Rectified Classifier Chains for Prediction of Antibiotic Resistance From Multi-Labelled Data With Missing Labels

Mukunthan Tharmakulasingam, Brian Gardner, Roberto La Ragione, and Anil Fernando

Abstract—Predicting Antimicrobial Resistance (AMR) from genomic data has important implications for human and animal healthcare, and especially given its potential for more rapid diagnostics and informed treatment choices. With the recent advances in sequencing technologies, applying machine learning techniques for AMR prediction have indicated promising results. Despite this, there are shortcomings in the literature concerning methodologies suitable for multi-drug AMR prediction and especially where samples with missing labels exist. To address this shortcoming, we introduce a Rectified Classifier Chain (RCC) method for predicting multi-drug resistance. This RCC method was tested using annotated features of genomics sequences and compared with similar multi-label classification methodologies. We found that applying the eXtreme Gradient Boosting (XGBoost) base model to our RCC model outperformed the second-best model, XGBoost based binary relevance model, by 3.3% in Hamming accuracy and 7.8% in F1-score. Additionally, we note that in the literature machine learning models applied to AMR prediction typically are unsuitable for identifying biomarkers informative of their decisions; in this study, we show that biomarkers contributing to AMR prediction can also be identified using the proposed RCC method. We expect this can facilitate genome annotation and pave the path towards identifying new biomarkers indicative of AMR.

Index Terms—Multi-label classification, Classifier Chain, Multi-drug AMR, Missing labels, Semi-supervised model, Feature selection

1 Introduction

Antibiotic Resistance is an Antimicrobial Resistance (AMR) in bacteria and a growing public and veterinary health concern due to the increasing spread of resistant bacteria. Some intensive farming practices have led to the inappropriate use of antibiotics to prevent bacterial diseases, and this has intensified the issue. The World Health Organization (WHO) has listed AMR as one of the three most critical health issues of the 21st century [1]. Currently, 700000 people die every year worldwide due to AMR, and it is predicted to be five million death by 2050 unless urgent actions are taken [2]. Despite efforts to identify AMR and prevent its spread, significant issues still exist in terms of establishing its genetic factors.

Promising solutions to address these concerns include developing novel strategies to identify AMR presence in bacteria, for example via antibiotic susceptibility testing *in-vitro* [3]. Advantageously, this approach facilitates a personalised treatment plan, thus avoiding unnecessary antibiotic use that would otherwise exert selective pressure for resistance emergence. The standard method for identifying AMR consists of exposing bacterial isolates to different concentrations of antibiotics and measuring their growth under laboratory environments. However, this approach is time-consuming and expensive [4].

Alternative approaches include detecting the presence of AMR through analysing genomic sequences of the pathogens of interest. The development of Next-Generation Sequencing (NGS) technologies have enabled rapid and more cost-friendly profiling of whole genomes [5]. Coupling NGS data with machine learning for the accurate prediction of AMR from bacterial genome sequences shows strong promise and may encourage better use of antimicrobial agents [6].

Since one bacterium can be resistant to multiple antibiotics at the same time, there is a need to predict multi-drug resistance simultaneously associated with a single sequence. This multi-drug resistance for a single genomic sequence makes a genomic sequence a multi-label dataset. Even though increasing genomic data availability opens the path for machine learning-based prediction, many AMR phenotypes are missing in multi-labelled datasets due to the complex and time-consuming process involved in identifying

- *Mukunthan Tharmakulasingam is with the Centre for Vision, Speech and Signal Processing, University of Surrey, GU2 7XH Guildford, Surrey, U.K. E-mail: m.tharmakulasingam@surrey.ac.uk.*
- *Brian Gardner is with the School of Veterinary Medicine, University of Surrey, GU2 7XH Guildford, Surrey, U.K. E-mail: b.gardner@surrey.ac.uk.*
- *Roberto La Ragione is with the School of Veterinary Medicine and School of Biosciences and Medicine, University of Surrey, GU2 7XH Guildford, Surrey, U.K. E-mail: r.laragione@surrey.ac.uk.*
- *Anil Fernando is with the Department of Computer and Information Sciences, University of Strathclyde, G1 1XQ Glasgow, U.K. E-mail: anil.fernando@strath.ac.uk.*

Manuscript received 10 March 2021; revised 25 November 2021; accepted 27 January 2022. Date of publication 7 February 2022; date of current version 3 February 2023.
This work was supported by the University of Surrey and Zoetis through the vHive Initiative. Brian Gardner was supported by the European Union's Horizon 2020 Research and Innovation Programme under Grant 773830.
(Corresponding author: Mukunthan Tharmakulasingam.)
Digital Object Identifier no. 10.1109/TCBB.2022.3148577

different types of resistance. As samples were collected at different times and laboratory-based experiments were done only for the susceptibility of selected antibiotics, the susceptibilities of all antibiotics are not known. Therefore, the unknown labels are treated as missing labels in this paper. This causes difficulty in training the multi-label model as current multi-label identification methods rely on fully labelled data for input classification. Other issues with the current multi-label classification methods are results dependence on the label order and lack of interpretability of the results derived from classification models. In this paper, we propose a Rectified Classifier Chain (RCC) method to overcome the issues arising from missing labels, label order ambiguity and model interpretability.

This paper is structured as follows. Section 2 covers the background of this study, Section 3 describes the proposed RCC, Section 4 describes the methodology and the metrics used to measure the performance, Section 5 reports the test results, Section 6 discusses the results and Section 7 finishes with concluding remarks.

2 Background

2.1 AMR Prediction

Genes are segments of genome sequences that contribute to an organism's phenotypic traits. Therefore, identifying genes from genomes that are associated with AMR helps to predict the phenotype of pathogens. To this end, annotated genome sequences available from public databases[1, 2] are used to annotate genes in a new sequence [7]. Basic Local Alignment Search Tool (BLAST) [8] is an application used to compare biological sequences, identify similar sequences from the database and annotate. Specifically, BLAST calculates the statistical significance of these compared sequences to decide if they match one another [7]. Accordingly, BLAST is used to compare an examined genomic sequence against existing ones in the gene database, and AMR genes in this sequence are annotated based on identified matches [9]. Once genes are identified using the BLAST, machine learning plays a role in phenotypic prediction [10]; several studies have applied machine learning algorithms with the extracted gene data to predict AMR from annotated genes as discussed above [11], [12], [13], [14], [15].

A pan genome-based machine learning method was proposed to predict Antimicrobial Resistance (AMR) in *E. coli* strains [12]. Machine-learning algorithms were developed to predict the antimicrobial resistivity and susceptibility in specific strains, and the *E. coli* dataset was used to validate them. Four types of machine learning algorithms, namely Support Vector Machine (SVM), Naïve Bayes (NB), Random Forest (RF), and AdaBoost, were used to predict 39 different antibiotic resistance types from genomic data collected from PAThosystems Resource Integration Centre (PATRIC) database [6]. BLAST algorithm [8] was used to annotate the sequences for these algorithms. Aside from these machine learning methods, a genetic algorithm (GA) was applied to identify the best set of genes using Comprehensive Antibiotic Resistance Database (CARD) annotations to predict antimicrobial resistivity and susceptibility of the strains [12]. In this work, the Area Under the Receiver Operating Characteristic (AUROC) curve metric was used to validate the models. The models were built separately for each AMR type and features significantly contributing to classification decisions were not reported. With the identifications of new types of AMR, it is essential to build a single model to predict different types of AMR and to identify the genes contributing to those decisions.

A deep learning system to predict genes related to AMR has been proposed with two different implementations as applied to processing short-read sequences and long-read sequences, each of which was tested using data from the Comprehensive Antibiotic Resistance Database (CARD), Antibiotic Resistance Genes Database (ARDB), and UNIversal PROTein Resource (UNIPROT) database [16]. A high precision and recall of 97% and 91%, respectively, were achieved using this system. BLAST was used to annotate the sequences to identify AMR genes. This method identified different genes from genome reads yet failed to identify different AMR types as gene presence does not directly imply resistance for any specific type of antibiotic.

A custom-built AdaBoost machine learning classifier was proposed in another study for identifying different antimicrobial resistance genes from various species and predicting AMR phenotype using the PATRIC database [13]. This study also developed separate models for distinct AMR types.

Multi-drug resistance is becoming a critical issue in human and animal health systems [14]. It is therefore important to consider multi-label classification models which can predict whether genome sequences are susceptible or resistant to many types of antibiotics. To our best knowledge, there has been little effort in applying multi-label classification models to predict multi-AMR types using a single model.

