Comparative Analysis of Machine Learning Algorithms for Predicting Drug Mechanism of Action

Hana Alalawi, Manal Alsuwat, Amani Alsabeie and Sarah Al-Shareef,

Umm Al-Qura University, Makkah, Saudi Arabia

Abstract
Predicting the Mechanism of Action (MoA) of drugs is a crucial step in drug discovery, influencing both the efficacy and safety of therapeutic interventions. This study undertakes a comparative analysis of four machine learning algorithms—K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Decision Trees (DT), and Random Forest (RF)—to identify the most effective method for MoA prediction. Employing Classifier Chains and Binary Relevance techniques, we explore the impact of feature selection and data balancing strategies on the performance of these algorithms. Results demonstrate that SVM and RF generally provide the best performance, especially in handling complex, feature-rich datasets. The study highlights the importance of tailored data preprocessing and balancing to optimize algorithmic predictions in pharmacological applications. Our findings offer significant insights into machine learning implementations in drug discovery, providing a foundation for further research into advanced predictive models.

Keywords:
Drug Mechanism of Action, MoA, Machine Learning, Multi-Label Classification, Feature Selection, Data Balancing Techniques.

1. Introduction

Understanding the Mechanism of Action (MoA) of drugs is pivotal in pharmacology, guiding the development of new therapeutic agents and enhancing the efficacy and safety of existing treatments. The MoA describes how a drug interacts at the molecular level within the body to exert its effects, typically involving interactions with specific biomolecules such as receptors or enzymes. Accurately predicting the MoA can significantly streamline drug discovery processes, reducing the time and cost associated with experimental assays [1].

Despite its importance, predicting the MoA of drugs remains a complex challenge due to the intricate nature of biological systems and the vast diversity of drug structures [2]. Traditional methods rely heavily on biochemical experiments that are time-consuming, costly, and limited in their throughput. With the arrival of big data in biomedicine, machine learning algorithms have emerged as powerful tools capable of uncovering patterns from large datasets that are not immediately apparent to human researchers [3].

This study addresses the need for advanced computational approaches to predict the MoA of drugs more efficiently. We focus on applying and comparing several classification algorithms, including K-Nearest Neighbors (KNN), Support Vector Machines (SVM), and Decision Trees, to determine the most effective methods for this task. Each algorithm has been chosen for its unique ability to handle different aspects of the complex data typically involved in MoA studies, such as gene expression profiles and cellular viability metrics. The primary objectives of this paper are:

- To evaluate the accuracy of various classification algorithms in predicting the MoA of drugs from pharmacological data.
- To determine the impact of different feature selection techniques on the performance of these algorithms.
- To identify the algorithm that provides the best balance between prediction accuracy and computational efficiency.

Our approach is novel in its comprehensive comparison of multiple machine-learning techniques tailored specifically for MoA prediction. Additionally, this study innovates in applying feature selection methods that enhance model performance by identifying the most informative biological markers relevant to drug action mechanisms.

This paper is structured as follows: The next section reviews relevant literature on MoA prediction and the application of machine learning in drug discovery. We then detail our methodology, including data collection, preprocessing, and model building. The results section presents our findings on the performance of each algorithm, followed by a discussion that interprets these results within the broader context of pharmacological research. Finally,

Manuscript received May 5, 2025
Manuscript revised May 20, 2025
https://doi.org/10.22937/IJCSNS.2025.25.5.1

we conclude with the implications of our study for future drug discovery efforts and potential areas for further research.

By establishing a robust computational framework for MoA prediction, this work aims to contribute significantly to computational biology and pharmacology, providing insights that may accelerate the discovery and development of new drugs..

2. Related Work

Advances in machine learning techniques have significantly improved our understanding of drug MoA, particularly in handling the complex biological datasets generated in drug discovery. Previous studies have leveraged various non-deep learning algorithms to predict MoA [3], focusing on classification and feature selection methods that accommodate the high-dimensional nature of biological data.

Traditional supervised machine learning methods like Support Vector Machines (SVM) [3,4], Decision Trees [6], and Ensemble Methods [7] have been widely employed in MoA prediction. For instance, SVMs have demonstrated effectiveness in binary classification tasks involved in determining whether a compound is active or inactive based on bioactivity data [2,4]. Decision Trees and Random Forests have been particularly noted for their interpretability and robustness in handling biological datasets, which often contain irrelevant and redundant features [7,8]. Moreover, unsupervised machine learning methods, such as KNearest neighbor (KNN) [9], were also employed in the drug discovery realm.

