IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 31, 2020, accepted September 9, 2020, date of publication September 14, 2020,
date of current version September 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3023662

Investigating the Impact of Gene Cofunctionality in Predicting Gene Mutations of *E. coli*

MICHAEL OKWORI, (Graduate Student Member, IEEE), AND ALI ESLAMI, (Member, IEEE)
Department of Electrical Engineering and Computer Science, Wichita State University, Wichita, KS 67260, USA

Corresponding author: Michael Okwori (mxokwori@shockers.wichita.edu)

This work was supported in part by the National Aeronautics and Space Administration (NASA) under Award 80NSSC20M0133, in part by the National Science Foundation under Award OIA-1656006, and in part by the State of Kansas through the Kansas Board of Regents. The work of Michael Okwori was supported by the Petroleum Technology Development Fund (PTDF).

ABSTRACT Machine learning algorithms (MLAs) have recently been applied to predict gene mutations of *Escherichia coli (E. coli)* under different exposure conditions, with room for improvement in performance. In a bid to improve performance, we hypothesize that incorporating the interactions between genes will help MLAs make better predictions. To investigate this, we integrated protein-coding gene cofunctional networks into a mutation dataset of *E. coli* exposed to different conditions. Also, we proposed a feature-selection algorithm based on gene cofunctional networks to pick the most relevant exposure conditions. Then, we used the extended dataset to train a support vector classifier, an artificial neural network, and an ensemble of both MLAs. Separate models were trained for each of the protein-coding genes. Validation results showed that our approach improved both the area under the receiver operating characteristic (ROC) curve (AUC) and the area under the precision-recall curve (AUPRC). A peak increase of 8.20% in AUPRC was observed. A similar analysis on selected genes, with ten or more mutation points for each gene, also showed improvement in the general performance of the MLAs. Out-of-sample testing on adaptive laboratory evolution experiments curated from the literature provided further evidence of an enhanced mutation-prediction performance, where a maximum 8.74% boost in the AUC was observed. Finally, we highlighted the genes with the most improved and most degraded predictions due to the additional information of the cofunctional genes. This work suggests that the functional relationship between genes may play a role in gene mutation and illustrates how the relationships might help to improve mutation prediction.

INDEX TERMS Mutation prediction, *E. coli* gene interactions, feature selection, machine learning, artificial neural network, support vector classifier.

I. INTRODUCTION

A. MOTIVATION

Mutations are alterations in the genetic sequence of an organism and when passed over several generations can result in the acquisition of adaptive features by an organism [1]. The acquired adaptation can be either beneficial or detrimental to the survival of the organism. Exposure to specific environments can be the driver of harmful mutations that confer undesirable resistance to drugs on viruses and bacteria, and can also result in cancerous growths. These negative consequences can cause devastating effects such as ineffective medications and may ultimately lead to death if not mitigated. The environmental conditions of an organism can play a role in deciding if any mutations are passed to the next generation depending on the effect of the mutation on the organism's fitness. As such, it is worthwhile to be able to predict mutation outcomes following exposure to a given environment. The knowledge of mutation-inducing exposure and particularly susceptible genes not only enables measures to be taken to avoid the mutation but also facilitate carrying out targeted corrective actions, such as the recurrent modifications to vaccines as a result of the pathogens mutating and becoming resistant. This knowledge will also be beneficial in strain engineering and fermentation settings for microbial biotechnology, where desired qualities can be the target of induced mutations.

As highlighted in the work of [2], fields such as strain engineering and microbial biotechnology have a need to investigate a high number of exposure conditions, making it

The associate editor coordinating the review of this manuscript and approving it for publication was Xiangtao Li.

challenging to obtain the actual genomic sequence and to make use of already existing pipelines in order to detect mutations for the vast combinatorial space. As such, this work focuses on training machine models with data of mutations derived from the state-of-the-art pipelines on genomic sequencing with the aim of predicting mutations for novel exposure conditions. This is also with the aim of reducing the need to perform the traditional equipment-demanding and time-consuming genomic sequencing. This use of trained machine models can be employed in preliminary investigations to reduce the search space, i.e., conditions under considerations, before using the more accurate traditional approach. The model organism *E. coli* is used in this work, but a similar approach can be developed and applied to other organisms of interest.

