National College of Ireland

Sub-Optimal Hyperparameter Selection for Multi-Label Classifier Chains Predicting Cardiotoxicity from Gene-Expression Data

MSc Research Project
Data Analytics

Christopher Signorelli
Student ID: 19181027

School of Computing
National College of Ireland

Supervisor: Dr. Vladimir Milosavljevic

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Christopher Signorelli
Student ID:	19181027
Programme:	Data Analytics
Year:	2022
Module:	MSc Research Project
Supervisor:	Dr. Vladimir Milosavljevic
Submission Due Date:	19/9/2022
Project Title:	Sub-Optimal Hyperparameter Selection for Multi-Label Classifier Chains Predicting Cardiotoxicity from Gene-Expression Data
Word Count:	7218
Page Count:	22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	18th September 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Sub-Optimal Hyperparameter Selection for Multi-Label Classifier Chains Predicting Cardiotoxicity from Gene-Expression Data

Christopher Signorelli
19181027

Abstract

Robust multi-label classifier chains are difficult to optimise, due to the large search space of base model types and hyperparameters. This project demonstrates how robust MLC models can be trained within a subspace, using a pragmatic approach. A solution is proposed for training a robust multi-label classifier chain that predicts cardiotoxicity forms from known gene-expression data and drug properties. Empirical gene-expression data from the LINCS L1000 project have been joined with drug compound properties, curated by other researchers to create the model training set. The training set includes features for gene-expression responses to drug perturbation of cancerous cells, molecular descriptors, 79-bit estate fingerprints, and perturbation times. The multiple binary labels represent various cardiotoxicity outcomes for each of the drugs. An automated hyperparameter search was conducted on a relatively small search space, and led to the creation of a robust MLC chain, called *Best Means Chain*, with its performance ranking in the top-3 of 100 other chains. The main finding is that for this cardiotoxicity application, training a robust MLC chain can be achieved in a relatively straightforward manner in reasonable time, albeit with the trade-off of being sub-optimal compared to searching the full hyperparameter search space.

1 Introduction

This research project is primarily focused on developing a robust multi-label classifier (MLC) in the field of cardiotoxicity prediction, using empirical pharmacogenomics (PGx) data. Application of MLC models to cardiotoxicity forms is a relatively new area, and automated hyperparameter searching for MLC models is still an open issue (Wever et al.; 2021). The proposed solution builds on state-of-the-art research by Mamoshina et al. (2020) in developing a robust model, and exploits the gap regarding automated hyperparameter searching within a time-constrained framework (Wever et al.; 2021).

Cardiotoxicity refers to heart damage caused by cancer treatment. This makes empirical datasets, such as the LINCS L1000 (Subramanian et al.; 2017) increasingly important. With the recent growth of PGx experiments, now is a fertile time for discovery in this area. There are several inter-dependent cardiotoxicity forms, making MLC chains theoretically well-suited for their prediction. However, creating robust MLC models is challenging due to the complex hierarchy in the hyperparameter space (Wever et al.; 2021). Robustness of MLC models is important, especially when the hyperparameter

space has to be limited for computational feasibility. Given the large search space and resource demand for training classifier chains, automation is crucial, since many days or weeks of continuous training may be involved. Also, randomly generated hyperparameter sets can easily lead to model training divergence, which needs to be interrupted to allow the remaining experiments to run. Therefore, robustness needs to be introduced into the hyperparameter search logic.

Research Question: How can the hyperparameter search of a multi-label classifier, used to predict multiple cardiotoxicity outcomes from gene-expression data, be automated to train a robust model in reasonable time, given only a small search space?

Research Objectives:

- OBJ-1: Create a sufficiently large training dataset that combines gene-expression data, drug chemical properties, and cardiotoxicity outcome labels.
- OBJ-2: Perform feature selection for dimension reduction, and minimal model performance degradation.
- OBJ-3: Implement a MLC to match the state-of-the-art, capable of predicting multiple cardiotoxicity outcomes.
- OBJ-4: Automate the hyperparameter search for optimising the MLC, for training a robust MLC using the optimised hyperparameters.
- OBJ-5: Demonstrate the efficacy of the robust MLC and automation process, relative to other models.
- OBJ-6: Benchmark the computation times of the training experiments.

Contribution to the Scientific Literature:

- CONTR-1: A simple and pragmatic approach has been developed for automating the hyperparameter selection process for cardiotoxicity MLC, extending the prior art in (Mamoshina et al.; 2020).
- CONTR-2: This work contributes to the area of Auto-ML in MLC, where the proposed solution could serve as a starting point for machine learning practitioners, before attempting more complex approaches, which are still open issues (Wever et al.; 2021).
- CONTR-3: The proposed solution is expected to be generalisable beyond the scope of this project, and useful for applications where complex Auto-ML solutions are unnecessary, or not yet fully developed.

2 Related Work

2.1 Pharmacogenomics Overview

Pharmacogenomics (PGx) is a field of precision medicine that links genomics and drugs, with the aim of minimising adverse events in tailored drug therapy, while also maximising efficacy. PGx has experienced huge growth following the success of the Human Genome Project, and given the abundance of emerging data, modern advances in genomics and machine learning (ML) are stimulating new research (Pandi et al.; 2021). Before the era of precision medicine there were many adverse side-effects and ineffective drug responses, where variability of drug efficacy and toxicity was a significant challenge (Arbitrio et al.;

2021). While PGx has made it possible to identify predictive biomarkers, the validation process must be robust, and clinical-grade assays need to comply with strict regulations (Arbitrio et al.; 2021). Some studies (Sharifi-Noghabi et al.; 2021) have even begun to propose guidelines for the development of ML predictors of drug sensitivity. PGx covers a wide range of areas and applications, a small fraction of which are presented below.

2.2 Pharmacogenomics Research Areas

Oncology: Tran et al. (2021) investigated how deep learning (DL) has been applied in the field of oncology, for example cancer diagnosis, prognosis and treatment management. The authors stated that deep neural networks are well matched to oncology based models, where input data of different types can be combined, such as, clinical data, medical images, and genomic profiles. However data variability is a major challenge, for example where different laboratories have differently calibrated measuring devices. Other barriers include the cost of acquiring high-quality data with omic (genomic, methylation and transcriptomic) profiling, and the need for AI explainability in clinical settings, noting that advances in this area have been recently accelerated (Tran et al.; 2021). For DL models, a large amount of robust, well phenotyped data is needed for good model generalisability, and the typical data sources that satisfy this constraint are not publicly available (Tran et al.; 2021). This highlights the need for ML models to be made as robust as possible.

Antidepressant Treatments: Lin et al. (2021) reviewed ML and DL techniques used in PGx for major depressive disorder (MDD) antidepressant treatments, using neuroimaging and multi-omics data. Lin et al. (2021) stated that while research in this field has encountered challenges, ML and DL techniques are uncovering genomic variants and biomarkers linked to antidepressant treatments for MDD. They also suggested that future research should investigate multi-layer feedforward neural networks (MFNN) and generative adversarial networks (GAN) for predicting antidepressant treatment response and remission.

Single-Nucleotide Polymorphism Prediction: Poplin et al. (2018) implemented supervised deep learning using convolutional neural networks (CNN) to perform variant calling of single-nucleotide polymorphisms (SNP) from next-generation-sequencing (NGS) data. Variant calling is the process of detecting gene variants in DNA sequences. The DL model (called DeepVariant) was trained with no specialist genomics knowledge, and had high sensitivity, but low specificity (Poplin et al.; 2018). While the approach was very successful in predicting positive cases, there is potential to improve the prediction of negative cases.

DNA/RNA Protein Binding Site Prediction: Shadab et al. (2020) used two DL models to identify DNA-binding proteins (DBP), using only protein sequences. One of these (DeepDBP-ANN) was a traditional neural network and the other was a CNN (DeepDBP-CNN). The DeepDBP models seemed to have a noticeable difference between the training and test performance scores (in the order of 10%). This suggests that overfitting of the training set may have been an issue, with a potential lack of robustness in the model. Shadab et al. (2020) also reported that in similar ML studies, overfitting was very common.