Recent reviews, e.g. [2,3], have highlighted the importance of integrating different types of biological data, such as genomics, proteomics, and metabolomics, to enhance the prediction accuracy of MoA models. This integration helps in capturing the comprehensive biological interactions of compounds, aiding in more accurate MoA predictions. Furthermore, methods like feature selection and dimensionality reduction are crucial in managing the vast datasets typically involved in MoA studies, as they help focus the learning algorithms on the most informative features [8,10].

In addressing data imbalance, which is a common issue in MoA prediction due to the varied frequency of MoA classes, techniques such as Synthetic Minority Over-sampling Technique (SMOTE) and adaptive sampling have been useful [11]. These methods help in creating balanced training sets, which are vital for improving the performance of machine learning models.

In conclusion, while deep learning methods are increasingly popular in bioinformatics, traditional machine learning approaches remain invaluable in MoA predictions due to their effectiveness in smaller datasets, interpretability, and fewer computational requirements. Future research could focus on hybrid models that integrate both deep learning and traditional machine learning techniques to leverage their respective strengths in handling complex and large-scale biological data.

In light of these advancements, our study aims to address gaps by comparing multiple ML techniques and exploring the impact of feature selection in MoA prediction. By focusing on these elements, we contribute to refining the predictive capabilities of ML models in pharmacological applications, potentially leading to more targeted and effective therapeutic interventions.

Our approach is particularly novel in its extensive comparison of ML algorithms, tailored for MoA prediction, along with a deep dive into feature selection techniques that enhance model performance. This work not only builds on but also significantly extends the methodologies discussed in existing literature.

3. MoA Dataset

The dataset utilized in this study is sourced from the Kaggle, specifically designed for MoA prediction challenges[1]. It comprises various drug profiles that include gene expression data and cell viability metrics. The dataset features approximately 876 variables, including gene expression levels (g-0 to g-772), cell viability scores (c-0 to c-99), and treatment conditions (cp_type, cp_time, cp_dose). The cp_type variable indicates whether the samples were treated with a compound (trt_cp) or a control perturbation (ctl_vehicle).

The dataset includes 207 target labels, such as inhibitors, activators, agonists, etc. The majority of target labels fall into three categories: inhibitors, agonists, or antagonists. Figure 1 illustrates the

[1] https://www.kaggle.com/competitions/lish-moa

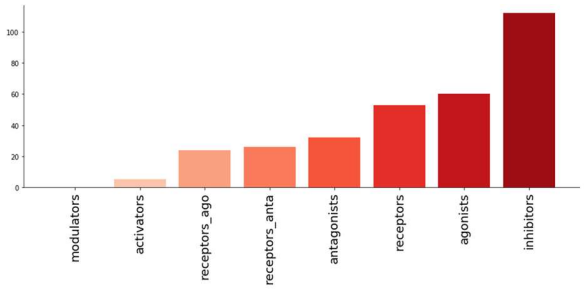

Figure 1: Distribution of the target types in the MoA dataset. The *y*-axis represents the frequency of the type in the dataset.

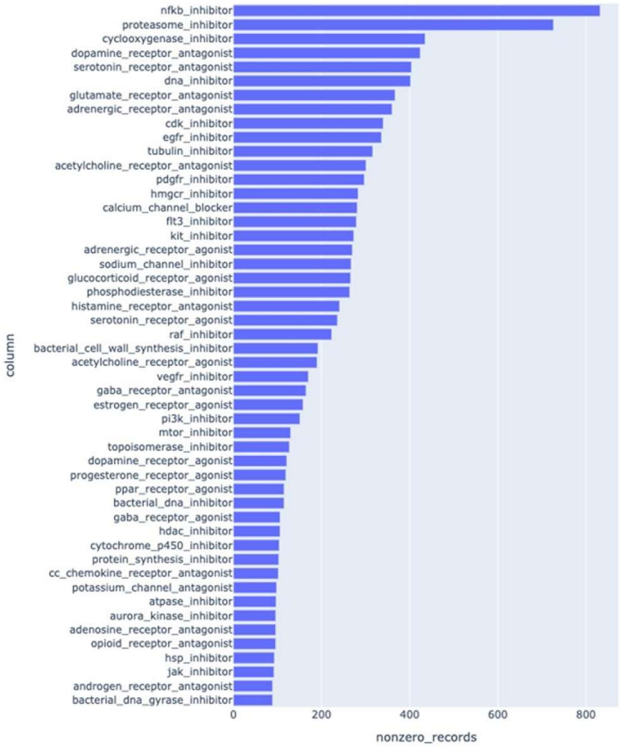

Figure 2: The 50 most frequent labels in the target dataset.