2.2 Multi-Label Classification

Most traditional machine learning algorithms are developed for single-label classification problems; hence, datasets need to be transformed, or learning models adapted in order to be compatible with multi-label datasets. Multi-label classification models can be grouped into two categories: problem transformation, which transforms the multi-label dataset into several single label datasets, and algorithm adaptation method which modifies the conventional algorithms to deal with multi-labelled datasets [17], [18], [19], [20]. Binary Relevance (BR) [21], Label Powerset (LP), Classifier Chain (CC), and pair-wise approach are a few examples of a problem transformation approach, and decision trees, SVM kernel-based approaches, and lazy learning, including ML-kNN, are some examples of an algorithm-based adaptation approach [19]. Algorithm adaption approaches are often better suited to specific domains due to less flexibility and high computational complexity [20].

In the context of the AMR multi-label dataset, each label is treated as a binary classification since it is represented as one of two values: 'Resistance' or 'Susceptible.' Other values such as 'Intermediate' and 'Susceptible-dose dependent' are normally converted to 'Resistant' and 'Susceptible' respectively.

[1] ftp://ftp.patricbrc.org/
[2] https://ftp.ncbi.nih.gov/genomes/

The standard approach in the literature trains a separate binary classifier for each AMR label, by splitting the original multi-label problem into many single-label problems. However, these approaches have several drawbacks. First, it is not straightforward to feed the data into multiple trained models and predict each AMR type separately, as well as identify biomarkers contributing to them, as there are more than 30 types of AMR to classify. Second, these approaches do not consider the possible dependencies between the different AMR labels; these labels have dependencies between each other due to shared predictive genes and AMR generating mechanisms. Therefore, predicting all AMR labels using the same model should perform.

The Label Powerset method is another multi-label classification approach that considers each member of the power set of labels in the training set as a single label and trains them with traditional models [18]. However, the Label Powerset approach is computationally expensive for larger label datasets and results in diminished performance for imbalanced datasets. As AMR labels are correlated and depend on each other, there is a low chance of obtaining a balanced dataset after the multi-label to single label transformation using a Label power set.

Algorithm 1. The Algorithm for Training a Rectified Chain Classifier. It Returns a Vector of Trained Models to be Used for Prediction

```
Require:
  $X$→ L x d matrix of input instances, where L is the number of samples and d is the number of features for one instance.
  $Y$→ L x m matrix of outputs, where L is the number of samples and m is the number of labels for one instance.
  ModelList → 1xm vector of models
Ensure:
  for $i$ 0 → $m$-1 do
    NanId,NonNanId ← extractEmptydata($Y$, $i$) {extractEmptydata method returns Nan and NonNan indexes for $i^{th}$ labels of $Y$}
    Xtrain ← X[NonNanId] {Get NonNan indexes as train feature}
    Ytrain ← $Y_{,i}$ [NonNanId] {Get NonNan indexes of $i^{th}$ label from Y}
    $model_i$ →train(Xtrain, Ytrain) {Train Classifier for $i^{th}$ label}
    Xempty ← X[NanId] {Get Nan indexes as test feature}
    Yempty ← ($model_i$→predict(Xempty)) {Predict missing labels using trained model}
    $Y_{,i}$ [NanId] ← Yempty { Assign the model to model chain to use the same model in the prediction stage}
    X ←addColumn(X,$Y_{,i}$) {Add current labels as feature to next model}
  end for
  return ModelList
```

Binary relevance is an ensemble of single-label binary classifiers that are trained independently on the original dataset to predict sample membership for each class, and the results are combined at the end to give the multi-label output [19]. Even though it has linear complexity with the number of labels, this model does not consider label correlation. ML-kNN is a lazy learning approach that determines the label set of the test instance, based on prior and posterior probabilities for the frequency of each label within the k nearest neighbours in the training set [20]. This method is computationally expensive, and label correlations are not considered as well.

Classifier chains [22] method is an improved binary relevance method by linking each binary classifier as a chain to capture the dependencies between labels. Each linked binary relevance model makes a prediction in the order specified by the chain using all the available features plus the predictions of previous classifiers in the chain, where the number of classifiers is set equal to the number of labels. Classifier (C_j) relevance to the j^{th} label (y_j) will take the set of features, X, and labels from y_1 to $y_{(j-1)}$ as input to predict target y_j. Therefore, label prediction accuracy will depend on the label order, and the optimal label order is difficult to predict. Therefore, an Ensemble of Classifier Chains (ECC) was proposed combining the predictions of different label orders with different samples of the training data to train each member of the ensemble [22]. A few studies were published on optimising Ensemble of Classifier Chains (ECC) efficiency and accuracy [23]. It has been found that the ECC method needs a larger number of data samples as they split the data and train each split data in a different label order to get better accuracy, however, ECC based approaches are slow to train due to their ensemble-based training setup. This poses practical challenges to applying ECC for predicting multilabel datasets containing limited samples with insufficient computational resources.

Algorithm 2. The Algorithm for Predicting Labels for Given Input X Using Rectified Chain Classifier

```
Require:
  $X$→ L x d matrix of input instances, where L is the number of samples and d is the number of features for one instance.
  ModelList → 1xm vector of models
  $Y$→ L x m matrix of outputs, where L is the number of samples and m is the number of labels for one instance.
Ensure:
  for $i$ 0 → $m$-1 do
    $model_i$←ModelList[i]
    $Y_i$ ← ($model_i$ → predict(X)) {Predicting $i^{th}$ label with $i^{th}$ classifier}
    X ←addColumn(X,$Y_{,i}$) {Add current labels as feature to next classifier}
  end for
  return $Y$
```

Comparatively few studies have been published on applying multi-label classification methods to genomics data for predicting multiple types of AMR [15], [20], [21], [22], [27], [28]. DeepGo [28] and DeepGoplus [27] are two such popular methods, as applied to predicting multi-label protein classes from genomic sequences using deep learning methodologies; however, they do not predict AMR specifically nor handle missing labels as part of their operation. Although there are a few multi-label classification methods proposed in the literature in related areas, there is little effort in comparing the different multi-label methods for predicting multi-drug resistance. Normally, AMR labels have strong interdependencies due to common genes

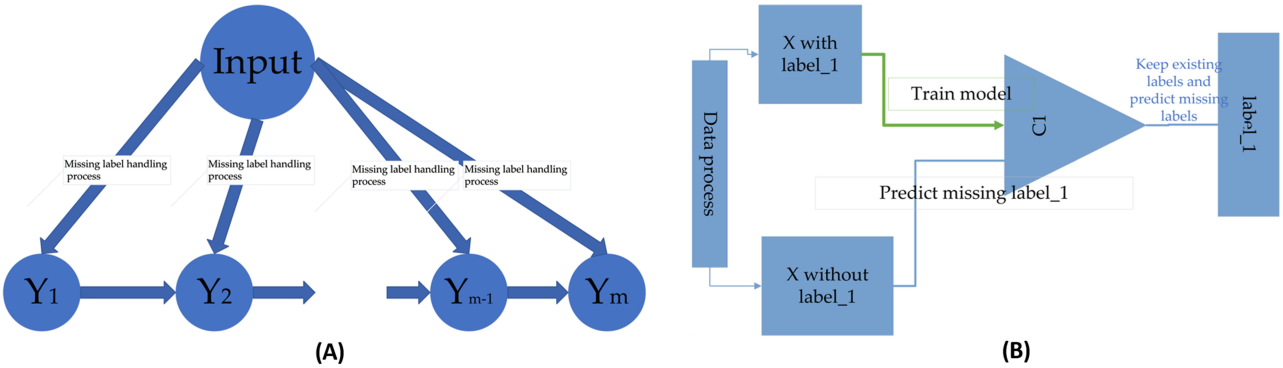
Input
Missing label handling process
Missing label handling process
Missing label handling process
Missing label handling process
Y_1
Y_2
Y_{m-1}
Y_m
(A)
X with label_1
Train model
Keep existing labels and predict missing labels
label_1
C1
Data process
Predict missing label_1
X without label_1
(B)

Fig. 1. (A) The overall structure of proposed Rectified Chain Classifier with m labels. (B) Matching label handling process algorithm overview in each classifier.

shared. As per our knowledge, this nature of the data is not reflected in any literature on predicting multi-AMR phenotypes. Even though there are several AMR prediction methodologies available to choose from for a genomic dataset, none of these utilizes the label interdependencies for multi-label data based on our knowledge, handles the missing labels scenarios, and explains the results derived from the model. Despite this, a few studies have been published on missing labels in multi-label classification methods in other research areas [29], [30], [31]. In these studies, different methodological approaches have been applied such as nested stacking and subset correction to overcome attribute noise that is caused due to erroneous prediction of labels in the previous classifiers [31]. Even though the use of predicted values as a substitute for the actual labels addresses the issue of missing values to some extent, missing labels at the first classifier in a chain should be imputed and that may result in a wrong model for the first classifier. The error in predicted values due to the wrong first model may cause to build wrong model for other classifiers due to the chaining effect. Therefore, these models are not suitable for the data with a higher amount of missing label.