The pattern of mutation occurrence due to exposure to varying environmental conditions can be a random and complex process. The question as to whether this process is repeatable and predictable has provided an interesting debate in the literature. The authors of [3] highlighted the challenges of the predictability of evolution, and then addressed the doubt by using examples of viral and microbial systems, populations of cancer cells, and repertoires of immune receptors to show that the competition of clades within a population creates deterministic evolutionary dynamics. MLAs are good candidates for learning and predicting complex processes and have been applied to predict various biological phenomena. These encompases investigation into the relatedness of genes, long non-coding RNAs (lncRNAs) and microRNAs (miRNAs) [4], [5], predicting phenotypic characteristics and environmental conditions from gene expression profiles [6], antibiotic resistance acquisition [7], analysis of cancers, survival outcome and disease-pathway associations [8]–[10], diagnosis of mutations in epidermal growth factors [11], and classifying protein binding activity to DNA [12], [13].

Also crucial in this area of research is the type of MLA and evaluation metrics utilized. Many MLAs have been used in the prediction of bioactivities, ranging from Naive Bayes, support vector classifiers, and decision trees to deep neural networks [4], [6], [7]. The particular algorithm that is used depends on criteria such as type of data, processing constraints, and the algorithm's performance in similar problems. Single-threshold measures such as accuracy, error rate, specificity, recall (sensitivity), and precision can be used to evaluate the performance of MLA. Threshold-free metrics, such as the area under the receiver operating characteristic (ROC) curve (AUC) and area under the precision-recall curve (AUPRC) are recommended because they avoid the impact of selecting a wrong threshold on the evaluation process [14].

A common practice in bioinformatics is the combination of different data types to boost the performance of models. This is because one kind of data may not be sufficient to capture the dynamics driving the variation in the process under investigation. For example, a combination of gene mutation and gene expression data was used to infer gene network rewiring [15], gene expression and DNA methylation were used to improve the indentification of clusters in cancer patients with different survival rates, and the integration of miRNA, messenger RNA (mRNA), and protein data was used to better identify pathways and networks [16]. The particular type of datasets that are combined are determined by the existing knowledge about the process under investigation. The current existing knowledge from adaptive laboratory evolution experiments has demonstrated that genotypic adaptation via mutation can occur on a functional level, and interactions among gene mutations can result in complex fitness effects [17], [18]. This cognition drives our conjecture that adding the functional interactions among genes may improve the prediction performance.

B. RELATED WORKS

In this section, we present an overview of related research. The focus of this work is not on improving the vast amount of machine learning models that have been used to investigate different biological processes, some of which are itemized in the introduction, but rather to better predict the occurrence of mutation. Although considerable research has gone into predicting the impact of mutations [19]–[22], only a few studies have considered predicting the occurrence of mutations.

Two main approaches have been applied to predicting mutation occurrence: use of descriptive analysis of testable hypothesis on mutations [3], [23], [24], and building predictive models [2]. The authors of [3] used viral and microbial systems, populations of cancer cells, and repertoires of immune receptors to show the predictability of evolution. They made the claim that the competition of clades within a population creates deterministic evolutionary dynamics, which forms the basis for models that predict evolution. The nucleotide substitution rate was used to investigate adaptive evolution in [23]. Here, the phylogenic tree of the evolutatonary history of a gene was reconstructed and then used to estimate the nucleotide substition rate over time. The hypothesis that postitive selection was responsible for a high rate of non-synonymous substitutions over synonymous substitutions in human influenza haemogglutinnin was confirmed to be true for a number of years. A mechanistic model of three regulatory pathways of the model bacterium *Pseudomonas fluorescens* was presented in [24] and then used to predict the mutational rate of each route and mutational targets.

In [2], a dataset of gene mutations in *Escherichia coli* (*E. coli*) due to exposure to various conditions was collected and then used to train an ensemble involving an artificial neural network (ANN), support vector classifier (SVC), and Naive Bayes algorithm. The prediction performance obtained was promising, with possible room for improvement. This work is our benchmark for comparison. In a bid to investigate whether gene interactions can help make better predictions, we studied the impact of incorporating gene interaction on predicting mutations. This holds the promise of making it possible to accurately predict the genomic state of other

genes from the genomic state and exposure conditions of a few key genes. This will address the gap of sub-optimal performance in mutation prediction and confirm features that can be utilized in addition to the already established features.