distribution of these feature types. Inhibitors constitute most of the target variables, while the numbers of agonists and antagonists are not equivalent.

These target features are used to discover 23.8k drug samples in the dataset. Each drug may be associated with multiple MoA target labels; however, almost half of the samples have only one MoA target label, and around 38% are not associated with any labels. Figure 2 lists the 50 most frequent labels in the target dataset. The nfkb_inhibitor appears at the top of the list, indicating it has the highest occurrence.

4. Methodology

4.1 Data Processing and Feature Engineering

1) Reducing the target set: The presence of a large number of labels can complicate the modeling process. To streamline this, all target MoA labels are categorized into one of seven groups: receptor antagonists, receptor agonists, receptors, agonists, antagonists, activators, and inhibitors. Figure 3 demonstrates the distribution of these categories within the target MoAs. As a result, these categories have been adopted as the primary target labels for our analysis. Finally, an OR Boolean operation was performed across all target MoA labels of the same category, enabling each sample to be labeled under the newly defined target set.

2) Selecting discriminative features: The features used in training significantly impact the achievable outcomes, as models trained with unnecessary features can suffer from decreased accuracy due to reliance on irrelevant data. Feature selection offers multiple advantages, including improved accuracy, reduced overfitting, and shorter training times. In this project, selecting the most important features was based on analyzing the correlation between features, beginning by categorizing features into types marked with prefixes -g and -c, then combining columns and

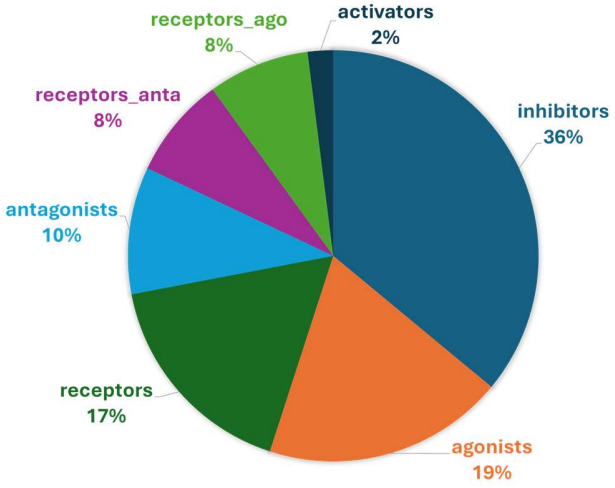

Figure 3: Distribution of MoA categories within the dataset.

selecting random features. An iterative process tested

each pair for correlation to identify features with minimal correlation, aiming to select distinct features since a strong correlation between two features suggests redundancy—where one can effectively replace the other, thus optimizing memory use and reducing training time. A feature was chosen if its correlation was less than an absolute value of 0.0003—a threshold determined to be optimal through various experiments, ensuring that the selected features are almost unique, with only weak similarities to others. From a total of 875 features, 461 were selected based on these criteria.

3) Overcoming class imbalance: An imbalanced dataset can significantly impact algorithm performance by leading to the neglect of the minority class, which is not adequately trained. Two common resampling techniques address this: oversampling and undersampling, each with its own set of benefits and drawbacks. Resampling helps balance the dataset either by adding cases to the minority class or removing cases from the majority class, thus facilitating the development of more effective machine learning models. Oversampling increases the representation of the minority class by duplicating samples, as was done with the 'cp_type' column using Random Oversampling. While helpful in balancing classes, this method can lead to overfitting due to the duplication of records and increased dataset size, which may slow down the training process. Conversely, a simple undersampling approach involves randomly removing records from the majority class, which can reduce training time and memory usage but risks losing important information. Hence, RandomOverSampler and RandomUnderSampler are used to achieve balance. Initially, there were 21,948 instances in Class 1 and 1,866 in Class 0. Post-balancing with RandomOverSampler, each class had 17,576 instances, as shown in Figure 28. RandomUnderSampler reduced both classes to 1,475 instances each.