Aside from this, machine learning algorithms are appealing tools in this context to identify AMR, since they facilitate identifying a distinct feature set that can be further interpreted by domain experts. Therefore, identifying biomarkers from genomic sequences, contributing to the decision, and applying dimensional reduction techniques for this data is crucial for achieving higher accuracy in predicting AMR from genome sequence data [7]. Existing multi-label classification models are deficient in returning the feature set and weights contributing to arriving at the classification decision, thereby making these models non-interpretable.

Hence, there is a need for a comprehensive overview of current multi-label methods, how to handle missing labels, identifying biomarkers contributing to the multi-AMR prediction model, and metrics that are used to measure performance on an imbalanced dataset. Therefore, we propose a Rectified Classifier Chain (RCC) method to predict multi-drug resistance with missing labels and to identify biomarkers contributing to decision on RAST [33], [34] based annotated *Escherichia coli* (*E. coli*) genomic data to improve classifier accuracy and interpretability.

3 Proposed Methodologies

As discussed in Section 2, Classifier Chains (CC) capture the dependencies between labels on multi-label prediction. The potential pitfall of the CC approach is that classifier accuracy heavily depends on the label order and accuracy of the label values. One way to handle this pitfall is dropping data with missing labels before applying the CC method and that option is not feasible for a dataset that has a high level of missing labels. This limitation prevents applying CC to multi-label genomic data for AMR prediction as these datasets tend to have a lot of missing labels.

To overcome these challenges, we propose a semi-supervised based RCC which has an inbuilt mechanism to predict missing labels and impute with the predicted value in the training process of other label predictions. One of the challenges of this approach is the effect of attribute noise which is in direct correspondence with the accuracy of the binary classifiers along the chain in RCC. Therefore, identifying optimal label order plays a key role in obtaining better performance. This study also proposes a few approaches to optimize the label in order to avoid the effect of attribute noise and improve the classifier performance along with the capability to identify biomarkers contributing to those predictions.

3.1 Multi-Label Classification

The Proposed RCC consists of binary classifiers equal to the number of labels in the dataset. The feature space for each binary model is extended with the predicted labels of all previous classifiers; thus, forming a chain similar to the conventional classifier chain as shown in Fig. 1A [22], [35]. Each classifier C_i in the chain is learning and predicting label Ci given the set of input features, X, and extended by all the predictions from prior classifiers in the chain: $\{y_1, \ldots, y_{(i-1)}\}$. The features of each classifier in the chain are extended with binary-valued labels (y_j = {0, 1}) corresponding to each prior classifier, {C_j} for j < i. Conventionally, classifier chains take this type of input and are trained assuming that all the training labels are available. By contrast, and as a novel aspect of RCCs, RCCs will predict the missing labels as shown in Fig. 1B, which in turn are used to inform part of the input for the next classifier in the chain. By this approach, multi-labels with missing labels can be used to train the model.

3.2 Training Phase

Each model in RCC is trained individually in order from C_1 to C_m, as shown in Fig. 1. The first Classifier (C_1) takes the features as its input, drops samples that miss the first label (*label_1*) and is trained with the remaining data which have the first label. Once the classifier is trained, it is then used to predict the missing label (*label_1*) and impute the missing labels with the predicted labels in order to train the second classifier. The second classifier takes features and the first label which is updated by the as its inputs. The second classifier drops the samples that miss the second label (label_2) and is trained with the remaining data. Once the second classifier is trained, it is then used to predict the missing second labels and impute the missing second labels. Those imputed labels will be used to train the third classifier as described above. These steps continue till training the final classifier as shown in Algorithm 1. Once all the classifiers are trained, they are then kept in the List structure as different models built for each label and they need to be used in the prediction step.

3.3 Prediction

The proposed RCC uses classifiers in the previously described List structure to predict labels as shown in Algorithm 2. Each classifier receives input features and the predicted values of all previous classifiers as inputs to predict the current labels.

3.4 Optimised Label Order

The order in which classifiers are chained together is important in predicting multi-labelled data with more accuracy as labels may be conditional upon one another, and there may be errors in the prediction of labels throughout the chains as predicted values are used for the missing labels. For example, if an erroneous label is imputed by a classifier near the beginning of the chain, it will act as noise to other classifiers along the chain and the accuracy of the prediction will be reduced [31]. To help mitigate this ordering of labels issue, using the Conditional Entropy (CE), Conditional Probability (CP) and Missing Ratio (MR) of labels are proposed as part of this paper to decide the order of labels for our proposed model. Conditional entropy ($H(Y|X)$) is calculated according to Equation (1), where the conditional entropy of a variable Y given another variable X provides how much uncertainty remains in Y after using the information that X gave it [36]. Conditional entropy value for a label is calculated by calculating the average of conditional entropy values for the particular label given each label as shown in Equation (2). This conditional entropy value for each label pair is calculated only if there are more than 50 samples for those label pairs to get a less biased estimate. The label with the lowest value will be assigned for prediction to the first classifier in the chain as it has the least uncertainty with respect to the other labels. An example for the calculation can be viewed in Supplementary File 1, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TCBB.2022.3148577.

$$\mathrm{H(Y|X)} = \sum_{x,y} p(x,y) \log \left(\frac{1}{p(y|x)} \right) \tag{1}$$

$$\boldsymbol{CE}(Label_i) = \frac{\sum_{j\ != \ i} CE\big(Label_i | Label_j\big)}{Number\ of\ pairs\ with\ more\ than\ 50\ samples} \tag{2}$$

As shown in Equation (3), the conditional probability for each label is calculated. Then, conditional probability value for a particular label is calculated by calculating the average of conditional probability values for the particular label given each label as shown in Equation (4). This calculation only considers the label pair with more than 50 samples similar to conditional entropy. The influence of the label on other labels is decided based on the calculated conditional probability value for each label. An example for the calculation can be viewed in Supplementary File 1, available online.

$$\begin{aligned} P(Y|X) = P(X=S) * P(Y=S|X=S) \\ +P(X=R) * \ P(Y=R|X=R) \end{aligned} \tag{3}$$

$$\boldsymbol{CP}(Label_i) = \frac{\sum_{j\ != \ i} CP\big(Label_i | Label_j\big)}{Number\ of\ pairs\ with\ more\ than\ 50\ samples} \tag{4}$$

Since a large number of labels are missing in the multilabel AMR dataset, using labels with a higher number of missing labels in the initial classifier will cause to build the less accurate model as missing labels are dropped in the training process of RCC and then, those missing values are predicted using the trained model. These imputed labels will act as noise to other models in the chain and cause reduced accuracy of the prediction [31]. Therefore, the missing label ratio for each label can be also used to determine a more optimal label order. These three label order optimization techniques are validated in our experiment.

3.5 Significant Features

It is challenging to identify features and their contributions informative of the classification decisions since many classifiers are applied in the chain. As part of this RCC, significant features are calculated in two approaches. In one approach, the top contributing features for the classification and their contributions are calculated for each model in the RCC. The absolute values of these contributions are added together for each model, and the overall top feature contributions are calculated. This approach helps to identify the most indicative features of multi-drug resistance prediction and reduces the effort needed to annotate key features for prediction purposes. In the other approach, SHapley Additive exPlanations (SHAP) [37] values are calculated and their mean values are used to identify significant features. SHAP is a unified way to interpret the predicted values using the Shapley value as the sum of the attribution value of each input feature. SHAP values can be used as the global interpretability since SHAP captures how much each predictor contributes, either positively or negatively, to the target variable [37], [38]. Each model in the RCC is integrated with the SHAP approach to identify the significant features. SHAP values calculated for each model is added to identify the significant features of the RCC.