Although the performance of models predicting the effect of mutations are appreciable, the only work that built machine learning models to predict the occurrence has room for improvement in the performance. This work attempts to improve the performance by using the interactions among the protein coding genes as an additional training feature. A key, and crucial, assumption in this work is that the mutation state of the cofunctional genes are known a priori. We believe that this assumption is appropriate at the current level of investigation, which is the focus of this work. Before practical applications are developed, this issue will be addressed. A number of potential approaches to addressing this assumption are briefly highlighted in the discussion of future work in section VII.

C. NOVELTY AND CONTRIBUTION

There is a limited amount of research work on predicting mutation from exposure-conditions. Benchmark work, the only literature we could find, recently collated the first dataset aimed at boosting research in this field. Our work is different in our data setup and in the feature-selection process. Our main contributions in this work are as follows:

- We incorporate the cofunctionality relationship of genes into our base dataset with a gene mutation pattern of 13 strains of *E. coli* when exposed to varying environmental conditions. In doing so, we have identified cofunctionally connected genes to each gene in the base dataset, determined the state (i.e., whether mutated or not) of such connected genes for each input condition, and then added these states to the expanded input.
- We propose a feature-selection process based on the feature significance of the exposure-condition features of each gene and its cofunctional genes. For each gene, we utilize a number of decision trees to determine the selection metric of all the features. Features with metrics above a set threshold are then selected for the gene under consideration. In addition, the feature-selection process adds the features of the cofunctional genes with metrics above the threshold.
- We train and validate an ANN and SVC on the expanded database. We have also determined the best weights in which to combine the ANN and the SVC for an ensemble model. In the validation process, we investigate optimizing parameters and the ensemble weights at the gene level or at the entire $1,561$ protein-coding genes level. We report the validation performance over the entire $1,561$ protein-coding genes and over 112 protein-coding genes that were mutated at least ten times in the database.
- We evaluate our models on five adaptive laboratory evolution (ALE) experiments curated from the literature, and present the analysis of the results obtained. From the AUC and AUPRC of both the validation process and the out-of-sample testing, we subsequently investigate the impact of our feature selection and the expanded dataset. Then, we proffer inferences from the validation and test results.
- We classify *E. coli* genes based on the impact of incorporating the cofunctional gene network on the training and subsequent predictions of machine learning algorithms. We carry out gene ontology GO) enrichment analysis and highlight the cellular components, biological processes, and molecular functions of the top genes in each class.

D. OVERVIEW OF WORK AND PAPER ORGANIZATION

A schematic of the processes in this work is shown in Fig. 1a. First, the training data was set up by concatenating dataset 1 and dataset 2. Section II describes both datasets and data concatenation using Algorithm 1. Then, the newly set up dataset was split into a training set and and a validation set. Subsequently, feature selection and oversampling was performed on the training set. Section III presents the details of the feature-selection Algorithm 2 and the oversampling procedure. ANN and SVC models, described in section IV, with various parameter settings were trained using the selected features and the oversampled training dataset. The training and testing procedure are outlined in section V. The best-performing model parameters on the validated set, including the best weights to combine the predicted mutation probabilities of both the ANN and SVC for the ensemble model, were selected as the final model parameters. The models with selected parameters were then trained on the entire training dataset. We tested the performance on two test sets, and then analyzed the mutation prediction results in Section VI. Section VII concludes the paper and highlights future work.

II. DATASET DESCRIPTIONS AND DATA SETUP

Here, we discribe the datasets used in this work. Our entire training dataset is comprised of: dataset 1 and dataset 2. The test dataset is five adaptive laboratory evolution experiments curated from the literature.

A. DATASET 1

The dataset collated in the work of [2] was the base dataset for this work. This dataset came from a curation of *E. coli* adaptive laboratory evolution experiments and provided mutation states of $1,991$ genes or intergenic regions for 178 conditions (across 13 strains, 19 media, and 49 stresses). Out of the $1,991$ genes or intergenic regions, $1,561$ were protein-coding genes. Out of the protein-coding genes, 112 had ten or more mutation hits.