4.2 Classification Algorithms

1) K-Nearest Neighbor (KNN): KNN is based on the principle that similar items are located near each other. It is a robust supervised classification method that predicts the target label by storing all available data and classifying new cases based on similarity measures such as Manhattan, Euclidean, Minkowski, and Hamming distances. KNN's simplicity and non-parametric nature make it well-suited as a baseline model for complex datasets, particularly useful in biological contexts where relationships can be non-linear. In MoA prediction, KNN leverages localized data characteristics effectively, especially in scenarios involving multi-class classification, by identifying patterns from the nearest neighbors in a feature space defined by drug properties and cellular responses.

2) Support Vector Machine (SVM): SVM is a robust supervised learning algorithm used for classification and regression tasks. It excels in high-dimensional spaces, such as those typical in drug MoA prediction, involving complex and large datasets. SVM operates by identifying the optimal hyper-plane that separates different classes with the maximum margin, utilizing support vectors that are the nearest data points to the hyper-plane. For multi-label classification challenges, such as predicting drug classes, SVM can be enhanced with techniques like classifier chains and binary relevance. These methods, coupled with SVM's ability to adapt its decision boundaries through various kernel functions, allow for precise and nuanced discrimination between classes. SVM's kernel trick, which enables data handling in high-dimensional spaces, makes it particularly valuable in genomic or high-throughput screening data analysis, where subtle distinctions between MoAs are crucial for accurate classification.

3) Decision Tree (DT): DT is a classification algorithm that constructs a tree-like structure, enabling the selection among various outcomes. It builds a regression model by progressively splitting data into smaller branches, culminating in a decision node. This intuitive approach creates a visual map of decisions and their potential consequences, making it straightforward to understand and interpret. However, decision trees can be susceptible to overfitting, particularly as the complexity of the tree increases.

4) Random Forest (RF): RF is an ensemble learning technique ideal for classification and regression, which works by building numerous decision trees on random data subsets and averaging their predictions to enhance accuracy and control overfitting. Especially effective in drug discovery for MoA prediction, it handles high-dimensional biological data adeptly, managing the complexities of genomic data or high-throughput screening. Random Forest's ability to capture non-linear relationships and provide interpretable results—identifying key features

influencing predictions—makes it particularly valuable in biomedical research, where understanding biological pathways is crucial. It is evaluated using accuracy, precision, recall, and F1-score, ensuring robust performance assessment in multi-label classification scenarios.

4.3 Multi-label Classification Techniques

In the study of multi-label classification, commonly applied in fields such as drug discovery and bioinformatics, methodologies are categorized mainly into problem transformation methods and algorithm adaptation methods. Problem transformation methods simplify multi-label problems into several single-label problems, enabling the application of traditional algorithms that are not originally equipped to handle multiple labels simultaneously. In contrast, algorithm adaptation methods modify existing algorithms to directly accommodate multi-label data, thus inherently managing the complexities and dependencies between multiple labels.

Our focus was on problem transformation techniques, specifically employing Binary Relevance and Classifier Chains. Binary Relevance treats each label as a distinct binary classification task, effectively ignoring label dependencies. Classifier Chains enhance this approach by considering the interdependencies among labels, where each classifier in the chain uses the predictions of previous classifiers as additional features, thereby improving the prediction accuracy for subsequent labels. These methods transform complex multi-label tasks into more tractable binary problems, suitable for conventional machine learning algorithms.

In this study, we focused on problem transformation techniques:

1) Binary Relevance: This method treats each label as a separate single binary classification problem. It involves decomposing the multi-label task into multiple binary tasks, where each task predicts the presence or absence of a specific label. Despite its simplicity, binary relevance is effective but does not account for label correlations, treating each label independently.

2) Classifier Chains: To capture label interdependencies, we utilized Classifier Chains, which build upon the concept of binary relevance. In this method, the multi-label problem is converted into a chain of binary classification problems. Each classifier in the chain predicts a label, incorporating the outputs of previous classifiers in the chain as additional features. This sequential approach allows each classifier to leverage the label associations learned by its predecessors, enhancing the predictive performance by considering the relationships between labels.