3.6 Complexity Analysis

The computational complexity of the proposed RCC is very close to that of a CC, depending on the total number of labels, the individual complexity of the underlying learner, the ratio of missing labels and the complexity of missing label

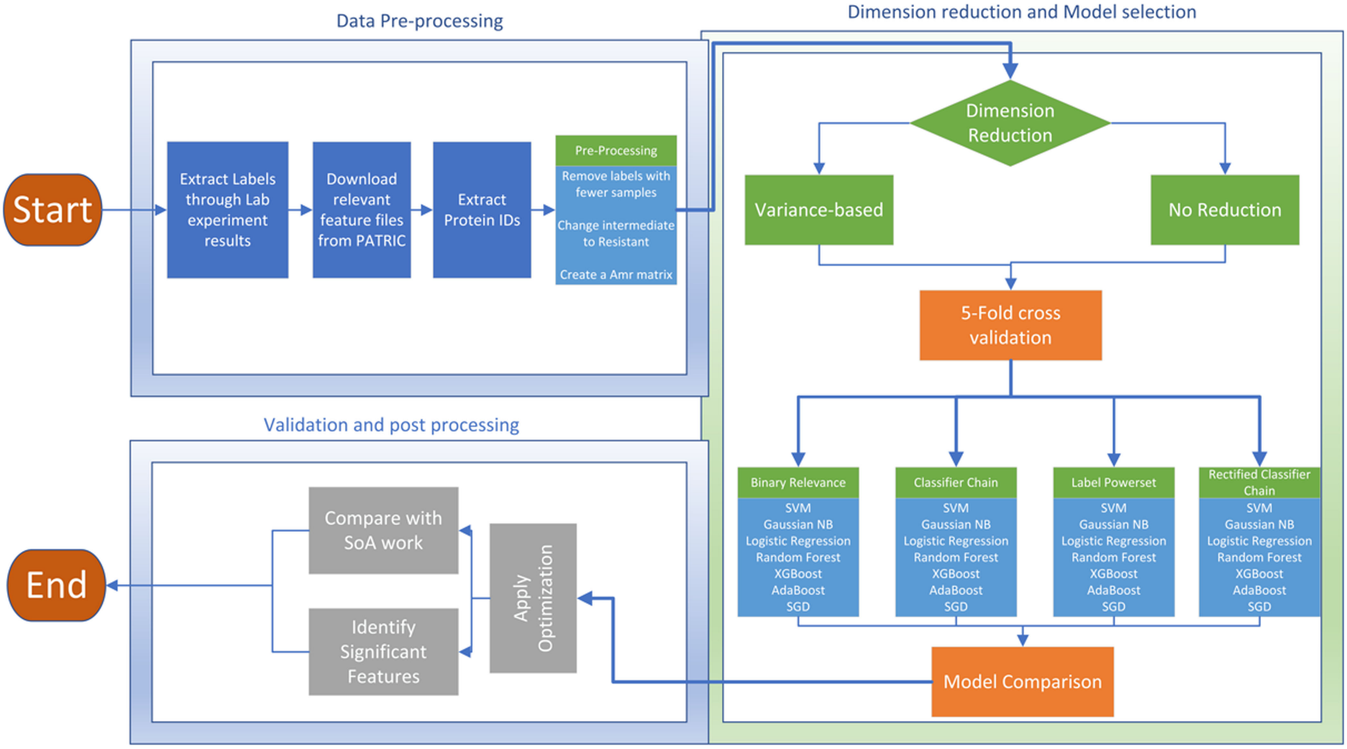

Fig. 2. Visualization of the data analysis study done for this research.

prediction. CC's complexity is $O(|L| \times f(|X| + |L|, |D|))$, where $f(|X|+|L|, |D|)$ is the complexity of the underlying learner having $|L|$ labels as additional attributes with $|X|$ features for $|D|$ number of training samples [22]. Using the same notation, RCC's complexity can be represented as $O(|L| \times (f(|X| + |L|, |D_1|)+ g(|X| + |L|, |D_2|))$, where D_1 is an average number of samples with labels present and D_2 is an average number of samples with label missing. As the complexity of the underlying learner is usually higher than the complexity of prediction using the underlying learner, the Complexity of the RCC is less than the complexity of CC as shown in Equations (5), (6), (7), and (8).

$$g(|X|+ |L|, |D_2|) \langle \; f(|X|+ |L|, |D_2|) \tag{5}$$

Therefore,

$$\begin{aligned} &O\ (|L|\ \times\ (f(|X|\ +\ |L|,\ |D_1|) + g(|X|\ +\ |L|,\ |D_2|)) \\ &\quad < O\ (|L|\ \times\ (f(|X| + |L|,\ |D_1|) + \ f(|X| + |L|,\ |D_2|)) \end{aligned} \tag{6}$$

For instance, assuming a linear base learner and $|D_1| + |D_2| = |D$

$$\begin{aligned} &O\ (|L|\ \times\ (f(|X|\ +\ |L|,\ |D_1|) + g(|X|\ +\ |L|,\ |D_2|)) \\ &< O\ (|L|\ \times\ (f(|X| + |L|,\ |D|)) \end{aligned} \tag{7}$$

Therefore,

$$RCC's\ complexity\ \ <\ \ CC's\ Complexity \tag{8}$$

As RCC uses a training subset with a correct label in each label model training phase and use those correct label and predicted labels for the missing labels, the complexity of RCC will be slightly lower than that of CC. However, with respect to non-linear models which may have a similar prediction time compared to training, RCC complexity can be a little more. Even though $|L|/2$ features are added to each instance on average in addition to the $|X| features$, RCC has a negligible impact on complexity as $|L|$ is invariably limited in practice compared to the features.

4 Experimentations With the Proposed Methodology

This section describes our experiment setup and results for *Escherichia coli* (*E. coli)* and *Salmonella* annotated genomic datasets that are publicly available on PATRIC. The proposed algorithm was implemented in Python using the Scikit-learn [39] library; our source code, the genome IDs we used for these experiments, and the pre-processed datasets are made available on GitHub[3].

As shown in Fig. 2, we applied a multi-label classification model for the pre-processed dataset using our proposed method with different base classifiers. The results we obtained from applying different base classifiers on the benchmark PATRIC dataset were compared with the following, traditional, multi-label learning models: Binary Relevance (BR), Label Powerset (LP), and Classifier Chains (CC). Then, the best model based on the above result was compared against similar works in multi-label classification and multilabel AMR prediction from conserved genes in the literature [15], [30], [33]. Following these steps, our model with the best performing base classifier was further analysed according to the classifier order optimisation techniques as discussed in Optimised label order (Section 3.4). The key biomarkers

[3] https://github.com/mukunthan/Rectified-Classifier-Chain

contributing to the decision were also reported using the proposed feature selection method (Section 3.5) for this model.

4.1 Dataset

The PATRIC database is one of the most comprehensive antibiotic resistance databases that collect genes, proteins, and genomic information relating to the resistance or susceptibility of pathogens to various antibiotic drugs [40]. All genomes in PATRIC were annotated using RAST [34], the Rapid Annotations using Subsystems Technology. As AMR genes may not be suitable for use when genomes are incomplete [15], Protein genus-specific families are identified using the RAST annotations for each genome sequence which were used in our experiments.

With respect to data preparation, all the *E. coli* sample genome ids and their associated laboratory experiment results were extracted, and their relevant 2775 genomic feature files marked against 32 AMR types as mentioned in Supplementary Table 1, available online, were downloaded from the PATRIC FTP site[4]. Relevant genome ids used for this experiment can be found in the GitHub[5] data folder.

Since our experimentation is for binary classification, strains labelled with intermediate levels of resistance and Susceptible-dose dependent labels were converted as resistance and susceptible labels respectively. AMR types that had less than 200 laboratory experiment results were dropped from our experiment as there were no sufficient data points to build a model for those types. Feature files downloaded from PATRIC were pre-processed to obtain protein genus-specific families and 16345 Protein genus-specific families (PLfams) were extracted as input features. After these pre-processing steps, these features were used to set up a binary-valued matrix that indicated the presence/absence of Protein genus-specific families for each genome sequence. Label distribution and missing label percentage label can be viewed in Supplementary Table 1, available online. As indicated in the supplementary table, available online, some labels had nearly more than 50% of missing labels (NaN) while some had few missing labels.

4.2 Dimension Reduction and Model Selection

It is important to select the most informative features contributing to a classifier's decision to reduce the input dimensionality and avoid overfitting issues. Filters, wrappers, and embedded methods are three types of feature selection processes [41]; filter methods measure the intrinsic properties of features based on univariate statistics metrics such as variance, mutual information, Chi-square(χ^2), information gain, correlation, etc. Wrapper methods, such as sequential feature selection and heuristic search algorithms, select a subset of features by applying an evaluation function that is optimised using a machine learning technique: forward selection, backward elimination and recursive feature elimination are a few examples of sequential feature selection methods [42], [43], [44]. Wrapper methods select the features by measuring their contribution to classifier performance in an iterative manner; therefore, wrapper methods are more computationally complex as compared to filter methods since they require repeated steps and cross-validation. The third type of feature selection, embedded methods, works similarly to wrapper methods, however, they utilize an intrinsic model for building metrics during the learning process to reduce its computational complexity. In our case, considering the complexity and size of the studied datasets, we selected a filter-based approach for feature selection.