Drug-Target Interaction: The prediction of drug-target interaction can help screen new potential drugs for market, prior to full clinical trials, where many side-effects could be predicted at early stages of the drug discovery process (Vaz and Balaji; 2021). Tsubaki et al. (2018) combined a graph neural network (GNN) and CNN for predicting drug-target interaction, using a single-label modelling approach, where the GNN was used for the drugs, and the CNN was used for the targets. Also, in contrast to other research, such as (Mamoshina et al.; 2020), their proposed GNN and CNN models did not rely on any chemical or biological features, such as fingerprint data. Also, the data was not empirical, where an assumption was made that similar known drug-target interactions are valid for labelling unknown interactions. The results were compared to those from traditional ML models, such as random forest, k-NN, SVM, and L2-logistic regression. It was reported that using a low-dimensional, shallow neural network has the potential to perform better than prior methods for both balanced and unbalanced datasets. While the results in Tsubaki et al. (2018) were positive, and the approach rigorous, potential restructuring of the single-label classifier (SLC) into a multi-label classifier (MLC), such as the classifier chain in (Mamoshina et al.; 2020) may allow correlations between different drug interactions to further increase the model performance and / or robustness.

2.3 Experimental Gene-Expression Studies

The National Institutes of Health (NIH)[1] established a gene-expression platform, called LINCS L1000 that led to the enhancement of a large public dataset called Connectivity Map (CMap), linking genes, drugs, and disease states (Subramanian et al.; 2017). The CMap dataset has been used by many researchers, and referenced $1,053$ times. In simple terms, the L1000 assay platform has provided gene-expression measurements for a range of cell lines in response to over $42,000$ chemical perturbagens. The current CMap data has over 1.3 million gene-expression profiles, where a profile refers to the identification of genes in a cell that make messenger RNA. Messenger RNA are the molecules that carry the genetic information to create proteins from DNA.

The data includes the gene-expression values and meta data describing how each sample in the assay was perturbed, in terms of chemical compound used, dosage amounts, dose times, and cell lines. Due to the large size of the dataset, the meta data is usually first queried to extract ID numbers for the desired chemicals and genes. The gene-expression data is then extracted using those IDs. The data is provided in 5 levels, each differing by the amount of post-processing involved. Level-1 is the expression data in their rawest form, and the level-5 is fully cleaned and processed. All levels are publicly available. The CMap data is one of the key datasets used in this project, based on the work by Mamoshina et al. (2020), discussed in Section 2.5.

Various other research has been performed with this data, including: 1) Application of deep learning for gene-expression inference (Chen et al.; 2016). 2) Building graph neural networks for identifying which approved drugs may be repurposed for breast cancer (Cui et al.; 2021). 3) Building binary classifiers to detect molecular initiating events (Bundy et al.; 2022).

[1] http://commonfund.nih.gov/lincs/

2.4 Classifier Chains

Classifier chains are highly regarded multi-label classifiers, and have been investigated with keen interest over the last decade (Read et al.; 2021). They are structured such that multiple binary classifiers are chained together in series, where each classifier predicts a single label. Each classifier appends the predicted label from the preceding classifier to its feature set. This allows inter-dependencies between labels to be captured by the model. Any binary classifier can be used as a base model in the chain, such as support vector machines, decision trees, logistic regression, and random forests) (Read et al.; 2021; Wever et al.; 2020).

Classifier chains have often shown to outperform individual classifiers when the labels are independently evaluated (Read et al.; 2021), and if necessary, chain ensembles can be used to reduce the error variance, albeit at the cost of average performance (Read et al.; 2021). Classifier chains are at risk of performing poorly when the labels are highly imbalanced (Read et al.; 2021), which is an issue for some of the labels in this project. Another challenge with classifier chains is their quadratic complexity with feature space expansion, however this is not a problem when the number of labels is less than ten (Read et al.; 2021).

In contrast to Auto-ML efforts for SLC models (Probst et al.; 2019; Bergstra et al.; 2015), research into Auto-ML for MLC models, such as classifier chains, is still in early stages, with recent work being done by Wever et al. (2020, 2021). So far, Auto-ML has only been applied to MLC models with genetic algorithms, grammar-based genetic programming, hierarchical task network planning, and a classifier specific approach with neural networks (Wever et al.; 2021). Wever et al. (2021) found that all methods struggled with the search space, and also that replacing these complex search methods with simpler ones, such as random searching, comes at the cost of not finding the globally optimal hyperparameters. Wever et al. (2021) note that currently, complex methods are still mostly infeasible out of the box. As such, simpler approaches still offer a reasonable tradeoff.

2.5 Cardiotoxicity Prediction from Gene-Expression Data

Concept: Mamoshina et al. (2020) focused on multi-label prediction of cardiotoxicity outcomes, using the L1000 gene-expression data, joined with drug properties. They comprehensively tackled the problem, devising a novel multi-label classifier, expanding the group of cardiotoxicity outcomes being predicted. This project builds on that research, proposing a complementary approach for increasing model robustness. The salient details of Mamoshina et al. (2020) are discussed below.

Datasets: To link gene-expression data with cardiotoxicity outcomes, Mamoshina et al. (2020) sourced and curated the various data from public databases including: Connectivity Map (CMap)[2], Medical Subject Headings (MESH)[3], Side Effect Resource (SIDER)[4], Chemical Translation Service (CTS)[5], Medical Dictionary for Regulatory

[2]https://clue.io/

[3]https://www.ncbi.nlm.nih.gov/mesh

[4]http://sideeffects.embl.de/

[5]http://cts.fiehnlab.ucdavis.edu/

Activities Terminology (MedDRA)[6], DrugBank[7], and 'R' CDK library[8]. These datasets allowed the L1000 experiments to be linked through the Broad Institute drug IDs to side-effects, cardiotoxicity drug states, chemical interaction compound identifiers, molecular descriptors, compound fingerprints, and cardiac / vascular disorders. The theory behind the fingerprints is described in (Hall and Kier; 1995).

The drug dataset curated by Mamoshina et al. (2020) consisted of $9,933$ samples, split into training (291 drugs, $8,237$ samples) and test sets (66 drugs, $1,696$ samples). These were collapsed into matrices of size 340 x $1,154$ and 340 x 746 respectively, where each row corresponded to gene-expression values for individual cell lines, dosage amounts, and dose times. The final 'Mamoshina' training set, made available for this project, consisted of 340 features, 7 labels, and $1,154$ rows. The test set consisted of 746 rows with drugs not present in the training set. The features consisted of 254 CMap gene-expression values, 7 molecular descriptors, and a 79-bit estate fingerprints. The molecular descriptors were molecular weight (MW), partition coefficient (XLogP), atomic polarisabilities (apol), topological polar surface area (TopoPSA), polar surface area expressed as a ratio to molecular size (tpsaEfficiency), Ghose-Crippen LogKow (ALogP) and molar refractivity (AMR) (Mamoshina et al.; 2020).

Modelling Approach: Due to observed correlation between the multiple cardiotoxicity labels, Mamoshina et al. (2020) selected a classifier chain as the model framework, within which they implemented various base classifiers, including elastic net logistic regression (ELNET), random forest (RF), gradient boosting (GBM), and categorical boosting (CATBOOST). They also compared the classifier chain results with single binary classifiers. The MLC performance was evaluated using Matthews correlation coefficient (MCC), averaged across each label. Note that the MCC is mostly intended for SLC models, rather than MLC ones. In constrast, the Jaccard score is suited to both SLC and MLC models. Cohen's Kappa score was also used to estimate the accuracy of each label, and also between transcriptional features. With regards to hyperparameter selection, Mamoshina et al. (2020) described that they selected optimal values, and provided these in the supplementary material, however a discussion on the selection process was not presented. This gap can be addressed with automated hyperparameter searching.

Feature Selection and Robustness: After establishing that the RF model was the best performer, Mamoshina et al. (2020) experimented with different combinations of feature subsets on the RF classifier chain, including: 1) Gene-expression features only, and 2) Molecular descriptors and fingerprints only. It was found that models trained on all features performed better than any subset. For robustness, different cross-validation methods were trialled, including: 1) Random validation sets, and 2) Leave-drug-out validation sets. It was found that random cross-validation led to overly high performance due to overfitting, whereas the leave-drug-out approach showed less overfitting, with lower performance. This was the approach to model robustness.

Further Observations: Mamoshina et al. (2020) established the first machine learning approach capable of predicting six forms of drug-induced cardiotoxicity from both gene expression and molecular descriptors data. They reported that the RF classifier chain showed good predictive ability for cardiotoxicity forms, outperforming SLC models, and that coupling the gene-expression data with the molecular descriptors improved predictive power.