These techniques were chosen for their ability to adapt conventional binary classifiers to the multi-label setting effectively. By employing these methods, we could transform complex multi-label tasks into simpler binary problems that are more tractable for traditional machine learning algorithms. Our implementation details and the effectiveness of these approaches are further explored in the results section.

4.4 Evaluation Metrics

To assess the performance of the classification algorithms employed in this study, we utilized a comprehensive set of evaluation metrics as defined in [12]. These metrics are crucial for determining the effectiveness of our models across various dimensions of accuracy and error measurement.

Accuracy: Defined as the proportion of correctly predicted labels to the total number of labels (both predicted and actual) for each instance, with the overall accuracy being the average across all instances.

Precision: Also known as the positive predictive value, precision is the ratio of correctly predicted positive labels to the total number of actual positive labels, averaged over all instances.

Recall: It measures the proportion of actual positive labels that are correctly predicted as such, averaged over all cases.

F1 Score: Also known as balanced F-score, is the harmonic mean of precision and recall, offering a single metric that balances both. It is particularly useful in the context of uneven class distributions, where one class may dominate over others.

Hamming Loss: This metric represents the fraction of the wrong labels to the total number of labels, indicating the overall error rate in the label prediction.

Beyond these instance-based metrics, we also employed label-based measures that evaluate the performance for each label separately before averaging them:

Macro Averaging: This approach calculates metrics independently for each class and then takes the

average, treating all classes equally regardless of their frequency. This method is beneficial for ensuring that minority classes are considered fairly in the evaluation process.

Micro Averaging: In contrast, micro averaging aggregates the contributions of all classes to compute the average metrics. This method calculates the total counts of false positives, false negatives, and true positives across all classes, then derives metrics like recall, precision, and F1-score.

For this study, we particularly focused on macro averaging due to its effectiveness in handling class imbalances by assigning equal importance to each category. This choice reflects our aim to develop a model that performs consistently well across all classes, including those that are less represented in the dataset. Through these metrics, we aim to comprehensively understand the strengths and weaknesses of our models, ensuring they are robust and effective across diverse multi-label classification scenarios.

5. Experimental Design

The experimental setup for this study was carefully designed to ensure robust evaluation and reproducibility of results. All experiments were implemented in Python, utilizing the widely recognized scikit-learn library [13], which provided a comprehensive framework for machine learning tasks. To guarantee an unbiased assessment and generalizability of the outcomes, we employed stratified 10-fold cross-validation across all models. This method preserves the proportion of each class within each fold, thereby maintaining the distribution integrity of the original dataset.

Hyperparameter tuning was systematically conducted using a grid search approach, allowing us to explore a range of possible configurations and identify the optimal settings for each model.

In total, eight models were developed for each classification algorithm to thoroughly assess performance across various scenarios:

- The first and second models utilized all available features (807 in total). The first model applied the classifier chain method targeting the 7-category MoA labels, while the second used binary relevance under the same conditions.
- The third and fourth models were trained using a reduced feature set (461 features), following the same methodology as the first and second models, respectively, to evaluate the impact of feature reduction on model performance.
- The remaining models incorporated strategies to balance the dataset, with six additional models created to address potential biases introduced by class imbalances.
    - Models five and six replicated the first and second models but utilized undersampling to balance the dataset.
    - Models seven and eight mirrored the third and fourth models but used undersampling on the reduced feature set.
    - Models nine and ten applied oversampling to the full feature set using classifier chain and binary relevance, respectively.
    - Models eleven and twelve used oversampling with the reduced feature set following the same respective methodologies.

This systematic approach to experimental design, spanning multiple models and configurations, was chosen to comprehensively evaluate the effectiveness of each classification strategy under varying data conditions. The results of these experiments are intended to provide insightful conclusions about the scalability and adaptability of the proposed methods in the context of multi-label classification for drug MoA prediction.

6. Results

The results of this study provide a comprehensive evaluation of four machine learning algorithms—KNN, SVM, DT, and RF —across various experimental setups designed to understand their efficacy in multi-label classification for predicting drug MoA. Each model was tested using two primary techniques: Classifier Chains and Binary Relevance, and each technique was evaluated under different conditions of feature selection and dataset balancing methods. The performance metrics used for evaluation included precision, recall, F1-score, Hamming Loss, and overall accuracy, as detailed in Tables I to III.