The variance-based statistical method measures variability in each feature while the mutual information-based approach measures mutual dependence between label and feature [45]. As our data have a high number of missing labels, a variance-based method was applied for the extracted PLfams matrix to reduce the features from the input PLfams matrix. we used a variance of less than 0.01 as a threshold, which resulted in selecting 8763 as the most significant features from 16345 PLfams extracted in data preprocessing. Concerning the model setup, SVM, logistic regression, Gaussian Naive Bayes, XGBoost, AdaBoost and Stochastic gradient descent (SGD) were each, separately, used to define the set of base classifiers constituting our proposed RCC model which was tasked with predicting the 32 AMR phenotypes from the selected features. A five-fold cross-validation step with a total of 2775 genome data samples was performed on the data to evaluate each of these models.

4.3 Hyper-Parameter Configuration

For each of the tested models, parameters were optimised from preliminary experiments and according to those reported in the literature [15], [32]. The linear, kernel-based SVM was implemented using the Scikit-learn library with the regularization parameter set to 0.01. XGBoost (XGB) was implemented with the GB-Tree [46] type booster, with 0.0625 as the learning rate, an ensemble of trees as the model, and 16 as the maximum depth. BR, LP, and CC do not depend on any hyper-parameters. Our RCC model uses conditional entropy, conditional probability, and missing ratio as three different approaches to find optimal label order and all these three approaches were used for the experiment separately to find the best approach.

4.4 Evaluation Metrics

Label-based micro-averaging, macro averaging, Hamming-based accuracy, subset accuracy, and label-based micro-averaged F-measure are the most used evaluation metrics in multi-label classification. The Hamming distance- compares the actual labels with the predicted labels and count how many labels are incorrectly predicted. Therefore, the Hamming distance-based accuracy defined in Equation (9) evaluates how many labels are correctly classified in the total set of labels; however, this measure is not suitable for predicting the performance for an imbalanced dataset which have higher samples for particular label class and fewer samples for another label class. Since AMR data is imbalanced, it is vital to measure the recall and precision metrics to determine the performance of a predictive algorithm. Micro averaged F1-score aggregates the

[4] ftp://ftp.patricbrc.org/

[5] https://github.com/mukunthan/Rectified-Classifier-Chain

TABLE 1
Hamming Accuracy for Different Multi-Label Methods With Different Base Models

Method	BR	CC	LP	RCC(MR)	RCC(CE)	RCC(CP)
SVM	84.10 ± 0.32	84.09 ± 0.29	74.28 ± 0.87	86.92 ± 0.55	86.86 ± 0.62	86.68 ± 0.50
Gaussian NB	70.26 ± 0.56	69.53 ± 0.50	67.19 ± 1.85	76.23 ± 1.08	76.24 ± 1.03	76.21 ± 1.03
Logistic Regression	84.04 ± 0.73	84.19 ± 0.66	81.98 ± 0.95	86.91 ± 0.49	86.86 ± 0.59	86.86 ± 0.81
Random Forest	79.75 ± 0.70	79.96 ± 0.60	77.91 ± 0.92	86.22 ± 0.75	86.08 ± 0.58	86.45 ± 0.69
XGB	87.38 ± 0.67	87.41 ± 0.60	84.38 ± 0.56	90.70 ± 0.70	90.39 ± 0.71	90.54 ± 0.65
AdaBoost	85.26 ± 0.55	85.11 ± 0.72	68.22 ± 1.65	89.30 ± 0.79	88.76 ± 0.81	89.23 ± 0.65
SGD	82.12 ± 0.52	82.56 ± 0.29	78.14 ± 1.14	85.87 ± 0.40	85.89 ± 0.53	85.80 ± 0.29

BR-Binary Relevance, CC- Classifier Chain, LP- Label Powerset, RCC (MR) – Rectified Classifier chain with Missing label, RCC (CE) – Rectified Classifier Chain with Conditional Entropy, RCC (CP) – Rectified Classifier Chain with Conditional probability. Accuracy values are reported by mean of the repeated k-fold experiment with standard deviation as the error. (I.e., mean ± standard deviation).

contributions of all classes to compute the average F1-score based on recall and precision. F1-score can be used as a score that can be used as an average of both precision and recall scores [47].

We used micro-averaged Hamming distance-based accuracy metrics and micro averaged F1-score for our experiments. The Hamming accuracy (HA) is calculated as shown in Equation (9), where N refers to the total number of samples and M the total number of available labels as it avoids comparing the missing labels with the predicted values in measuring the performance.

$$HA = \frac{1}{N}\sum_{i \in sample}\frac{1}{M}\sum_{j \in label}\left(X_{i,j}^{True} == X_{i,j}^{predict}\right) \tag{9}$$

The True Positive (TP) is the number of correctly predicted positive classes as defined in Equation (10) and, the True Negative (TN) is the number of correctly predicted negative classes as defined in Equation (11). The False Positive (FP) is the number of incorrectly predicted positive classes as defined in Equation (12) and the False Negative (FN) is the number of incorrectly predicted negative classes as defined in Equation (13). Precision is the proportion of positive class predictions that are correct as defined in Equation (14) and recall is the proportion of actual positive classes that are predicted correctly as defined in Equation (15). The F1 score conveys the balance between the precision and the recall, and the average F1-score is calculated as shown in Equation (16). All these calculations were done by considering the labels that were present only; the missing labels were avoided.

$$TP_i = \sum_{j \in labels}\left(X_{i,j}^{True} == 1 \;\&\&\; X_{i,j}^{predict} == 1\right) \tag{10}$$

$$TN_i = \sum_{j \in labels}\left(X_{i,j}^{True} == 0 \;\&\&\; X_{i,j}^{predict} == 0\right) \tag{11}$$

$$FP_i = \sum_{j \in labels}\left(X_{i,j}^{True} == 0 \;\&\&\; X_{i,j}^{predict} == 1\right) \tag{12}$$

$$FN_i = \sum_{j \in labels}\left(X_{i,j}^{True} == 1 \;\&\&\; X_{i,j}^{predict} == 0\right) \tag{13}$$

$$Precision_i = \frac{TP_i}{TP_i + FP_i} \tag{14}$$

$$Recall_i = \frac{TP_i}{TP_i + FN_i} \tag{15}$$

$$F1\ score = \sum_{i \in samples}\left(\frac{2 * Precision_i * Recall_i}{Precision_i + Recall_i}\right) \tag{16}$$

5 Results

This section presents the results of the evaluation experiments that were conducted. The Hamming loss-based metric and F1-score accuracies for the different models were analysed, and the results are summarised in Tables 1 and 2. The XGBoost (XGB) classifier performed best for most of the multi-label classification models in terms of both the Hamming loss-based evaluation metric and F1-score. The Classifier chain (CC) performed better compared to the Binary Relevance (BR), and Label Power (LP) set methods. Our proposed RCC method outperformed all the other methods used in these experiments. Moreover, the RCC with XGB as

TABLE 2
F1 Score Accuracy for Different Multi-Label Methods With Different Base Models

Method	BR	CC	LP	RCC (MR)	RCC (CE)	RCC (CP)
SVM	57.83 ± 1.66	57.88 ± 1.58	39.31 ± 1.32	63.89 ± 0.89	63.92 ± 1.33	63.86 ± 1.25
Gaussian NB	45.41 ± 1.08	43.29 ± 1.25	23.09 ± 1.99	50.61 ± 0.79	50.52 ± 0.77	50.54 ± 0.77
Logistic Regression	56.69 ± 1.80	56.88 ± 1.73	53.09 ± 0.46	64.29 ± 0.14	64.28 ± 1.30	64.25 ± 1.51
Random Forest	48.28 ± 1.51	48.48 ± 1.40	40.97 ± 0.94	63.29 ± 1.61	63.16 ± 1.30	63.60 ± 0.79
XGB	62.00 ± 1.56	61.93 ± 1.55	55.85 ± 1.31	69.76 ± 0.69	69.28 ± 0.79	69.77 ± 0.74
AdaBoost	58.40 ± 1.28	58.20 ± 1.64	26.17 ± 1.66	67.71 ± 0.83	66.92 ± 0.81	67.69 ± 0.75
SGD	54.27 ± 1.44	54.95 ± 1.18	46.29 ± 1.40	63.13 ± 1.31	63.05 ± 1.31	62.91 ± 1.15

BR-Binary Relevance, CC- Classifier Chain, LP- Label Powerset, RCC (MR) – Rectified Classifier chain with Missing label, RCC (CE) – Rectified Classifier Chain with Conditional Entropy, RCC (CP) – Rectified Classifier Chain with Conditional probability. Accuracy values are reported by mean of the repeated k-fold experiment with standard deviation as the error. (I.e., mean ± standard deviation).