B. DATASET 2

Adaptive laboratory evolution experiments have shown that genotypic adaptation via mutation can occur on the functional level, and interactions among gene mutations can result in complex fitness effects [17]. Hence, we hypothesize that

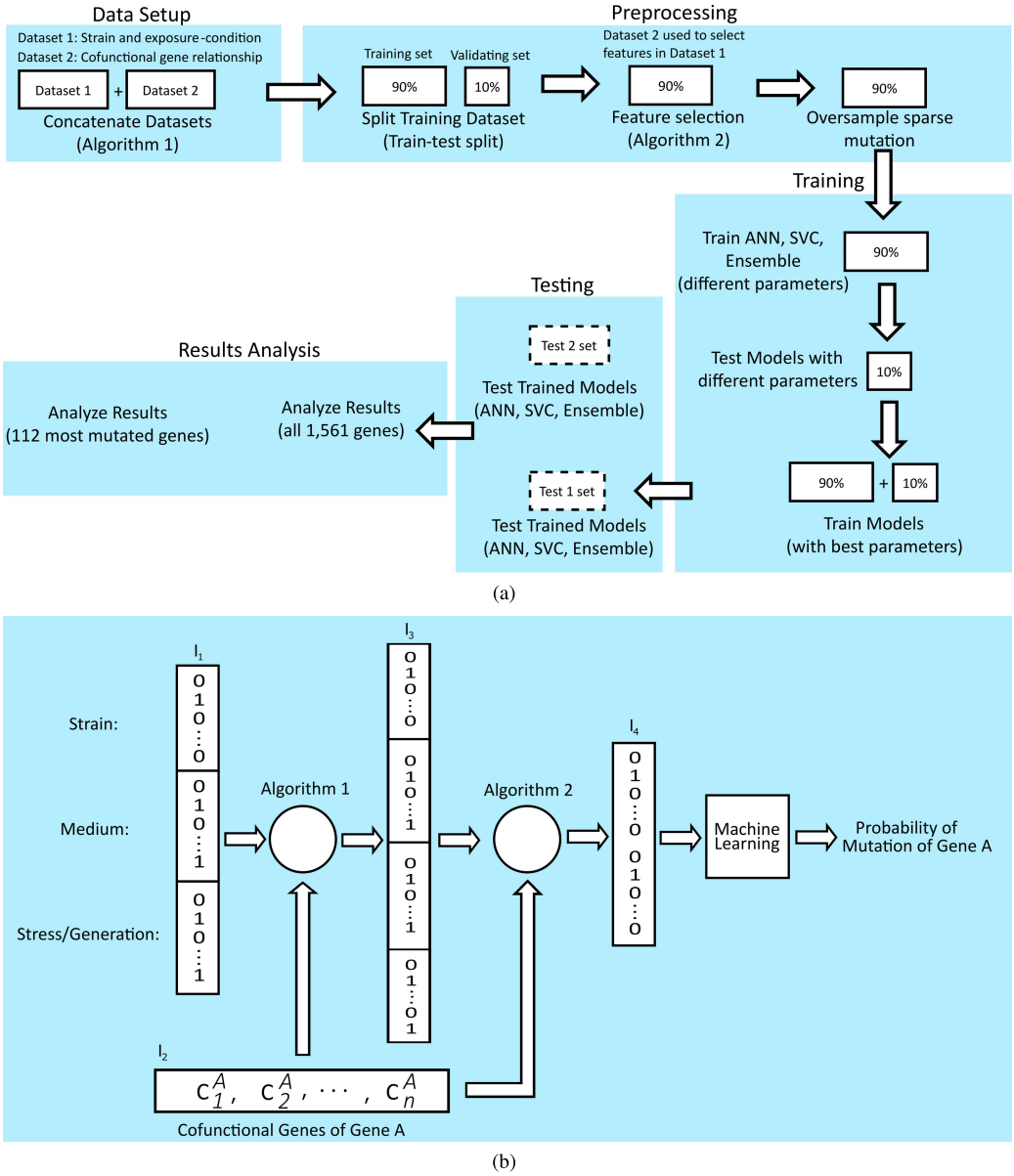

FIGURE 1. Process flow of method illustrating various stages, and example of input and output of machine model: (a) Solid boxes are related to training data, dotted boxes are testing data, and beside each box is process performed on each set of data. Flow shows how information leakage was prevented during both validation and testing processes. (b) Binary numbers representing strain of *E. coli*, medium, stress, and number of generations of exposure (I_1) are inputted into Algorithm 1. List of cofunctional genes (Cs) of Gene A (I_2) also provided to both algorithms. Algorithm 1 produces I_3, which is concatenation of I_1 and binary coded I_2, and output of the feature selection in Algorithm 2 (I_4) is final input to machine learning model. Model output is probability of mutation of Gene A.

adding functional interactions among the genes may improve the prediction performance. To investigate and test this theory, we incorporated a second dataset of gene cofunctional relationships [25] into the dataset.