[6]https://bioportal.bioontology.org/ontologies/MEDDRA

[7]https://www.drugbank.com/

[8]https://cran.r-project.org/web/packages/rcdk/index.html

2.6 Identified Research Gaps

Several research gaps have been identified from the literature and preliminary analysis.

Feature Selection: This was somewhat investigated in (Mamoshina et al.; 2020) by experimenting with a few large feature subset blocks, namely gene-expression data, molecular descriptors, and 79-bit fingerprints. The gap being addressed in this research is to fully test all possible feature combinations, using recursive feature elimination prior to training the model. Investigating further reductions in feature count dramatically helps speed up training.

Training Set: The size of the 'Mamoshina' training set is considered quite small for the purpose of this study. This gap is addressed by re-sourcing a larger portion of the L1000 CMap data with an increased number cell lines.

Choice of Performance Metric: Rather than use the SLC metrics (MCC and Cohen Kappa), as in (Mamoshina et al.; 2020), the Jaccard score is considered to be more suitable for this project, due to its suitability for both SLC and MLC models.

Automated Hyperparameter Selection: A core aspect of the research question is automating the hyperparameter search for the robust MLC. Building on the approach in (Mamoshina et al.; 2020), and considering that Auto-ML hyperparameter selection for MLC models is still an open issue (Wever et al.; 2021), the proposed solution addresses this gap.

Experiment Interruption: A requirement for automation is the continuous training of several experiments without getting stuck in infinite loops. The proposed solution implements timed interruption of the training process to ensure that the overall experiment process can complete in reasonable time.

3 Methodology

3.1 Datasets

Curation: Data were curated for this project from CLUE CMap[9], and (Mamoshina et al.; 2020). The CMap data contains only gene-expression data, measured and processed through the LINCS L1000 project. The gene-expression data represents the extent to which a variety of gene assay samples have been experimentally expressed using controlled drug stimuli. The genes span many biological cell lines, and other features in the drug perturbation stimuli include dosage amount and dose time.

The data from (Mamoshina et al.; 2020) were supplied on request by email, and contain a small subset of gene-expression data from the CMap source, joined with separately curated cardiotoxicity labels, drug compound molecular descriptors and 79-bit fingerprints. The gene-expression data in the provided Mamoshina training dataset were only associated with one cell line, having approximately 1000 observations. This was considered too small to obtain sufficiently accurate results in this study, so to increase the sample size, the gene-expression data was replaced with newly collected CMap data with more cell lines (each having a row count of at least 100 cases). This increased the dataset to approximately 9000 observations, where a feature selection process was adopted to automatically remove data with poor predictive ability.

[9] https://clue.io/

Choice of Raw CMap Dataset: The CMap data are publicly available at different levels (1 to 5), differing in terms of the degree of processing from the raw laboratory experiment readings. Level-1 is the most raw, and level-5 is the most processed. The creators of the CMap data suggest using Level-5 data, however Mamoshina et al. (2020) reported using Level-3 data. On close inspection, the provided Mamoshina dataset appears to be more closely matched with the Level-2 data. This led to choosing the Level-2 data in this project with some further standardisation.

Joining CMap and Mamoshina Data: The two datasets are inner-joined using the drug identification numbers. Based on the exploratory data analysis (EDA), the data are also filtered to keep only rows where the dose time was 6 or 24 hours, and the dosage had a molar concentration of $10\mu M$. The joined dataset has 354 features, 7 labels, and 9106 rows. It is split into training and test datasets to have the same respective drugs as in the Mamoshina training and test data. Finally, the data are standardised by removing the mean, and scaling to unit variance.

Feature Selection: Feature selection is used to reduce the dimensionality of the feature set, and increase the predictive power of the trained model. This is implemented in Python using random forest binary classifiers and recursive feature elimination, with one model per label. The process is applied to each independent label, and the reduced feature set that leads to the highest model performance is selected as the final set. Assessment is based on the highest Area Under the Curve (AUC) score, also factoring in the F1 scores.

3.2 Modelling and Training

Random Forest Classifier Chains: The chosen model for this project is a classifier chain to closely match the one used in (Mamoshina et al.; 2020). Figure 1 shows the model structure, comprising several random forest binary classifiers chained together, where the output predictions from one stage are appended to the feature set of the following stage. The base features, before chain linking, are obtained from the feature selection process, and each model in the chain predicts one of the multiple labels. To mimic the approach in (Mamoshina et al.; 2020), the order of the labels (and chain stages) is maintained as: 1) *Vascular.disorders*, 2) *Cardiac.disorder.signs.and.symptoms*, 3) *Cardiac.arrhythmias*, 4) *Heart.failures*, 5) *Coronary.artery.disorders*, 6) *Pericardial.disorders*, 7) *Myocardial.disorders*. Each stage has a training and validation phase, using 5-fold cross-validation. This validation phase deviates slightly to the leave-drug-out validation in (Mamoshina et al.; 2020), due to difficulty in applying the Mamoshina approach with the Python library module used.

Best Means Chain: The novel work in this project is the creation of a robust random forest classifier chain, building on the work in (Mamoshina et al.; 2020). The concept of *Best Means* is that the hyperparameters that on average lead to increased model performance (measured by the Jaccard score), are selected to train a single classifier chain, called the *Best Means Chain*. The best hyperparameters are selected by grouping the results from 100 classifier chain training experiments, and calculating the mean Jaccard scores for each hyperparameter value. The values with the highest mean score are then chosen. The idea is that using the set of hyperparameters which generally yield the highest model performance will increase robustness of the model when unseen data is used to predict cardiotoxicity. As with any robust modelling approach, this robust model is expected to perform highly most of the time but not necessarily always. Section 6 presents the results of the hyperparameter selection process, arising from 100 individually

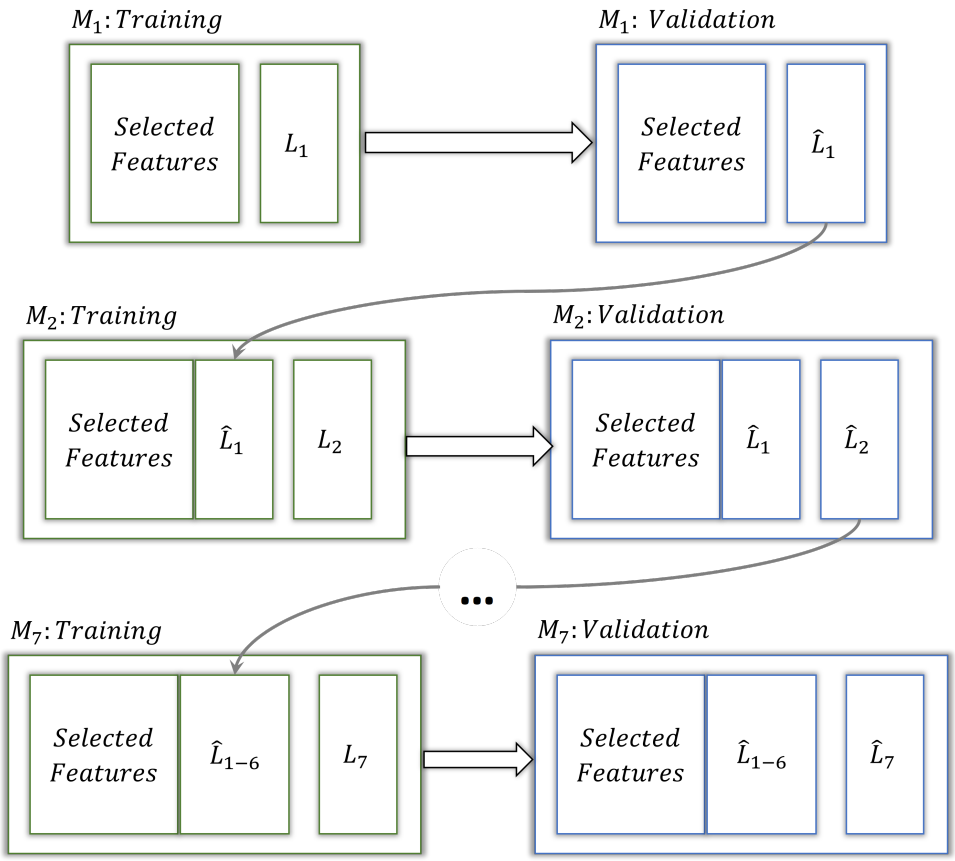


Figure 1: Classifier Chain Structure.

trained classifier chains. The hyperparameters include:

- *n_estimators*: number of trees in the forest.
- *max_features*: maximum number of features considered for splitting a node.
- *max_depth*: maximum number of levels in each decision tree.
- *bootstrap*: method for sampling data points.
- *min_samples_leaf*: minimum number of data points allowed in a leaf node.
- *min_samples_split*: minimum number of data points placed in a node before the node is split.