TABLE 3
Time Taken in Seconds With the Studied AMR Dataset for Different Multi-Label Methods With Different Base Models

Method	BR	CC	LP	RCC (MR)	RCC (CE)	RCC (CP)
SVM	714	690	462	239	240	240
Gaussian NB	13	18	18	27	28	29
Logistic Regression	1385	1342	14102	470	470	475
Random Forest	48	29	3	25	26	27
XGB	5227	4533	13744	2709	2801	2767
AdaBoost	1162	573	40	303	304	305
SGD	277	294	289	162	161	164

BR-Binary Relevance, CC- Classifier Chain, LP- Label Powerset, RCC (MR) – Rectified Classifier chain with Missing label, RCC (CE) – Rectified Classifier Chain with Conditional Entropy, RCC (CP) – Rectified Classifier Chain with Conditional probability. This time is measured as mean time of training and testing time in seconds in each fold in 5-fold validation steps.

a base classifier outperformed the second-best classifier chain with XGB as a base model by 3.3% in Hamming accuracy and 7.8% in F1-score based accuracy. The conditional entropy, conditional probability, and missing ratio were used to find the best label order for the RCC, and the missing ratio approach slightly outperformed other approaches for most of the base classifiers in terms of the Hamming accuracy and F1-score evaluation metrics.

As the algorithm complexity is also a key factor in model selection, the times taken for training and testing were recorded for each of the models implemented in our study: these results are summarized for a Windows 10 machine with an Intel Core^{TM} i9-9900 CPU @ 3.10GHz processor in Table 3. These results indicate that Random Forest is the fastest base classifier. The binary Relevance method and the Classifier Chain (CC) took a similar time for the training and testing. All three variants of the proposed RCC had a better execution time compared to other approaches for most of the base classifiers, in addition to their better performance measure in terms of Hamming accuracy and F1-score as shown in Tables 1 and 2. The RCC outperformed the CC model, which gave a second-best result in terms of Hamming accuracy and F1-score, by reducing the execution time to half of that of the CC approach.

Even though XGBoost based approaches were given better performance in terms of Hamming accuracy and F1-score, they took longer to train, and test compared to other base classifier models. This resulted in more time taken to train XGBoost, although it returned the best performance in terms of accuracy and F1 score. Overall, our proposed RCC model performed better than other models with SVM, Logistic Regression, XGBoost and SGD classifiers, while it took slightly more time to run than Random Forest, Gaussian NB and AdaBoost classifiers.

As XGBoost based RCC with missing ratio-based label order selection gave the best result, that model is used to compare with the models in the results. As shown in Table 4, the results show that our proposed method outperformed the XGBoost based AMR prediction from conserved genes [15], AdaBoost model [33] and the nested stacking with subset correction-based improvement for CC model [30] for the E. coli dataset. These models were tested with the *Salmonella* dataset as well, and the XGB based binary relevance method outperformed our method as that dataset had many labels with fewer resistant labels as shown in Supplementary Table 2, available online.

6 Discussions

As reported in the results section, the RCC model outperformed the second-best model CC in Hamming accuracy, F1 score and execution time. RCC only uses fewer data samples compared to CC as it uses data that have labels in the training phase of each classifier and uses the trained model to predict missing values to impute the label in the next classifier as indicated in the proposed methodologies section (Section 3). This result indicates the efficiency of the proposed method in terms of algorithmic complexity as well. Missing Ratio (MR), Conditional Entropy (CE), and Conditional Probability (CP) based approaches on selecting optimal label order had nearly the same time taken to train and retrieve the results. The experiments with the label order

TABLE 4
Comparison of Hamming Accuracy and F1 Score Accuracy With State of Art Different Multi-Label Method and AMR Prediction Method in Literature Which Uses Conserved Genes

Method	Our Proposed XGB based RCC (MR)		Nested Stacking and Subset correction for CC with XGB [30]		XGBoost based AMR prediction from conserved Genes (BR) [15]*		AdaBoost based AMR prediction [33]*	
	Hamming Accuracy	F1 Score	Hamming Accuracy	F1 Score	Hamming Accuracy	F1 Score	Hamming Accuracy	F1 Score
E. coli dataset	90.70±0.70	69.76±0.69	86.36±0.47	62.86±1.41	86.23±0.35	62.27±1.31	84.37±0.81	59.35±0.81
Salmonella dataset	85.33±0.17	57.78±0.85	85.52±0.49	56.38±0.65	85.82±0.55	57.25±1.00	84.83±0.57	54.39±0.54

BR-Binary Relevance, CC- Classifier Chain, RCC (MR) – Rectified Classifier chain with Missing label. These performances are measured as mean values in each fold in 5-fold validation steps.
Changes were done to accommodate given dataset.

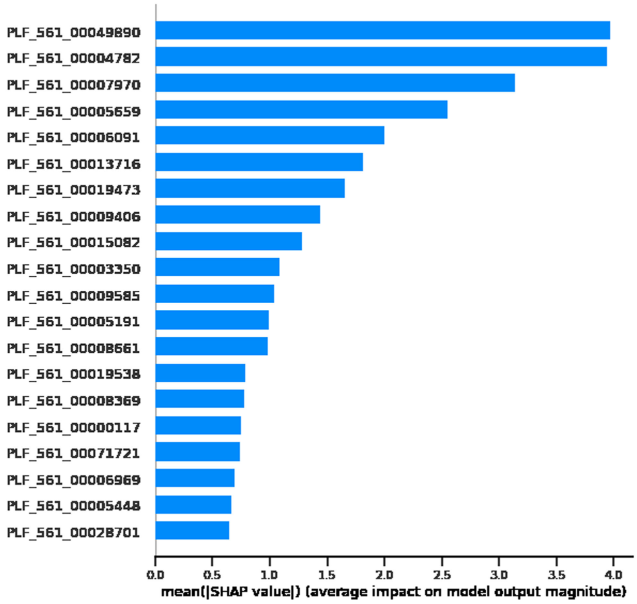

Fig. 3. Significant Features contributing to XGBoost based Rectified Classifier chain multi-label classification decision.

showed that the missing ratio approach is slightly better compared to others as there is a higher number of missing labels in predicting labels on this *E. coli* AMR dataset.

When comparing the base classifier performance, XGBoost gave the best results. XGBoost took more time to train the model and predict the test data set while Random Forest took less time. The XGBoost and Random Forest classifiers are built based on trees. However, XGBoost is boosting based approach which builds trees sequentially using information from previous trees. As the Random Forest classifier builds trees in parallel while XGBoost builds it sequentially, XGBoost took more time compared to Random Forest.

In the literature on predicting AMR from the conserved gene dataset, methodologies have been proposed for predicting labels using XGBoost but predicting labels separately similar to the Binary relevance approach [15]. As our work is also focused on conserved genes instead of well-known AMR genes, we used their proposed method [15] with our data and compared it with our proposed method. Another work similar to our work is using AdaBoost with the k-mers to predict AMR [33]. Even though that work was done based on k-mers, they considered a multi-label scenario. Therefore, we used the same approach with the dataset we used and compared these previous results with the results obtained through our methodology. As reported in Table 4, our RCC model outperformed those models for the *E. coli* dataset.

Nested stacking with subset correction [30] is one of the improvements suggested in the literature to improve the classifier chain drawbacks. This model stacks the predicted labels with true labels and do the subset label correction to overcome issues due to the erroneous prediction of labels in the previous classifiers [30]. Even though substituting predicted values for the labels overcomes the issue of missing values to some extent, the occurrence of missing labels at the first classifier in the chain tends to result in a less accurate model as the missing values in the first label has to be imputed with a default value. That erroneous label due to the default imputation may cause learn a less accurate first classifier. That less accurate first classifier and its predicted values will result in a less accurate model. The results in the Table 4 confirms above as our RCC model performed better than the nested stacking with a label subset correction approach to the CC model.