The cofunctional gene networks in the work of [25] were constructed using gold standard positive links evidenced by shared GO annotation along with evidence inferred from direct essay, inferred from genetic interaction, or inferred from mutant phenotype. Gold standard negative links were also inferred from annotated genes that do not share any of the annotations. The gold standard links were then used to score links inferred by the coexpression pattern of genes, cooccurrence of protein domains among coding genes, similar genomic contexts of bacterial orthologs of genes, high-throughput protein-protein interactions, small/medium-scale protein-protein interactions, similar phylogenetic profiles among genes, and cocitation of genes in the literature.

C. DATASET CONCATENATION

For each of the 1, 561 protein-coding genes in dataset 1, we added the extracted cofunctional related genes in dataset 2. We focused on just the protein-coding genes

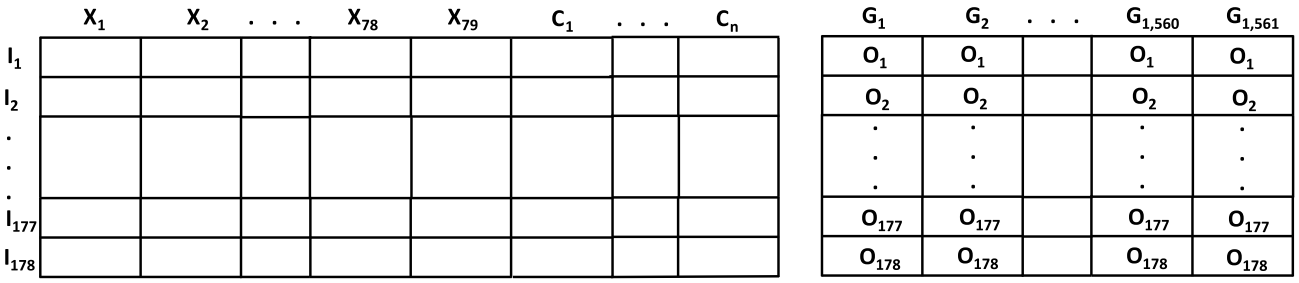

FIGURE 2. Structure of concatenated dataset showing extended features of input and range of output. Each of rows I_1 to I_{178} represents an input condition, with features varying across strain, medium, stress, generations of exposure (X_1 to X_{79}), and state of cofunctional related genes (C_1 to C_n). For each gene, G_1 to $G_{1,561}$, and for each of 178 input conditions, output O_1 to O_{178} is 1 if gene under consideration is mutated, and 0 if not mutated.

because dataset 2 provides links between the protein-coding genes. The integration of the connected gene to the base dataset was done using Algorithm 1. This process involves identifying inferred connected genes (line 2) and then determining the state (mutated or not) of the connected genes (line 5) for each of the 178 exposure conditions in the database. Here, we applied the assumption that the state, i.e., whether mutated or non-mutated, of the cofunctional genes are known a priori for each of the 178 exposure conditions. Finally, we added these state conditions to the training inputs of the considered gene (lines 6 or 7). The structure of the resulting extended dataset used in the training is depicted in Fig. 2.

Algorithm 1 Gene Connectivity Information Extraction and Integration Algorithm

Input: Dataset1 ($D1$), Dataset2 ($D2$)

// loop through the 1, 561 genes and get cofunctional genes

1: **for** $i \leftarrow 1, N$ (where N is the total genes in D1) **do**

2: From D2, Get $C \leftarrow [C_1, C_2 \ldots., C_n]$,
where C: set of the related genes,
and n: number of genes cofunctionally related to Gene$_i$

// loop through cofunctional genes of each gene

3: **for** $k \leftarrow 1, n$ **do**

// loop through exposure conditions

4: **for** $j \leftarrow 1, E$ (where E is the total 178 exposure conditions in D1) **do**

// concatenate the mutation state of each cofunctional gene (C_k^i) to each of the exposure conditions (I_j^i)

5: **if** C_k^i is mutated in I_j^i **then**
append 1 to I_j^i

6: **else**
append 0 to I_j^i

7: **end if**

8: **end for**

9: **end for**

10: **end for**

Output: Extended Database for Each Gene with Connectivity Information, (DE)

D. TEST DATASET

We curated adaptive laboratory evolution experiments from the literature to form the test dataset. Table 1 shows the literature from which the test dataset was obtained.