Mamoshina et al. (2020) used a subset of these hyperparameters (*ntree* and *mtry*) in their 'R' implementation for *n_estimators* and *max_features* respectively. *ntree* is equivalent to *n_features*, however *mtry* differs from *max_features* slightly, in that it is defined as an exact value rather than a range of values. Note also that *n_estimators* and *max_features* are commonly described as being the most influential hyperparameters for the random forest algorithm. The remaining ones used in this study were chosen from commonly found online examples. The decision to use randomly selected hyperparameters within relevant ranges addresses the small search space goal in this project, as a practical trade-off against running all possible combinations, which would be time-prohibitive. The ranges for *n_estimators* and *max_features* are based on those in (Mamoshina et al.; 2020) to mimic the search space as closely as possible. The additional hyperparameters are included to deepen the hyperparameter search space, from which the small subset is sampled.

Ensemble Chain: To help benchmark the *Best Means Chain*, an alternative classifier chain is created. This is a direct-voting ensemble of the 100 individual chain predictions. The algorithm averages the output probabilities for each label across the individual chains, and applies a 0.5 probability threshold to set the binary output to either zero or one.

Training Time: Training experiment runs reveal that training duration varies dramatically, with times ranging between 0.5 and 2 hours, sometimes diverging. A Python class-based multi-threading approach is used to automatically halt the training process after 1 hour and continue with the next experiment. The whole training process for the 100 experiments runs continuously for approximately 1 week. Each classifier chain is saved to a single model output file, for subsequent post-processing.

3.3 Post-Processing and Evaluation

Assembly of the Training Results: The first stage of post-processing involves loading each of the model artifact files, and linking the models, hyperparameters and experiment performance results. Each row in the results data frame corresponds to a single model within each chain, leading to 700 rows from the 100 chains and 7 labels. The calculated metrics include precision, recall, F1 score, MCC score, and Jaccard score. Note that the metrics for each row correspond to the random forest binary classifiers at each chain stage. Correlation statistics between each of the performance metrics are presented in Section 6.

Performance Metric Calculations: The second stage of post-processing is to group the rows by each hyperparameter, and calculate the aggregate mean Jaccard scores. The Jaccard score has been chosen to assess relative chain performance, as it can easily be applied to both single-label and multi-label outputs. It is also easily interpretable, ranging between 0 and 1, with 1 being a perfect score. In contrast, Mamoshina et al. (2020) used the MCC on the chain's random forest binary classifiers, then averaged them to score the overall model. Section 6 shows the grouped Jaccard scores for two hyperparameter examples, plotted in descending order to clearly show the best performing hyperparameter values across the 100 chains. The best performing set of hyperparameters are then used to train the *Best Means Chain*. The individual random forest chains, *Best Means Chain* and *Ensemble* chain are then ranked according to their Jaccard scores. The results are presented in Section 6.

Computation Time and Ensemble Size: The trained model artifact properties are analysed to determine the computation time of each training run and file size. Correlation statistics are also calculated between the Jaccard scores, computation time (training + file writes), and file size. Further analysis is conducted to search for diminishing returns in the computation time and model performance to determine if there is an optimal time at which computation should be halted. Diminishing return analysis is carried out to investigate if there exists an ensemble size beyond which, additional benefit in training more chains is limited. The process involves recalculating the ensemble output for increasing numbers of chains between 2 and 100, then plotting the Jaccard score vs. ensemble size. The results are presented in Section 6.

4 Design Specification

Hardware and Software Platforms: All code was executed on a standard spec HP Pavilion laptop, running Windows 10, with a 6-core AMD Ryzen 5 4500U processor, 2.375 Ghz, Radeon Graphics card, 8GB RAM, and a 500GB hard drive. Anaconda and Jupyter Notebooks were used as the programming framework, with all code being written in Python. Standard library packages, such *pandas, numpy, pickle*, etc. were used for the common tasks, and specialist packages including scikit-learn[10] and cmapPy[11] were used for machine learning related tasks and working with the CMap data.

Data Curation and Merging: The CMap gene-expression data was sourced from the CLUE CMap[12] project. The data consists of three files: 1) Perturbation information, *GSE92742_Broad_LINCS_sig_info.txt*, 2) Gene information *GSE92742_Broad_LINCS_gene_info.txt*, and 3) Gene-expression data *GSE92742_Broad_LINCS_Level2_GEX_epsilon_n1269922x978.gctx*. The first two are relatively small files, however the gene-expression data is close to 5 GB in size. Therefore, the information files are used to obtain the indexes of only the required gene-expression data. These are then used to ingest the gene-expression values.

The Mamoshina dataset was sourced directly from the authors of (Mamoshina et al.; 2020), who provided a link to *.csv* files via email. The genes and drugs in the Mamoshina datasets are used to collect the meta data from the CMap datasets before ingesting the gene-expression data as described above. The curated and merged datasets are saved into Python binary 'pickle' files called *train_v2_2.pkl* and *test_v2_2.pkl*. These data are also standardised, using Python's *scikit-learn* package, where the standard scaler object is stored in *scaler.pkl*.

Exploratory Data Analysis: The merged data is analysed for distribution characteristics in the numerical features, and value counts for the categorical features. The histograms and bar charts are created using Python's *matplotlib* package. The final datasets formed for model training and validation are saved into 'pickle' files *v22_train_x_cc.pkl*, *v22_train_y_cc.pkl*, *v22_test_x_cc.pkl*, and *v22_test_y_cc.pkl*.

Feature Selection: Feature selection pipelines, using a random forest classifier and recursive feature elimination are trained separately for each of the labels, requiring approximately 30 min for each set of features to be reduced. *Scikit-learn* is used for both training and evaluation.

Modelling and Training: The training process involves stepping through a large number of randomly selected hyperparameter combinations until 100 random forest classifier chains are successfully trained. *Scikit-learn* is used to train the models and extract the final performance metrics. Early training experiments showed that some hyperparameter combinations unpredictably led to insolvable models. A class-based processor threading method remedies this by automatically interrupting the experiments and moving on to the next run. Using this approach, the overall training process continues uninterrupted with the 100 experiments completing in approximately one week.

Post-Processing and Evaluation: The final results are gathered into *pandas* data frames and saved to *all_results.pkl*. The graphical outputs (bar charts, box plots and correlation charts) are created using *matplotlib*.

Feature Expansion: Due to the feature selection process, the current design is

[10] https://scikit-learn.org/stable/
[11] https://clue.io/cmapPy/
[12] https://clue.io/

