = \frac{(R^2 - R_0'^2)}{R^2}$	(III.15)	<0.1
Adjusted (R_m^2)	$R_m^2 = R^2 \times \left(1 - \sqrt{\lvert R^2 - R_0^2 \rvert}\right)$	(III.16)	≥0.5
Bias Factor (B_f)	$B_f = 10^{\left(\frac{\sum_{i=1}^{N} log \frac{Y_i^{exp}}{Y_i^{cal}}}{N}\right)}$	(III.17)	Closer to 1

Accuracy Factor (A_f)	$A_f = 10^{\left(\frac{\sum_{i=1}^{N}\left\lvert log\frac{Y_i^{exp}}{Y_i^{cal}}\right\rvert}{N}\right)}$	(III.18)	Closer to 1
Akaike Information Criterion (AIC)	$AIC = N\ln\left(\frac{SSE}{N}\right) + 2N_P + \frac{2N_P(N_P+1)}{N-(N_P+1)}$	(III.19)	Lower values indicate a better balance of fit and complexity
	$SSE = \sum_{i=0}^{N}(Y_i^{exp} - Y_i^{cal})^2$	(III.20)	
	$\bar{Y}^{exp} = \frac{\sum_{i=1}^{N} Y_i^{exp}}{N}\bar{Y}^{cal} = \frac{\sum_{i=1}^{N} Y_i^{cal}}{N}$	(III.21)	

Where N is the number of data points, y^{exp} is the value of experimental datasets at the sampling point i, y_i^{cal} is the i_{th} value of the predicted datasets and $\bar{Y}^{exp}$and $\bar{Y}^{cal}$are the average of the experimental and predicted data respectively.

III.4. Results and Discussion

This section presents the results obtained from building QSTR models to predict the toxicity of ionic liquids (ILs) using Support Vector Machine (SVM) combined with three meta-heuristic optimization algorithms: Dragonfly Algorithm (DA), Grey Wolf Optimizer (GWO), and Moth-Flame Optimization (MFO). The models were developed using two different descriptor selection strategies: Genetic Algorithm combined with Multiple Linear Regression (GA-MLR), and the Boruta algorithm.

III.4.1. Optimized SVM Hyperparameters and Comparisons of Performances

As described in Section III.3.6, the integrated meta-heuristic SVM models involved optimizing the key SVM hyperparameters: the box constraint (C), epsilon (ϵ), and the kernel scale (σ).

The optimal hyperparameter values found by each meta-heuristic algorithm for the SVM models built with the GA-MLR and Boruta selected descriptors are presented in Table III.4, based on the optimization process.

Table III.4: Optimized SVM hyperparameters for GA-MLR and Boruta descriptors.

Model	Descriptor Set	C (Box Constraint)	σ (Kernel Scale)	ϵ (Epsilon)	Search Agents	Iterations
DA-SVM	GA-MLR	123.6106	2	0.0060	10	200
	Boruta	333.1510	2	0.0263	10	200
MFO-SVM	GA-MLR	194.7156	3	0.0073	10	200
	Boruta	85.0253	2	0.0300	10	200
GWO-SVM	GA-MLR	64.1816	2	0.0066	10	200
	Boruta	59.3288	1.6481	0.0041	10	200

Figure III.2 and Figure III.3 illustrate the optimization progress of the three meta-heuristic algorithms (DA, GWO, and MFO) during the SVM hyperparameter tuning process for GA-MLR and Boruta selected descriptors, respectively. These convergence curves represent the best Root Mean Squared Error (RMSE) achieved by each algorithm over 200 iterations. As observed in both figures, all three algorithms effectively reduced the RMSE over successive iterations, demonstrating their capability to explore the solution space.

For the GA-MLR selected descriptors (Figure III.2), the hybrid DA-SVM model achieved the lowest final RMSE of 0.1787, indicating its superior ability to identify optimal SVM hyperparameters. The GWO-SVM model followed with a final RMSE of 0.1811, while MFO-SVM resulted in a slightly higher RMSE of 0.1853.

Similarly, for the Boruta-selected descriptors (Figure III.3), the DA algorithm again demonstrated strong performance, achieving the lowest final RMSE of approximately 0.1827. The GWO algorithm also performed well, converging to an RMSE of around 0.1838, while MFO converged to approximately 0.1911.

In both descriptor selection scenarios, this convergence analysis highlights the comparatively faster and more stable performance of the DA algorithm in minimizing the error function, suggesting its stronger search capability within the defined iteration limit and parameter space.

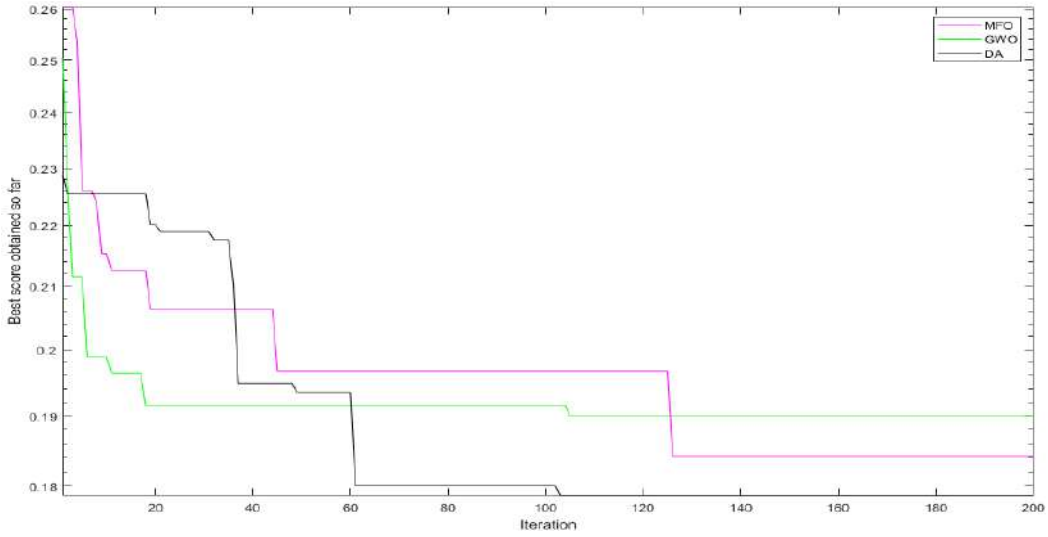


Figure III.2. Convergence curve of the three algorithms (using GA_MLR descriptors).

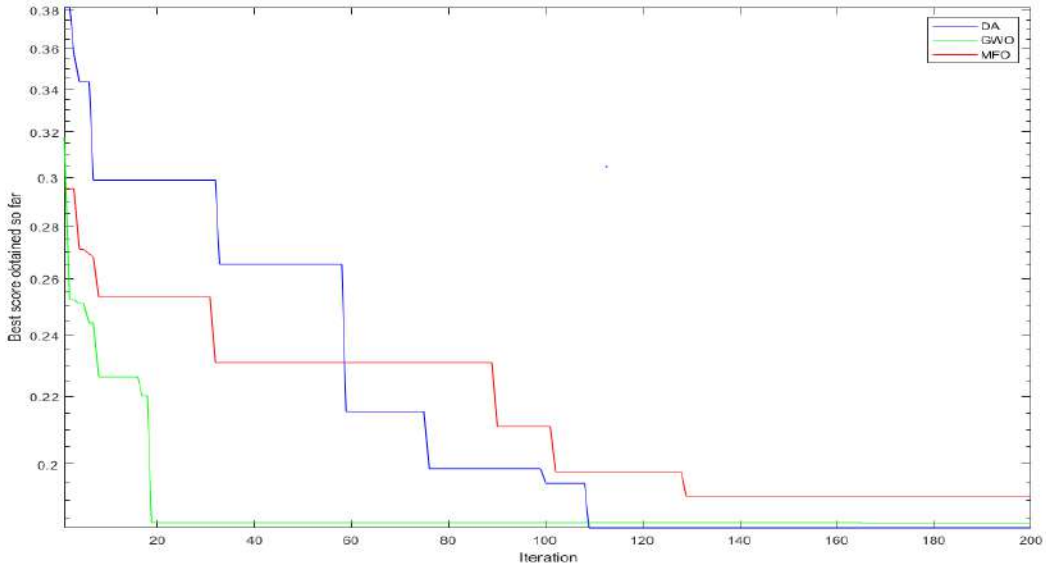
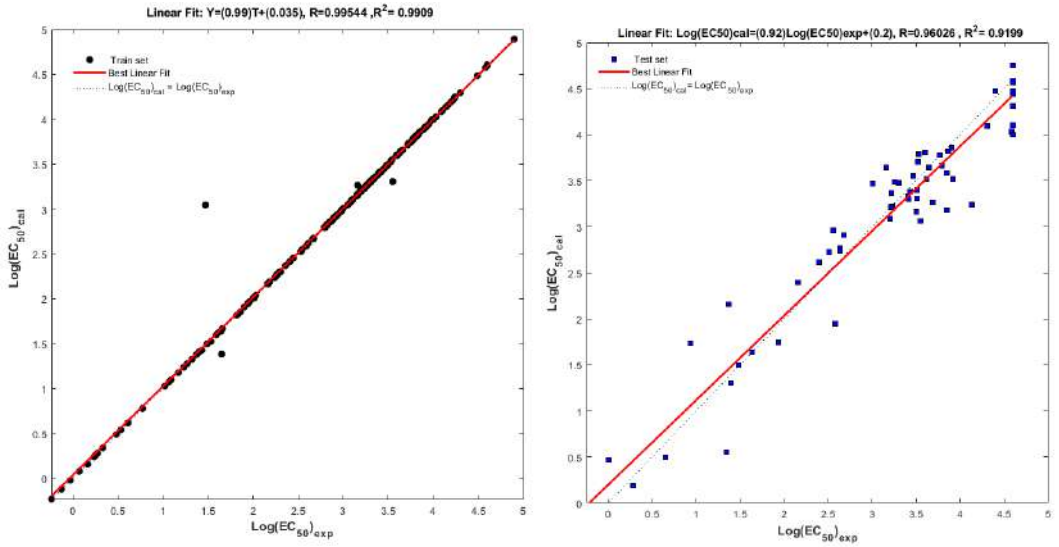


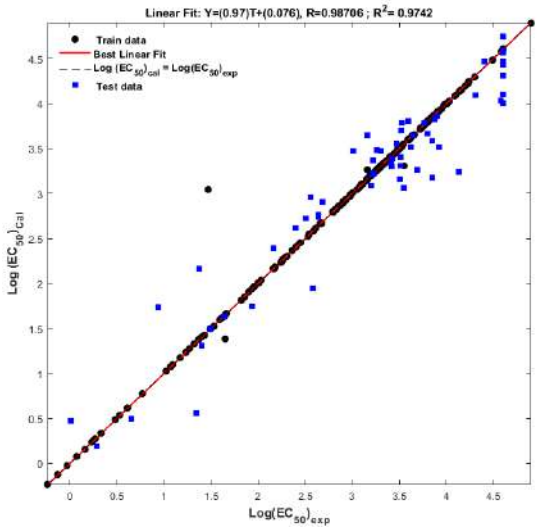
Figure III.3. Convergence curve of the three algorithms (using Boruta descriptors).

To visually evaluate the predictive performance of the developed QSTR models built with the GA-MLR and Boruta selected descriptors, scatter plots were generated to illustrate the correlation between the experimentally observed and model-predicted toxicity values for the training, test, and global datasets of ionic liquids. As illustrated in Figures III.4–III.6 (GA-MLR) and Figures III.7–III.9 (Boruta), these diagrams clearly show the alignment between the calculated values and the experimental values, indicated by a tight dispersion of the points around the ideal diagonal line (the first bisector). Notably, the DA-SVM model demonstrates superior alignment across both descriptor selection methods

compared to the GWO-SVM and MFO-SVM models, visually suggesting its higher predictive accuracy. While some outlier points are visible in each modeling approach, their deviation from the expected range remains acceptable and does not significantly impact the overall performance of the models.



(a) (b)



(c)

Figure III.4. Scatter plots predicted vs. observed values for the (a) train, (b) test and (c) global sets, for DA_SVM model (GA_MLR).

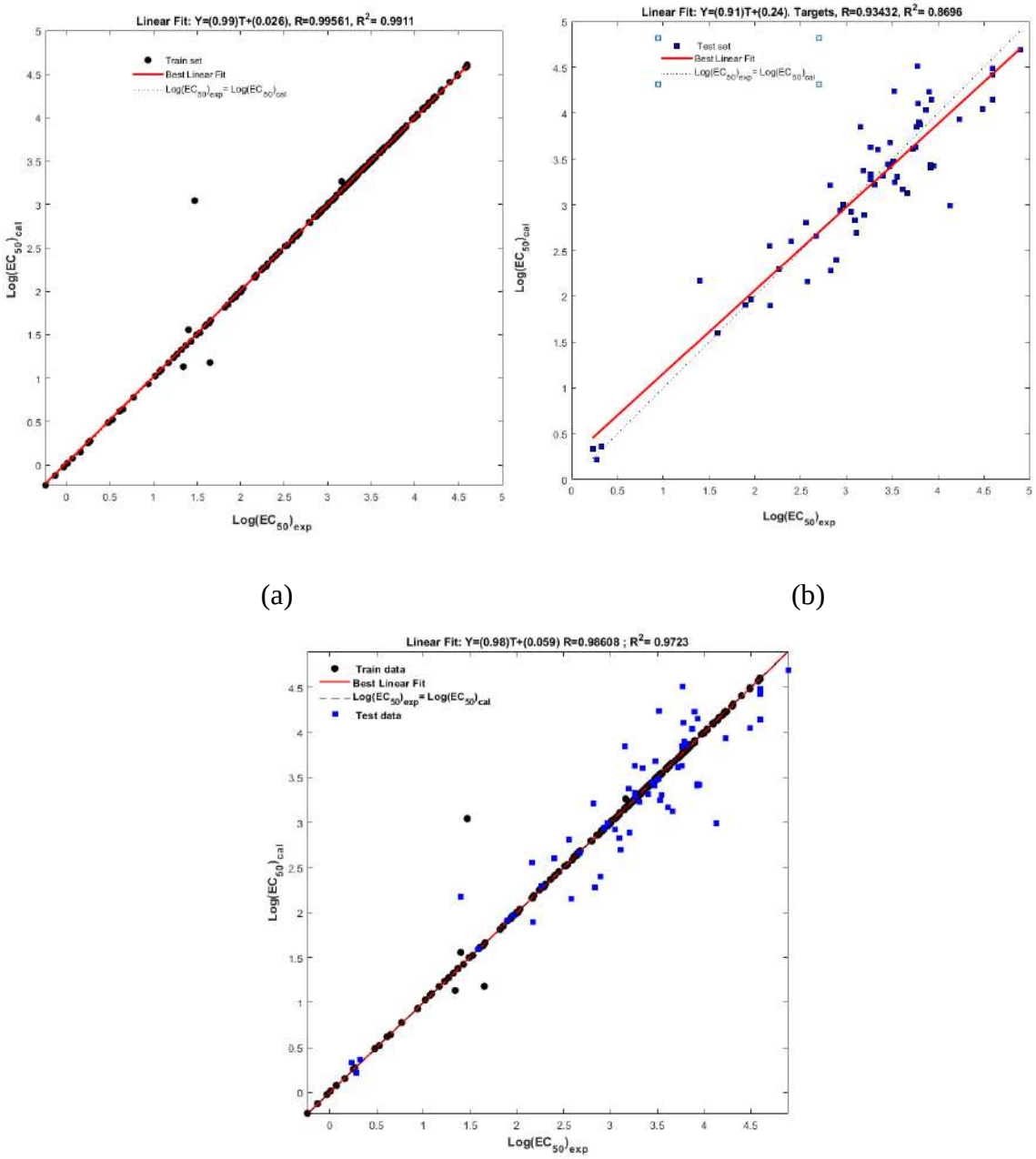
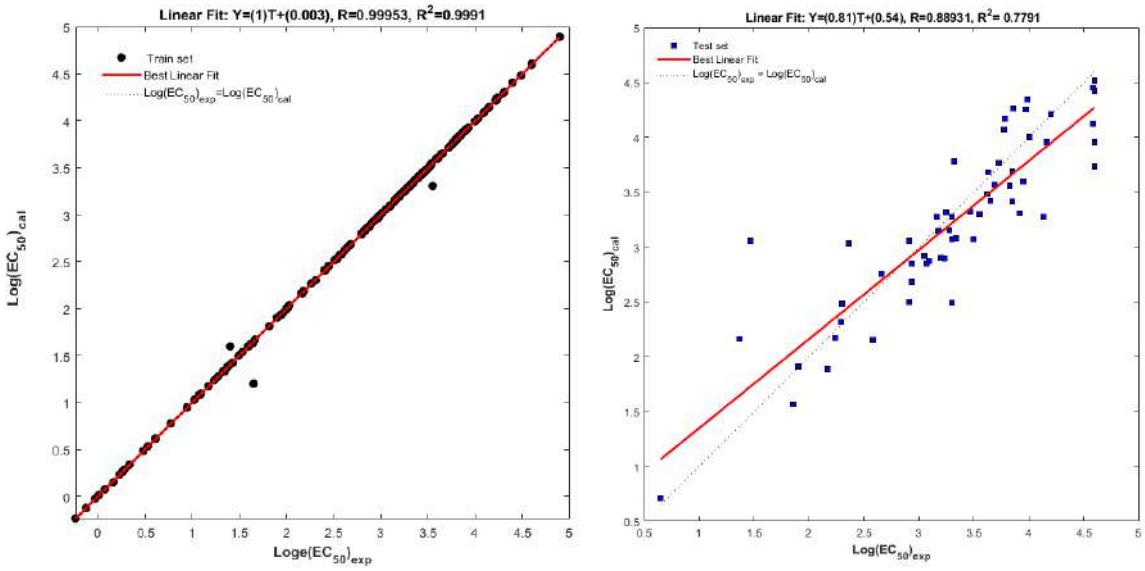
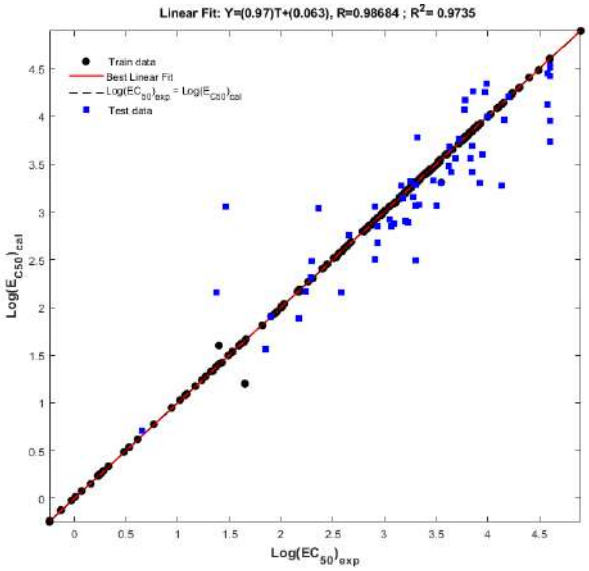


Figure III.5. Scatter plots predicted vs. observed values for the (a) train, (b) test and (c) global sets, for MFO_SVM model (GA_MLR).



(a) (b)



(c)

Figure III.6. Scatter plots predicted vs. observed values for the (a) train, (b) test and (c) global sets, for GWO_SVM model (GA_MLR).

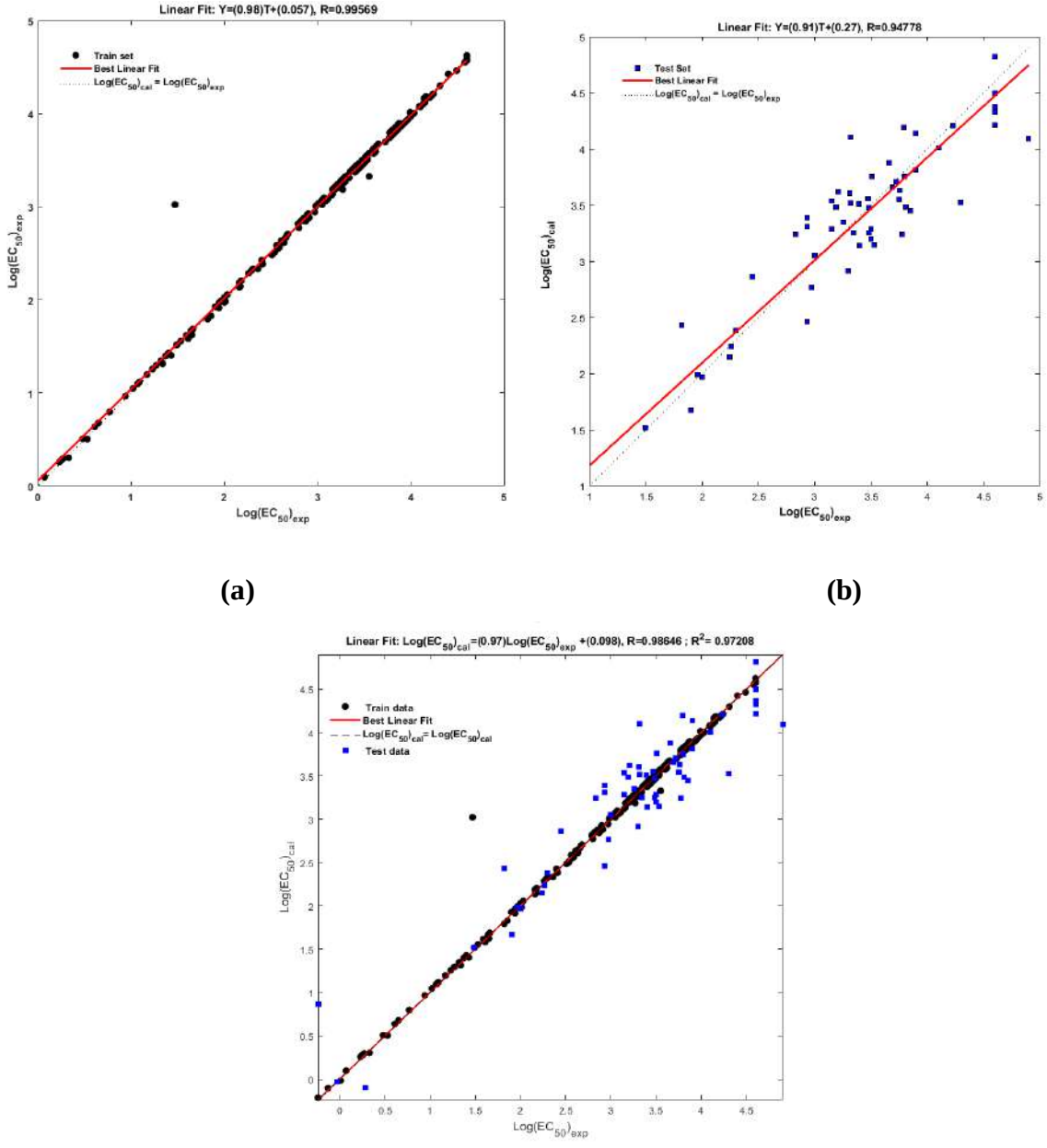
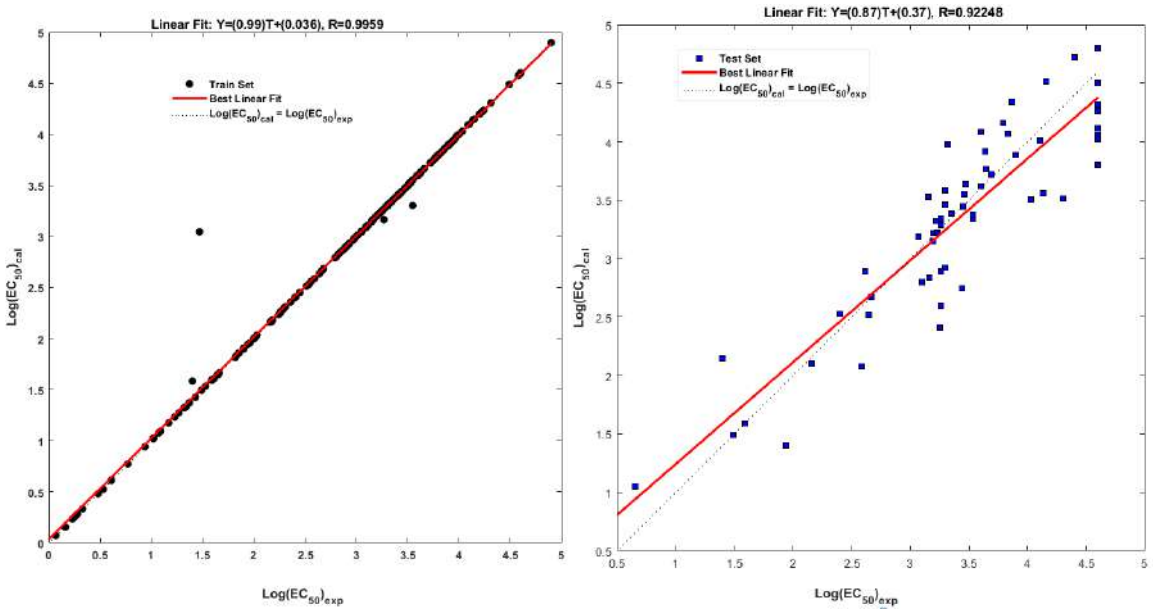
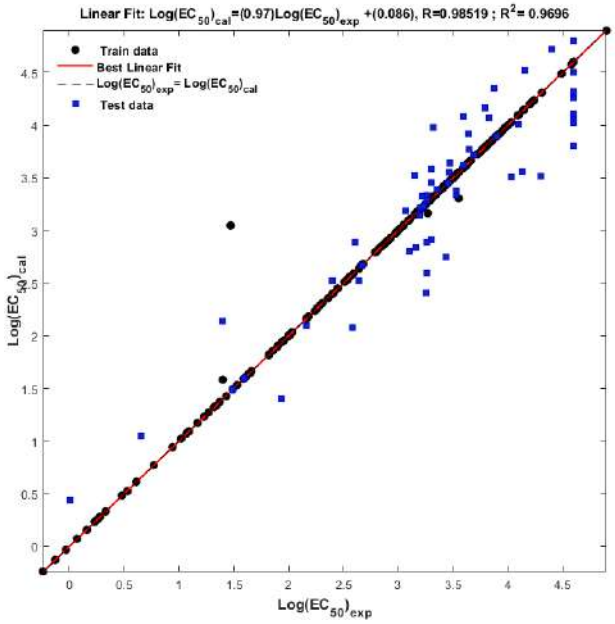


Figure III.7. Scatter plots predicted vs. observed values for the (a) train, (b) test and (c) global sets, for DA_SVM model (Boruta).