Table I: Result of the multi-label MoA classification using unbalanced data with two techniques (Classifier Chains and Binary Relevance).

	Classifier Chains					*Binary Relevance*				
Models	All features									
	precision	recall	f1-score	Hamming Loss	accuracy	precision	recall	f1-score	Hamming Loss	accuracy
KNN	0.42	0.11	0.16	0.15	0.53	0.42	0.11	0.16	0.15	0.53
SVM	0.51	0.085	0.13	0.15	0.56	0.50	0.083	0.132	0.15	0.55
DT	0.45	0.05	0.09	0.15	0.51	0.45	0.05	0.09	0.15	0.51
RF	0.48	0.08	0.12	0.14	0.56	0.48	0.08	0.12	0.14	0.55
	Best features									
Models	precision	recall	f1-score	Hamming Loss	accuracy	precision	recall	f1-score	Hamming Loss	accuracy
KNN	0.42	0.11	0.16	0.15	0.55	0.42	0.11	0.16	0.15	0.54
SVM	0.47	0.086	0.13	0.14	0.57	0.50	0.083	0.14	0.15	0.56
DT	0.35	0.06	0.10	0.16	0.51	0.35	0.06	0.10	0.16	0.51
RF	0.62	0.08	0.12	0.14	0.55	0.48	0.48	0.8	0.13	0.55

Table II: Result of the multi-label MoA classification using undersampled data with two techniques (Classifier Chains and Binary Relevance).

	RandomUnderSampler - Classifier Chains					*RandomUnderSampler - Binary Relevance*				
Models	All features									
	precision	recall	f1-score	Hamming Loss	accuracy	precision	recall	f1-score	Hamming Loss	accuracy
KNN	0.16	0.06	0.06	0.16	0.47	0.16	0.05	0.06	0.16	0.47
SVM	0.092	0.03	0.092	0.16	0.48	0.1	0.27	0.03	0.16	0.48
DT	0.05	0.00	0.00	0.16	0.45	0.05	0.00	0.00	0.163	0.45
RF	0.09	0.01	0.01	0.17	0.45	0.05	0.00	0.00	0.163	0.45
	Best features									
Models	precision	recall	f1-score	Hamming Loss	accuracy	precision	recall	f1-score	Hamming Loss	accuracy
KNN	0.59	0.08	0.12	0.15	0.53	0.59	0.08	0.12	0.15	0.53
SVM	0.12	0.023	0.12	0.15	0.56	0.10	0.04	0.12	0.15	0.55
DT	0.12	0.02	0.03	0.15	0.48	0.12	0.02	0.03	0.15	0.48
RF	0.12	0.04	0.05	0.15	0.51	0.12	0.04	0.05	0.15	0.51

Table III: Result of the multi-label MoA classification using oversampled data with two techniques (Classifier Chains and Binary Relevance).

	RandomOverSampler - Classifier Chains					*RandomOverSampler - Binary Relevance*				
Models	All features									
	precision	recall	f1-score	Hamming Loss	accuracy	precision	recall	f1-score	Hamming Loss	accuracy
KNN	0.19	0.06	0.07	0.16	0.45	0.19	0.06	0.07	0.16	0.45
SVM	0.094	0.033	0.094	0.16	0.49	0.098	0.027	0.043	0.16	0.48
DT	0.6	0.00	0.00	0.16	0.45	0.06	0.00	0.00	0.16	0.45
RF	0.09	0.01	0.02	0.017	0.44	0.06	0.00	0.00	0.16	0.45
	Best features									
Models	precision	recall	f1-score	Hamming Loss	accuracy	precision	recall	f1-score	Hamming Loss	accuracy
KNN	0.38	0.09	0.14	0.15	0.53	0.38	0.09	0.14	0.15	0.53
SVM	0.10	0.027	0.14	0.15	0.48	0.10	0.27	0.13	0.15	0.48
DT	0.37	0.03	0.06	0.15	0.46	0.00	0.00	0.00	0.166	0.44
RF	0.37	0.04	0.07	0.17	0.45	0.11	0.02	0.03	0.17	0.45

techniques yielded diverse outcomes, demonstrating the complex dynamics of model performance in multi-label settings.