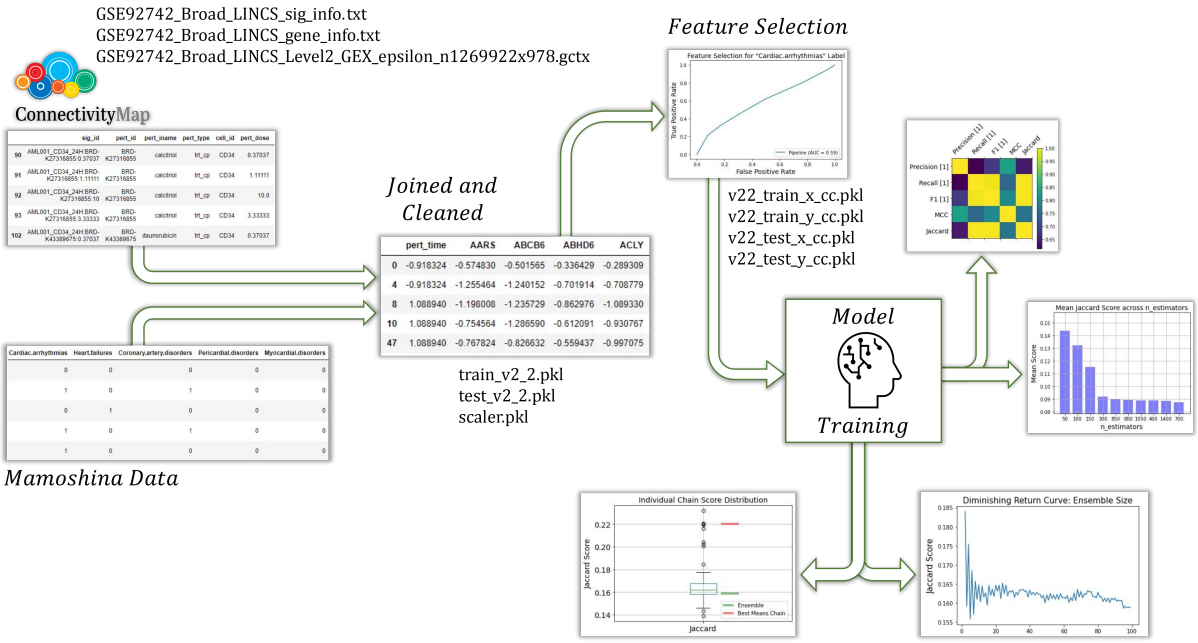


Figure 2: End-to-End Design.

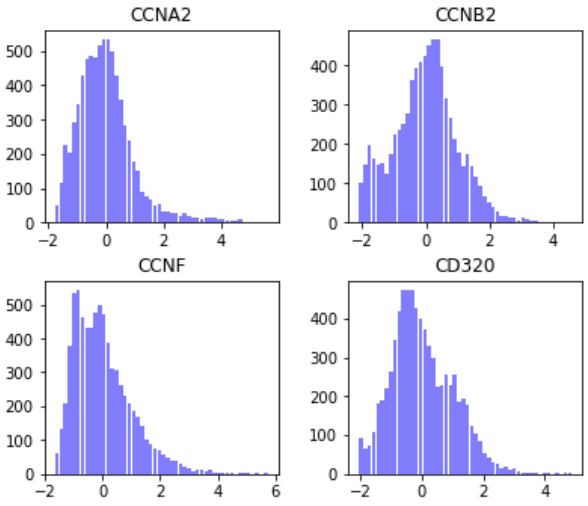
suitable for a specific subset of genes, molecular descriptors, and estate fingerprint bits. If additional genes, chemical properties, and / or drugs are required, the training and test datasets would need to be rebuilt and the model retrained. Also, if new drugs are introduced, they would need to be accompanied with labels for each of the seven cardiotoxicity outcomes. The feature selection process would also need to be redone. Figure 2 shows the overall end-to-end process described above.

5 Implementation

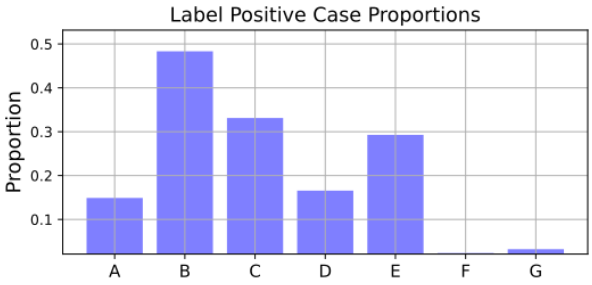
Concept: Following a process of data curation, transformation, and training, the final solution in this study is to predict seven cardiotoxicity outcomes based on measured gene expression data for known drugs and their chemical properties. To do this, a robust random forest classifier chain is trained, based on hyperparameter values, proven on average to yield high model performance.

Best Means Chain Model Predictions: After training, the *Best Means Chain* model is validated using a test set to predict cardiotoxicity outcomes with drugs unseen by the trained model. All performance metrics are calculated based on these predictions and known true values in the test set.

Evaluation and Benchmarking: The *Best Means Chain* performance is compared to the 100 individual chains and the direct-voting *Ensemble* chain. The test set is used to obtain the predictions from all models, with the Jaccard similarity score being used to assess performance. In addition to model performance, measurements are collected for the trained model artifact sizes, and computation times (training + file write operations). These are analysed to assess potential associations between model performance and computation times and artifact storage requirements. The motivation is to identify potential time and storage savings to be made during the model training process. Ensemble size is also investigated for potential computation time savings at the post-processing stage.



(a) Example Gene-Expression Distributions.



(b) Positive Case Proportions for each Label: Vascular.disorders (A), Cardiac.disorder.signs.and.symptoms (B), Cardiac.arrhythmias (C), Heart.failures (D), Coronary.artery.disorders (E), Pericardial.disorders (F), Myocardial.disorders (G).

Figure 3: Feature and Label Distributions.

6 Results and Discussion

6.1 Training Dataset

Discrepancies between Reported and Provided Mamoshina Datasets: The training set in the Mamoshina data obtained for this project, matches the reported dimensions in (Mamoshina et al.; 2020), however the provided test set size differs slightly. Also, the entire Mamoshina dataset of $9,933$ samples is not available, so it cannot be validated how the entire dataset was collapsed into the smaller training sets. Further, the size of the smaller training set is considered too small for the purpose of this project. This is the main motivation for re-sourcing the CMap gene-expression data and joining it with the drug-related portions of the Mamoshina dataset.

Note also that Mamoshina et al. (2020) reported using L1000 Level 3a - NORM data, however on inspection, similar gene-expression values could not be found between the Mamoshina dataset and the online CMap source, where values differed by one to two orders of magnitude. This gap is reduced by using the Level 2 CMap data, although it is not normalised at source. It is supposed that Mamoshina et al. (2020) carried out transformations on the Level 3a - NORM data, however this could not be confirmed. This unresolved discrepancy makes direct performance comparison with this project difficult. Focus in this project has been firmly placed on relative performance of the *Best Means Chain* with the other chains in this project, rather than absolute comparisons with the Mamoshina model performance.

6.2 Exploratory Data Analysis and Transformations

Figure 3a shows example gene expression distributions in the CMap dataset. All other genes show similar distributions, most of which are nearly normal. Some do have some slight right-skew, although most of the skewness is due to outliers. Figure 3b shows the distribution of labels from the Mamoshina dataset, where a large variability can be seen between the level of imbalance between the positive and negative cases. The proportions refer to the proportion of positive cases to all cases.

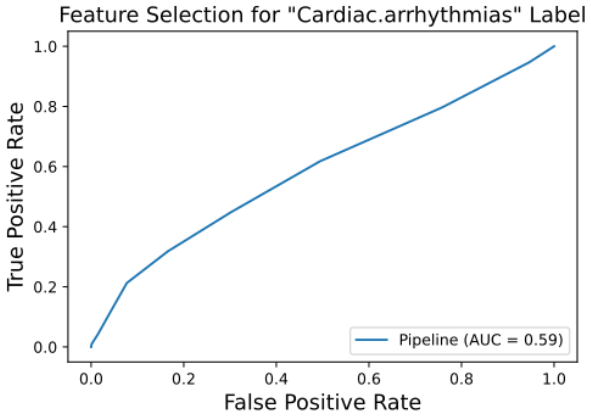
Confusion Matrix:
[[825 69]
 [562 152]]

Classification Report:

	precision	recall	f1-score	support
0	0.59	0.92	0.72	894
1	0.69	0.21	0.33	714
accuracy			0.61	1608
macro avg	0.64	0.57	0.52	1608
weighted avg	0.64	0.61	0.55	1608

Number of selected features: 221

(a) Performance Metrics.



(b) ROC Curve and AUC Score.

Figure 4: Best Feature Selection Model.

6.3 Feature Selection Results

Figure 4 shows the best performance results from the feature selection process, and corresponds to the *Cardiac.arrhythmias* label. Figure 4a shows the confusion matrix and the classification report. From left to right, top to bottom, the confusion matrix shows the [*True-Negative*, *False-Positive*, *False-Negative*, *True-Positive*] having the values [825, 69, 562, 152] respectively. The row of highlighted metrics (precision, recall, f1-score) corresponds to cardiotoxicity outcomes being positive, i.e. drugs with reported cardiotoxicity. The support values show that there are 714 out of 1608 occurrences in the test set with a positive value. The number of selected features for this model is 221, reduced from the full set of 354. Figure 4b shows the Receiver Operating Characteristic (ROC) curve for the model, with an AUC score of 0.59, 9% better than randomly guessing the outcome.