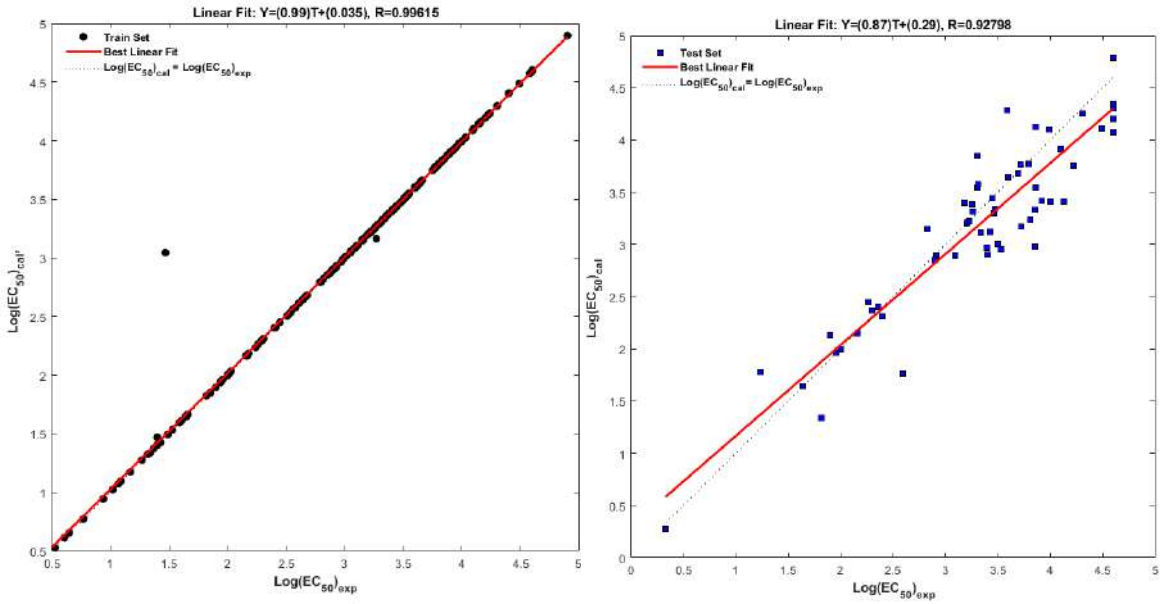


(a) **(b)**

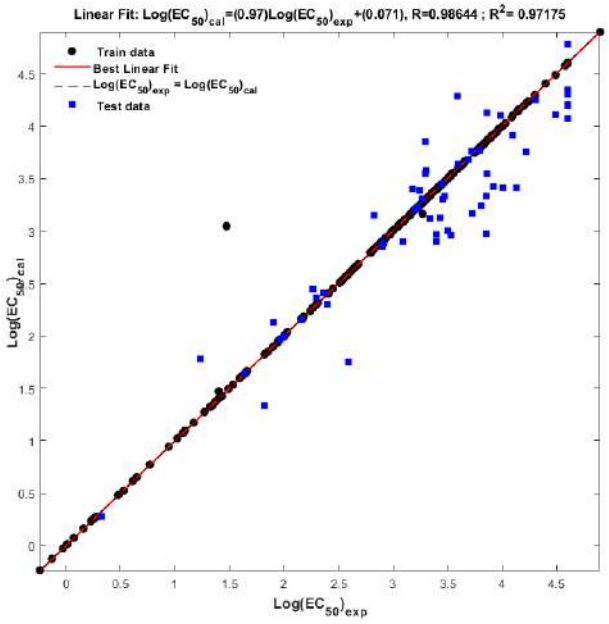


(c)

Figure III.8. Scatter plots predicted vs. observed values for the (a) train, (b) test and (c) global sets, for MFO_SVM model (Boruta).



(a) **(b)**



(c)

Figure III.9. Scatter plots predicted vs. observed values for the (a) train, (b) test and (c) global sets, for GWO_SVM model (Boruta).

Table III.5: Different errors of the three SVM models.

Model	DescriptorSet	RMSE Train	RMSE Test	RMSE Global	MSE Train	MSE Test	MSE Global	MAE Train	MAE Test	MAE Global
DA-SVM	**GA-MLR**	0.1038	0.3435	0.1787	0.0108	0.1180	0.0319	0.0148	0.2567	0.0625
	Boruta	0.1038	0.3540	0.1827	0.0108	0.1253	0.0334	0.0328	0.2732	0.0802
MFO-SVM	**GA-MLR**	0.1068	0.3572	0.1853	0.1068	0.1276	0.0344	0.0174	0.2683	0.0669
	Boruta	0.1031	0.3765	0.1911	0.0106	0.1418	0.0365	0.0117	0.2930	0.0672
GWO-SVM	**GA-MLR**	0.0357	0.4012	0.1811	0.0113	0.1610	0.0328	0.0101	0.2916	0.0656
	Boruta	0.0326	0.3104	0.1838	0.0010	0.0963	0.0302	0.0094	0.2219	0.0598

Table III.6: AIC values of the three SVM model.

Model	Descriptor Set	AIC Value
DA-SVM	**GA-MLR**	-1019.7694
	Boruta	-1014.173
MFO-SVM	**GA-MLR**	-997.5725
	Boruta	-970.235
GWO-SVM	**GA-MLR**	-1011.6243
	Boruta	-993.107

Beyond visual assessment, a quantitative evaluation of the QSTR models' predictive performance was conducted using critical statistical metrics, calculated for the training, test, and global sets. These metrics provide a comprehensive understanding of each model's fit, internal consistency, and generalization capability.Table III.5 and Table III.6 present the calculated error metrics (RMSE, MSE, MAE) and Akaike Information Criterion (AIC) values, respectively, for the three SVM models built with both GA-MLR and Boruta selected descriptors. Analysis of these metrics indicates that all three models, regardless of the descriptor selection method (GA-MLR or Boruta), demonstrated strong performance on the training set, exhibiting very low RMSE, MSE, and MAE values, which confirms their ability to fit the development data effectively.However, when assessing generalization to unseen data (test set) and considering the overall performance across the global dataset, the DA-SVM model consistently showed superior predictive accuracy. For GA-MLR descriptors, DA-SVM outperformed both the MFO-SVM and GWO-SVM models on the test set, maintaining its lead across the global dataset with the

lowest RMSE of 0.1787, MSE of 0.0319, and MAE of 0.0625. Similarly, when using Boruta descriptors, DA-SVM also presented highly competitive error metrics on the test (RMSE = 0.3540, MSE = 0.1253, MAE = 0.27322) and global (RMSE = 0.1827, MSE = 0.0334, MAE = 0.080243) datasets. This consistent accuracy across different data partitions and descriptor sets reinforces its robustness.Furthermore, the lowest AIC value was recorded for the GA-MLR DA-SVM model (–1019.7694), indicating the best trade-off between model accuracy and complexity. The Boruta-based DA-SVM model followed slightly with an AIC of –1014.173, reinforcing its effectiveness. Overall, these combined error and AIC metrics highlight the robustness and efficiency of DA-SVM, particularly when paired with GA-MLR descriptors, while also showing that Boruta is a valid alternative with slightly reduced but still solid predictive performance.To provide a more exhaustive assessment of the models' predictive power and robustness, a range of advanced statistical metrics were also evaluated for each meta-heuristic SVM model under both GA-MLR and Boruta descriptor sets. These include various Q^2 variants, the correlation coefficient (R), the coefficient of determination (R^2), and other validation parameters, as summarized in Table III.7.

Table III.7: Comprehensive statistical metrics for SVM models (GA-MLR and Boruta Descriptors).

Metrics	**Dataset**	**DA-SVM**	**DA-SVM**	**MFO-SVM**	**MFO-SVM**	**GWO-SVM**	**GWO-SVM**
		GA-MLR	**Boruta**	**GA-MLR**	**Boruta**	**GA-MLR**	**Boruta**
	Train	-	-	-	-	-	-
Q^2F_1	Test	0.9201	0.8987	0.8717	0.8641	0.8063	0.8549
	Global	-	-	-	-	-	-
	Train	-	-	-	-	-	-
Q^2F_2	Test	0.9119	0.8980	0.8696	0.8466	0.7791	0.8390
	Global	-	-	-	-	-	-
	Train	-	-	-	-	-	-
Q^2F_3	Test	0.9002	0.8990	0.9019	0.8904	0.8803	0.9023
	Global	-	-	-	-	-	-
	Train	-	-	-	-	-	-
Q^2_{CCC}	Test	0.9583	0.9472	0.9332	0.9191	0.8803	0.9154
	Global	-	-	-	-	-	-
	Train	0.9909	0.9913	0.9911	0.9918	0.9991	0.9923

R^2	Test	0.9199	0.8980	0.8696	0.8466	0.7791	0.8390
	Global	0.9742	0.9721	0.9723	0.9696	0.9735	0.9717
	Train	0.9954	0.9957	0.9956	0.9959	0.9995	0.9961
R	Test	0.9603	0.9478	0.9343	0.9225	0.8893	0.9280
	Global	0.9871	0.9865	0.9861	0.9852	0.9868	0.9864
	Train	0.9989	1.2002	0.9986	0.9986	1.2008	0.9986
k	Test	1.0165	0.9992	1.1985	1.0154	1.0248	1.0384
	Global	1.2024	1.2000	1.2005	1.2023	1.1958	1.1967
	Train	1.2001	0.9988	1.2004	1.2003	0.9991	1.2005
k'	Test	0.9737	0.9896	0.9803	0.9736	0.9631	0.9537
	Global	0.9946	0.9969	0.9963	0.9943	0.9913	0.9903
	Train	1.2000	1.2000	1.2000	1.2000	1.2000	1.2000
R^2_0	Test	0.9978	1.2000	0.9992	0.9966	0.9888	0.9775
	Global	0.9999	1.2000	1.2000	1.2000	0.9997	0.9996
	Train	1.2000	1.2000	1.2000	1.2000	1.2000	1.2000
$R^{2'}_0$	Test	0.9947	0.9990	0.9956	0.9908	0.9779	0.9687
	Global	0.9997	0.9999	0.9999	0.9997	0.9993	0.9992
	Train	-0.0092	-0.0088	-0.0088	-0.0083	-0.0009	-0.0078
M	Test	-0.0847	-0.1136	-0.1489	-0.1773	-0.2690	-0.1651
	Global	-0.0264	-0.0277	-0.0285	-0.0303	-0.0269	-0.0277
	Train	-0.0092	-0.0088	-0.0089	-0.0083	-0.0009	-0.0078
N	Test	-0.0813	-0.1125	-0.1448	-0.1704	-0.2551	-0.1546
	Global	-0.0262	-0.0276	-0.0284	-0.0301	-0.0265	-0.0272
	Train	0.8963	0.8989	0.8985	0.9020	0.9684	0.9052
R_m	Test	0.6632	0.6112	0.5566	0.5186	0.4224	0.5267
	Global	0.8181	0.8134	0.8105	0.8041	0.8161	0.8131
	Train	1.1983	1.0304	1.0131	1.1961	1.1990	1.1977
	Test	1.2025	1.2719	1.1000	1.1696	1.1027	1.0935
A_f	Global	1.0440	1.0741	1.0297	1.0365	1.0268	1.0240
	Train	1.2016	1.1961	1.2029	1.1938	1.2007	1.1943
B_f	Test	1.0580	1.1940	0.9929	1.0497	0.9806	0.9607
	Global	1.0125	1.1973	1.2009	1.0127	0.9967	0.9955

In light of these findings, and across all datasets (train, test, and global), the DA-SVM model consistently demonstrated strong predictive ability and fitted the data exceptionally

well. Both DA-SVM configurations exhibited excellent statistical performance, with metrics such as Q^2CCC, R^2, R, and R^2p0 consistently showing high values close to 1. For the GA-MLR descriptors, key statistics included a test Q^2CCC of 0.9583, a global R^2 of 0.9742, a global R of 0.9871, and a global R^2p0 of 0.9997, alongside an R_m of 0.8181. The Boruta version preserved this strong performance, with test Q^2CCC of 0.9472, global R^2 of 0.9721, global R of 0.9865, global R^2p0 of 0.9999, and an R_m of 0.8134.

The MFO-SVM models also exhibited good to excellent statistical performance, showing strong internal consistency and reasonable external predictability with both descriptor sets. The GWO-SVM model, on the other hand, displayed varying generalization capabilities, its GA-MLR configuration showed a tendency towards overfitting, whereas the Boruta-based GWO-SVM achieved a more balanced and generalizable performance. Overall, the statistical metrics confirm that DA-SVM remains the top-performing model across all criteria and descriptor sets, with both GA-MLR and Boruta based models validating their utility for reliable toxicity prediction and reinforcing the robustness and versatility of the modeling framework.

The distribution of relative errors for the DA-SVM model was visualized using two types of diagrams. Figure III.10 and Figure III.11 illustrate the relative error as a function of toxicity values ($Log(EC_{50})$) for the GA-MLR and Boruta configurations, respectively. For both descriptor sets, the majority of data points were clustered around the zero-error line, confirming a good model fit.In contrast, Figure III.12 presents the overall distribution of relative errors for the two descriptor selection methods. Notably, 77.96% of predictions from the model using GA-MLR descriptors fall within the highly accurate [0–1%] relative error interval, compared to 59.21% for the version using Boruta descriptors. Furthermore, while 7.89% of predictions from the GA-MLR configuration and 9.54% from the Boruta configuration exceed a 10% relative error, both remain within acceptable predictive bounds for practical modeling applications. Together, these results reinforce the conclusion that both descriptor selection approaches enable accurate toxicity predictions. The GA-MLR configuration shows a slight advantage in overall precision, while the Boruta configuration maintains robust performance across the full toxicity range, validating its effectiveness as a reliable alternative.

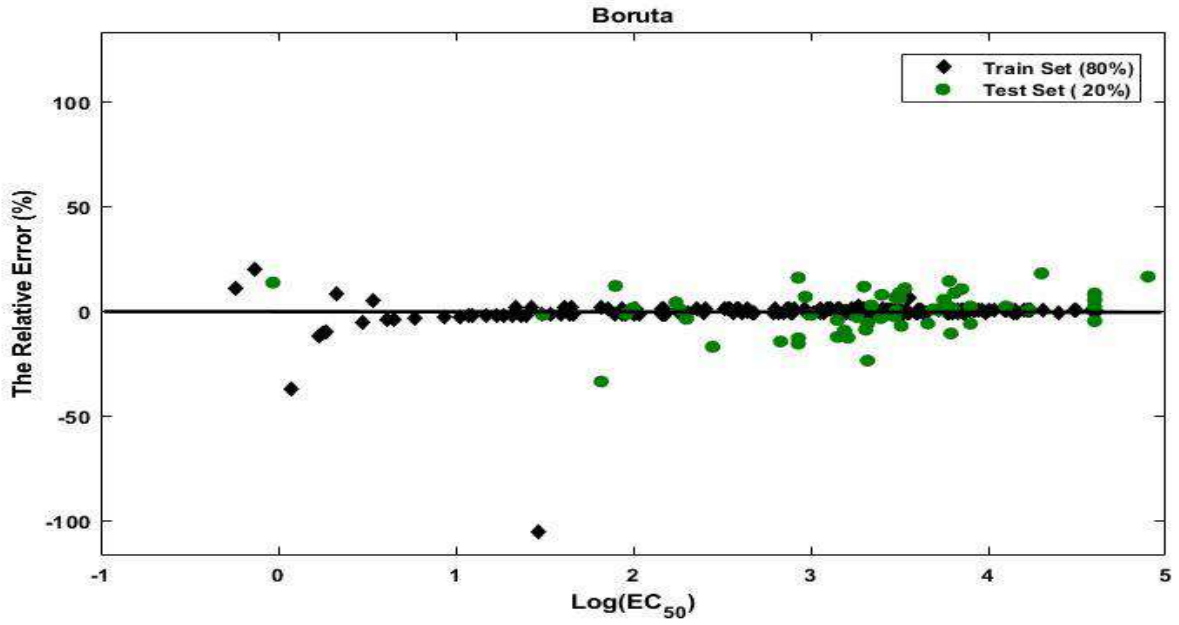


Figure III.10. The relative error (%) of the toxicity by the DA_SVM model (GA_MLR descriptors).

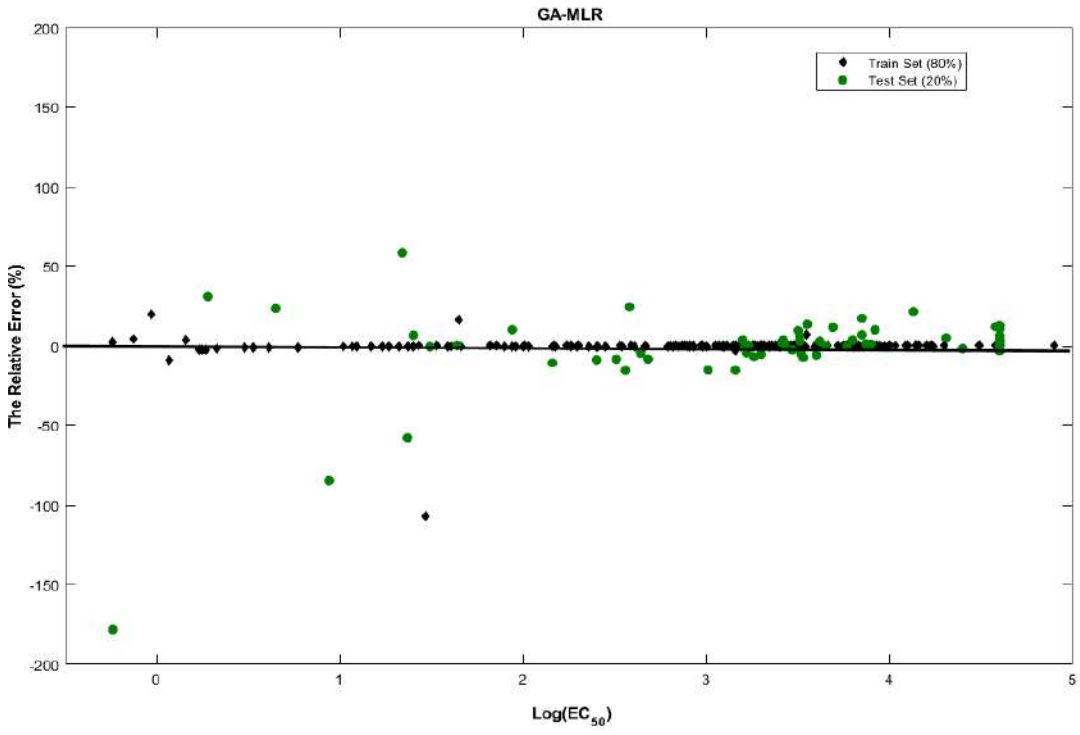


Figure III.11. The relative error (%) of the toxicity by the DA_SVM model (Boruta descriptors).

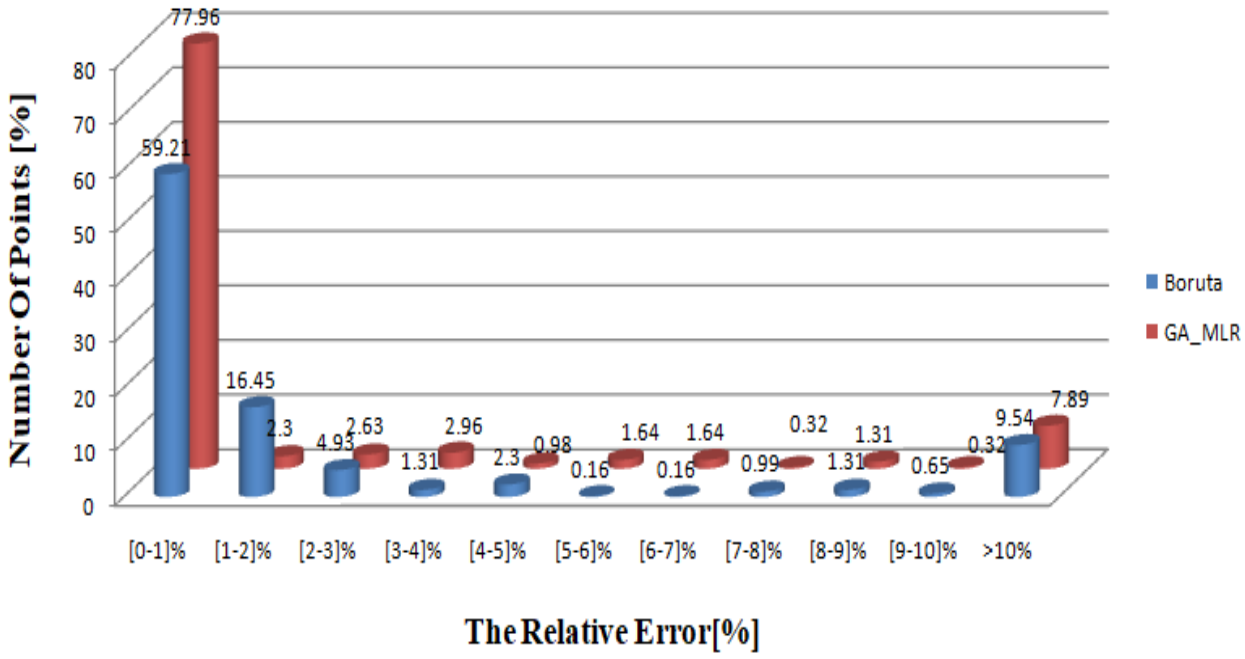


Figure III.12. Distribution diagrams of relative error (%) of the DA_SVM model.

III.4.2. Influence of Molecular Descriptors

The relevance factors quantify the influence of each descriptor on the model's predictive output. Positive values indicate a direct contribution to toxicity, while negative values suggest an inverse relationship. Figure III.13 and Figure III.14 illustrate the relevance factors of the molecular descriptors used in the DA-SVM models constructed with GA-MLR and Boruta-selected features, respectively.