Even though the RCC model performed better for the *E. coli* dataset, the performance of RCC is not the best with the range of the value with the *Salmonella* dataset, this is not significant. As shown in Supplementary Table 2, available online, the *Salmonella* dataset has AMR labels where resistance values are very low compared to susceptible values. This imbalanced data will train classifiers with lower accuracy for those labels. As RCC uses the trained classifier to predict missing labels, classifiers with lower accuracy may predict wrong values for the imputation, and that will lower the overall accuracy. F1-score is the best parameter to measure when there is a data imbalance. Our proposed algorithm slightly outperforms other approaches when we compare the F1-score. Therefore, our experiments confirmed our RCC model with XGBoost as the base classifier performed better in terms of classification accuracy and F1 score.

Our RCC model with XGBoost base classifier was selected for further experiments to ascertain the significant features contributing to classification by measuring SHAP values for each feature as shown in Fig. 3. The Plfams ID identified as significant features in decision making were checked against known genes, and the identified genes are reported in Supplementary Table 3, available online.

Although it identified the well characterised *Tet(B)*, *TetR*, *OmpD* gene as a significant gene, it also identified some unfamiliar genome subsets reported in supplementary Table-3, available online, which might have biological relevance in predicting AMR. Here, Protein genus-specific families (Plfams) IDs were used as referred to in the PATRIC database, and details of that Plfams ID can be viewed in Supplementary Table 3, available online.

7 Conclusion

An extensive experimental evaluation of multi-label classification methods, including techniques for handling missing labels, in predicting multidrug AMR was undertaken in this study. The topic of multidrug AMR has recently received significant research interest. However, a more comprehensive experimental comparison of different multi-label methods and missing label handling is still lacking in the literature. Here, the RCC model is proposed to handle multilabel classification for genus-specific protein features data with missing labels and evaluate them with the most appropriate methods.

In this study, we demonstrated that the proposed RCC model can provide high Hamming-loss-based accuracy and F1 score with low algorithmic complexity, compared with other models. RCC has a novel mechanism to train the base classifiers with the labels present and predict the missing label to impute the missing labels and use the imputed values in the training stage of the next classifier. Other than that, RCC has a novel approach to decide the best label order to capture the label dependencies and to reduce the missing label effect. Furthermore, our proposed model can identify features contributing to its decisions. To the best of our knowledge, this study is the first to apply multi-label classification methods to handling missing labels and reporting the significant features for protein annotated datasets. The identified features will help reduce the complexity of annotation to identify AMR and provide new knowledge on biomarkers identifying multi-drug AMR. As there are thousands of features in the genomic feature dataset, it is important to identify important features to avoid overfitting and to improve the AMR prediction results as well. The comparative analysis conducted on the *E. coli* dataset illustrated that the proposed Rectified Classifier Chains method is a promising approach for multi-drug resistance prediction. Newly identified features offer pathologists an opportunity to analyse the contributions of those genome subsets in AMR.

Our study has a few limitations. As our model used predicted values to fill missing values, our method does not work well with an imbalanced dataset which has an imbalance in the number of samples for each class. Future work is required to address the problem of class imbalanced data in-depth to improve performance. These experiments were conducted with binary classification models; however, the RCC approach we used can be applied to multi-label classifications as well by selecting base classifiers that can support multi-label classification. Our study focuses on predicting AMR from simply annotated Protein genus-specific families (PLfams); however, these annotations require high computational power and laboratory-based experiments to obtain reference genomes. Therefore, identifying the genome using the k-mer approach [33] should help mitigate these limitations in the future.

References

[1] World Health Organization, Ed., *Antimicrobial Resistance: Global Report On Surveillance*. Geneva, Switzerland: World Health Organization, 2014.

[2] D. Jasovský, J. Littmann, A. Zorzet, and O. Cars, "Antimicrobial resistance—A threat to the world's sustainable development," *Upsala J. Med. Sci.*, vol. 121, no. 3, pp. 159–164, Aug. 2016, doi: 10.1080/03009734.2016.1195900.

[3] C.-A. D. Burnham, J. Leeds, P. Nordmann, J. O'Grady, and J. Patel, "Diagnosing antimicrobial resistance," *Nat. Rev. Microbiol.*, vol. 15, no. 11, pp. 697–703, Nov. 2017, doi: 10.1038/nrmicro.2017.103.

[4] P.-J. Van Camp, D. B. Haslam, and A. Porollo, "Bioinformatics approaches to the understanding of molecular mechanisms in antimicrobial resistance," *Int. J. Mol. Sci.*, vol. 21, no. 4, Feb. 2020, Art. no. 1363, doi: 10.3390/ijms21041363.

[5] W. Gu, S. Miller, and C. Y. Chiu, "Clinical metagenomic next-generation sequencing for pathogen detection," *Annu. Rev. Pathol.*, vol. 14, pp. 319–338, 2019, doi: 10.1146/annurev-pathmechdis-012418-012751.

[6] A. Drouin, F. Raymond, G. L. St-Pierre, M. Marchand, J. Corbeil, and F. Laviolette, "Large scale modeling of antimicrobial resistance with interpretable classifiers," Dec. 2016. Accessed: Apr. 13, 2019. [Online]. Available: http://arxiv.org/abs/1612.01030

[7] D. Wheeler and M. Bhagwat, *BLAST QuickStart*. Totowa, NJ, USA: Humana Press, 2007. Accessed: Aug. Nov., 2020. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK1734/

[8] S. F. Altschul *et al.*, "Gapped BLAST and PSI-BLAST: A new generation of protein database search programs," *Nucleic Acids Res.*, vol. 25, no. 17, pp. 3389–3402, Sep. 1997.

[9] M. Krauthammer, A. Rzhetsky, P. Morozov, and C. Friedman, "Using BLAST for identifying gene and protein names in journal articles," *Gene*, vol. 259, no. 1, pp. 245–252, Dec. 2000, doi: 10.1016/S0378-1119(00)00431-5.

[10] E. Avershina *et al.*, "AMR-Diag: Neural network based genotype-to-phenotype prediction of resistance towards β-lactams in escherichia coli and klebsiella pneumoniae," *Comput. Struct. Biotechnol. J.*, vol. 19, pp. 1896–1906, Jan. 2021, doi: 10.1016/j.csbj.2021.03.027.

[11] E. S. Kavvas *et al.*, "Machine learning and structural analysis of mycobacterium tuberculosis pan-genome identifies genetic signatures of antibiotic resistance," *Nature Commun.*, vol. 9, no. 1, Oct. 2018, Art. no. 4306, doi: 10.1038/s41467-018-06634-y.

[12] H.-L. Her and Y.-W. Wu, "A pan-genome-based machine learning approach for predicting antimicrobial resistance activities of the escherichia coli strains," *Bioinformatics*, vol. 34, no. 13, pp. i89–i95, Jul. 2018, doi: 10.1093/bioinformatics/bty276.

[13] N. Macesic, O. J. Bear Don't Walk, I. Pe'er, N. P. Tatonetti, A. Y. Peleg, and A.-C. Uhlemann, "Predicting phenotypic polymyxin resistance in klebsiella pneumoniae through machine learning analysis of genomic data," *mSystems*, vol. 5, no. 3, 2019, Art. no. e00656, doi: 10.1128/mSystems.00656-19.

[14] J. Kim *et al.*, "VAMPr: VAriant mapping and prediction of antibiotic resistance via explainable features and machine learning," *PLoS Comput. Biol.*, vol. 16, no. 1, Jan. 2020, Art. no. e1007511, doi: 10.1371/journal.pcbi.1007511.

[15] M. Nguyen, R. Olson, M. Shukla, M. VanOeffelen, and J. J. Davis, "Predicting antimicrobial resistance using conserved genes," *PLoS Comput. Biol.*, vol. 16, no. 10, Oct. 2020, Art. no. e1008319, doi: 10.1371/journal.pcbi.1008319.

[16] G. Arango-Argoty, E. Garner, A. Pruden, L. S. Heath, P. Vikesland, and L. Zhang, "DeepARG: A deep learning approach for predicting antibiotic resistance genes from metagenomic data," *Microbiome*, vol. 6, no. 1, Dec. 2018, Art. no. 23, doi: 10.1186/s40168-018-0401-z.

[17] P. Boerlin *et al.*, "Antimicrobial resistance and virulence genes of escherichia coli isolates from swine in ontario," *Appl. Environ. Microbiol.*, vol. 71, no. 11, pp. 6753–6761, 2005, doi: 10.1128/AEM.71.11.6753-6761.2005.

[18] D. Heider, R. Senge, W. Cheng, and E. Hüllermeier, "Multilabel classification for exploiting cross-resistance information in HIV-1 drug resistance prediction," *Bioinformatics*, vol. 29, 2013, pp. 1946–1952, doi: 10.1093/bioinformatics/btt331.