While there is a noticeable reduction in the number of features, the model training time could be significantly improved if the number of features could be reduced further, for example to less than 50. One possible approach might be to use Principle Component Analysis (PCA), which could potentially also lead to higher model accuracy from the projection of features onto orthogonal axes.

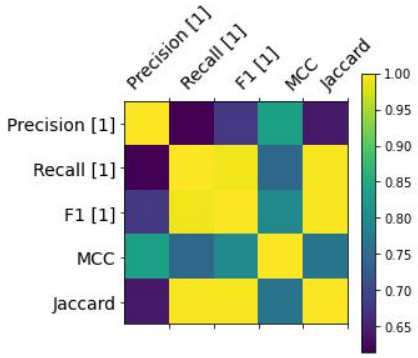
One issue that has not been dealt with in this project is that the optimal feature set depends on which label is used to train the feature selection model. However the current implementation only allows for one feature set to be applied to all classifiers in the chain. This causes the feature selection for the overall classifier chain to be sub-optimal. Improved results might be achieved if the classifier chain algorithm could be modified to allow each stage of the chain to have its own optimal base feature set.

6.4 Model Performance Metrics

Figure 5 shows the correlations between each of the classifier chain model performance metrics (precision, recall, F1, MCC, Jaccard), calculated for each model in each chain. Relative to the Jaccard score all metrics have a high correlation, with values above 0.64, two of which (recall and F1) have extremely high correlation values above 0.99. The MCC score has a high correlation of 0.76. Another benefit of choosing the Jaccard score to evaluate the models is that it has a very high correlation with recall, which when

	Precision [1]	Recall [1]	F1 [1]	MCC	Jaccard
Precision [1]	1.000000	0.614561	0.678289	0.833098	0.642144
Recall [1]	0.614561	1.000000	0.991341	0.745310	0.996319
F1 [1]	0.678289	0.991341	1.000000	0.798751	0.995641
MCC	0.833098	0.745310	0.798751	1.000000	0.764570
Jaccard	0.642144	0.996319	0.995641	0.764570	1.000000

(a) Correlation Coefficients.



(b) Graphical Representation.

Figure 5: Performance Metric Correlations.

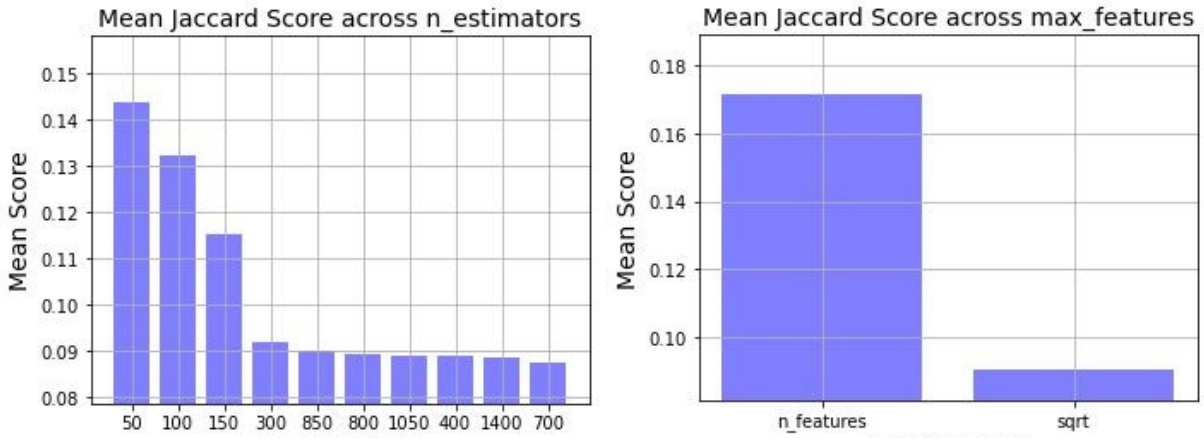
maximised, results in minimising false-negatives. This would be considered safer in a medical context such as cardiotoxicity prediction.

The choice of using the Jaccard score in this study deviates from (Mamoshina et al.; 2020), where MCC was used. This makes it difficult to make absolute performance comparisons, however it was decided that Jaccard score is more suitable for calculating performance metrics for multi-label classifiers. Given that the primary goal of this research is to create a robust classifier chain, only relative scores between the individual chains, *Best Means Chain* and the *Ensemble* are strictly needed. For that purpose, the choice of Jaccard score is considered appropriate.

6.5 Best Means Hyperparametmers

Figure 6a and Figure 6b show the mean Jaccard scores for two hyperparameters, $n\_estimators$ and $max\_features$, commonly used with random forest classifiers. It can be seen that $n\_estimators = 50$ leads to the highest mean score of approximately 0.143, and that $max\_features$ leads to a much higher mean Jaccard score with a value of $n\_features$. Note that $n\_features$ refers to the total number of features in the training set, and *sqrt* refers to the square root of that number.

The same approach was used with all other hyperparameters, leading to the set of *Best Mean Hyperparameters*, shown in Table 1, and used to train the *Best Means Chain*. Note that *Chain ID* and *Model ID* refer to the individual chain ID and model (or label) ID numbers. As discussed in Section 6.6, this *Best Means Chain* approach has proved very effective, resulting in a high Jaccard score, relative to the individual chains and *Ensemble* chain.



(a) Hyperparameter: $n\_estimators$. (b) Hyperparameter: $max\_features$.

Figure 6: Averaged Jaccard Scores across Grouped Training Results.

	Hyperparameter
Chain ID	131
Model ID	1
n_estimators	50
max_features	n_features
max_depth	40
bootstrap	False
min_samples_leaf	2
min_samples_split	2

Table 1: Best Means Hyperparameters.

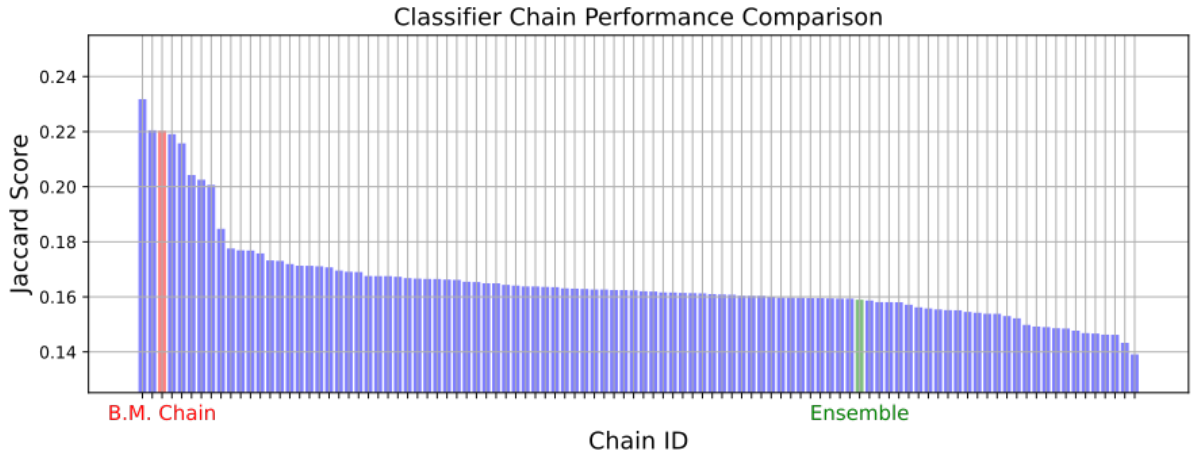


Figure 7: Chain Performance Rankings. Individual Chains (blue). Best Means Chain (red). Ensemble (green).

6.6 Chain Rankings

Figure 7 shows the ranking of all 100 individual chains (blue), the *Ensemble* of those chains (green), and the *Best Means Chain* (red). The Jaccard score here is the multi-label version, calculated for each chain. The *Best Means Chain* has outperformed most, which is also highlighted in the boxplot in Figure 8. The red marker shows that the *Best Means Chain* performance lies well into the upper outlier region, and the *Ensemble* is closely situated to the bottom of the interquartile range. These results demonstrate that the *Best Means* approach has been very effective at producing a robust model, in contrast to all other chains in this project.