Using the Classifier Chains technique, where models are designed to consider label dependencies, we observed varied results across the models (Table I to III). When employing all features, SVM and RF performed slightly better in terms of accuracy (0.56) compared to KNN and DT. Notably, Random Forest showed a balanced performance with respect to precision and F1-score when best features were utilized, achieving an accuracy of 0.55 and the highest precision among all models at 0.62, albeit with a low recall (Table I). This suggests that while RF can accurately predict the correct labels, it is conservative in labeling, missing several true labels.

When RandomUnderSampler was applied in conjunction with the Classifier Chain technique, all models suffered a decrease in performance (Table II). This was particularly evident in terms of precision and recall, indicating difficulty in managing the reduced sample size while maintaining the ability to predict correct labels across multiple categories.

The Binary Relevance technique, which treats each label as an independent binary classification, revealed similar trends (Table I to III). Here, SVM consistently showed moderate performance improvements over other models, particularly with all features, where it achieved an accuracy of 0.55 and a precision of 0.50. Interestingly, when best features were used, RF's recall dramatically increased to 0.48 with an F1-score of 0.80 (Table I), indicating a significant improvement in identifying true positives across the labels.

Under conditions of RandomOverSampler using Binary Relevance, the performance generally declined (Table III). This was evident from the increased Hamming Loss and decreased accuracy across all models. It suggests that while oversampling increases the dataset size by replicating labels, it does not necessarily contribute to learning new information, which might lead to overfitting and decreased model generalizability.

The impact of using all features versus best features was notable across both techniques. Models generally performed better with best features, particularly KNN and RF, which improved precision and recall (Tables I to III). This highlights the importance of feature selection in improving model accuracy and handling class imbalances effectively.

Models trained under RandomUnderSampler conditions generally showed poorer performance than those under natural class distributions, particularly in terms of F1-score and recall, suggesting that significant data reduction may lead to the loss of critical information necessary for accurate classification (Tables II).

Conversely, RandomOverSampler often led to higher Hamming Loss and lower overall accuracy (Tables III), indicating potential overfitting issues as models were likely learning from repeated instances rather than from new information.

These experiments demonstrate the complexities and challenges of applying machine learning algorithms to multi-label classification tasks in drug discovery. Each model's performance varied significantly based on the classification technique, feature selection, and data balancing approach used, underscoring the need for careful consideration of these factors in model deployment.

7. Discussion

The results underscore several key insights into the application of machine learning techniques in multi-label drug MoA prediction. Firstly, the interplay between Classifier Chains and Binary Relevance techniques with different algorithms suggests no one-size-fits-all solution; the choice of technique must be aligned with the specific characteristics of the dataset and the computational constraints.

Secondly, the impact of feature selection and balancing techniques on model performance highlights the critical role of data preprocessing in machine learning workflows. While feature optimization generally leads to performance gains, the method of balancing class distribution requires careful consideration to avoid undermining the model's ability to generalize from the training data.

Lastly, the varying performance across algorithms under different experimental setups calls for a nuanced approach to selecting and tuning machine learning models for drug discovery applications. Future research should focus on developing more sophisticated methods for handling imbalanced data without compromising the quality of the model's predictions.

This comprehensive evaluation provides a foundation for further exploration into the best practices for deploying machine learning algorithms

in the complex field of pharmacology, ensuring that the benefits of these computational tools can be fully realized in practical applications.

8. Conclusion

This study conducted a comprehensive comparative analysis of four machine learning algorithms —KNN, SVM, DT, and RF— within a multi-label classification framework for predicting drug Mechanisms of Action (MoA). By utilizing Classifier Chains and Binary Relevance techniques and examining various feature sets and data balancing approaches, the research highlighted the nuanced performances of these algorithms in pharmacology.

The findings indicate that SVM and Random Forest generally provided the most robust performance, particularly with accuracy and handling complex datasets. Feature selection was crucial, consistently enhancing model performance, while data balancing techniques like RandomUnderSampler and RandomOverSampler often negatively impacted outcomes.

Future research should explore advanced ensemble methods and innovative data balancing techniques to leverage strengths across algorithms and address class imbalances effectively. This study underscores the potential of machine learning in drug discovery, emphasizing the need for tailored algorithm selection and preprocessing strategies to maximize predictive accuracy and efficiency in real-world applications.