[19] P. El Kafrawy, A. Mausad, and H. Esmail, "Experimental comparison of methods for multi-label classification in different application domains," *Int. J. Comput. Appl.*, vol. 114, no. 19, pp. 1–9, 2015.

[20] M. P. El-Kafrawy, M. A. Sauber, and A. Khalil, "Multi-label classification for mining big data," pp. 75–80, 2015.

[21] E. A. Tanaka, S. R. Nozawa, A. A. Macedo, and J. A. Baranauskas, "A multi-label approach using binary relevance and decision trees applied to functional genomics," *J. Biomed. Inform.*, vol. 54, pp. 85–95, Apr. 2015, doi: 10.1016/j.jbi.2014.12.011.

[22] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, 2009, pp. 254–269.

[23] J. E. Heydorn, "An improved classifier chain ensemble for multi-dimensional classification with conditional dependence," Master's thesis, Dept. Comput. Sci., Brigham Young Univ., Provo, Utah, U.S., Jul. 2015. Accessed: Feb. 12, 2021. [Online]. Available: https://scholarsarchive.byu.edu/etd/5515

[24] F. Javed and M. Hayat, "Predicting subcellular localization of multi-label proteins by incorporating the sequence features into chou's PseAAC," *Genomic.*, vol. 111, no. 6, pp. 1325–1332, 2019, doi: 10.1016/j.ygeno.2018.09.004.

[25] S. Kouchaki *et al.*, "Multi-label random forest model for tuberculosis drug resistance classification and mutation ranking," *Front. Microbiol.*, vol. 11, 2020, Art. no. 667.

[26] X. Xiao, P. Wang, W.-Z. Lin, J.-H. Jia, and K.-C. Chou, "iAMP-2L: A two-level multi-label classifier for identifying antimicrobial peptides and their functional types," *Anal. Biochem.*, vol. 436, no. 2, pp. 168–177, 2013, doi: 10.1016/j.ab.2013.01.019.

[27] M. Kulmanov and R. Hoehndorf, "DeepGOPlus: Improved protein function prediction from sequence," *Bioinformatics*, vol. 36, no. 2, pp. 422–429, Jan. 2020, doi: 10.1093/bioinformatics/btz595.

[28] M. Kulmanov, M. A. Khan, and R. Hoehndorf, "DeepGO: Predicting protein functions from sequence and interactions using a deep ontology-aware classifier," *Bioinformatics*, vol. 34, no. 4, pp. 660–668, Feb. 2018, doi: 10.1093/bioinformatics/btx624.

[29] W. Bi and J. Kwok, "Multilabel classification with label correlations and missing labels," *Proc. AAAI Conf. Artif. Intell.*, vol. 28, no. 1, pp. 1680–1686, 2014.

[30] H.-F. Yu, P. Jain, P. Kar, and I. Dhillon, "Large-scale multi-label learning with missing labels," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 593–601.

[31] R. Senge, J. J. del Coz, and E. Hüllermeier, "Rectifying classifier chains for multi-label classification," Jun. 2019. Accessed: Oct. 10, 2021. [Online]. Available: http://arxiv.org/abs/1906.02915

[32] A. Drouin *et al.*, "Predictive computational phenotyping and biomarker discovery using reference-free genome comparisons," *BMC Genomic.*, vol. 17, no. 1, Dec. 2016, Art. no. 754, doi: 10.1186/s12864-016-2889-6.

[33] J. J. Davis *et al.*, "Antimicrobial resistance prediction in PATRIC and RAST," *Sci. Rep.*, vol. 6, Jun. 2016, Art. no. 27930, doi: 10.1038/srep27930.

[34] R. K. Aziz *et al.*, "The RAST server: Rapid annotations using subsystems technology," *BMC Genomic.*, vol. 9, no. 1, Feb. 2008, Art. no. 75, doi: 10.1186/1471-2164-9-75.

[35] J. Zhang, Z. Zhang, L. Pu, J. Tang, and F. Guo, "AIEpred: An ensemble predictive model of classifier chain to identify anti-inflammatory peptides," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 18, no. 5, pp. 1831–1840, Sep. 2021.

[36] X. Jun, Y. Lu, Z. Lei, and D. Guolun, "Conditional entropy based classifier chains for multi-label classification," *Neurocomputing*, vol. 335, pp. 185–194, 2019, doi: 10.1016/j.neucom.2019.01.039.

[37] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," Nov. 2017. Accessed: Oct. 21, 2021. [Online]. Available: http://arxiv.org/abs/1705.07874

[38] S. Chen, "Interpretation of multi-label classification models using shapley values," Apr. 2021. Accessed: Oct. 21, 2021. [Online]. Available: http://arxiv.org/abs/2104.10505

[39] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in python," *J. Mach. Learn. Res.*, vol. 12, no. 85, pp. 2825–2830, 2011.

[40] A. R. Wattam *et al.*, "PATRIC, the bacterial bioinformatics database and analysis resource," *Nucleic Acids Res.*, vol. 42, pp. D581–D591, Jan. 2014, doi: 10.1093/nar/gkt1099.

[41] P. R. Anukrishna and V. Paul, "A review on feature selection for high dimensional data," in *Proc. Int. Conf. Inventive Syst. Control*, 2017, pp. 1–4. doi: 10.1109/ICISC.2017.8068746.

[42] M. Tharmakulasingam, C. Topal, A. Fernando, and R. La Ragione, "Backward feature elimination for accurate pathogen recognition using portable electronic nose," in *Proc. IEEE Int. Conf. Consum. Electron.*, 2020, pp. 1–5.

[43] B. Gholami, I. Norton, A. R. Tannenbaum, and N. Y. R. Agar, "Recursive feature elimination for brain tumor classification using desorption electrospray ionization mass spectrometry imaging," in *Proc IEEE Conf. Eng. Med. Biol. Soc.*, 2012, pp. 5258–5261, doi: 10.1109/EMBC.2012.6347180.

[44] M. Tharmakulasingam, C. Topal, A. Fernando, and R. La Ragione, "Improved pathogen recognition using non-euclidean distance metrics and weighted kNN," in *Proc. 6th Int. Conf. Biomed. Bioinf. Eng.*, 2019, pp. 118–124. doi: 10.1145/3375923.3375956.

[45] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005, doi: 10.1109/TPAMI.2005.159.

[46] Q. Liu, X. Tan, F. Huang, C. Peng, Y. Yao, and M. Gao, "GB-Tree: An efficient LBS location data indexing method," in *Proc. 3rd Int. Conf. Agro-Geoinformat.*, 2014, pp. 1–5.

[47] A. Maxwell *et al.*, "Deep learning architectures for multi-label classification of intelligent health risk prediction," *BMC Bioinf.*, vol. 18, no. 14, Dec. 2017, Art. no. 523, doi: 10.1186/s12859-017-1898-z.

Mukunthan Tharmakulasingam (Member, IEEE) received the BSc degree (Hons.) in electronics and telecommunications engineering from the University of Moratuwa, Sri Lanka, in 2014. He is currently working toward the PhD degree with the Centre for Vision, Speech and Signal Processing, University of Surrey, U.K. His current research focuses on applying machine learning techniques to biological data.

Brian Gardner received the MPhys degree (Hons.) in physics from the University of Exeter, U.K., in 2011, and the PhD degree in computational neuroscience from the Department of Computer Science, University of Surrey, U.K., in 2016. He is currently a research fellow with the Department of Pathology and Infectious Diseases, School of Veterinary Medicine, University of Surrey. His research interests include modeling biological data using mechanistic and machine learning-based approaches.

Roberto La Ragione is currently a professor of veterinary microbiology and pathology, in the School of Veterinary Medicine and Head of The School of Biosciences and Medicine at the University of Surrey. He graduated in 1995 and then studied for a post-graduate degree in veterinary microbiology at the RVC and a PhD degree at Royal Holloway. His current research interests focus on AMR and understanding the pathogenesis of foodborne pathogens with a particular interest in the development of control and intervention strategies.

Anil Fernando (Senior Member, IEEE) received the BSc degree (Hons.) in electronics and telecommunications engineering from the University of Moratuwa, Sri Lanka, in 1995, the MEng degree (Hons.) in telecommunications from the Asian Institute of Technology, Thailand, in 1997, and the PhD degree in video coding from the Department of Electrical and Electronic Engineering, University of Bristol, U.K., in 2001. He is currently a professor with the Department of Computer Science, University of Strathclyde, Glasgow, U.K. His research interests include video processing/coding, resource optimisation, artificial intelligence, quality of experience, intelligent video encoding for wireless systems, and video communication in 5G.