In the GA-MLR-based model (Figure III.13), the toxicity of ionic liquids was significantly influenced by three descriptors: SIC2, B06[C-O], and P_VSA_ppp_N. These descriptors showed a direct correlation with toxicity. The remaining descriptors exhibited an inverse relationship to toxicity. Among all descriptors, SIC2 and B09[C-C] were identified as the most relevant input variables, with relevance factors of +0.5448 and -0.7165, respectively, indicating their strong impact on predicting toxicity levels. In contrast, descriptors MATS5i and P_VSA_ppp_N were deemed less relevant, with relevance factors of -0.0177 and +0.2236, respectively.

Conversely, the Boruta-based model (Figure III.14) incorporated a broader and more diverse set of descriptors, certain descriptors exhibit a strong influence on the model's performance. ESOL has the highest positive relevance factor at 0.5945, indicating

a strong direct correlation with toxicity. Notably, B09[C-C] once again stands out as a highly influential descriptor, exhibiting the most negative relevance factor at -0.7165, which signifies a substantial inverse relationship with toxicity, consistent with its role in the GA-MLR model. Other descriptors, such as MATS2e and JGI7, have relatively low relevance factors. Specifically, JGI7 has a very small relevance factor of 0.0031 indicating it has very little influence on the model.

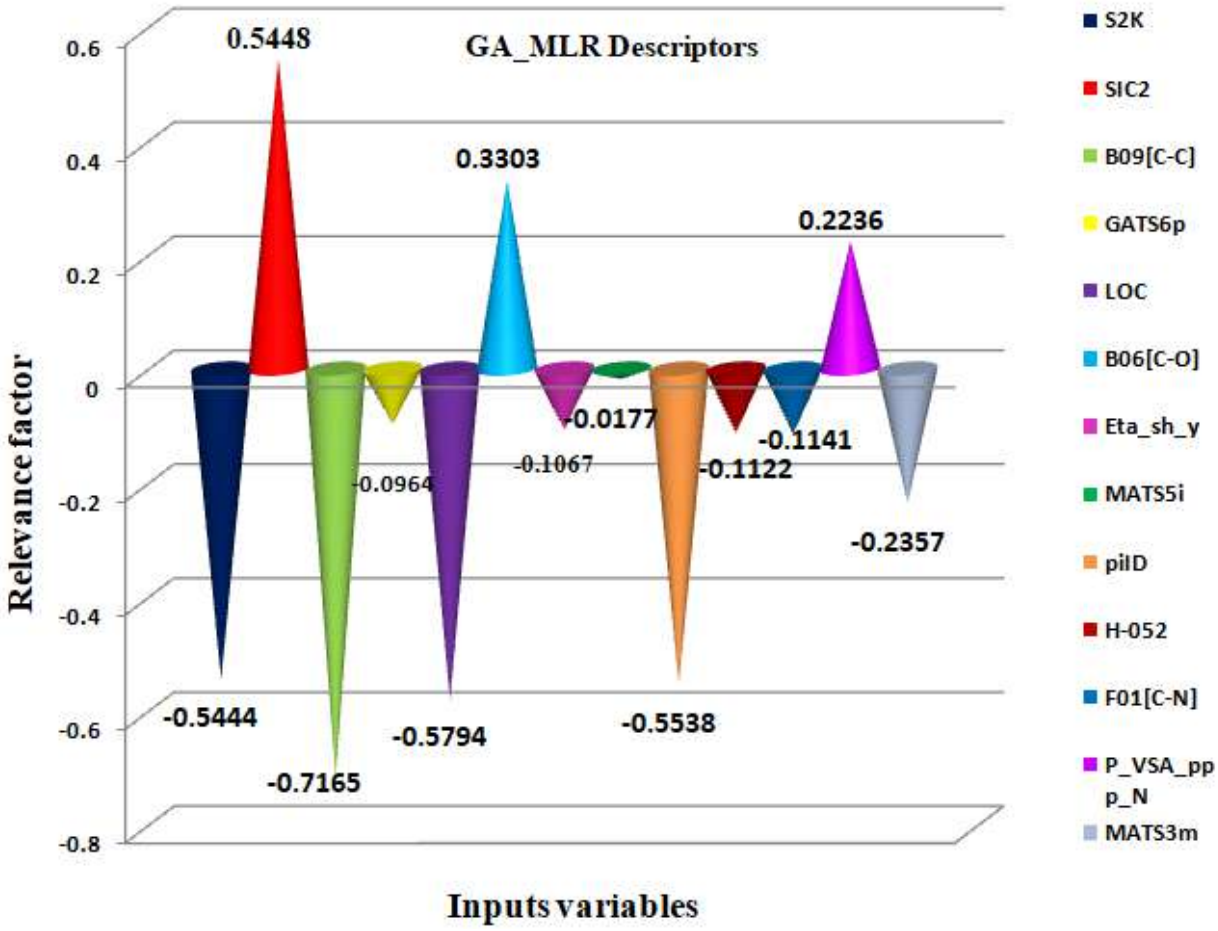


Figure III.13. The results of the relevance factor performed on the DA_SVM model (GA_MLR descriptors).

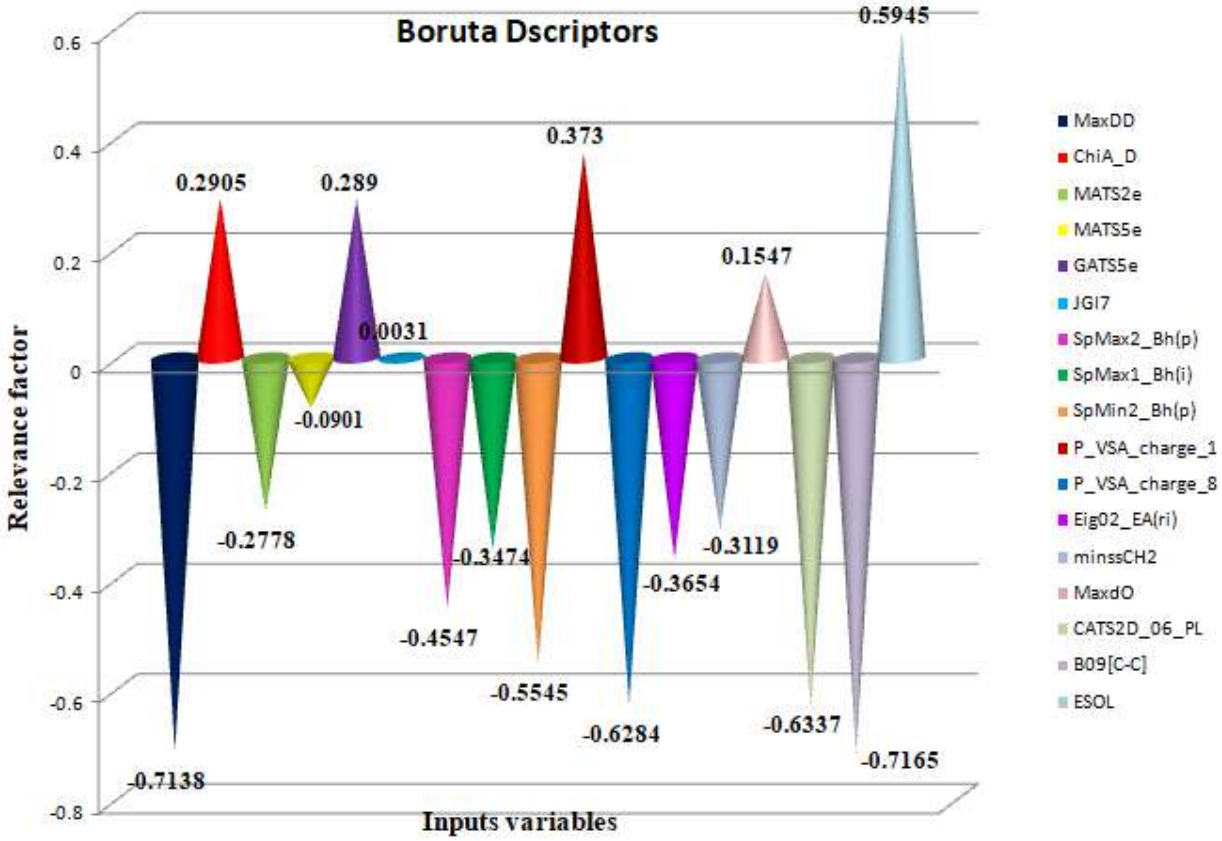


Figure III.14. The results of the relevance factor performed on the DA_SVM model (Boruta Descriptors).

Analyzing the correlation among selected molecular descriptors is essential to understand potential redundancies and assess multicollinearity within the feature set. A correlation coefficient with an absolute value greater than 0.75 is often considered indicative of a strong linear relationship that might lead to problematic multicollinearity in models[160]. Heat maps provide a visual representation of these inter-descriptor correlations.

Figure III.15 presents the correlation matrix for the GA-MLR-selected descriptors. The analysis reveals that this subset is generally well-balanced, with all absolute correlation coefficients remaining below the 0.75 threshold. For example, a moderate positive correlation is observed between LOC and S2K (0.56), while S2K and SIC2 exhibit a moderate negative correlation (–0.47). A weak correlation such as between S2K and B06[C–O] (0.00) further demonstrates minimal linear dependence. These findings confirm that the GA-MLR approach effectively filtered out highly correlated descriptors, ensuring that the selected variables provide complementary and non-redundant information for modeling.

A similar analysis was conducted for the Boruta-selected descriptors, as shown in Figure III.16. The resulting heat map reveals that the vast majority of descriptor pairs exhibit low to moderate correlations, reflecting a generally diverse and non-redundant feature set. Notably, only three descriptor pairs slightly exceed the commonly accepted threshold of 0.75. However, none of these correlations surpass 0.79, indicating that the observed linear associations are modest and unlikely to pose serious multicollinearity concerns. This outcome confirms that, overall, the Boruta algorithm was also successful in selecting descriptors with minimal redundancy, thereby maintaining the quality and robustness of the QSTR modeling process.

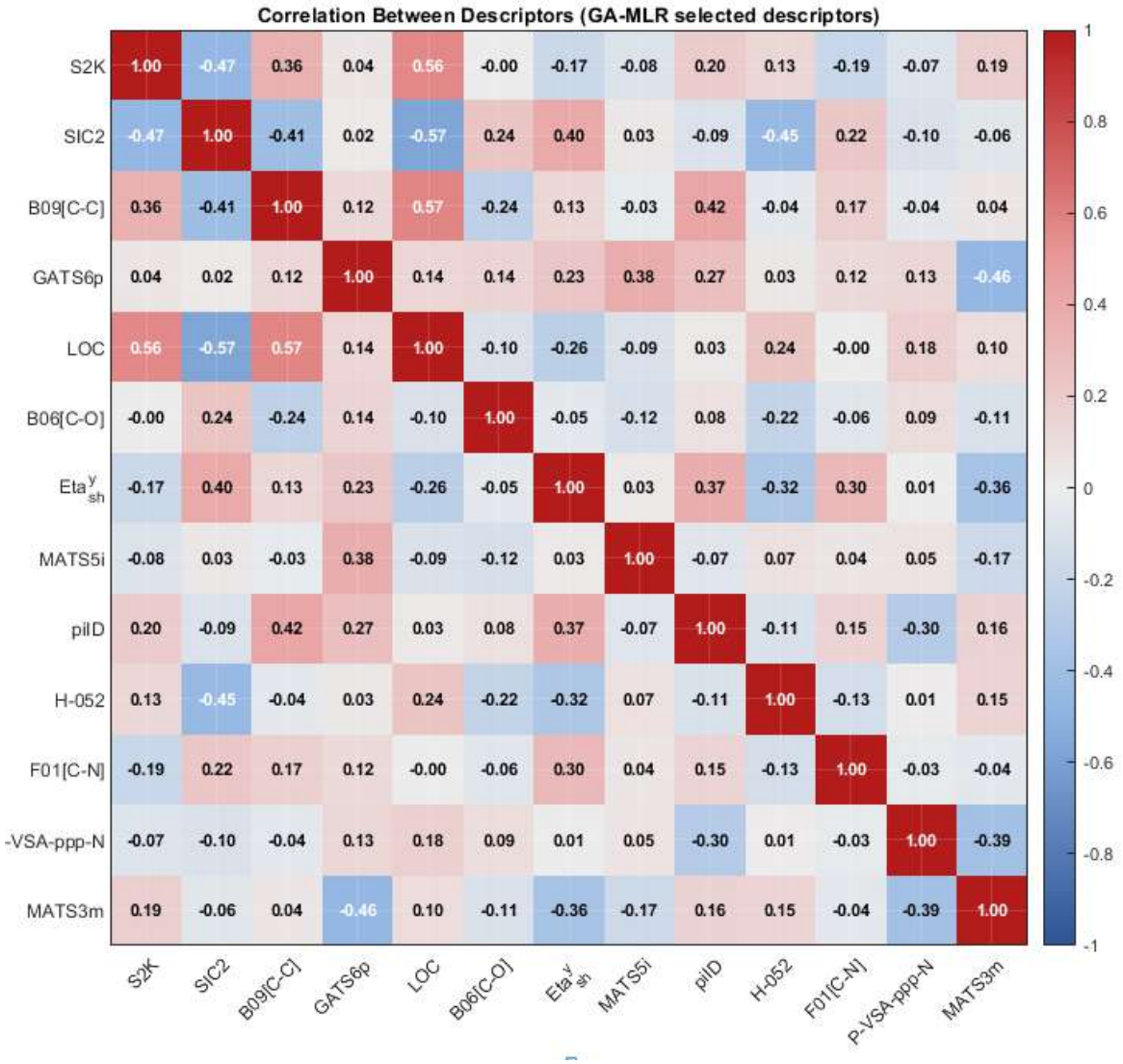


Figure III.15. Correlation matrix of the thirteen descriptors selected by GA_MLR.

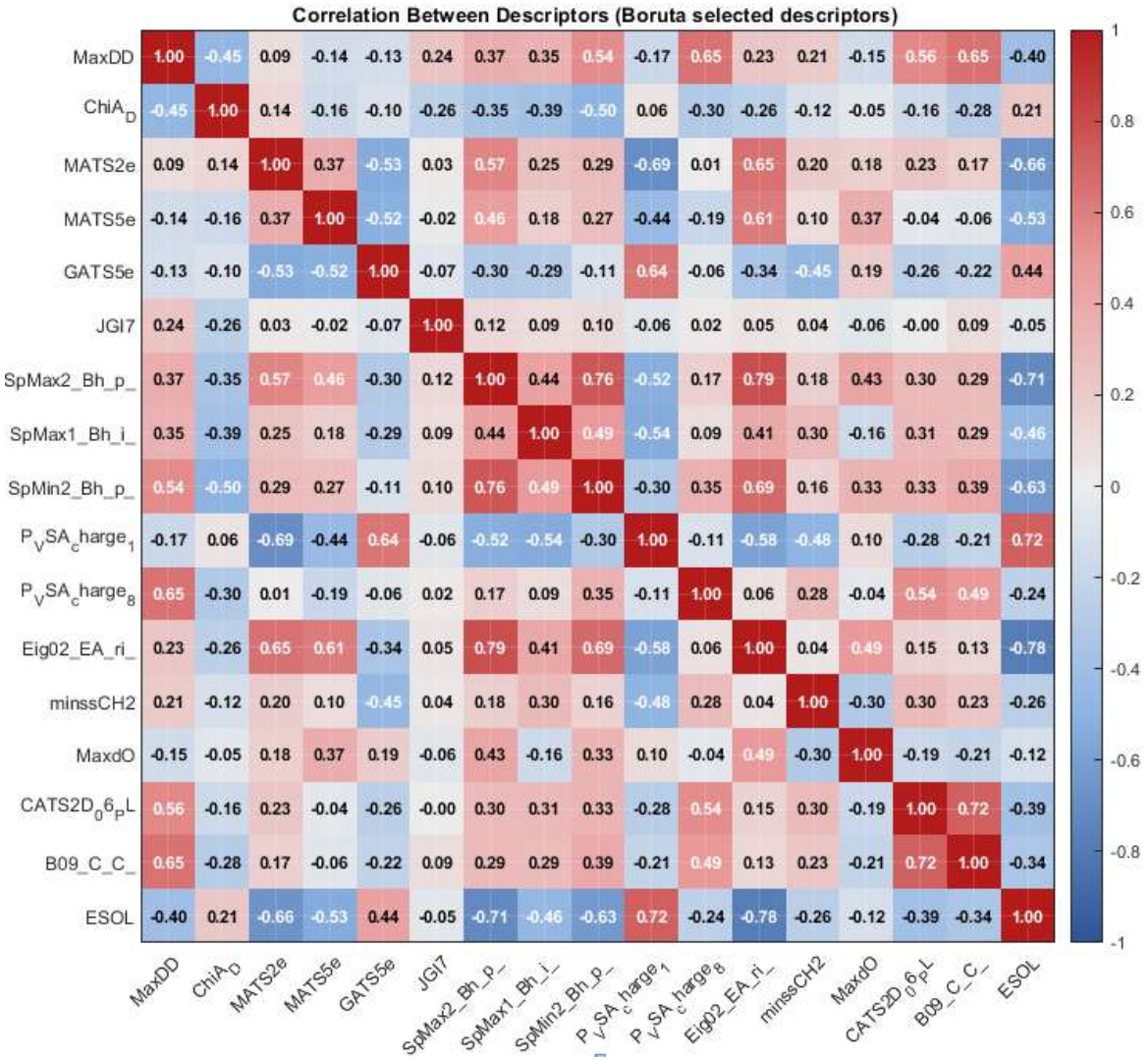


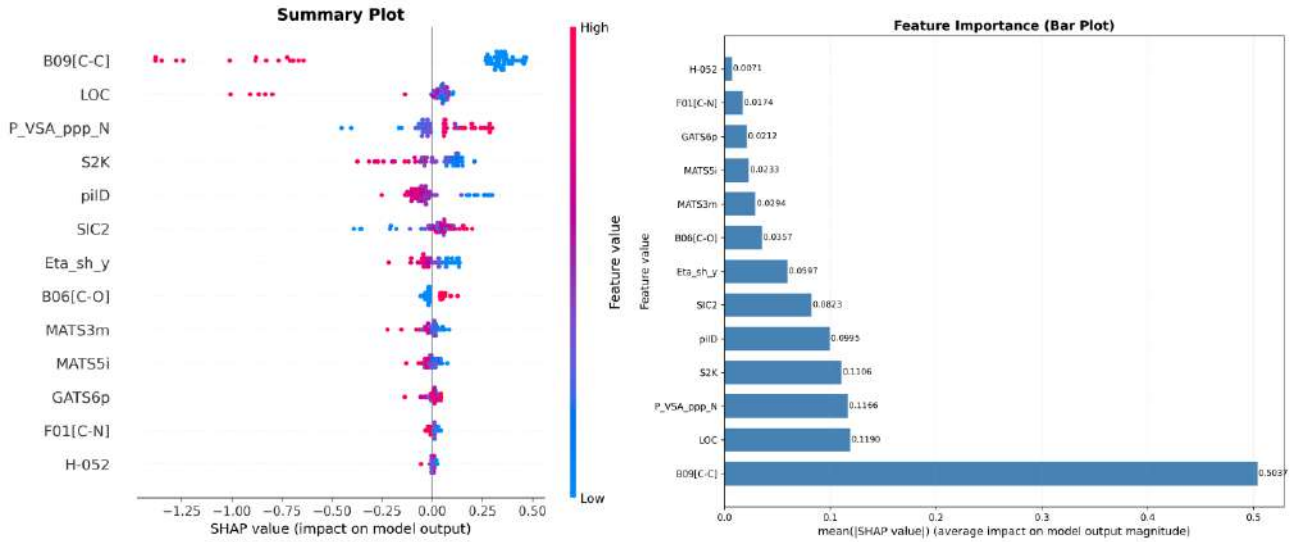
Figure III.16. Correlation matrix of the seventeen descriptors selected by Boruta.

III.4.3. SHAP Analysis

To understand the intricate relationships between molecular descriptors and ionic liquid toxicity, a detailed analysis of feature importance is essential. Beyond traditional correlation methods, SHAP (SHapley Additive exPlanations) analysis provides a powerful game-theoretic approach to interpret the output of machine learning models. SHAP values quantify the contribution of each feature to the prediction for a given instance, offering insights into both the magnitude and direction (positive or negative) of a feature's impact on the model's output[161–163].

The SHAP analysis for the model developed using the GA-MLR selected descriptors reveals the specific contributions of these features to the prediction of Log(EC_{50}) values.

Figure III.17 provides a comprehensive view of feature importance for the GA-MLR selected descriptors. The SHAP bar plot (b) ranks features by their overall importance (mean absolute SHAP value)[164], while the SHAP summary plot (a) details the individual impact of each feature on specific predictions. In the summary plot[163], each data point represents an ionic liquid instance, with its horizontal position indicating the feature's contribution to that prediction (positive values increase Log(EC_{50}), negative values decrease it).The color of each data point indicates the feature's actual value (red for high, blue for low), revealing directional relationships. Features are ordered by their average impact from top to bottom.

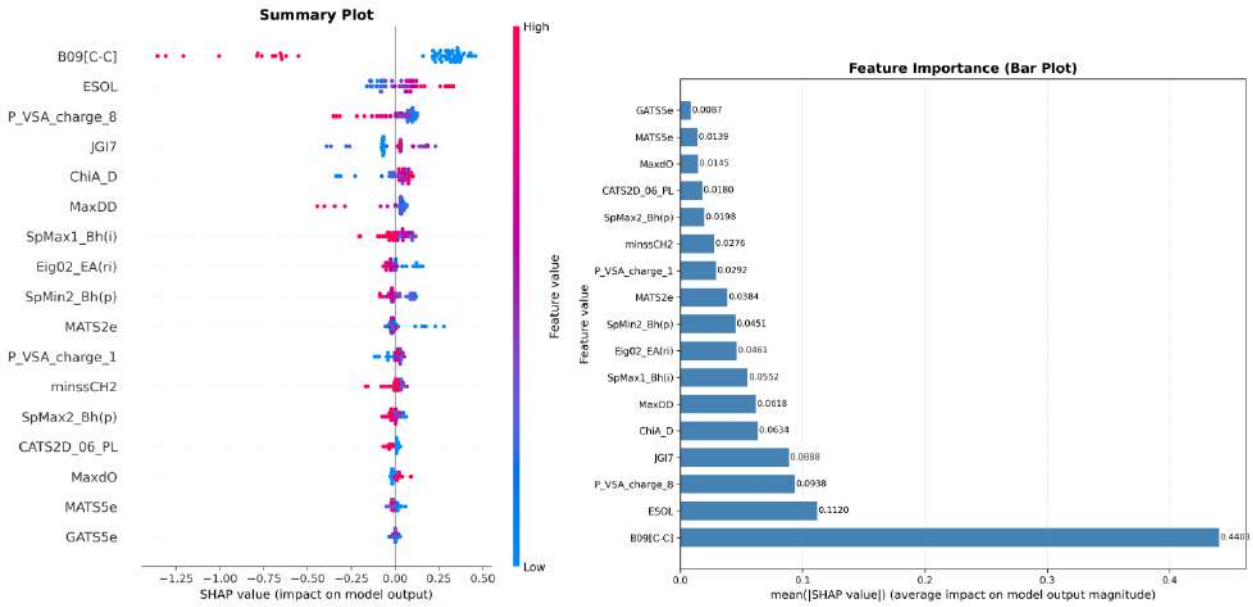


(a) (b)

Figure III.17. SHAP interpretability and features importance analysis for the detected Log(EC_{50}) DA_SVM model for GA_MLR selected descriptors: summary plots (a) and Bar plot (b).