References

[1] K. A. Berg and W. P. Clarke, "Making sense of pharmacology: inverse agonism and functional selectivity," International Journal of Neuropsychopharmacology, vol. 21, no. 10, pp. 962–977, 2018.

[2] J. Vamathevan, D. Clark, P. Czodrowski, I. Dunham, E. Ferran, G. Lee, B. Li, A. Madabhushi, P. Shah, M. Spitzer et al., "Applications of machine learning in drug discovery and development," Nature reviews Drug discovery, vol. 18, no. 6, pp. 463–477, 2019.

[3] M.-A. Trapotsi, L. Hosseini-Gerami, and A. Bender, "Computational analyses of mechanism of action (moa): data, methods and integration," RSC Chemical Biology, vol. 3, no. 2, pp. 170–200, 2022.

[4] S. Korkmaz, G. Zararsiz, and D. Goksuluk, "Drug/nondrug classification using support vector machines with various feature selection strategies," Computer methods and programs in biomedicine, vol. 117, no. 2, pp. 51–60, 2014.

[5] A. Yosipof, R. C. Guedes, and A. T. Garc´ıa-Sosa, "Data mining and machine learning models for predicting drug likeness and their disease or organ category," Frontiers in chemistry, vol. 6, p. 162, 2018.

[6] M. A. Mahyoub, L. A. Lekham, E. Alenany, L. Tarawneh, and D. Won, "Analysis of drug consumption data using data mining techniques and a predictive model using multi-label classification," in IIE Annual Conference. Proceedings. Institute of Industrial and Systems Engineers (IISE), 2019, pp. 864–869.

[7] M. R. Boland, F. Polubriaginof, and N. P. Tatonetti, "Development of a machine learning algorithm to classify drugs of unknown fetal effect," Scientific reports, vol. 7, no. 1, p. 12839, 2017.

[8] M. G¨utlein and S. Kramer, "Filtered circular fingerprints improve either prediction or runtime performance while retaining interpretability," Journal of cheminformatics, vol. 8, pp. 1–16, 2016.

[9] P. A. Sarkate and A. Deorankar, "Classification of chemical medicine or drug using k nearest neighbor (knn) and genetic algorithm," Int. Res. J. Eng. Technol, vol. 5, pp. 833–834, 2018.

[10] E. Pierson and C. Yau, "Zifa: Dimensionality reduction for zero-inflated single-cell gene expression analysis," Genome biology, vol. 16, pp. 1–10, 2015.

[11] S. Korkmaz, "Deep learning-based imbalanced data classification for drug discovery," Journal of chemical information and modeling, vol. 60, no. 9, pp. 4180–4190, 2020.

[12] M. S. Sorower, "A literature survey on algorithms for multi-label learning," Oregon State University, Corvallis, vol. 18, no. 1, p. 25, 2010.

[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

Hana Alalawi received the bachelor's degree in computer science from Umm Al-Qura University, in 2018. She is currently pursuing the master's degree in artificial intelligence. She has received three rewarding excellence regarding academic and research output in the past few years. Her research interests include machine learning, computer vision, speech and gesture recognition, and machine translation.

Manal Alsuwat received the bachelor's degree in computer science from Taif University, in 2019. She is currently pursuing the master's degree in artificial intelligence with Umm Al-Qura University. In the past few years, She has received four rewarding academic and research achievements. Her research interests include computer vision, machine learning, and speech recognition

Amani Alsabei received her B.S degree from Umm Al-Qura University, Makkah, Saudi Arabia, in 2017. She is currently a student in the M.S artificial intelligence at Computer Science Department at Umm Al-Qura University, Makkah, Saudi Arabia, and is expected to receive her degree in 2022

Sarah Al-Shareef received the B.S. degree in computer science from King Abdulaziz University, Jeddah, Saudi Arabia, in 2005 and the M.S. degree in advanced computer science from Sheffield University, Sheffield, United Kingdom, in 2009 and a PhD degree in computer science from Sheffield University, Sheffield, United Kingdom, in 2015. Currently, she works as Assistant Professor in Computer Science and Artificial Intelligence Department at Umm Al-Qura University, Makkah, Saudi Arabia. Her research interests include speech and Arabic technologies and especially automatic speech recognition ad acoustic modelling.