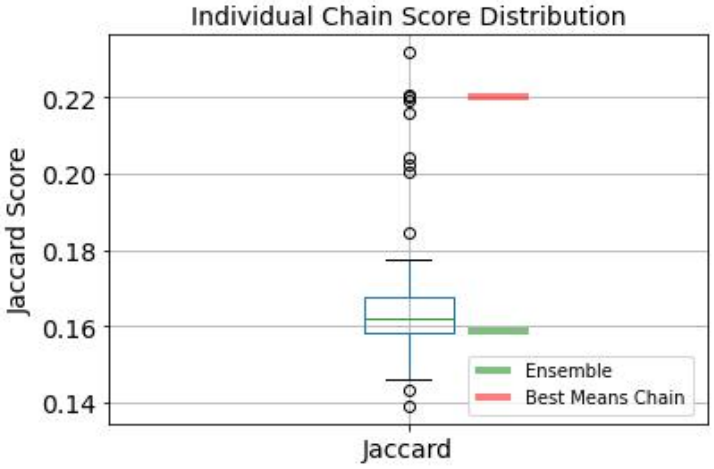


Figure 8: Boxplot Distribution of Chain Performance.

The configuration manual for this project (Signorelli; 2022) shows how the training experiments can be run in a *Short Demo Mode* that allows the code to execute in approximately 20 min, as opposed to the 1 week needed for the full set of experiments. Figure 9 shows that the *Best Means Chain* performs well even with a small number of individual chains, however the relative benefit of the approach increases with the number

of chains. This highlights generalisability of the approach. Future work could investigate the relationship between the number of chains and the *Best Means Chain* performance in more depth. As part of that work, a curve of diminishing returns would help identify the point at which there is no further benefit to increasing the number of chains, as the training time can be intensive.

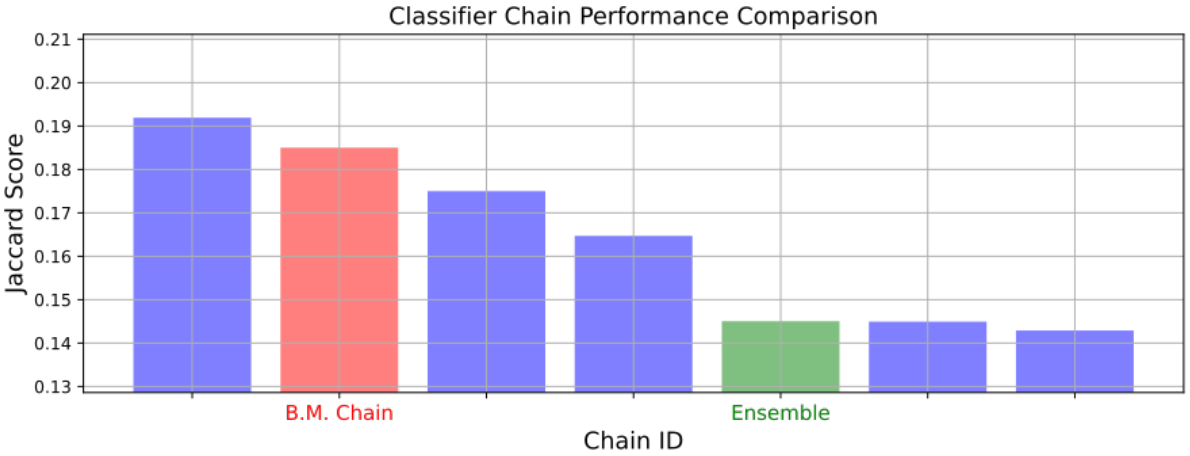


Figure 9: Chain Rankings (Short Demo).

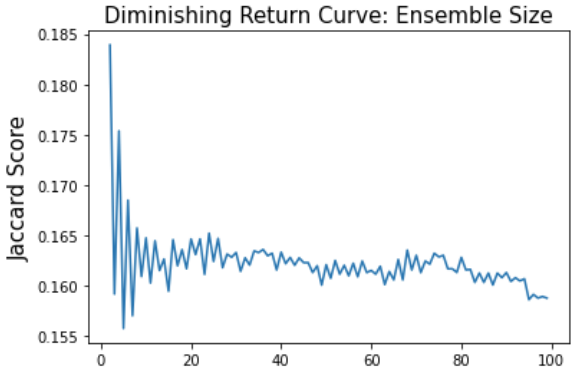
The positive trend of Jaccard performance on the test set suggests that the robustness of the *Best Means Chain* approach is generalisable to other similar data sets. Future work could look at how the approach works with unrelated datasets, to determine how well generalisability extends to a wider context. The approach to robustness in (Mamoshina et al.; 2020), was to use a leave-drug-out cross validation approach, where each fold (or sub-division) in the validation set did not contain drugs from any other fold. Mamoshina et al. (2020) reported that this improved the robustness of the model. Carrying on from this project, future work could investigate whether a combination of the two approaches leads to superior performance of the individual ones alone.

6.7 Ensemble Size and Computation Times

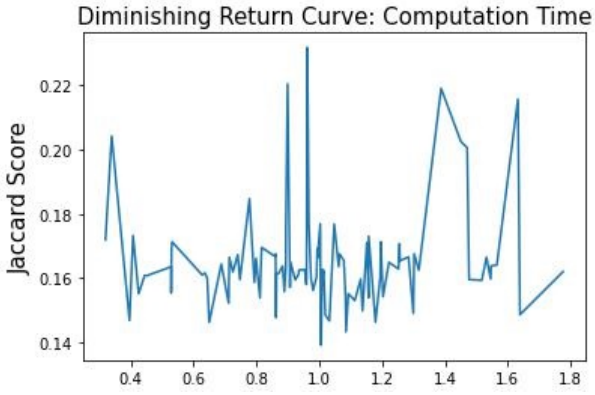
Ensemble Size: Figure 10a shows the diminishing return curve for ensemble size. Following some initial transient oscillations, the curve is quite flat, with a slight downward trend beyond approximately 20 chains. This suggests that for the current application, limited benefit can be expected from large ensemble sizes. Future work could investigate whether the random selection of individual chain subsets, used in each ensemble voting calculations might change the results. Additionally, individual chains with different classification models, for example support vector machines, k-nearest neighbours, and the various other models in (Mamoshina et al.; 2020) could be added to the ensemble. These different model types could potentially increase the *Ensemble* performance.

Computation Times: Figure 10b shows the *Computation Time* curve, and Figure 11 shows the correlation between the Jaccard performance metric, the computation time, and artifact size of the trained models. The Jaccard score can be seen to have no correlation with computation time, and a slight negative correlation with artifact size. The *Ensemble Size* curve also suggests that there is no clear relationship between Jaccard score and computation time, although there is potentially a small band of computation

times that exhibit high performances between approximately 1.3 and 1.7 hours. This should be investigated in future work, and considering that the *Best Means Hyperparameters* appear to be well correlated with Jaccard score, it would be insightful to search for correlations between the *Best Means Hyperparameters* and computation time. If no correlation exists, then future training experiments could be potentially interrupted earlier in the process. This could dramatically reduce the overall training time.



(a) Ensemble Size (Number of Individual Chains).



(b) Computation Time (hr).

Figure 10: Diminishing Return Curves for Ensemble Size and Computation Time.

	Computation Time (hr)	Size (MB)	Jaccard
Computation Time (hr)	1.000000	0.437228	0.089430
Size (MB)	0.437228	1.000000	-0.295665
Jaccard	0.089430	-0.295665	1.000000

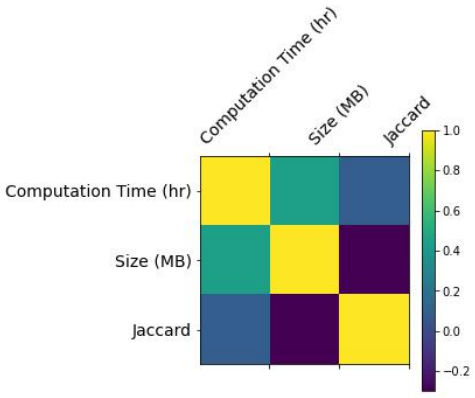


Figure 11: Correlations between Performance, Computation Time, and Artifact Size.