The SHAP summary plot (Figure III.17a) provides a detailed understanding of how each feature influences the model's predictions. Each dot on the plot represents a single data instance (an ionic liquid), and its horizontal position indicates the SHAP value, which is the impact of that feature on the model's output for that specific instance. A positive SHAP value pushes the prediction higher (towards higher Log(EC_{50})), while a negative SHAP value pushes it lower (towards lower Log(EC_{50})). The color of each dot corresponds to the feature's actual value for that instance: red signifies a high feature value, and blue signifies a low feature value. This coloring allows us to discern the directional relationship between a feature's magnitude and its impact on the prediction. Features are ranked

vertically by their overall importance, with the most impactful at the top. From these plots, B09[C-C] clearly stands out as the most influential descriptor. In the summary plot (b), high B09[C-C] values (represented by red data points) are consistently associated with large negative SHAP values, strongly pushing the predicted Log(EC_{50}) downward (implying higher toxicity). Conversely, low values (represented by blue data points) push predictions upward (lower toxicity), establishing a robust inverse relationship.LOC and P_VSA_ppp_N are also highly impactful. For LOC, the spread of red and blue data points across both positive and negative SHAP values indicates a more complex interaction, where its effect on toxicity might depend on other factors. P_VSA_ppp_N shows a tendency for higher values (red data points) to slightly increase predicted Log(EC_{50}) (positive SHAP values). Other descriptors, such as H-052, F01[C-N], and GATS6p, show minimal influence, with their SHAP values tightly clustered around zero and small average impacts in the bar plot (b). This granular analysis illuminates the specific contributions of the GA-MLR selected features to the toxicity predictions.The SHAP analysis was also performed for the model developed using the Boruta-selected descriptors to understand their individual contributions to the prediction of Log(EC_{50}) values, as illustrated in Figure III.18.



(a) (b)

Figure III.18. SHAP interpretability and features importance analysis for the detected Log(EC_{50}) DA_SVM model for Boruta selected descriptors: summary plots (a) and Bar plot (b).

As highlighted by the bar plot (b), B09[C-C] again emerges as the most significantly impactful feature. Its behavior in the summary plot (b) is strikingly consistent with the GA-MLR model: higher values (represented by red data points) primarily drive predictions towards lower Log(EC_{50}) (higher toxicity), while lower values (represented by blue data points) push predictions towards higher Log(EC_{50}) (lower toxicity). This reinforces B09[C-C]'s robust inverse relationship with toxicity, regardless of the feature selection method.Following B09[C-C], ESOL and P_VSA_charge_8 are also highly influential. For ESOL, higher feature values (represented by red data points) are consistently associated with positive SHAP values, meaning they tend to increase the predicted Log(EC_{50}) (decrease toxicity). Conversely, lower ESOL values (represented by blue data points) generally lead to negative SHAP values (increase toxicity). P_VSA_charge_8 shows a more dispersed impact, with both high and low values contributing to predictions in both directions, suggesting its influence is more nuanced or dependent on interactions with other features. Other features, such as JG17, ChiA_D, and MaxDD, also show notable contributions, with their red and blue data points indicating various directional influences. Features at the bottom of the bar plot (b), like GATS5e and MATS5e, exhibit minimal average impact, with their SHAP values tightly centered around zero in the summary plot (a). This comprehensive SHAP analysis for Boruta-selected features provides a clear, interpretable view of the key molecular properties driving the toxicity predictions.

III.4.4. Applicability Domain

QSTR model should have an applicability domain (AD), which is a theoretical region defined by the descriptors used in model and the studied response, which can be characterized in various ways. Two complementary approaches were employed in this study to evaluate the AD of the DA-SVM model developed with both GA-MLR and Boruta descriptor sets: William’s plot and the Insubria graph[165].

III.4.4.1.William’s Plot

The leverage approach (Williams plot) was used to verify the prediction reliability and identify outliers and the compounds residing outside the AD. In this plot, the two horizontal lines indicate the limit of normal values for Y outliers (i.e. samples with standardized residuals greater than 3.0 standard deviation units, ±3.0s), the vertical straight

lines indicate the limits of normal values for X outliers (i.e. samples with leverage values greater than the threshold value, h > h*) [166–168].

The leverage hi is defined as follows [169] :

$$hi = x_i(X^T X)^{-1} x_i$$

Where xi is the descriptor row-vector of the query and X is the (n × k) matrix of the training set, k is the number of model descriptors and n is the number of training set samples. The warning leverage h*(the limit of normal values for X outliers) is given by [166]:

$$h^* = \frac{3(p+1)}{n}$$

Where n is the total number of samples in the training set, and p is the number of descriptors involved in the correlation.

The Williams plots for the DA-SVM models based on GA-MLR and Boruta descriptor sets are shown in Figures III.19 and III.20, respectively. In these plots, the standardized residuals (Y-axis) are plotted against the leverage values (X-axis), serving as a diagnostic tool to identify influential data points and assess model reliability.

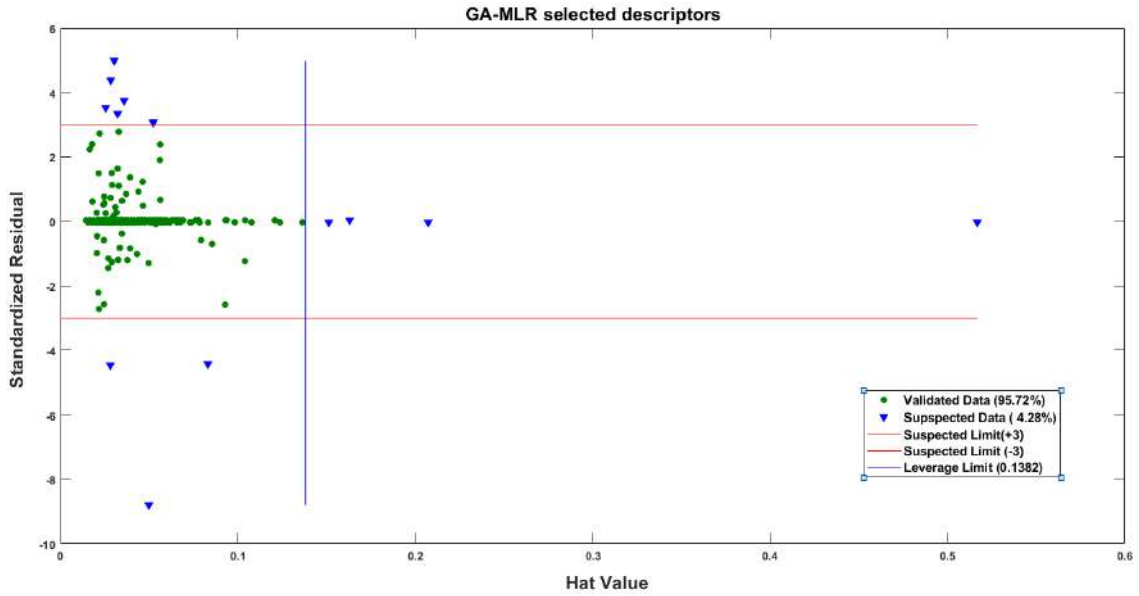


Figure III.19. William's plot for DA_SVM model (GA_MLR selected descriptors).

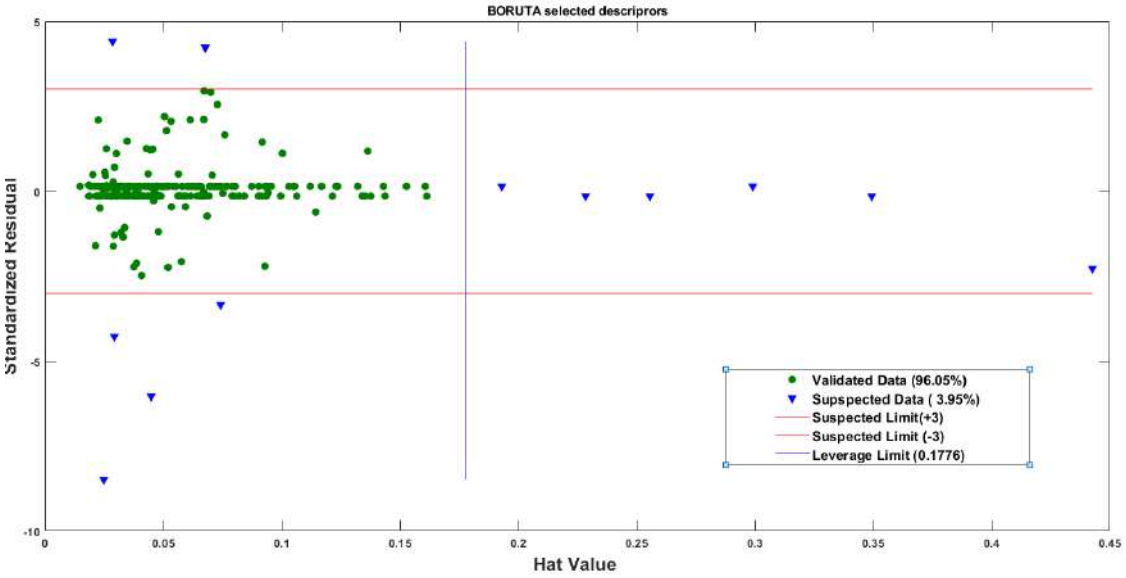


Figure III.20. William's plot for DA_SVM model (Boruta selected descriptors).

Upon examination, it is evident that the majority of compounds, specifically 95.72%, fall within the acceptable AD region (standardized residuals between ±3 horizontal lines and leverage below h* = 0.1382) for the GA-MLR model. However, only a small proportion of data points (4.28%) are observed beyond this range, indicating a well-constrained and reliable predictive space.

Similarly, in the Boruta-based model (Figure III.20), the leverage threshold was slightly higher (h* = 0.1776) due to the increased number of descriptors. Here, 96.05% of the data points are within the applicability domain, and only 3.95% were identified as potential outliers. Notably, the majority of these points designated as 'Suspected Data' had leverage values greater than h*, suggesting they might have structural properties not well-represented in the training set.

Overall, the Williams plots confirm that the DA-SVM models built with both descriptor selection methods operate within a well-defined applicability domain and demonstrate strong robustness in predicting the toxicity of ionic liquids.

III.4.4.2. Insubria Graph

Complementary to the Williams plot, the Insubria Graph provides an alternative visualization of the Applicability Domain. It provides a visualization of the interpolated and extrapolated predictions, the compounds with h>h* are considered outside of the structural domain, the graph is also a scatter plot in two dimensions. The x-axis is the

leverage value (Hat) , which is defined the same as that of the Williams graph, and the y-axis is the predicted Log(EC50) value of the model for both training and test set[35, 41, 42, 43]. The predicted results are reliable, if both conditions: hi< h* and $Log(EC50)_{(Min)}<Log(EC50)_{(Pred)}<Log(EC50)_{(Max)}$. The terms "$Log(EC_{50})_{(Min)}$ " and "$Log(EC_{50})_{(Max)}$" refer to the the minimum (-0.24) and maximum (4.9) value of observed toxicity in the training set, respectively.

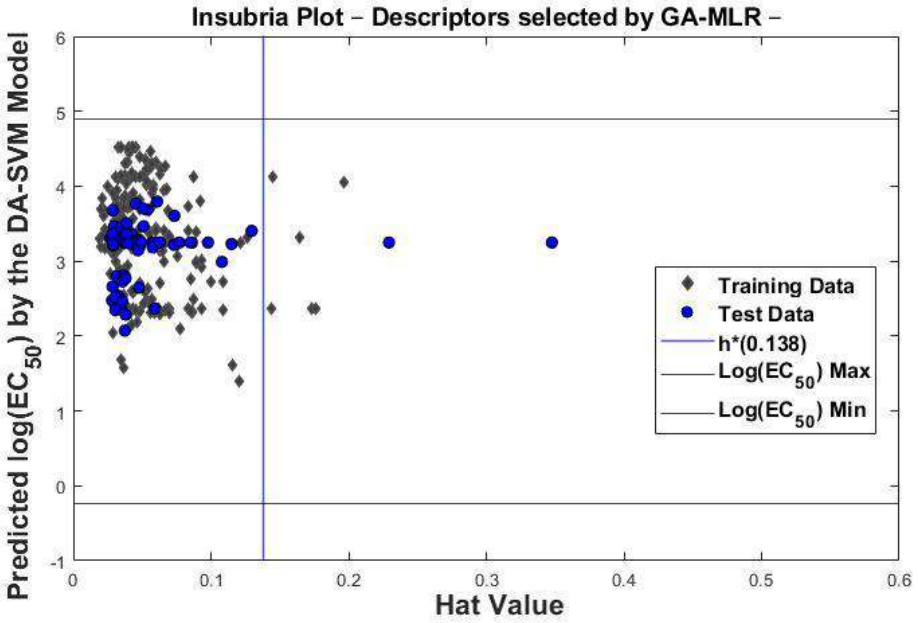


Figure III.21. Insubria graph for the DA_SVM model (descriptors selected by GA_MLR).

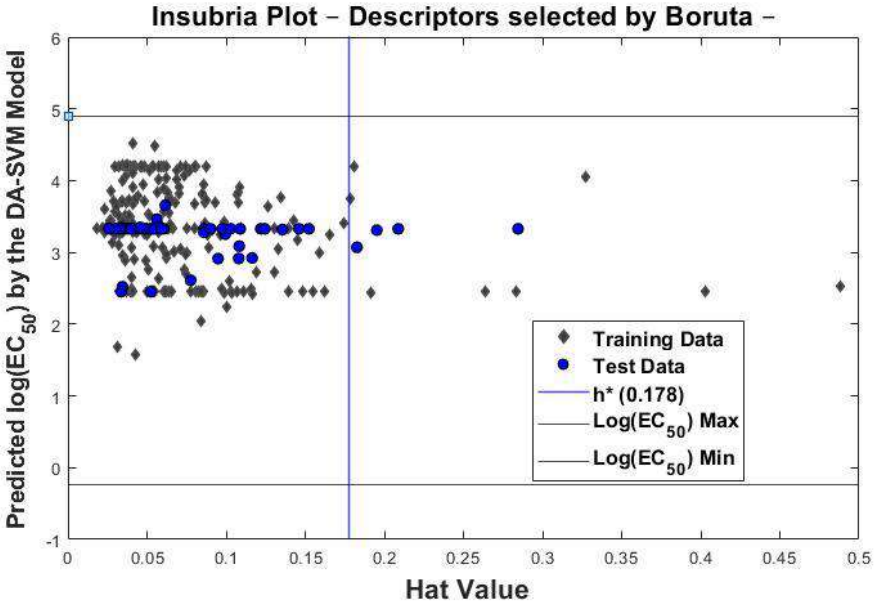


Figure III.22. Insubria graph for the DA_SVM model (descriptors selected by Boruta).

Figures III.21 and III.22 display the Insubria plots for the DA-SVM models constructed with GA-MLR and Boruta selected descriptors, respectively. Both models show strong internal applicability domain coverage. For the GA-MLR-based model, 96.7% of the test set and 97.1% of the training set samples lie within the AD, with only minor extrapolations. The Boruta-based model presents similarly robust results, with 93.3% of the test set and 96.7% of the training set falling within the AD. Only a small number of compounds exceeded the leverage threshold or fell outside the response limits, confirming the structural representativity of both training sets. These findings reinforce the reliability of both models across the majority of the chemical space, with slightly better coverage observed in the GA-MLR configuration.

III.4.5. External Test

External validation was conducted to assess the predictive generalization ability of the DA-SVM models on an independent set of 14 ionic liquids not involved in model training or internal testing. Figures III.23 and III.24 display the scatter plots of predicted versus experimental Log(EC_{50}) values for the GA-MLR-based and Boruta-based models, respectively.

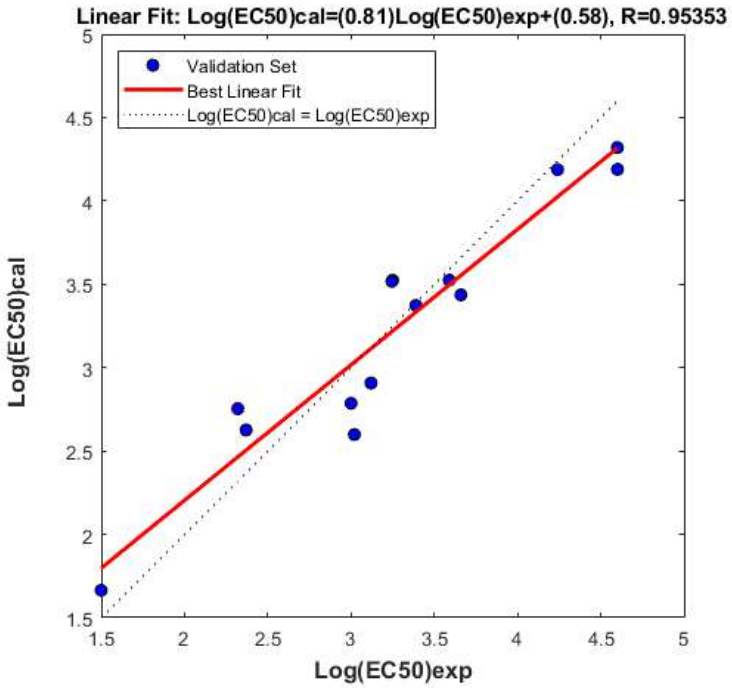


Figure III.23. Scatter Plot of Predicted vs. Observed Values for DA_SVM in External Testing (GA_MLR).

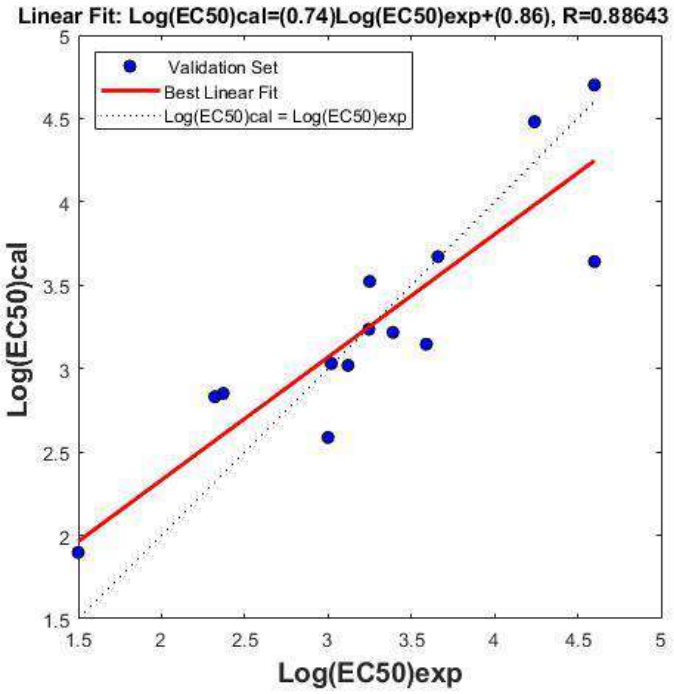


Figure III.24. Scatter Plot of Predicted vs. Observed Values for DA_SVM in External Testing (Boruta).

In Figure III.23, the GA-MLR-based DA-SVM model exhibits a strong linear relationship between predicted and observed values, with the data points closely aligned around the ideal diagonal line. This visual alignment indicates high predictive consistency and reliability across the external validation set, as indicated by a correlation coefficient of R = 0.95353.In comparison, Figure III.24 shows the Boruta-based model results, which reveal a slightly more dispersed distribution of data points around the diagonal. Although the alignment is slightly less strong, the general trend between predicted and observed toxicity values remains present, suggesting an acceptable level of prediction accuracy. To quantitatively support these observations, Table III.8 summarizes three key error metrics: MSE, RMSE, and MAE, for both models:

Table III.8: Prediction Errors for the External Validation Set for DA_SVM model.

Model	MSE	RMSE	MAE
GA–MLR(Selected Descriptors)	0.0716	0.2676	0.2356
Boruta(Selected Descriptors)	0.1514	0.3891	0.2945

The GA-MLR-based model achieved lower error values across all errors, confirming its superior performance in terms of external predictive accuracy and demonstrating that the model performs reliably even on unseen data. The Boruta-based model, while showing slightly higher errors, still delivered acceptable results, with all values remaining within reasonable limits for QSTR applications. These findings confirm that both models are capable of making reliable external predictions, with the GA-MLR configuration offering a more precise estimation and the Boruta configuration providing a viable alternative. Additionally, Figure III.25 presents violin plots of the residuals, which represent the differences between the observed and predicted toxicity (Log EC50)values generated both GA-MLR and Boruta-based models on the external validation set.Upon examination, the median line (white dot) for the GA-MLR model is closely aligned with zero, indicating that this model provided predictions with minimal bias. Moreover, the narrow width of the violin plot for GA-MLR, particularly around the median, signifies a high concentration of residuals very close to zero, suggesting excellent predictive consistency. It is also evident that the lines representing the 25% quartile (lower part of the black box) and the 75% quartile (upper part of the black box) are much closer to the median line for the GA-MLR model. This observation is crucial as it indicates a highly balanced and compact distribution of residuals around zero, demonstrating its accuracy in providing residuals with minimal magnitude between predicted and observed values.

In comparison, the violin plot for the Boruta model is wider and more spread out along the Y-axis. While its median is also centered near zero, the wider sections around the median and the greater distance between its 25th and 75th quartiles indicate a broader distribution of residual values. This suggests that the Boruta model's predictions, though acceptable, exhibit a larger variability in their errors compared to the GA-MLR model.

The results visually confirm that the GA-MLR model is superior in terms of residual distribution, exhibiting outstanding predictive accuracy and minimal bias with residuals consistently close to zero. This demonstrates its remarkable robustness and reliability in predicting the toxicity of ionic liquids.

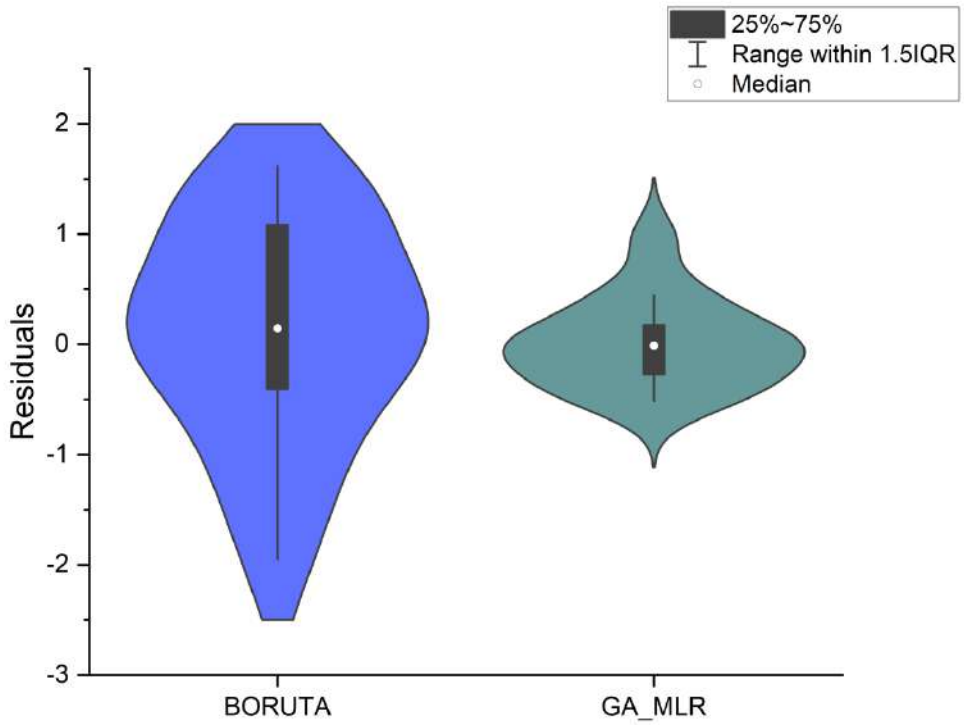


Figure III.25. Violin Plot for Comparative analysis of residuals in the external test set.

III.4.6. Analysis of Activity and Prediction Cliffs using the ARKA Method

This section employs the ARKA (Arithmetic Residuals in K-groups Analysis) method, a dimensionality reduction technique specifically designed for datasets with low sample sizes and multiple features, which often complicate modeling efforts. The ARKA method was applied to identify activity cliffs within a dataset of chemical compounds. Activity cliffs are regions where minor structural modifications result in substantial changes in biological activity, posing significant challenges in predictive modeling. By computing and analyzing $ARKA_1$ and $ARKA_2$ descriptors, this study highlights compounds that exhibit these critical activity cliffs, offering valuable insights for refining structure-activity relationship models.

III.4.6.1. ARKA descriptor computation and dataset description

To identify potential activity and prediction cliffs, ARKA descriptors were computed from the same subset of descriptors previously selected through the GA-MLR feature selection method. This ensures that the resulting $ARKA_1$ and $ARKA_2$ values retain the most relevant structural information already proven significant in toxicity modeling.