TABLE 1. Description of ALEs used for testing.

Test Sample	Literature Source
I	Culture in [2]
II	Glycerol culture in [26]
III	Xylose culture in [26]
IV	Culture in [27]
V	Culture in [28]

E. EXAMPLE OF MODEL INPUTS AND OUTPUT

An example of the model inputs and output for a sample Gene A is depicted in Figure 1b. The first input, $I_1 = (X_1, X_2, \ldots, X_{79})$, $X_i = 1$, for the *E. coli* strain from which Gene A is derived, and the remaining strain related $X_i = 0$. Similarly, all the X_i representing the medium, stress, and generations of exposure of the sample are set to 1, and the remaining are set to 0. A list of what each i represents can be found as features listed in Supplementary Data 3. As a further illustration, if $X_{10}, X_{17}, X_{28}, X_{34}, X_{46}, X_{74}, X_{78} = 1$, and other $X_i = 0$, then Gene A is from an MG1655 *E. coli* strain in an M9 medium with glucose as the carbon source, is exposed to ampicillin in the presence of oxygen at a temperature of 30°C, and is observed for 500 generations. The second input, I_2, is a list of cofunctional genes of Gene A. For example, *rpoC* will have its cofunctional genes of *groL, rpsF, rnr,* and *thiH* in I_2. Algorithm 1 represents the cofunctional genes in binary form depending on the state, i.e., whether mutated or not in condition I_1. Finally, the most relevant features from the concatenated features in I_3 are extracted to obtain I_4, which is then fed into the machine learning. The output of the model is the probability of mutation of Gene A, given the strain and exposure conditions (I_1), and the state of the cofunctional genes (I_2).

III. DATA PREPROCESSING

Here, we briefly describe the preprocessing procedures that were carried out on our dataset. We describe

the feature-selection process utilized and oversampling procedure.

A. FEATURE SELECTION

Given the high dimensions of the input data, we propose a feature-selection procedure to select the most relevant features. For each of the genes, we used the procedure outlined in the proposed feature-selection Algorithm 2 to select the most relevant features after the cofunctional genes were added. This algorithm selects the features of each gene and its cofunctional genes that have a *Feature Importance* greater than zero, set to the minimum to ensure that all informative features are selected. We compute the *Feature Importance* using XGBoost [29]. The XGBoost trains a number of decision trees, and each decision tree uses each of the features as the splitting point in the decision trees and measures how well each split reduces the impurity. The *Feature Importance* metric, a measure of how well each feature performed in the individual decision trees, is then summed across the trees and weighted by the number of trees. Result of the selection procedure and the biological backing are presented in section VI-A.

Algorithm 2 Feature-Selection Algorithm

Input: Dataset1 ($D1$), Dataset2 ($D2$)

for $i \leftarrow 1, N$ (where N is the number of genes in D1) **do**
  for $j \leftarrow 1, n$ (where n is the number of features (F) in D1) **do**
    Compute the metric of F_j^i: M_j^i
    if $M_j^i > 0$ **then**
      append F_j^i to $F_{Selected}^i$
    else discard F_j^i
    end if
  end for
  for $k \leftarrow 1, C$ (where C is the number of genes in D2) **do**
    for $l \leftarrow 1, n$ (where n is the number of features in D1) **do**
      Compute the metric of F_l^k: M_l^k
      if $M_j^k > 0$ **then**
        append F_l^k to $F_{Selected}^i$
      else discard F_l^k
      end if
    end for
  end for
  Remove duplicate features in $F_{Selected}^i$
  Add $F_{Selected}^i$ to $F_{Selected}$
end for

Output: Selected Features, ($F_{Selected}$)