7 Conclusions and Future Work

This research project has achieved all of the objectives for answering the research question: "How can the hyperparameter search of a multi-label classifier, used to predict multiple cardiotoxicity outcomes from gene-expression data, be automated to train a robust model in reasonable time, given only a small search space?". To achieve OBJ-1, empirical gene-expression (CMap) data from the NIH's L1000 platform was collected and joined with the Mamoshina data, creating a sufficiently large, cleaned dataset. To achieve OBJ-2, feature selection was performed, using recursive feature elimination with single-label random forest classifiers to reduce the dimensionality by approximately one-third. To satisfy OBJ-3, the Mamoshina MLC chain model was replicated as closely as possible, with a modified hyperparameter set for the random forest models. OBJ-4 was achieved by creating a process-interruptable framework for aborting computationally expensive training experiments, from which, the hyperparameters were decoupled to obtain the best performers. These were used in the *Best Means Chain*. OBJ-5 was achieved by comparing the Jaccard scores for all individual chains, the *Best Means Chain*, and *Ensemble*, demonstrating the *Best Means Chain* to be ranked in the top-3. OBJ-6 was achieved through correlation analysis between computation time, and model performance.

Hyperparameter automation has been achieved using the *Best Means Chain* approach that automatically decouples the hyperparameters across the 100 individual chains to find the highest performing ones on average. The small search space goal has been achieved, since the subspace of 100 hyperparameter combinations is much smaller than the set of possible combinations arising from six random forest hyperparameters that have either 2, 3, 5, or 11 potential values. Training time has been kept feasible, using an interrupt-based framework that prevents non-converging experiments to be halted and skipped. Robustness has been demonstrated through consistently high performance on small and large datasets (full experiment runs versus short demo runs). It has also been demonstrated by how the *Best Mean Chain* ranked in the top-3 of the 100 individual and *Ensemble* chains. Having achieved all of these outcomes and objectives, the research question has been answered.

It was found that, for this cardiotoxicity application, it is possible to find a relatively high performing robust MLC model from a relatively small hyperparameter search space, using the *Best Means Chain* (CONTR-1). Implications of the work are that, while Auto-ML for MLC models is still an open issue, the outcomes can be used to tune robust MLC models in a straightforward manner, at least for similar applications (CONTR-2). One limitation of this study is that each stage in the MLC chains implements the same base model, base features, and hyperparameters, which makes the solution sub-optimal. The high ranking performance of the *Best Means Chain*, and its consistent performance between the full experiment runs and the short demo runs, demonstrates generalisability of the approach (CONTR-3).

While this project has successfully demonstrated a technique for creating robust MLC chains for cardiotoxicity prediction, several avenues can still be investigated. Potential exists to further optimise the MLC performance by using different base models at each stage of the chain. Also, using different feature subsets at each stage may also increase performance. The leave-drug-out cross-validation approach could be implemented to investigate further improvements to robustness, when combined with the proposed *Best Means Chain* approach. Investigating variations of label order in the chain would add further depth to the hyperparameter search space. Searching for relationships between

the number of individual chains and the *Best Means Chain* performance could lead to computational savings, where the process-interrupt timeout period could potentially be reduced. While this work is expected to be generalisable beyond the scope of this project, more work is needed to confirm the extent. To gain a deeper understanding, the proposed solution could be applied to the other PGx applications presented in Section 2. Finally, further insights should be sought for the medical interpretations of expanding the training set to include more cell lines.

References

Arbitrio, M., Scionti, F., Martino, M. T. D., Caracciolo, D., Pensabene, L., Tassone., P. and Tagliaferri, P. (2021). Pharmacogenomics biomarker discovery and validation for translation in clinical practice, *Clinical and Translational Science* **14**(1): 113–119.

Bergstra, J., Komer, B., Eliasmith, C., Yamins, D. and Cox, D. D. (2015). Hyperopt: A python library for model selection and hyperparameter optimization, *Computational Science and Discovery* **8**(1).

Bundy, J. L., Judson, R., Williams, A. J., Grulke, C., Shah, I. and Everett, L. J. (2022). Predicting molecular initiating events using chemical target annotations and gene expression, *BioData Mining* **15**(1): 1–27.

Chen, Y., Li, Y., Narayan, R., Subramanian, A. and Xie, X. (2016). Gene expression inference with deep learning, *Bioinformatics* **32**(12): 1832–1839.

Cui, C., Ding, X., Wang, D., Chen, L., Xiao, F., Xu, T., Zheng, M., Luo, X., Jiang, H. and Chen, K. (2021). Drug repurposing against breast cancer by integrating drug-exposure expression profiles and drug–drug links based on graph neural network, *Bioinformatics* **37**(18): 2930–2937.

Hall, L. H. and Kier, L. B. (1995). Electrotopological state indices for atom types: A novel combination of electronic, topological, and valence state information, *Journal of Chemical Information and Computer Sciences* **35**(6): 1039 – 1045.

Lin, E., Lin, C.-H. and Lane, H.-Y. (2021). Machine learning and deep learning for the pharmacogenomics of antidepressant treatments, *Clinical Psychopharmacology and Neuroscience* **19**(4): 577–588.

Mamoshina, P., Bueno-Orovio, A. and Rodriguez, B. (2020). Dual transcriptomic and molecular machine learning predicts all major clinical forms of drug cardiotoxicity, *Frontiers in Pharmacology* **11**.

Pandi, M.-T., Koromina, M., Tsafaridis, I., Patsilinakos, S., Christoforou, E., van der Spek, P. J. and Patrinos, G. P. (2021). A novel machine learning-based approach for the computational functional assessment of pharmacogenomic variants, *Human Genomics* **15**(1).

Poplin, R., Chang, P.-C., Alexander, D., Schwartz, S., Colthurst, T., Ku, A., Newburger, D., Dijamco, J., Nguyen, N., Afshar, P., Gross, S., Dorfman, L., McLean, C. and Depristo, M. (2018). A universal snp and small-indel variant caller using deep neural networks, *Nature Biotechnology* **36**(10): 983.

Probst, P., Wright, M. N. and Boulesteix, A.-L. (2019). Hyperparameters and tuning strategies for random forest, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **9**(3).

Read, J., Pfahringer, B., Holmes, G. and Frank, E. (2021). Classifier chains: A review and perspectives, *Journal of Artificial Intelligence Research* **70**: 683 – 718.

Shadab, S., Alam Khan, M. T., Neezi, N. A., Adilina, S. and Shatabda, S. (2020). Deepdbp: Deep neural networks for identification of dna-binding proteins, *Informatics in Medicine Unlocked* **19**: 100318.

Sharifi-Noghabi, H., Jahangiri-Tazehkand, S., Smirnov, P., Hon, C., Mammoliti, A., Nair, S. S. K., Mer, A. S., Ester, M. and Haibe-Kains, B. (2021). Drug sensitivity prediction from cell line-based pharmacogenomics data: guidelines for developing machine learning models, *Briefings in bioinformatics* **22**(6).

Signorelli, C. (2022). *Research project: Configuration manual*, Master's thesis, NCI, Dublin.

Subramanian, A., Narayan, R., Corsello, S., Peck, D., Natoli, T. et al. (2017). A next generation connectivity map: L1000 platform and the first 1,000,000 profiles, *Cell* **171**(6): 1437–1452.e17.

Tran, K. A., Kondrashova, O., Bradley, A., Williams, E. D., Pearson, J. V. and Waddell, N. (2021). Deep learning in cancer diagnosis, prognosis and treatment selection, *Genome Medicine* **13**(1).

Tsubaki, M., Tomii, K. and Sese, J. (2018). Compound–protein interaction prediction with end-to-end learning of neural networks for graphs and sequences, *Bioinformatics* **35**(2): 309–318.

Vaz, J. and Balaji, S. (2021). Convolutional neural networks (cnns): concepts and applications in pharmacogenomics, *Molecular Diversity* **25**(3): 1569–1584.

Wever, M., Tornede, A., Mohr, F. and Hüllermeier, E. (2020). Libre: Label-wise selection of base learners in binary relevance for multi-label classification, *in* M. R. Berthold, A. Feelders and G. Krempl (eds), *Advances in Intelligent Data Analysis XVIII*, Springer International Publishing, Cham, pp. 561–573.

Wever, M., Tornede, A., Mohr, F. and Hullermeier, E. (2021). Automl for multi-label classification: Overview and empirical evaluation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **43**(9): 3037 – 3054.