The ARKA descriptors were generated using the ARKAdesc-v2.0 software, a Java-based tool developed by DTC Lab, which implements the Arithmetic Residuals in K-groups Analysis (ARKA) algorithm. This approach offers a supervised dimensionality reduction technique optimized for small datasets with a relatively high number of descriptors[173]. A simplified workflow illustrating this computation is presented in Figure III.26.

The methodology for calculating ARKA 1 and ARKA 2 descriptors primarily involves preparing the training set descriptors by normalizing them and then computing mean differences between active and inactive classes. These differences are used to cluster descriptors into two groups (Class 1 for positive differences, Class 2 for negative differences) and assign a specific weight to each. The weightage of a descriptor is calculated as the ratio of its individual mean difference value to the sum of the mean difference values of all descriptors within that particular class:

$$\boldsymbol{Weightage\ of\ a\ descriptor} = \frac{\boldsymbol{difference\ value\ of\ a\ escriptor}}{\sum \boldsymbol{difference\ values\ of\ all\ the\ descriptors\ in\ the\ particular\ class}} \text{(III.22)}$$

Subsequently, the original QSAR descriptors for both training and test sets are standardized. Finally, ARKA 1 and ARKA 2 are computed as the weighted sum of these standardized descriptors belonging to their respective classes, following the general formula:

$$\boldsymbol{ARKA_X} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} (\boldsymbol{Wi} \times \boldsymbol{Xi}) \text{(III.23)}$$

Where X is the descriptors selected by GA_MLR values and w is the corresponding weightage.

This systematic approach transforms the original set of features into two ARKA descriptors, which are then utilized for subsequent analysis of activity and prediction cliffs.

$$\boldsymbol{ARKA_1} = W_2 \times X_2 + W_6 \times X_6 + W_{12} \times X_{12} \quad \text{(III.24)}$$

$$\boldsymbol{ARKA_1} = W_2 \times SIC2 + W_6 \times B06[C\text{–}O] + W_{12} \times P_VSA_ppp_N \quad \text{(III.25)}$$

$$\boldsymbol{ARKA_2} = W_1 \times X_1 + W_3 \times X_3 + W_4 \times X_4 + W_5 \times X_5 + W_7 \times X_7 + W_8 \times X_8 + W_9 \times X_9 + W_{10} \times X_{10} + W_{11} \times X_{11} + W_{13} \times X_{13} \quad \text{(III.26)}$$

$$\boldsymbol{ARKA_2} = W_1 \times S2K + W_3 \times B09[C\text{–}C] + W_4 \times GATS6p + W_5 \times LOC + W_7 \times Eta_{sh_y} + W_8 \times MATS5i + W_9 \times piID + W_{10} \times H - 052 + W_{11}F01[C\text{–}N] + W_{13} \times MATS3m \quad \text{(III.27)}$$

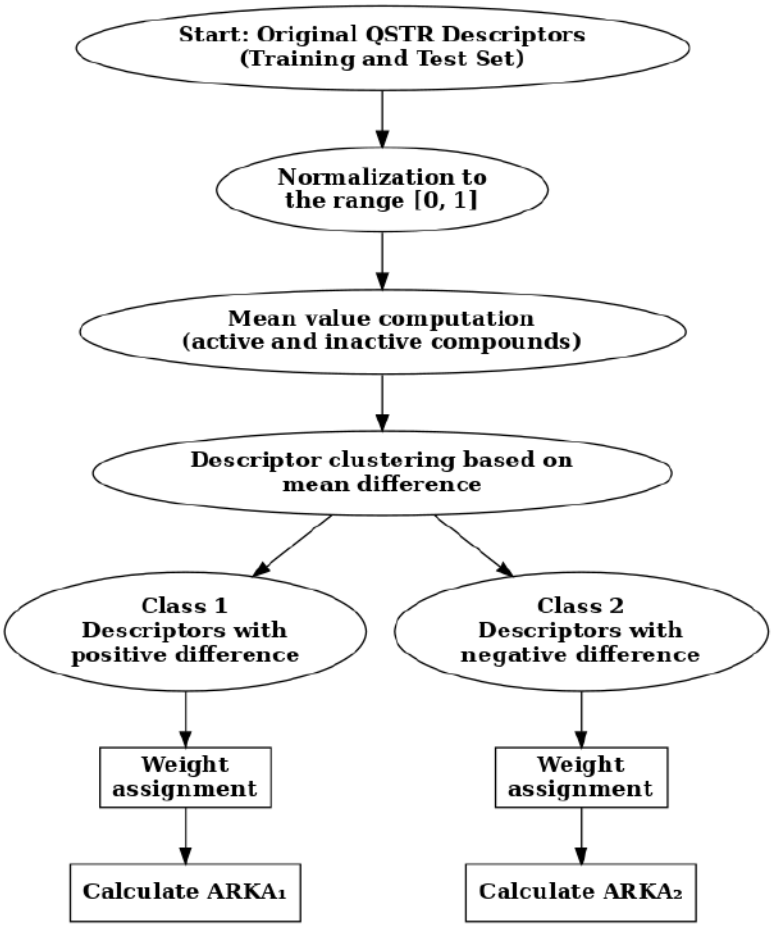


Figure III.26. Process for calculating ARKA descriptors.

III.4.6.2. Violin Plots (ARKA Descriptors vs. Activity)

To gain deeper insight into how ARKA descriptors distinguish between active and inactive compounds, violin plots were employed to visualize the distribution of $ARKA_1$ and $ARKA_2$ values across the two activity classes. These plots, presented in Figure III.27, represent data density, central tendency (median), and interquartile range, providing a comprehensive comparison between active and inactive groups in both the training and test sets. Analysis of these violin plots reveals distinct patterns. In the training set, $ARKA_1$ effectively separates active from inactive compounds: the active class is characterized by a positive median $ARKA_1$ value, whereas the inactive class displays a negative median. This contrast indicates that $ARKA_1$ captures structural variations correlated with biological activity, making it a valuable descriptor within the training context. However, in the test set, this separation becomes less pronounced; both activity classes show similar negative medians, suggesting that $ARKA_1$ may have limited generalization capacity when applied to new, unseen compounds.

In contrast, $ARKA_2$ demonstrates a more stable and consistent performance across both datasets. Within the training set, it differentiates between classes through a complex distribution among inactives and a clearly defined negative median for actives. Notably, this pattern persists in the test set, where $ARKA_2$ continues to display distinct behavior for each class. This stability highlights $ARKA_2$'s robustness as a descriptor and supports its utility in generalizing across datasets.

Overall, the violin plot analysis underscores the ARKA method's strength in capturing key structural drivers of activity variation. The superior stability and classification performance of $ARKA_2$, particularly across external data, affirms its relevance for identifying activity cliffs and enhancing QSTR modeling.

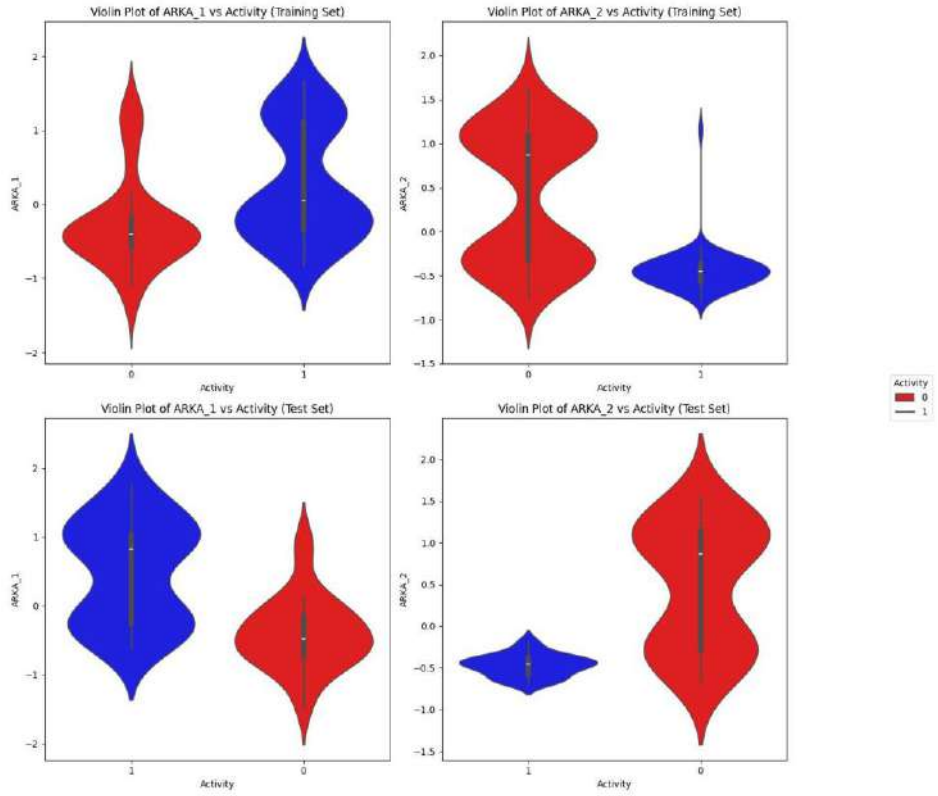


Figure III.27. Violin Plots of ARKA 1 vs Activity and ARKA 2 vs Activity for the training and test sets.

III.4.6.3. ARKA$_1$ vs ARKA$_2$ Scatter Plot (Activity and Prediction Cliffs)

The scatter plots of ARKA$_1$ versus ARKA$_2$, presented in Figure III.28 (training set) and Figure III.29 (test set)**,** serve as crucial tools for evaluating the spatial distribution of compounds. These plots allow for the identification of regions where subtle alterations in ARKA descriptors could lead to significant changes in biological activity, thereby revealing potential activity and prediction cliffs.

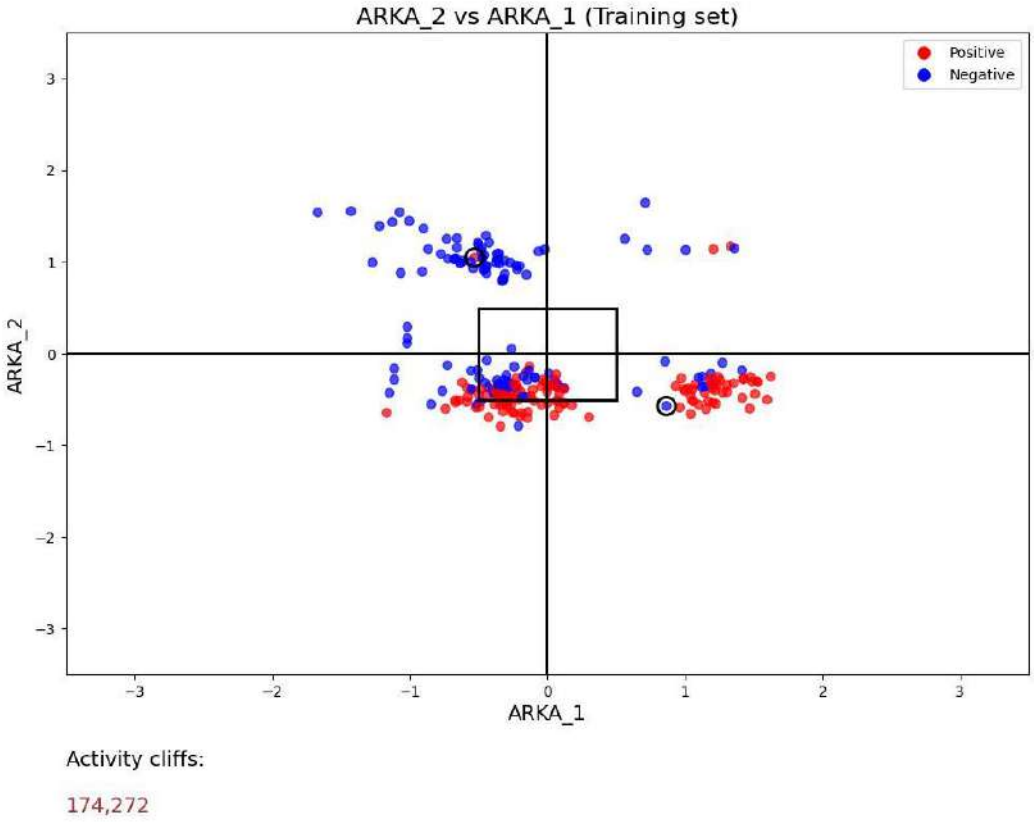


Figure III.28. $ARKA_1$ vs $ARKA_2$ in the training set.

In the training set, compounds that are confidently classified as active typically appear in the fourth quadrant (positive $ARKA_1$ and negative $ARKA_2$), while inactive compounds are generally found in the second quadrant (negative $ARKA_1$ and positive $ARKA_2$). The compounds located near the origin or along the X or Y axes exhibit lower decisiveness in classification, indicating that their $ARKA_1$ and $ARKA_2$ values are close to zero, which undermines their distinct categorization. Such points, especially when they deviate from the expected quadrant, are identified as potential activity cliffs. For example, active compounds appearing in the second quadrant or inactive compounds in the fourth quadrant are strong candidates for activity cliffs, suggesting that slight structural changes could result in significant biological activity alterations [173]. In our case, we found 2 points having a high potential activity cliff, the compound (174) a negative class compound yet appearing in the fourth and the compound (272) classified as active class compound yet falling under the second quadrant. These cliffs are critical for understanding the limitations in model ability, as they represent areas where the model may struggle to correctly classify compounds due to their borderline ARKA values.

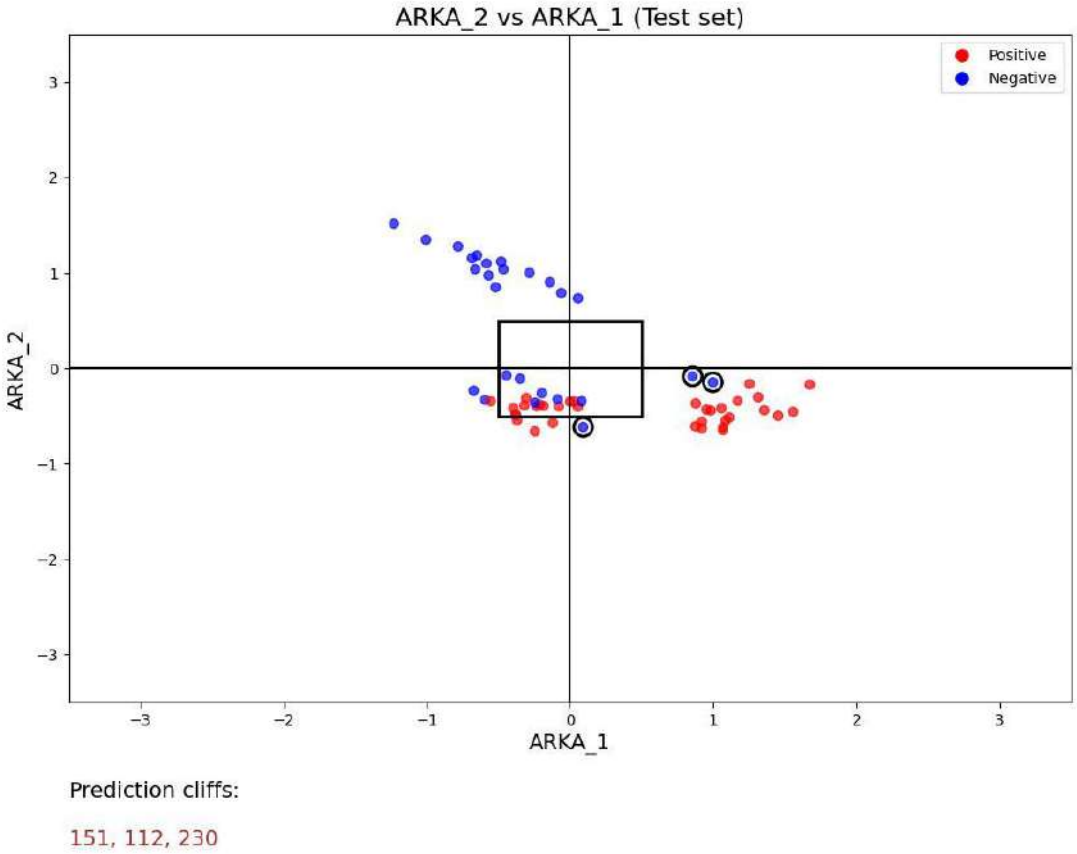


Figure III.29. ARKA 1 vs ARKA 2 in the test set.

In the test set, the scatter plot similarly reveals regions where the model's predictive performance may falter. Active compounds falling into the second quadrant, or inactive compounds residing in the fourth quadrant are identified as prediction cliffs. These points indicate potential inaccuracies in the model's predictions, reflecting cases where the ARKA descriptors do not strongly align with the expected activity. Additionally, compounds near the origin, where both ARKA 1 and ARKA 2 values are low, have a reduced likelihood of confident classification, as indicated by their proximity to the axes, for our test set we could identify the following compounds (151,112, 230) as potential prediction cliffs, since they are classified as negative and appearing in the 4th quadrant, they may require more investigation since they still fall under the borderline zone (not a reliable indicators). The identification of these prediction cliffs is vital, as it points out the areas where the model's performance might be compromised, necessitating further refinement or additional descriptors to enhance predictive accuracy.

III.4.6.4. Descriptor Contribution Analysis to ARKA Descriptors

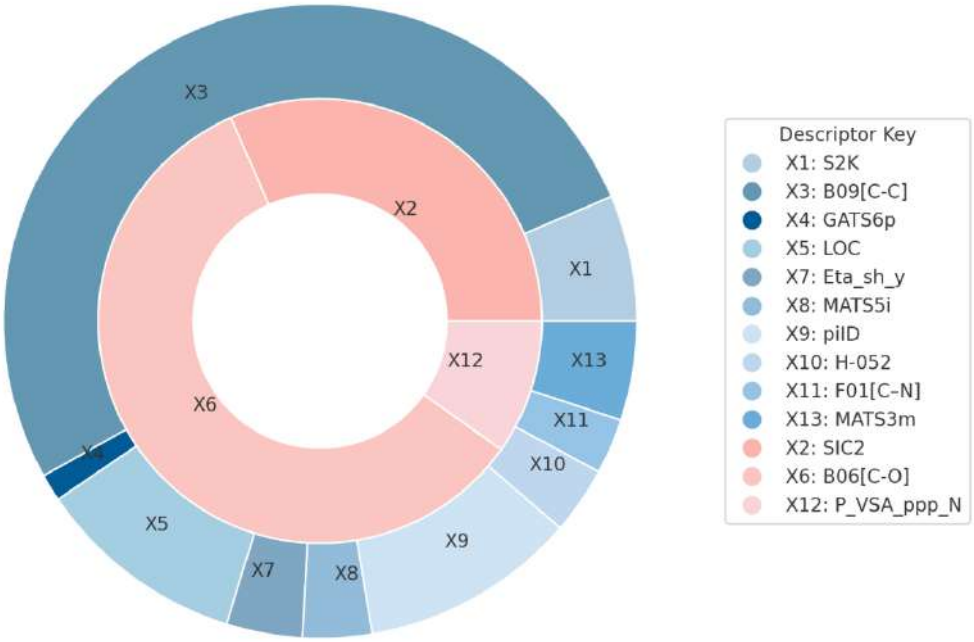


Figure III.30. Contributions of QSTR Descriptors to $ARKA_2$ (Outer Ring) and $ARKA_1$ (Inner Ring).

Figure III.30 illustrates the relative contributions of each QSTR descriptor to $ARKA_2$ (outer ring) and $ARKA_1$ (inner ring). The larger sections in $ARKA_2$, such as B09[C–C], LOC, and piID, indicate higher weightage in shaping specific molecular characteristics, while descriptors like B06[C–O], SIC2, and P_VSA_ppp_N dominate $ARKA_1$, reflecting distinct structural patterns. This visualization highlights the complementary roles of the descriptors in defining different chemical dimensions, offering insight into how the transformed features ($ARKA_1$ and $ARKA_2$) capture diverse and meaningful aspects of molecular behavior.

III.4.6.5. Comparative Modeling with XGBoost and Random Forest Using GA-MLR and ARKA Descriptors

Following the individual analyses of the GA-MLR selected descriptors and the newly derived ARKA descriptors, it is essential to conduct a comparative evaluation of their performance when integrated into advanced machine learning models. This section aims to

evaluate the predictive capabilities of two popular ensemble learning algorithms, XGBoost (Extreme Gradient Boosting) and Random Forest (RF), using both the established GA-MLR descriptor set and the novel ARKA descriptor set. The objective is to determine which descriptor representation yields more robust and accurate models for predicting ionic liquid toxicity across these powerful non-linear modeling frameworks. This comparative study will provide deeper insights into the effectiveness of ARKA as a dimensionality reduction technique in enhancing model performance and generalization.

III.4.6.5.1. Models Development and Hyperparameter Optimization

In this section, two ensemble learning algorithms, XGBoost and Random Forest (RF), were employed to model the log(EC_50) of ionic liquids using two distinct sets of molecular descriptors: the original GA-MLR-selected QSTR descriptors and the reduced ARKA descriptor set. All modeling and evaluation procedures were implemented in Python using the scikit-learn and xgboost libraries.To ensure robust comparisons, the same data splitting strategy was adopted for all models (80% for training and 20% for testing). Performance evaluation was conducted on individual sets (training and test) as well as the overall global dataset.Hyperparameter tuning was performed via grid search using 5-fold cross-validation, targeting optimal model configurations that minimize prediction error and avoid overfitting. For the Random Forest models, the hyperparameters tuned included n_estimators (the number of decision trees in the forest), max_depth (the maximum depth of each tree), min_samples_split (the minimum number of samples required to split an internal node), and min_samples_leaf (the minimum number of samples required to be at a leaf node). These parameters control the complexity and regularization of individual trees and the overall ensemble structure[95, 174], and their specific ranges are detailed in **Table III.9**.

Table III.9: Hyperparameter Value for Random Forest Model.

Hyperparameter	Values Explored
n_estimators	[50, 150, 300]
max_depth	[5, 10, 50]
min_samples_split	[2, 5, 10]
min_samples_leaf	[1, 2, 4]

For the XGBoost models, a more extensive set of hyperparameters was optimized. This included n_estimators (the number of boosting rounds), learning_rate (which shrinks the step size to prevent overfitting), max_depth (the maximum depth of each tree), subsample (the fraction of samples used for fitting the individual base learners), colsample_bytree (the fraction of features/columns to be used for fitting each tree), gamma (a minimum loss reduction required to make a further partition on a leaf node), and regularization terms reg_alpha (L1 regularization) and reg_lambda (L2 regularization) which help to prevent overfitting [175, 176], The specific ranges explored for each of these hyperparameters during the grid search are detailed in Table III.10.

Table III.10: Hyperparameter Value for XGBoost Model.

Hyperparameter	Values Explored
n_estimators	[100, 200, 300]
learning_rate	[0.01, 0.05, 0.1]
max_depth	[3, 4, 5]
subsample	[0.6, 0.8, 1.0]
colsample_bytree	[0.6, 0.8, 1.0]
gamma	[0, 0.1, 0.5]
reg_alpha	[0.0, 0.1, 0.5]
reg_lambda	[0.5, 1.0, 2.0]

III.4.6.5.2. Comparative Performance Analysis

To evaluate the modeling capacity of both descriptor sets (GA-MLR-selected QSTR descriptors and ARKA descriptors), a comparative performance analysis was conducted using two ensemble learning models: XGBoost and Random Forest (RF). The models were evaluated on training, testing, and global datasets using various error metrics (RMSE, MSE, MAE) and statistical metrics (R^2, R, and Q^2). Table III.10 summarizes the complete performance results.

Table III.11: Performance Metrics for XGBoost and Random Forest using GA-MLR and ARKA descriptors.

Model	Descriptor Set	Metric	Train	Test	Global
	ARKA	RMSE	0.3615	0.3875	0.3688
		MSE	0.1307	0.1502	0.1360
		MAE	0.2684	0.2966	0.2740
		R^2	0.8968	0.8673	0.8903
		R	0.9486	0.9318	0.9443
XGBoost		Q^2	-	0.8674	-
	GA-MLR	RMSE	0.3997	0.4848	0.4182
		MSE	0.1598	0.2350	0.1749
		MAE	0.2944	0.3778	0.3111
		R^2	0.8722	0.8038	0.8589
		R	0.9341	0.9053	0.9269
		Q^2	-	0.8040	-
	ARKA	RMSE	0.2227	0.4004	0.2680
		MSE	0.0496	0.1604	0.0718
		MAE	0.1554	0.3138	0.1872
		R^2	0.9608	0.8584	0.9421
		R	0.9809	0.9315	0.9707
Random Forest		Q^2	-	0.8584	-
	GA-MLR	RMSE	0.2750	0.4118	0.3074
		MSE	0.0756	0.1696	0.0945
		MAE	0.1827	0.2999	0.2062
		R^2	0.9403	0.8502	0.9238
		R	0.9716	0.9232	0.9629
		Q^2	-	0.8502	-

Across both algorithms, models based on ARKA descriptors consistently outperformed those using GA-MLR descriptors in terms of lower RMSE, MSE, and MAE, and higher R^2, R, and Q^2 values. For instance, Random Forest with ARKA achieved the best global R^2 of 0.9421, compared to 0.9238 with GA-MLR. Similarly, XGBoost with ARKA descriptors also showed improved predictive accuracy, achieving a lower test RMSE (0.3875) than the GA-MLR-based model (0.4848). These findings underscore the

utility of ARKA descriptors in enhancing model performance through more effective dimensionality reduction.

To visually complement the numerical data in Table III.11, Figure III.31 presents a comprehensive bar chart illustrating the RMSE, MSE, and MAE values for each model across the training, test, and global datasets.

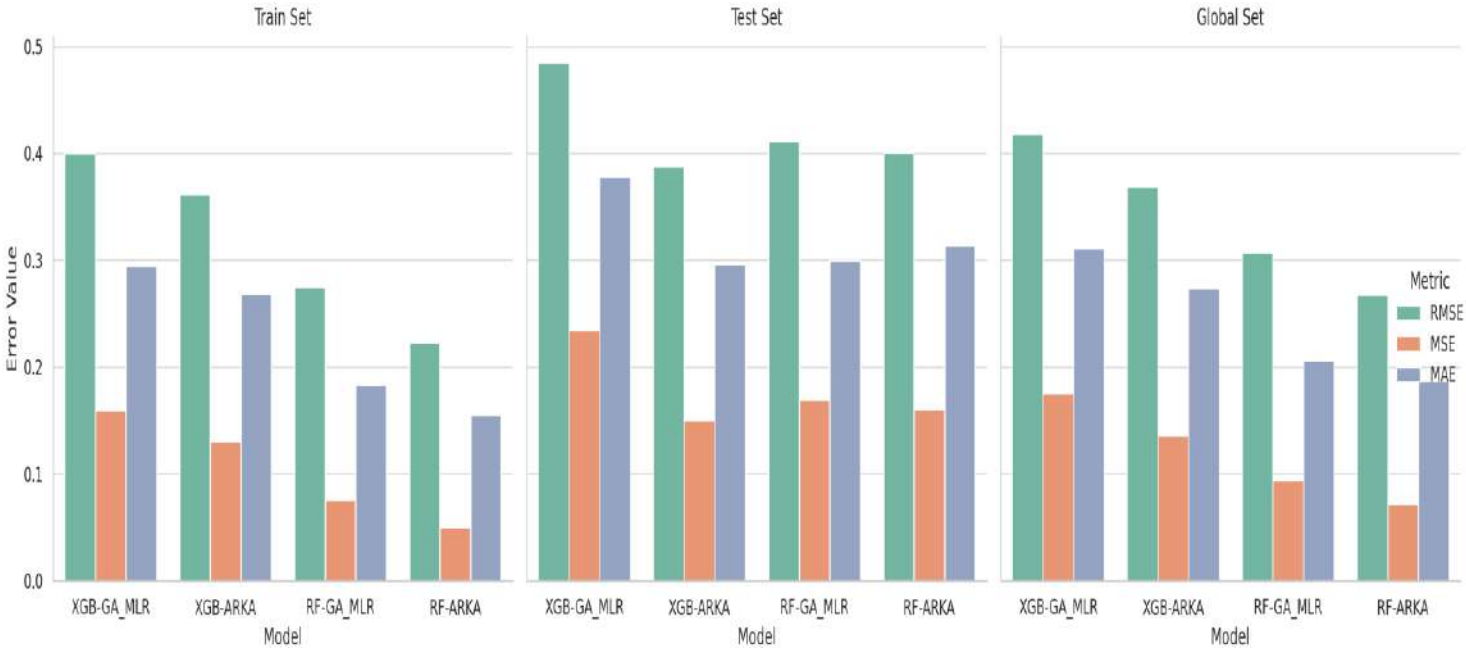


Figure III.31. Comparison of Error Metrics (RMSE, MSE and MAE) Across Models for Train, Test, and Global Sets.

Figure III.31 visually confirms the trends from Table III.8. It clearly shows that models developed with ARKA descriptors exhibit lower prediction errors compared to those using GA-MLR descriptors for both XGBoost and Random Forest algorithms, across all data sets. This visually reinforces that ARKA leads to improved model performance.

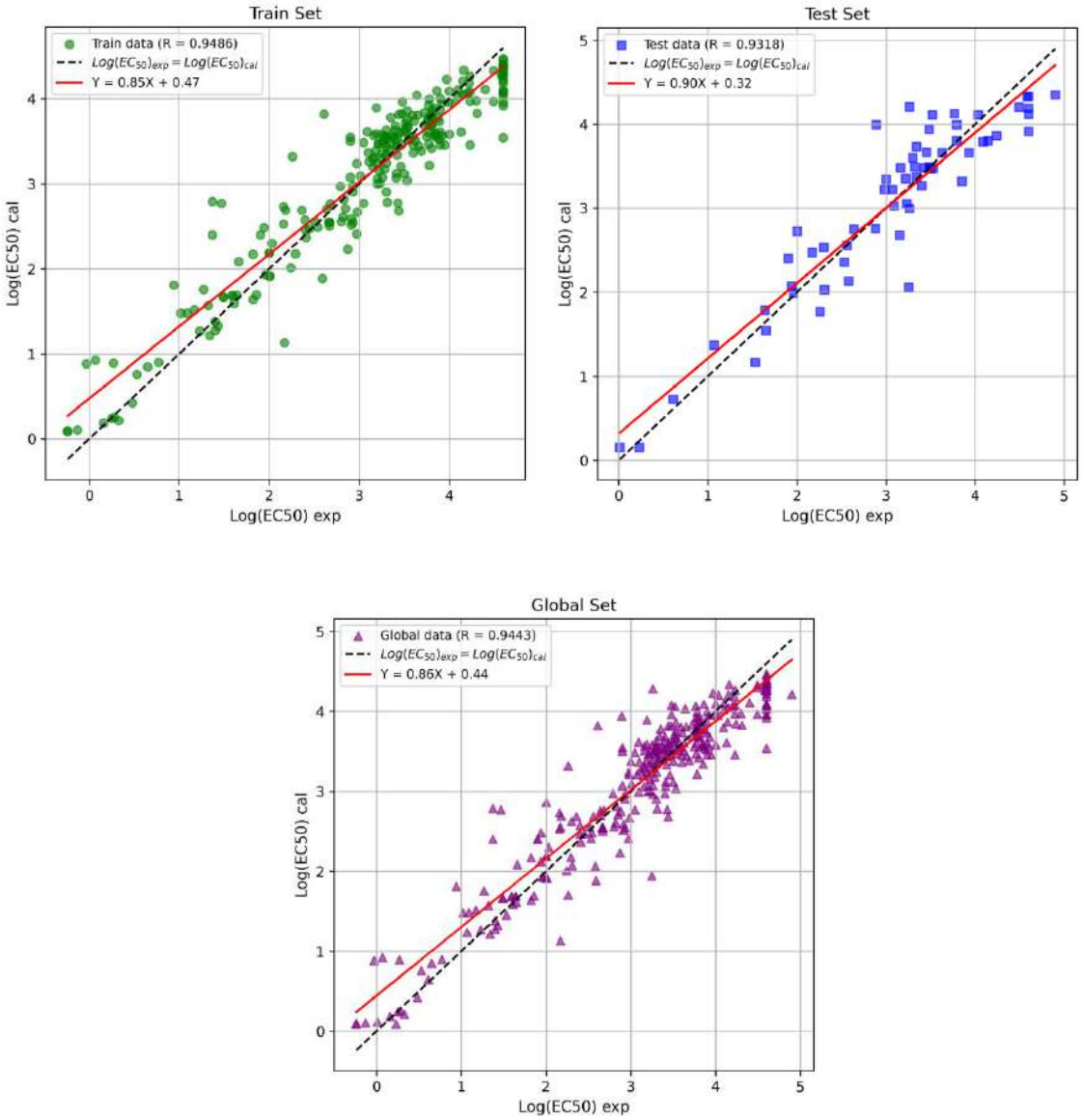


Figure III.32. Scatter plots predicted vs. observed values for XGBOOST model using ARKA descriptors.

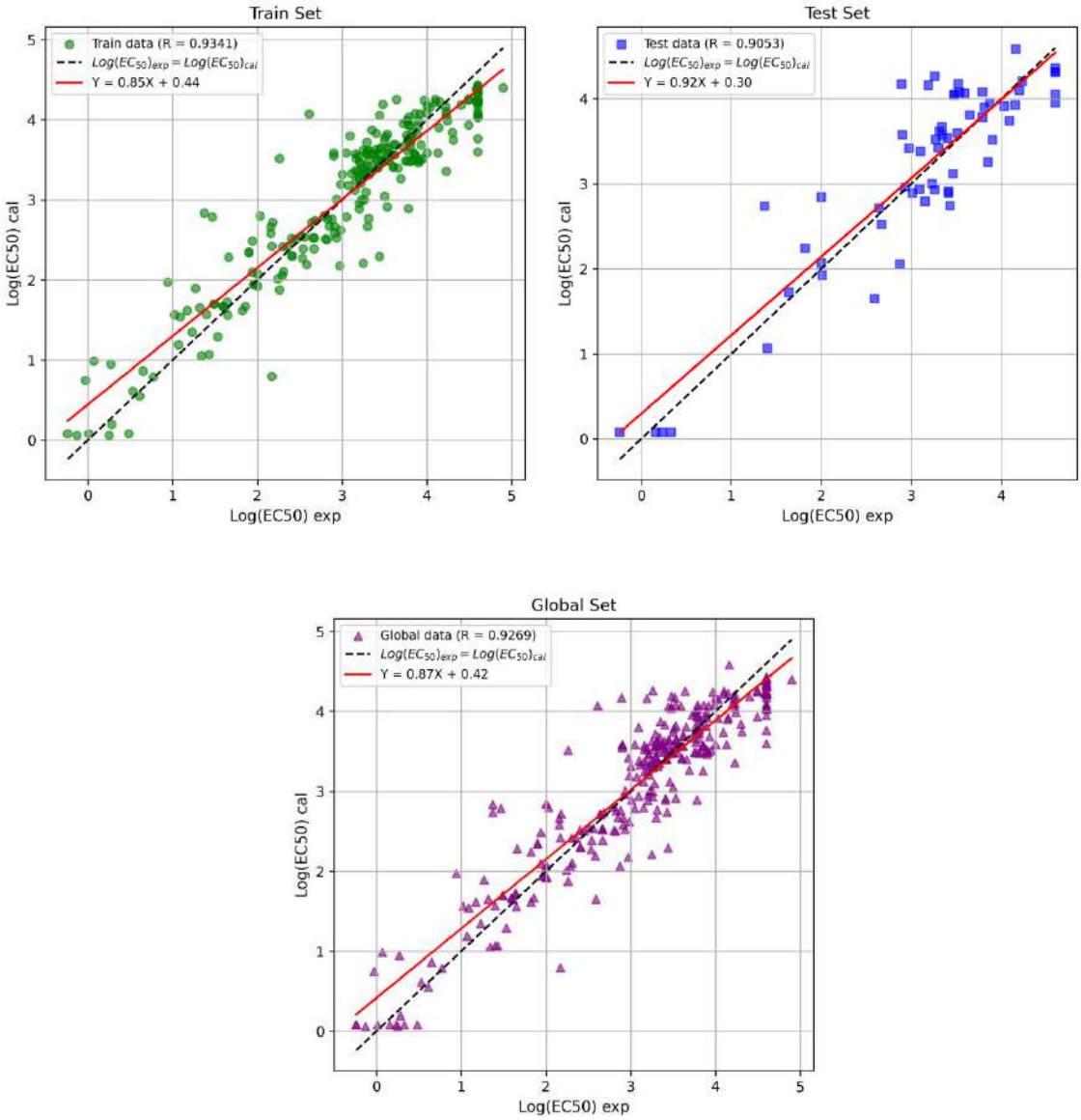


Figure III.33. Scatter plots predicted vs. observed values for XGBOOST model using GA_MLR descriptors.

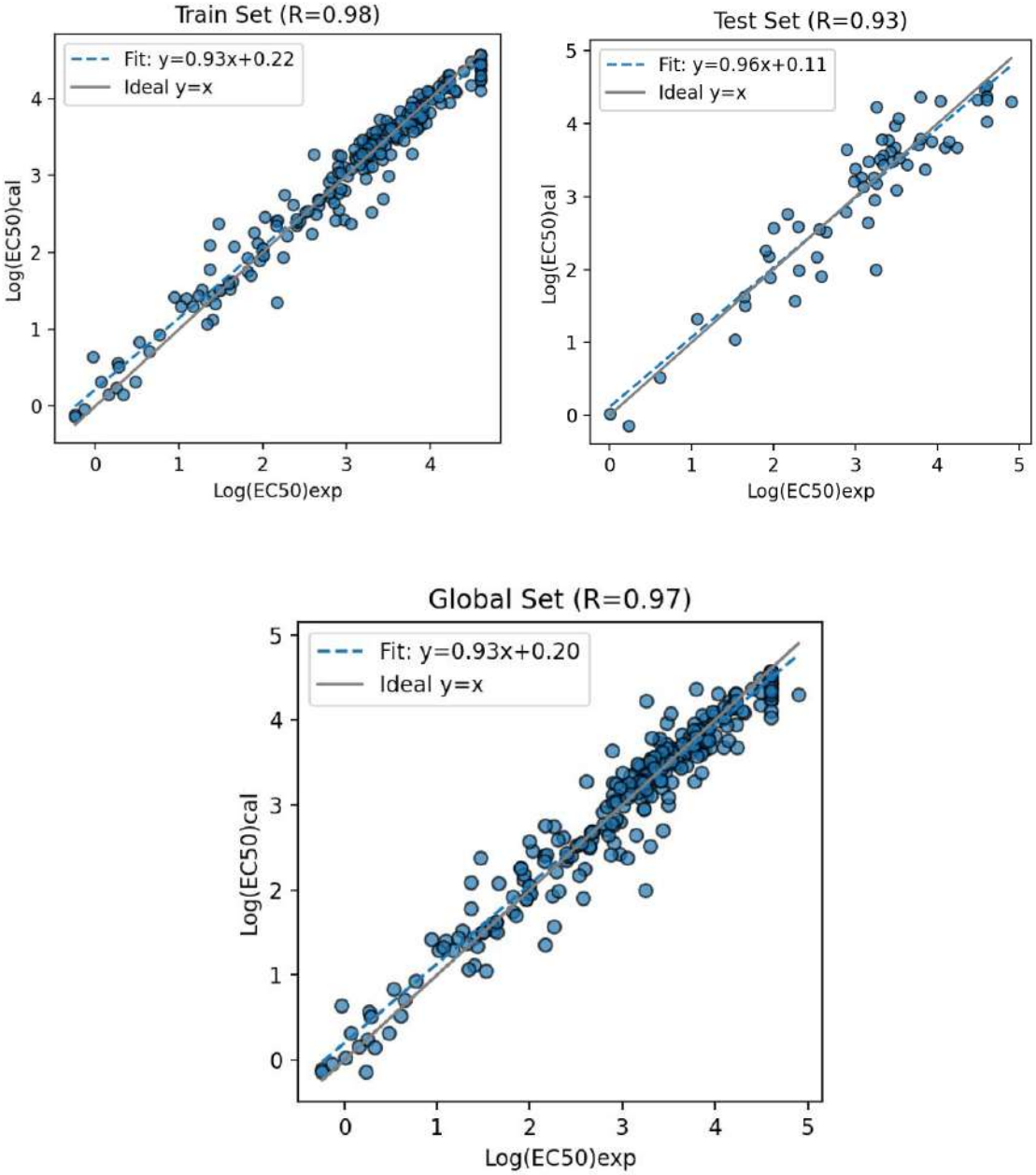


Figure III.34. Scatter plots predicted vs. observed values for RF model using ARKA descriptors.

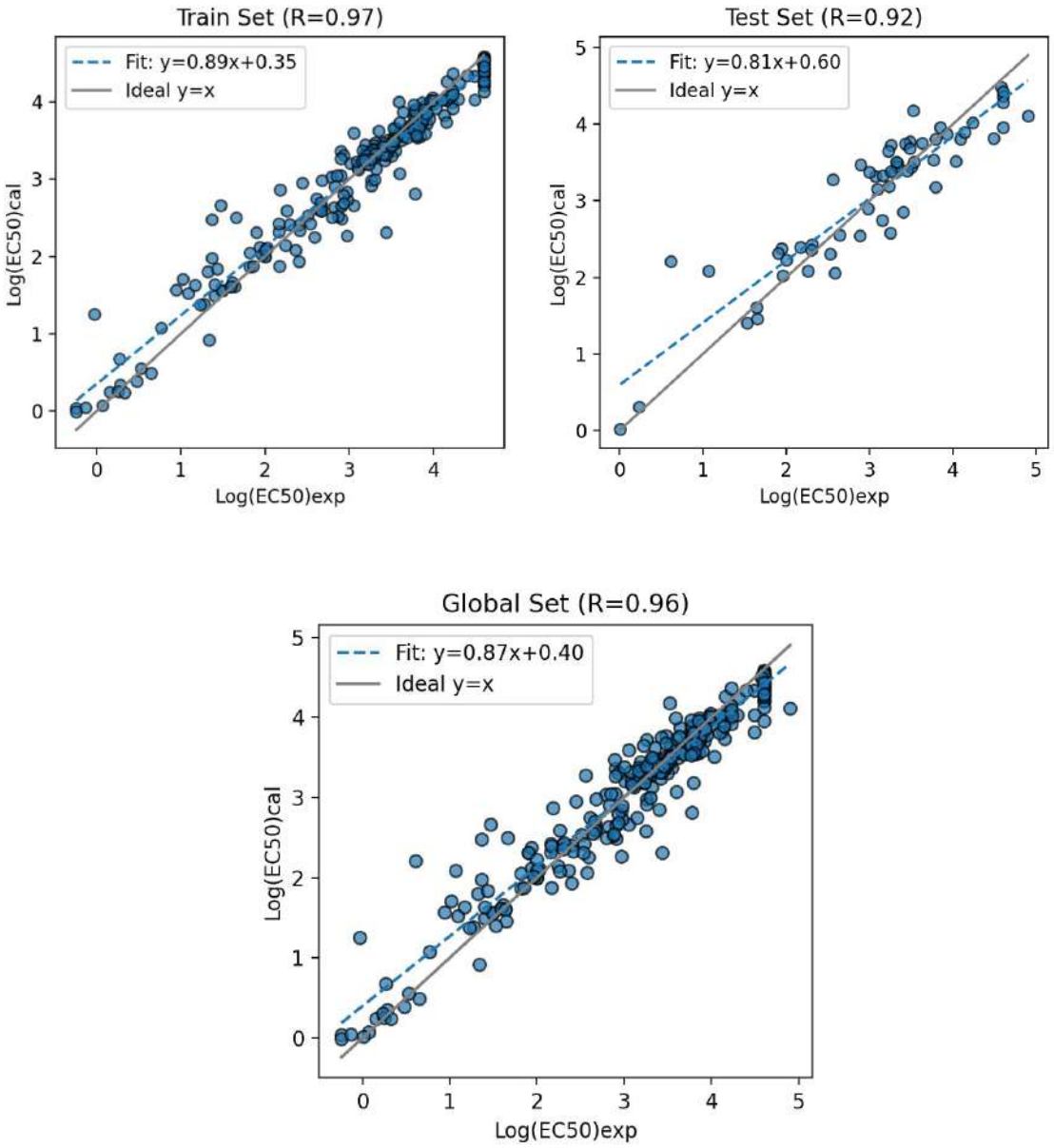


Figure III.35. Scatter plots predicted vs. observed values for RF model using GA_MLR descriptors.

Figures III.32-III.35 present the scatter plots of experimental versus predicted Log(EC_{50}) values for both XGBoost and Random Forest models using GA-MLR and ARKA descriptors. In all cases, the points are generally well aligned along the ideal diagonal, indicating that both models are able to capture the relationship between descriptors and toxicity. However, the predictions based on ARKA descriptors show a noticeably tighter clustering around the diagonal compared to GA-MLR, reflecting better accuracy and lower prediction error. While a few aberrant points are observed, they remain within an

acceptable range and do not significantly affect the overall model performance, this visual trend confirms that ARKA descriptors lead to improved predictive power for both machine learning models across the global dataset.

III.4.6.6. Applicability Domain Analysis (William's Plots)

Beyond evaluating predictive accuracy, defining the Applicability Domain (AD) of a QSTR model is essential to ensure the reliability of its predictions for new compounds. The Williams plot (a scatter plot of standardized residuals versus hat values (leverage)) is a widely accepted graphical tool for this purpose. It enables the identification of both outliers and high-leverage compounds that fall outside the model's defined AD [177, 178]. Figures III.36 and III.37 present the William's plots for the models built with ARKA descriptors using Random Forest and XGBoost, respectively.

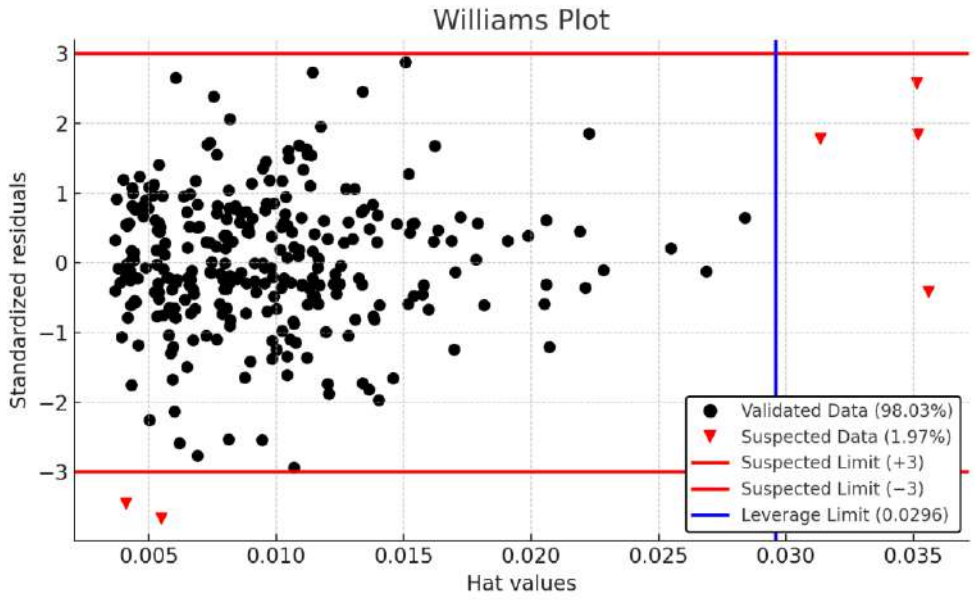


Figure III.36. Williams Plot for the RF Model Using ARKA Descriptors.

For the Random Forest model (Figure III.36), a significant majority of data points (98.03%) fall within the AD, remaining below the leverage threshold and within the ±3 standardized residual limits. Only 1.97% of the compounds are identified as suspected, highlighting the model's robustness and broad applicability across the chemical space.

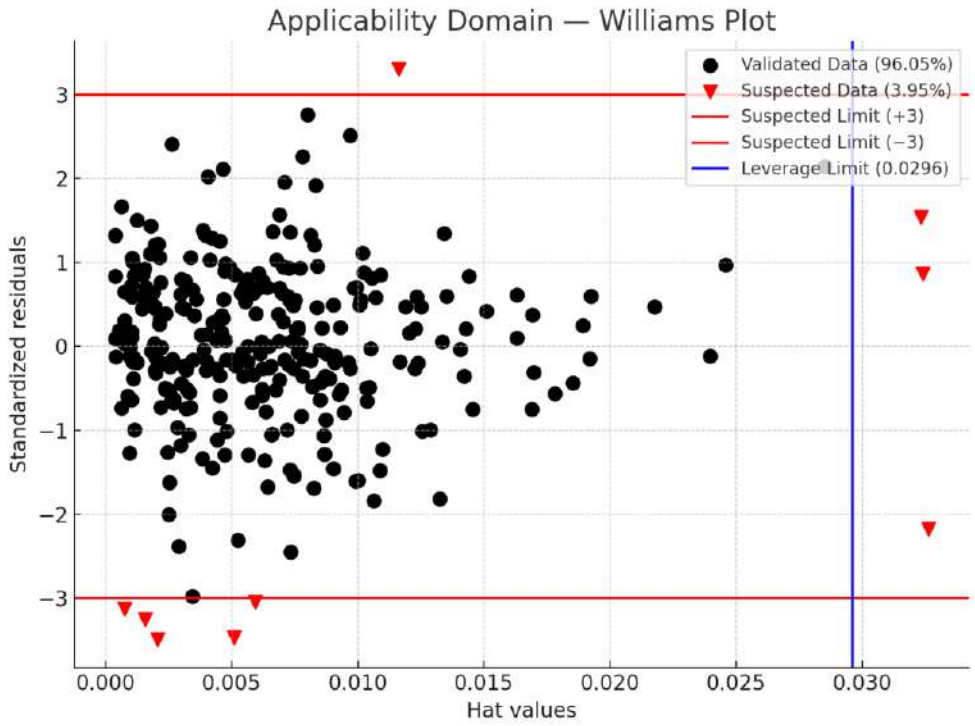


Figure III.37. Williams Plot for the XGBOOST Model Using ARKA Descriptors.

In contrast, the XGBoost model (Figure III.37) shows 96.05% of compounds within the domain and 3.95% outside. Although this represents a slightly narrower applicability domain, the proportion of suspected compounds remains low, suggesting that the model still provides reliable predictions. Overall, both models operate within well-defined and trustworthy applicability domains when using ARKA descriptors, with Random Forest offering slightly broader coverage.

III.5. Conclusion

This chapter successfully developed and validated robust QSTR models for predicting the toxicity of ionic liquids. Effective descriptor selection was achieved using both GA-MLR and Boruta algorithms, which identified the most relevant molecular features. SHAP analysis provided transparent and interpretable insights into the individual contributions of these descriptors.

In the first modeling stage, Support Vector Machine (SVM) models were optimized using three meta-heuristic algorithms: Dragonfly Algorithm (DA), Moth-Flame Optimization (MFO), and Grey Wolf Optimizer (GWO). Among these, the DA-SVM model consistently exhibited superior predictive accuracy and robustness, demonstrating

an optimal balance between model fit and generalization across both GA-MLR and Boruta-selected descriptor sets.

A pivotal finding from the second modeling stage was the pronounced effectiveness of the ARKA descriptors. When combined with advanced ensemble learning techniques, XGBoost and Random Forest models built upon ARKA descriptors consistently outperformed their GA-MLR counterparts, showcasing significantly enhanced predictive accuracy and generalization. This highlights ARKA's potential as a powerful dimensionality reduction strategy for improving QSTR model performance.

Crucially, the applicability domain analysis confirmed that all best-performing models operate within well-defined chemical space boundaries, ensuring the reliability of their predictions. Additionally, the identification of activity and prediction cliffs provided valuable insights that can inform the future design and assessment of ionic liquids. In essence, this chapter presents a reliable computational framework for ionic liquid toxicity prediction, underscoring the importance of advanced descriptor selection and machine learning integration in QSTR studies.

General Conclusion

General Conclusion

The accelerating development of versatile compounds like Ionic Liquids (ILs) presents vast opportunities across multiple industries while simultaneously emphasizing the critical need for comprehensive understanding of their physicochemical properties and potential environmental and health impacts. This thesis addresses these challenges by developing and validating advanced computational models capable of predicting both the toxicity and key molecular features of ILs. These models aim to serve as efficient in silico alternatives to traditional laboratory experiments, which are often costly and time-consuming.

The initial chapters of this thesis provide the necessary background. Chapter I offers an overview of ionic liquids (ILs) and a critical review of existing literature, while Chapter II explains the core principles underlying machine learning and optimization methods. Together, these chapters establish the theoretical and scientific foundation for the applied research.

The central focus of Chapter III is the development of Quantitative Structure–Toxicity Relationship (QSTR) models aimed at predicting the toxicity of ILs. Following rigorous data preprocessing, descriptor selection was performed using two complementary approaches: the Genetic Algorithm combined with Multiple Linear Regression (GA-MLR) and the Boruta algorithm. The optimization of Support Vector Machine (SVM) models using meta-heuristic algorithms proved particularly effective, with the Dragonfly Algorithm (DA), Moth-Flame Optimization (MFO), and Grey Wolf Optimizer (GWO) being employed. Among these, the DA-SVM model demonstrated superior predictive performance, achieving a global RMSE of 0.1787 and a test set Q^2CCC of 0.9583 using GA-MLR-selected descriptors.

Further advancements were achieved through the implementation of ARKA (Arithmetic Residuals in K-groups Analysis) descriptors. When integrated with ensemble learning algorithms such as XGBoost and Random Forest, these descriptors significantly enhanced predictive accuracy in comparison to classical molecular descriptors.

To ensure model transparency and robustness, several validation strategies were employed. Interpretability was enhanced through SHAP (SHapley Additive exPlanations) analysis, which elucidated the contribution of individual descriptors to toxicity predictions. Applicability domain assessments using Williams and Insubria plots confirmed the

reliability of predictions across diverse chemical structures. Additionally, the identification of activity cliffs provided valuable insight into structurally sensitive regions that contribute to sudden shifts in toxicity.

In summary, this thesis presents a comprehensive framework for building accurate, interpretable, and computationally efficient models to predict the toxicity of ionic liquids. The main contributions include:

- Enhanced Predictive Power: Among all models developed, the DA-optimized SVM using GA-MLR descriptors consistently outperformed others in terms of predictive accuracy, achieving the lowest errors and highest correlation metrics. Nevertheless, models built with ARKA descriptors—particularly in combination with XGBoost and Random Forest—demonstrated strong and promising performance, offering a viable alternative for toxicity modeling.
- Improved Interpretability: The application of SHAP analysis enabled a better understanding of how specific molecular features influence toxicity, transforming complex machine learning outputs into actionable insights for rational molecular design.
- Robust Validation: Comprehensive statistical assessments and applicability domain evaluations confirmed the reliability, generalizability, and practical relevance of the developed models, thereby supporting their deployment in predictive screening and chemical design.

Future Directions

While the present work provides a solid methodological framework, several avenues remain open for future exploration to further expand and refine predictive modeling in the field of ionic liquids:

- Dataset Expansion: Incorporating more diverse and representative experimental data will enhance model generalizability, improve robustness, and broaden the applicability domain.
- Inclusion of Additional Properties: Future research could target the prediction of other key physicochemical properties of ILs, such as viscosity, ionic conductivity, thermal stability, and CO_2 solubility. These properties are critical for IL

applications in catalysis, electrochemical systems, gas separation, and energy storage, and their inclusion would strengthen the practical value of predictive tools.

- Multi-objective Modeling: Developing multi-task learning frameworks capable of jointly optimizing multiple desirable IL attributes (e.g., low toxicity and high CO_2 absorption) could support more holistic and application-driven molecular design.
- Integration with Quantum Chemical Descriptors: Combining machine learning models with quantum chemically derived features or COSMO-RS-based parameters may provide deeper mechanistic understanding and improve model performance under varied conditions.
- Exploration of Deep Learning Approaches: Future studies may benefit from exploring advanced deep learning architectures such as artificial neural networks (ANNs) and convolutional neural networks (CNNs). These models are well-suited to capturing complex non-linear relationships in high-dimensional chemical data and may offer performance enhancements beyond those achieved with current models like SVM, Random Forest, and XGBoost…

Bibliography

1. Pan JY (2019) Engineering Chemistry Innovation. ACS Medicinal Chemistry Letters 10:703–707. https://doi.org/10.1021/acsmedchemlett.9b00096

2. Ivanković A (2017) Review of 12 Principles of Green Chemistry in Practice. International Journal of Sustainable and Green Energy 6:39. https://doi.org/10.11648/j.ijrse.20170603.12

3. Slootweg JC (2024) Sustainable chemistry: Green, circular, and safe-by-design. One Earth 7:754–758. https://doi.org/10.1016/j.oneear.2024.04.006

4. Savitskaya T, Kimlenka I, Lu Y, et al (2021) Green Chemistry: Process Technology and Sustainable Development

5. Bates JW, Ashton NW, Duckett JG (2018) Bryology for the Twenty-first Century. Bryology for the Twenty-first Century 460:1–382. https://doi.org/10.1201/9781315138626

6. Merlot C (2010) Computational toxicology-a tool for early safety evaluation. Drug Discovery Today 15:16–22. https://doi.org/10.1016/j.drudis.2009.09.010

7. Rusyn I, Daston GP (2010) Computational toxicology: Realizing the promise of the toxicity testing in the 21st century. Environmental Health Perspectives 118:1047–1050. https://doi.org/10.1289/ehp.1001925

8. Cherkasov A, Muratov EN, Fourches D, et al (2014) QSAR modeling: Where have you been? Where are you going to? Journal of Medicinal Chemistry 57:4977–5010. https://doi.org/10.1021/jm4004285

9. Huang T, Sun G, Zhao L, et al (2021) Quantitative structure-activity relationship (QSAR) studies on the toxic effects of nitroaromatic compounds (NACs): A systematic review. International Journal of Molecular Sciences 22:. https://doi.org/10.3390/ijms22168557

10. Papa E, Gramatica P (2010) QSPR as a support for the EU REACH regulation and rational design of environmentally safer chemicals: PBT identification from molecular structure. Green Chemistry 12:836–84. https://doi.org/10.1039/b923843c

11. Nasirpour N, Mohammadpourfard M, Zeinali Heris S (2020) Ionic liquids: Promising compounds for sustainable chemical processes and applications. Chemical Engineering Research and Design 160:264–300. https://doi.org/10.1016/j.cherd.2020.06.006

12. Plechkova N V., Seddon KR (2008) Applications of ionic liquids in the chemical industry. Chemical Society Reviews 37:123–150. https://doi.org/10.1039/b006677j

13. WELTON T (1999) Room-temperature ionic liquids. Solvents for synthesis and catalysis. Chemical Reviews 99:2071–2084. https://doi.org/10.1021/cr100194q

14. Wilkes JS (2002) A short history of ionic liquids - From molten salts to neoteric solvents. Green Chemistry 4:73–80. https://doi.org/10.1039/b110838g

15. Wang Y, He H, Wang C, et al (2022) Insights into Ionic Liquids: From Z-Bonds to Quasi-Liquids. JACS Au 2:543–561. https://doi.org/10.1021/jacsau.1c00538

16. Flieger J, Feder-kubis J, Tatarczak-michalewska M (2020) Chiral Ionic Liquids : Structural Diversity , Properties and Applications in Selected Separation Techniques. International journal of molecular sciences 11:4253.

17. Wang YL, Li B, Sarman S, et al (2020) Microstructural and Dynamical Heterogeneities in Ionic Liquids. Chemical Reviews 120:5798–5877. https://doi.org/10.1021/acs.chemrev.9b00693

18. Zhao D, Wu M, Kou Y, Min E (2002) Ionic liquids: Applications in catalysis. Catalysis Today 74:157–189. https://doi.org/10.1016/S0920-5861(01)00541-7

19. Hayes R, Warr GG, Atkin R (2015) Structure and Nanostructure in Ionic Liquids. Chemical Reviews 115:6357–6426. https://doi.org/10.1021/cr500411q

20. Vishwakarma S (2014) Ionic Liquids-Designer Solvents for Green Chemistry. International Journal of Basic Sciences and Applied Computing 1:1–4

21. Freemantle M (1998) Designer solvents—Ionic liquids may boost clean technology development. Chemical & Engineering News, 76:32–37

22. Kianfar E (2020) Ionic Liquids : Properties , Application , and Synthesis. Fine Chemical Engineering 2:21–29

23. Ghandi K (2014) A Review of Ionic Liquids , Their Limits and Applications. Green and sustainable chemistry 2014:44–53

24. Zhang S, Sun N, He X, et al (2006) Physical Properties of Ionic Liquids : Database and Evaluation. Journal of physical and chemical reference data 35:1475-1517. https://doi.org/10.1063/1.2204959

25. Cvjetko Bubalo M, Radošević K, Radojčić Redovniković I, et al (2014) A brief overview of the potential environmental hazards of ionic liquids. Ecotoxicology and Environmental Safety 99:1–12. https://doi.org/10.1016/j.ecoenv.2013.10.019

26. Thuy Pham TP, Cho CW, Yun YS (2010) Environmental fate and toxicity of ionic liquids: A review. Water Research 44:352–372. https://doi.org/10.1016/j.watres.2009.09.030

27. Hidalgo JM, Collado-gonz M, Montalb MG, Díaz FG (2016) Assessing chemical toxicity of ionic liquids on Vibrio fi scheri : Correlation with structure and composition. Chemosphere 155:405e414 Contents. https://doi.org/10.1016/j.chemosphere.2016.04.042

28. Grzonkowska M, Sosnowska A, Barycki M, et al (2016) How the structure of ionic liquid affects its toxicity to Vibrio fi scheri ? Chemosphere 159:199–207. https://doi.org/10.1016/j.chemosphere.2016.06.004

29. Flieger J (2020) Ionic Liquids Toxicity — Benefits and Threats. International Journal of Molecular Sciences 21:6367

30. Wang Z, Song Z, Zhou T (2021) Machine learning for ionic liquid toxicity prediction. Processes 9:1–10. https://doi.org/10.3390/pr9010065

31. Chipofya M, Tayara H, Chong KT (2022) Deep Probabilistic Learning Model for Prediction of Ionic Liquids Toxicity. International Journal of Molecular Sciences 23:5258. https://doi.org/10.3390/ijms23095258

32. Zhao Y, Zhao J, Huang Y, et al (2014) Toxicity of ionic liquids: Database and prediction via quantitative structure-activity relationship method. Journal of Hazardous Materials 278:320–329. https://doi.org/10.1016/j.jhazmat.2014.06.018

33. Raies AB, Bajic VB (2016) In silico toxicology : computational methods for the

prediction of chemical toxicity. 6:147–172. https://doi.org/10.1002/wcms.1240

34. Kumar A, Kumar P (2020) Quantitative structure toxicity analysis of ionic liquids toward acetylcholinesterase enzyme using novel QSTR models with index of ideality of correlation and correlation contradiction index. Journal of Molecular Liquids 318:114055. https://doi.org/10.1016/j.molliq.2020.114055

35. Lei Z, Chen B, Koo YM, Macfarlane DR (2017) Introduction: Ionic Liquids. Chemical Reviews 117:6633–6635. https://doi.org/10.1021/acs.chemrev.7b00246

36. ROGERS, Robin D., VOTH G (2007) Ionic liquids. Accounts of chemical research 40:1077-1078.

37. Welton T (2018) Ionic liquids: a brief history. Biophysical Reviews 10:691–706. https://doi.org/10.1007/s12551-018-0419-2

38. Rogers RD, Seddon KR (2003) Ionic Liquids - Solvents of the Future? Science 302:792–793. https://doi.org/10.1126/science.1090313

39. Hallett JP, Welton T (2011) Room-temperature ionic liquids: Solvents for synthesis and catalysis. 2. Chemical Reviews 111:3508–3576. https://doi.org/10.1021/cr1003248

40. Welton T (2011) Ionic liquids in Green Chemistry. Green Chemistry 13:225. https://doi.org/10.1039/c0gc90047h

41. Kar M, Plechkova N V., Seddon KR, et al (2007) Ionic Liquids-Further Progress on the Fundamental Issues. Australian Journal of Chemistry 72:3–10. https://doi.org/10.1071/CH18541

42. GREER, Adam J., JACQUEMIN, Johan, et HARDACRE C (2020) Industrial applications of ionic liquids

43. Chiappe C, Pieraccini D (2005) Ionic liquids: Solvent properties and organic reactivity. Journal of Physical Organic Chemistry 18:275–297. https://doi.org/10.1002/poc.863

44. Ghanem O Ben, Mutalib MIA, Lévêque JM, El-Harbawi M (2017) Development of QSAR model to predict the ecotoxicity of Vibrio fischeri using COSMO-RS

descriptors. Chemosphere 170:242–250. https://doi.org/10.1016/j.chemosphere.2016.12.003

45. Patel DD, Lee JM (2012) Applications of ionic liquids. Chemical Record 12:329–355. https://doi.org/10.1002/tcr.201100036

46. Zhang S, Sun N, He X, et al (2006) Physical properties of ionic liquids: Database and evaluation. Journal of Physical and Chemical Reference Data 35:1475–1517. https://doi.org/10.1063/1.2204959

47. Anthony JL, Anderson JL, Maginn EJ, Brennecke JF (2005) Anion Effects on Gas Solubility in Ionic Liquids. The Journal of Physical Chemistry B 109:6366–6374. https://doi.org/10.1021/jp046404l

48. Zeng S, Zhang X, Bai L, et al (2017) Ionic-Liquid-Based CO2 Capture Systems: Structure, Interaction and Process. Chemical Reviews 117:9625–9673. https://doi.org/10.1021/acs.chemrev.7b00072

49. Grewal J, Khare SK, Drewniak L, Pranaw K (2022) Recent perspectives on microbial and ionic liquid interactions with implications for biorefineries. Journal of Molecular Liquids 362:119796. https://doi.org/10.1016/j.molliq.2022.119796

50. Jastorff B, Störmann R, Ranke J, et al (2003) How hazardous are ionic liquids? Structure-activity relationships and biological testing as important elements for sustainability evaluation. Green Chemistry 5:136–142. https://doi.org/10.1039/b211971d

51. Zhao D, Liao Y, Zhang ZD (2007) Toxicity of ionic liquids. Clean - Soil, Air, Water 35:42–48. https://doi.org/10.1002/clen.200600015

52. Pretti C, Chiappe C, Baldetti I, et al (2009) Acute toxicity of ionic liquids for three freshwater organisms: Pseudokirchneriella subcapitata, Daphnia magna and Danio rerio. Ecotoxicology and Environmental Safety 72:1170–1176. https://doi.org/10.1016/j.ecoenv.2008.09.010

53. Fatemi MH, Izadiyan P (2011) Cytotoxicity estimation of ionic liquids based on their effective structural features. Chemosphere 84:553–563. https://doi.org/10.1016/j.chemosphere.2011.04.021

54. Paterno', A., Scire, S. & M (2016) A QSPR approach to the ecotoxicity of ionic liquids (Vibrio fischeri) using VolSurf principal properties. Toxicology Research 5:1090–1096. https://doi.org/10.1039/C6TX00071A

55. Luis P, Garea A, Irabien A (2010) Quantitative structure-activity relationships (QSARs) to estimate ionic liquids ecotoxicity EC50 (Vibrio fischeri). Journal of Molecular Liquids 152:28–33. https://doi.org/10.1016/j.molliq.2009.12.008

56. Yan F, Shang Q, Xia S, et al (2015) Topological study on the toxicity of ionic liquids on Vibrio fischeri by the quantitative structure – activity relationship method. Journal of Hazardous Materials 286:410–415. https://doi.org/10.1016/j.jhazmat.2015.01.016

57. Yan F, Shang Q, Xia S, et al (2015) Topological study on the toxicity of ionic liquids on Vibrio fischeri by the quantitative structure-activity relationship method. Journal of Hazardous Materials 286:410–415. https://doi.org/10.1016/j.jhazmat.2015.01.016

58. Couling DJ, Bernot RJ, Docherty KM, et al (2006) Assessing the factors responsible for ionic liquid toxicity to aquatic organisms via quantitative structure–property relationship modeling. Green Chemistry 8:82–90. https://doi.org/10.1039/b511333d

59. Luis P, Ortiz I, Aldaco R, Irabien A (2007) A novel group contribution method in the development of a QSAR for predicting the toxicity (Vibrio fischeri EC50) of ionic liquids. Ecotoxicology and Environmental Safety 67:423–429. https://doi.org/10.1016/j.ecoenv.2006.06.010

60. Ranke J, Müller A, Bottin-Weber U, et al (2007) Lipophilicity parameters for ionic liquid cations and their correlation to in vitro cytotoxicity. Ecotoxicology and Environmental Safety 67:430–438. https://doi.org/10.1016/j.ecoenv.2006.08.008

61. García-Lorenzo A, Tojo E, Tojo J, et al (2008) Cytotoxicity of selected imidazolium-derived ionic liquids in the human Caco-2 cell line. Sub-structural toxicological interpretation through a QSAR study. Green Chemistry 10:508–51. https://doi.org/10.1039/b718860a

62. Torrecilla JS, García J, Rojo E, Rodríguez F (2009) Estimation of toxicity of ionic liquids in Leukemia Rat Cell Line and Acetylcholinesterase enzyme by principal

component analysis , neural networks and multiple lineal regressions. Journal of Hazardous Materials 164:182–194. https://doi.org/10.1016/j.jhazmat.2008.08.022

63. Ismail Hossain M, Samir BB, El-Harbawi M, et al (2011) Development of a novel mathematical model using a group contribution method for prediction of ionic liquid toxicities. Chemosphere 85:990–994. https://doi.org/10.1016/j.chemosphere.2011.06.088

64. Yan F, Xia S, Wang Q, Ma P (2012) Predicting the toxicity of ionic liquids in leukemia rat cell line by the quantitative structure-activity relationship method using topological indexes. Industrial and Engineering Chemistry Research 51:13897–13901. https://doi.org/10.1021/ie301764j

65. Cho CW, Ranke J, Arning J, et al (2013) In silico modelling for predicting the cationic hydrophobicity and cytotoxicity of ionic liquids towards the Leukemia rat cell line, Vibrio fischeri and Scenedesmus vacuolatus based on molecular interaction potentials of ions. SAR and QSAR in Environmental Research 24:863–882. https://doi.org/10.1080/1062936X.2013.821092

66. Das RN, Roy K (2014) Predictive modeling studies for the ecotoxicity of ionic liquids towards the green algae Scenedesmus vacuolatus. Chemosphere 104:170–176. https://doi.org/10.1016/j.chemosphere.2013.11.002

67. Ma S, Lv M, Deng F, et al (2015) Predicting the ecotoxicity of ionic liquids towards Vibrio fischeri using genetic function approximation and least squares support vector machine. Journal of Hazardous Materials 283:591–598. https://doi.org/10.1016/j.jhazmat.2014.10.011

68. Sosnowska A, Grzonkowska M, Puzyn T (2017) Global versus local QSAR models for predicting ionic liquids toxicity against IPC-81 leukemia rat cell line: The predictive ability. Journal of Molecular Liquids 231:333–340. https://doi.org/10.1016/j.molliq.2017.02.025

69. Cao L, Zhu P, Zhao Y, Zhao J (2018) Using machine learning and quantum chemistry descriptors to predict the toxicity of ionic liquids. Journal of Hazardous Materials 352:17–26. https://doi.org/10.1016/j.jhazmat.2018.03.025

70. Yan F, Lan T, Yan X, et al (2019) Norm index-based QSTR model to predict the

eco-toxicity of ionic liquids towards Leukemia rat cell line. Chemosphere 234:116–122. https://doi.org/10.1016/j.chemosphere.2019.06.064

71. Zhu P, Kang X, Zhao Y, et al (2019) Predicting the toxicity of ionic liquids toward acetylcholinesterase enzymes using novel QSAR models. International Journal of Molecular Sciences 20:2186. https://doi.org/10.3390/ijms20092186

72. Wu T, Li W, Chen M, et al (2020) Estimation of Ionic Liquids Toxicity against Leukemia Rat Cell Line IPC-81 based on the Empirical-like Models using Intuitive and Explainable Fingerprint Descriptors. Molecular Informatics 39:1–7. https://doi.org/10.1002/minf.202000102

73. Kang X, Chen Z, Zhao Y (2020) Assessing the ecotoxicity of ionic liquids on Vibrio fischeri using electrostatic potential descriptors. Journal of Hazardous Materials 397:122761. https://doi.org/10.1016/j.jhazmat.2020.122761

74. Peng D, Picchioni F (2020) Prediction of toxicity of Ionic Liquids based on GC-COSMO method. Journal of Hazardous Materials 398:122964. https://doi.org/10.1016/j.jhazmat.2020.122964

75. Tabaaza GA, Tackie-Otoo BN, Zaini DB, et al (2023) Application of machine learning models to predict cytotoxicity of ionic liquids using VolSurf principal properties. Computational Toxicology 26:100266. https://doi.org/10.1016/j.comtox.2023.100266

76. Paternò A, Bocci G, Cruciani G, et al (2016) Cyto- and enzyme toxicities of ionic liquids modelled on the basis of VolSurf+ descriptors and their principal properties. SAR and QSAR in Environmental Research 27:221–244. https://doi.org/10.1080/1062936X.2016.1156571

77. Danush S, Dutta A (2023) Machine learning-based framework for predicting toxicity of ionic liquids. Materials Today: Proceedings 72:175–180. https://doi.org/10.1016/j.matpr.2022.06.380

78. Fan D, Xue K, Liu Y, et al (2023) Modeling the toxicity of ionic liquids based on deep learning method. Computers and Chemical Engineering 176:108293. https://doi.org/10.1016/j.compchemeng.2023.108293

79. Fan D, Xue K, Zhang R, et al (2024) Application of interpretable machine learning models to improve the prediction performance of ionic liquids toxicity. Science of the Total Environment 908:168168. https://doi.org/10.1016/j.scitotenv.2023.168168

80. Al-Jabri MM, Anshasi H (2025) Performance of machine and deep learning models for predicting delirium in adult ICU patients: A systematic review. International Journal of Medical Informatics 203:106008. https://doi.org/https://doi.org/10.1016/j.ijmedinf.2025.106008

81. Dei Cas D, Di Camillo B, Fadini GP, et al (2025) The impact of clinical history on the predictive performance of machine learning and deep learning models for renal complications of diabetes. Computer Methods and Programs in Biomedicine 268:108812. https://doi.org/https://doi.org/10.1016/j.cmpb.2025.108812

82. Rehman ZU, Chaozhe J, Jamal A, et al (2025) Unveiling pedestrian injury severity risk factors: A comparative analysis of machine learning and deep learning methods. Transportation Engineering 21:100352. https://doi.org/https://doi.org/10.1016/j.treng.2025.100352

83. Wang M, Nie Z, He Y, et al (2025) Deep learning methods for protein representation and function prediction: A comprehensive overview. Engineering Applications of Artificial Intelligence 155:110977. https://doi.org/https://doi.org/10.1016/j.engappai.2025.110977

84. Earnest A, Jones TW, Chee M, Holmes-Walker DJ (2025) Machine learning techniques to predict diabetic ketoacidosis and HbA1c above 7% among individuals with type 1 diabetes — A large multi-centre study in Australia and New Zealand. Nutrition, Metabolism and Cardiovascular Diseases 35:103861. https://doi.org/https://doi.org/10.1016/j.numecd.2025.103861

85. Finn C, Goodfellow I, Levine S (2016) Unsupervised Learning for Physical Interaction through Video Prediction. https://doi.org/10.48550/arXiv.1605.07157

86. Guo K, Buehler MJ (2021) Materials Horizons Artificial intelligence and machine learning in design of mechanical materials. Materials Horizons 8:1153–1172. https://doi.org/10.1039/d0mh01451f

87. Raza AA, Naseem M, Sharma S (2021) Machine Learning for Solving a Plethora of

Internet of Things Problems. Intelligent Communication and Automation Systems CRC Press 255–274. https://doi.org/10.1201/9781003104599-18

88. Pugliese R, Regondi S, Marini R (2021) Machine learning-based approach: Global trends, research directions, and regulatory standpoints. Data Science and Management 4:19–29. https://doi.org/10.1016/j.dsm.2021.12.002

89. Xu C, Nait M, Mohammed A, et al (2020) Evolving support vector regression using Grey Wolf optimization ; forecasting the geomechanical properties of rock. Engineering with Computers 30:39288–39318. https://doi.org/10.1007/s00366-020-01131-7

90. Boswell D (2002) Introduction to Support Vector Machines. Departement of Computer Science and Engineering University of California San Diego 11:16–17

91. Amroune M, Bouktir T, Musirin I (2018) Power System Voltage Stability Assessment Using a Hybrid Approach Combining Dragonfly Optimization Algorithm and Support Vector Regression. Arabian Journal for Science and Engineering 43:3023-3036. https://doi.org/10.1007/s13369-017-3046-5

92. Benimam H, Moussa CS, Hentabli M, et al (2020) Dragon fly-Support Vector Machine for Regression Modeling of the Activity Coefficient at Infinite Dilution of Solutes in Imidazolium Ionic Liquids Using σ - Profile Descriptors. Journal of Chemical & Engineering Data 65:3161-3172. https://doi.org/10.1021/acs.jced.0c00168

93. Euldji I, Belghait A, Amrane A (2023) A new hybrid quantitative structure property relationships-support vector regression (QSPR-SVR) approach for predicting the solubility of drug compounds in supercritical carbon dioxide. AIChE Journal 69:1–17. https://doi.org/10.1002/aic.18115

94. Wang H, Hu D (2005) Comparison of SVM and LS-SVM for Regression. nternational conference on neural networks and brain 1:279–283

95. Breiman L (2001) Random forests. Machine learning. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 45:5-32. https://doi.org/10.1007/978-3-030-62008-0_35

96. BREIMAN L (1996) Bagging predictors. Machine learning 24:123–140. https://doi.org/10.3390/risks8030083

97. Liu Y, Wang Y, Zhang J (2012) New machine learning algorithm: Random forest. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 7473 LNCS:246–252. https://doi.org/10.1007/978-3-642-34062-8_32

98. Song L, Langfelder P, Horvath S (2013) Random generalized linear model: A highly accurate and interpretable ensemble predictor. BMC Bioinformatics 14:5. https://doi.org/10.1186/1471-2105-14-5

99. Ho TK (1998) The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence 20:832–844. https://doi.org/10.1109/34.709601

100. Segal MR (2004) Machine learning benchmarks and random forest regression. *Center for Bioinformatics and Molecular Biostatistics, UC San Francisco, USA.* Biostatistics 1–14

101. CHEN, Tianqi et GUESTRIN C (2016) Xgboost: A scalable tree boosting system. Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining 785–794

102. Joshi A, Vishnu C, Mohan CK, Raman B (2024) Application of XGBoost model for early prediction of earthquake magnitude from waveform data. Journal of Earth System Science 133:. https://doi.org/10.1007/s12040-023-02210-1

103. Ibrahem Ahmed Osman A, Najah Ahmed A, Chow MF, et al (2021) Extreme gradient boosting (Xgboost) model to predict the groundwater levels in Selangor Malaysia. Ain Shams Engineering Journal 12:1545–1556. https://doi.org/10.1016/j.asej.2020.11.011

104. Noorunnahar M, Chowdhury AH, Mila FA (2023) A tree based eXtreme Gradient Boosting (XGBoost) machine learning model to forecast the annual rice production in Bangladesh. PLoS ONE 18:1–15. https://doi.org/10.1371/journal.pone.0283452

105. Lashkarbolooki M, Vaferi B, Rahimpour MR (2011) Comparison the capability of

artificial neural network (ANN) and EOS for prediction of solid solubilities in supercritical carbon dioxide. Fluid Phase Equilibria 308:35–43. https://doi.org/10.1016/j.fluid.2011.06.002

106. Meraihi Y, Dalila AR, Mohammed A (2020) Dragonfly algorithm : a comprehensive review and applications. Neural Computing and Applications 32:16625–16646. https://doi.org/10.1007/s00521-020-04866-y

107. Mafarja M, Heidari AA, Faris H, et al (2020) Dragonfly Algorithm : Theory , Literature Review , and Application in Feature Selection. Nature-Inspired Optimizers: Theories, Literature Reviews and Applications, 47–67. https://doi.org/10.1007/978-3-030-12127-3

108. Mirjalili S (2016) Dragonfly algorithm : a new meta-heuristic optimization technique for solving single-objective , discrete , and multi-objective problems. Neural Computing and Applications 27:1053–1073. https://doi.org/10.1007/s00521-015-1920-1

109. Yasen M (2018) Optimizing Neural Networks using Dragonfly Algorithm for Medical Prediction. 2018 8th International Conference on Computer Science and Information Technology (CSIT) 71–76

110. Hammouri AI, Mafarja M, Al-betar MA, et al (2020) Knowledge-Based Systems. Knowledge-Based Systems 203:106131. https://doi.org/10.1016/j.knosys.2020.106131

111. Elaziz MA, Khasawneh AM, Alabool H, Hamad H Al (2021) Dragonfly algorithm : a comprehensive survey of its results , variants , and applications. Multimedia Tools and Applications 80:14979–15016

112. Salam MA, Zawbaa HM, Emary E, et al (2016) A hybrid dragonfly algorithm with extreme learning machine for prediction. International symposium on innovations in intelligent systems and applications (INISTA) 1–6

113. Kumar A, Pant S, Ram M (2017) System Reliability Optimization Using Gray Wolf Optimizer Algorithm. Quality and Reliability Engineering International 33:1327–1335. https://doi.org/10.1002/qre.2107

114. Nadimi-Shahraki MH, Taghian S, Mirjalili S (2021) An improved grey wolf optimizer for solving engineering problems. Expert Systems with Applications 166:113917. https://doi.org/10.1016/j.eswa.2020.113917

115. Teng Z jun, Lv J ling, Guo L wen (2019) An improved hybrid grey wolf optimization algorithm. Soft Computing 23:6617–6631. https://doi.org/10.1007/s00500-018-3310-y

116. Dada EG, Joseph SB, Oyewola DO, et al (2022) Application of Grey Wolf Optimization Algorithm: Recent Trends, Issues, and Possible Horizons. Gazi University Journal of Science 35:485–504. https://doi.org/10.35378/gujs.820885

117. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey Wolf Optimizer. Advances in Engineering Software 69:46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007

118. Emary E, Yamany W, Hassanien AE, Snasel V (2015) Multi-Objective Gray-Wolf Optimization for Attribute Reduction. Procedia Computer Science 65:623–632. https://doi.org/10.1016/j.procs.2015.09.006

119. Nadimi-Shahraki MH, Fatahi A, Zamani H, et al (2021) Migration-based moth-flame optimization algorithm. Processes 9:1–28. https://doi.org/10.3390/pr9122276

120. Khalilpourazari S, Khalilpourazary S (2019) An efficient hybrid algorithm based on Water Cycle and Moth-Flame Optimization algorithms for solving numerical and constrained engineering optimization problems. Soft Computing 23:1699–1722. https://doi.org/10.1007/s00500-017-2894-y

121. Mirjalili S (2015) Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. Knowledge-Based Systems 89:228–249. https://doi.org/10.1016/j.knosys.2015.07.006

122. Settoul S, Chenni R, Hasan HA, et al (2019) MFO Algorithm for Optimal Location and Sizing of Multiple Photovoltaic Distributed Generations Units for Loss Reduction in Distribution Systems. Proceedings of 2019 7th International Renewable and Sustainable Energy Conference, IRSEC 2019 1–6. https://doi.org/10.1109/IRSEC48032.2019.9078241

123. Li Y, Zhu X, Liu J (2020) An improved moth-flame optimization algorithm for

engineering problems. Symmetry 12:1234. https://doi.org/10.3390/SYM12081234

124. Shehab M, Abualigah L, Al Hamad H, et al (2020) Moth–flame optimization algorithm: variants and applications. Neural Computing and Applications 32:9859–9884. https://doi.org/10.1007/s00521-019-04570-6

125. Sayed GI, Hassanien AE (2018) A hybrid SA-MFO algorithm for function optimization and engineering design problems. Complex & Intelligent Systems 4:195–212. https://doi.org/10.1007/s40747-018-0066-z

126. Laidi M, Hanini S (2013) Optimal solar COP prediction of a solar-assisted adsorption refrigeration system working with activated carbon/methanol as working pairs using direct and inverse artificial neural network. International Journal of Refrigeration 36:247–257. https://doi.org/10.1016/j.ijrefrig.2012.09.016

127. Ayinde AS, Huaming Y, Kejian W (2024) Review of machine learning methods for sea level change modeling and prediction. Science of The Total Environment 954:176410. https://doi.org/https://doi.org/10.1016/j.scitotenv.2024.176410

128. Aliferis C, Simon G (2024) Overfitting, Underfitting and General Model Overconfidence and Under-Performance Pitfalls and Best Practices in Machine Learning and AI

129. Mauri A (2020) alvaDesc: A tool to calculate and analyze molecular descriptors and fingerprints. Methods in Pharmacology and Toxicology 801–820. https://doi.org/10.1007/978-1-0716-0150-1_32

130. Aher, R. B., Khan, K., & Roy K (2020) A brief introduction to quantitative structure-activity relationships as useful tools in predictive ecotoxicology

131. Hamadache M, Hanini S, Benkortbi O, Amrane A (2016) Artificial Neural Network-Based Equation to Predict the Toxicity of Herbicides on Rats. Chemometrics and Intelligent Laboratory Systems. https://doi.org/10.1016/j.chemolab.2016.03.007

132. SUKUMAR, N., PRABHU, Ganesh, et SAHA P (2014) Applications of genetic algorithms in QSAR/QSPR modeling. In : Applications of metaheuristics in process engineering. Cham : Springer International Publishing 9783319065:315-324.

https://doi.org/10.1007/978-3-319-06508-3

133. Saxena AK, Prathipati P (2003) Comparison of MLR, PLS and GA-MLR in QSAR analysis. SAR and QSAR in Environmental Research 14:433–445. https://doi.org/10.1080/10629360310001624015

134. Albadr MA, Tiun S, Ayob M, Al-Dhief F (2020) Genetic algorithm based on natural selection theory for optimization problems. Symmetry 12:1–31. https://doi.org/10.3390/sym12111758

135. Dolatabadi M, Nekoei M, Banaei A (2010) Prediction of antibacterial activity of pleuromutilin derivatives by genetic algorithm-multiple linear regression (GA-MLR). Monatshefte fur Chemie 141:577–588. https://doi.org/10.1007/s00706-010-0299-z

136. Katoch S, Chauhan SS, Kumar V (2021) A review on genetic algorithm: past, present, and future. Multimedia Tools and Applications

137. Goodarzi M, Saeys W, Deeb O, et al (2013) Particle swarm optimization and genetic algorithm as feature selection techniques for the QSAR modeling of imidazo[1,5-a]pyrido[3,2-e]pyrazines, inhibitors of phosphodiesterase 10A. Chemical Biology and Drug Design 82:685–696. https://doi.org/10.1111/cbdd.12196

138. Rajabi, Mehdi and FS (2019) QSAR models for predicting aquatic toxicity of esters using genetic algorithm-multiple linear regression methods. Combinatorial Chemistry & High Throughput Screening 22:317-325. https://doi.org/10.2174/1386207322666190618150856

139. Gramatica P (2020) Principles of QSAR modeling: comments and suggestions from personal experience. International Journal of Quantitative Structure-Property Relationships 5:61–97. https://doi.org/10.4018/ijqspr.20200701.oa1

140. Hakim H (2025) A genetic algorithm-multiple linear regression model to quantify the toxicity of nitrobenzene derivatives Um modelo de regressão linear múltipla com algoritmo genético para quantificar a toxicidade dos derivados do nitrobenzeno Un modelo de regresión line. Studies in Engineering and Exact Sciences e13504–e13:1–16. https://doi.org/10.54021/seesv6n1-015

141. Eriksson L, Jaworska J, Worth AP, et al (2003) Methods for reliability and uncertainty assessment and for applicability evaluations of classification- and regression-based QSARs. Environmental Health Perspectives 111:1361–1375. https://doi.org/10.1289/ehp.5758

142. Joudaki D, Shafiei F (2019) QSPR Models to Predict Thermodynamic Properties of Cycloalkanes Using Molecular Descriptors and GA-MLR Method. Current Computer-Aided Drug Design 16:6–16. https://doi.org/10.2174/1573409915666190227230744

143. Kursa MB, Jankowski A, Rudnicki WR (2010) Boruta - A system for feature selection. Fundamenta Informaticae 101:271–285. https://doi.org/10.3233/FI-2010-288

144. Kursa MB, Rudnicki WR (2010) Feature selection with the boruta package. Journal of Statistical Software 36:1–13. https://doi.org/10.18637/jss.v036.i11

145. Kryshchyshyn A, Devinyak O, Kaminskyy D, et al (2018) Development of Predictive QSAR Models of 4-Thiazolidinones Antitrypanosomal Activity Using Modern Machine Learning Algorithms. Molecular Informatics 37:1–14. https://doi.org/10.1002/minf.201700078

146. Jakkula V (2006) Tutorial on Support Vector Machine (SVM). School of EECS, Washington State University 37:3

147. Wainer J, Cawley G (2017) Empirical evaluation of resampling procedures for optimising SVM hyperparameters. Journal of Machine Learning Research 18:1–35

148. Abbas F, Zhang F, Ismail M, et al (2023) Optimizing Machine Learning Algorithms for Landslide Susceptibility Mapping along the Karakoram Highway, Gilgit Baltistan, Pakistan: A Comparative Study of Baseline, Bayesian, and Metaheuristic Hyperparameter Optimization Techniques. Sensors 23:. https://doi.org/10.3390/s23156843

149. Bajaj NS, Patange AD, Jegadeeshwaran R, et al (2023) Application of metaheuristic optimization based support vector machine for milling cutter health monitoring. Intelligent Systems with Applications 18:200196. https://doi.org/10.1016/j.iswa.2023.200196

150. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Advances in engineering software 69:46–61

151. Roy K, Roy PP, Leonard JT (2008) On some aspects of validation of predictive QSAR models. Chemistry Central Journal 2:1–1. https://doi.org/10.1186/1752-153X-2-S1-P9

152. Roy K, Das RN, Ambure P, Aher RB (2016) Be aware of error measures. Further studies on validation of predictive QSAR models. Chemometrics and Intelligent Laboratory Systems 152:18–33. https://doi.org/https://doi.org/10.1016/j.chemolab.2016.01.008

153. Plevris V, Solorzano G, Bakas NP, Ben Seghier MEA (2022) Investigation of Performance Metrics in Regression Analysis and Machine Learning-Based Prediction Models. World Congress in Computational Mechanics and ECCOMAS Congress 0–25. https://doi.org/10.23967/eccomas.2022.155

154. de Lange, A. M. G., Anatürk, M., Rokicki, J., Han, L. K., Franke, K., Alnæs, D., ... & Cole JH (2022) Human Brain Mapping - 2022 - Lange - Mind the gap Performance metric evaluation in brain-age prediction.pdf. 3113-3129.

155. Hair JF (2021) Next-generation prediction metrics for composite-based PLS-SEM. Industrial Management and Data Systems 121:5–11. https://doi.org/10.1108/IMDS-08-2020-0505

156. Todeschini R, Ballabio D, Grisoni F (2016) Beware of Unreliable Q2! A Comparative Study of Regression Metrics for Predictivity Assessment of QSAR Models. Journal of Chemical Information and Modeling 56:1905–1913. https://doi.org/10.1021/acs.jcim.6b00277

157. Soleimani R, Saeedi Dehaghani AH, Shoushtari NA, et al (2018) Toward an intelligent approach for predicting surface tension of binary mixtures containing ionic liquids. Korean Journal of Chemical Engineering 35:1556–1569. https://doi.org/10.1007/s11814-017-0326-4

158. Roy K, Mitra I (2011) On Various Metrics Used for Validation of Predictive QSAR Models with Applications in Virtual Screening and Focused Library Design. Combinatorial Chemistry & High Throughput Screening 14:450–474.

https://doi.org/10.2174/138620711795767893

159. Consonni V, Ballabio D, Todeschini R (2009) Comments on the definition of the Q2 parameter for QSAR validation. Journal of Chemical Information and Modeling 49:1669–1678. https://doi.org/10.1021/ci900115y

160. Shrestha N (2020) Detecting Multicollinearity in Regression Analysis. American Journal of Applied Mathematics and Statistics 8:39–42. https://doi.org/10.12691/ajams-8-2-1

161. Park J, Kim J, Ryu D, Choi HY (2023) Factors related to steroid treatment responsiveness in thyroid eye disease patients and application of SHAP for feature analysis with XGBoost. Frontiers in Endocrinology 14:1–11. https://doi.org/10.3389/fendo.2023.1079628

162. Parsa, A. B., Movahedi, A., Taghipour, H., Derrible, S., & Mohammadian AK (2020) Toward safer highways, application of XGBoost and SHAP for real-time accident detection and feature analysis. Accident Analysis & Prevention, 136:105405

163. Ponce-Bobadilla, A. V., Schmitt, V., Maier, C. S., Mensing, S., & Stodtmann S (2024) Practical guide to SHAP analysis: explaining supervised machine learning model predictions in drug development. Clinical and Translational Science, 17:e70056.

164. Schindele A, Krebold A, Heiß U, et al (2025) Interpretable machine learning for thyroid cancer recurrence predicton: Leveraging XGBoost and SHAP analysis. European Journal of Radiology 186:. https://doi.org/10.1016/j.ejrad.2025.112049

165. Sahigara F, Mansouri K, Ballabio D, et al (2012) Comparison of different approaches to define the applicability domain of QSAR models. Molecules 17:4791–4810. https://doi.org/10.3390/molecules17054791

166. Zhao X, Pan Y, Jiang J, et al (2017) Thermal Hazard of Ionic Liquids: Modeling Thermal Decomposition Temperatures of Imidazolium Ionic Liquids via QSPR Method. Industrial and Engineering Chemistry Research 56:4185–4195. https://doi.org/10.1021/acs.iecr.6b04762

167. Hamadache M, Hanini S, Benkortbi O, et al (2016) Artificial neural network-based equation to predict the toxicity of herbicides on rats. Chemometrics and Intelligent Laboratory Systems 154:7–15. https://doi.org/10.1016/j.chemolab.2016.03.007

168. de Haas EM, Eikelboom T, Bouwman T (2011) Internal and external validation of the long-term QSARs for neutral organics to fish from ECOSAR tm. SAR and QSAR in Environmental Research 22:545–559. https://doi.org/10.1080/1062936X.2011.569949

169. Kardani MN, Baghban A, Sasanipour J, et al (2018) Group contribution methods for estimating CO2 absorption capacities of imidazolium and ammonium-based polyionic liquids. Journal of Cleaner Production 203:601–618. https://doi.org/10.1016/j.jclepro.2018.08.127

170. Sangion A, Gramatica P (2016) PBT assessment and prioritization of contaminants of emerging concern: Pharmaceuticals. Environmental Research 147:297–306. https://doi.org/10.1016/j.envres.2016.02.021

171. Gramatica P, Cassani S, Roy PP, et al (2012) QSAR modeling is not "Push a button and find a correlation": A case study of toxicity of (Benzo-)triazoles on Algae. Molecular Informatics 31:817–835. https://doi.org/10.1002/minf.201200075

172. Gramatica P, Cassani S, Chirico N (2014) QSARINS-chem: Insubria datasets and new QSAR/QSPR models for environmental pollutants in QSARINS. Journal of Computational Chemistry 35:1036–1044. https://doi.org/10.1002/jcc.23576

173. Banerjee A, Roy K (2024) ARKA: a framework of dimensionality reduction for machine-learning classification modeling, risk assessment, and data gap-filling of sparse environmental toxicity data. Environmental Science: Processes and Impacts 26:991–1007. https://doi.org/10.1039/d4em00173g

174. Probst P, Wright MN, Boulesteix AL (2019) Hyperparameters and tuning strategies for random forest. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 9:1–19. https://doi.org/10.1002/widm.1301

175. Chen T, Guestrin C (2016) XGBoost : A Scalable Tree Boosting System. Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining 785–794. https://doi.org/10.1145/2939672.2939785

176. Dalal S, Onyema EM, Malik A (2022) Hybrid XGBoost model with hyperparameter tuning for prediction of liver disease with better accuracy. World Journal of Gastroenterology 28:6551–6563. https://doi.org/10.3748/wjg.v28.i46.6551

177. Gadaleta D, Mangiatordi GF, Catto M, et al (2016) Applicability Domain for QSAR Models. International Journal of Quantitative Structure-Property Relationships 1:45–63. https://doi.org/10.4018/ijqspr.2016010102

178. Kar S, Roy K, Leszczynski J (2018) pplicability domain: a step toward confident predictions and decidability for QSAR modeling. Computational toxicology: methods and protocols, 1800:141-169.