B. DATA OVERSAMPLING

To take care of class imbalance due to the sparse mutation points, prior to the training of models for each gene, we oversampled the sparse points of mutation to equal the number of times the gene was not mutated. This is to prevent the models, particularly the ANN, from being trained on more of any of either the two classes. Training on the majority of a class, in this case the non-mutated points, is capable of resulting in a bias towards predicting only the majority non-mutated class. A random resampling of the mutation points to equal the number of non-mutated points for each of the 1, 561 protein-coding genes was adopted. For example, out of the 178 initial data points, gene *rpoC* was mutated 20 times and was not mutated in 158 points. As such, these 20 points were randomly sampled 138 times in order to equal the 158 non-mutated points. This increased the number of available data for *rpoC* from 178 to 336. This procedure was repeated for all 1, 561 protein-coding genes, which ensured a 1:1 ratio of the mutated points to the non-mutated points for each gene. To prevent information leak, we only oversampled the training set, and not the validation set or the test dataset.

IV. MODEL DESCRIPTION

Two models were trained in this work: an artificial neural network and a support vector classifier. After training both models, we found the best weighted ensemble of both models. Here, we briefly describe both models, the working principle, and our ensemble model.

A. ARTIFICIAL NEURAL NETWORK (ANN)

Our ANN uses one or two layers of neurons to model the relationship between input and output. The output of each neuron is

$$f(\boldsymbol{X}, \boldsymbol{w}, b) = \phi(\boldsymbol{X}.\boldsymbol{w} + b) = \phi\Big(\sum_{i}^{p} (X_i.w_i + b)\Big), \quad (1)$$

where X, w, and b are, respectively, the input, weight, and bias vectors of each neuron; p is the number of inputs to each neuron; and $\phi(.)$ is the activation function of each neuron. Our input (X) to the neural network is structured as follows:

$\{X_1 - X_{14}\} \rightarrow$ Strain related
$\{X_{15} - X_{33}\} \rightarrow$ Medium related
$\{X_{34} - X_{79}\} \rightarrow$ Stress/generations related
$\{X_{80} - X_N\} \rightarrow$ State of cofunctional related genes

where N is the total number of training features and varies for each gene as the number of the inferred related gene varies. To determine the model parameters, such as the number of hidden layers, the number of nodes, or the dropout rate, we performed the model validation process described in section V-A. The ANN implementation was done using the Keras with Tensorflow, a machine learning open-source platform written in Python [30].

B. SUPPORT VECTOR CLASSIFIER (SVC)

Our SVC has a decision function of

$$f(x) = \text{sgn}\left(\sum_{i=1}^{n} \alpha_i y_i \boldsymbol{K}(X, X_i) + b\right), \quad (2)$$

and Lagrange multipliers $\alpha_i > 0$ for the few support points, and the majority of $\alpha_i = 0$; X, y, and b are the input, output, and bias, respectively. $\boldsymbol{K}$ is the kernel function. The input (X) is structured as described for the ANN. The SVC was implemented using the scikit-learn library in Python [31], and the best parameters were determined in the validation process.

C. ENSEMBLE MODEL

Ensemble models combine two or more other models in a bid to produce optimal predictions. Different combining strategies are adopted in ensemble models, including unweighted votes and weighted sums [32]. Our ensemble model used a weighted sum of the probability predictions of both the ANN and the SVC:

$$\begin{pmatrix} P_1 \\ P_2 \\ \vdots \\ P_m \end{pmatrix} = \begin{pmatrix} \alpha_1 P_1^S \\ \alpha_2 P_2^S \\ \vdots \\ \alpha_m P_m^S \end{pmatrix} + \begin{pmatrix} (1-\alpha_1) P_1^A \\ (1-\alpha_2) P_2^A \\ \vdots \\ (1-\alpha_m) P_m^A \end{pmatrix}, \tag{3}$$

where P_i is the ensemble model's predicted probability of the mutation of gene i, α_i is the weight of gene i, P_i^A is the ANN predicted probability of the mutation of gene i, P_i^S is the SVC predicted probability of mutation of gene i, and m is the 1, 561 protein-coding genes. The best α for each gene was determined during the validation process, as explained in section V-A.

V. MODEL TRAINING AND TESTING

In this section, we describe our model training procedures and the testing performed. Training involved model validation prior to final training. In the validation process, we tested a number of hyperparameters of the models to select the best. Finally, we tested a number of weights to determine the best combination ratio for the ensemble. We then highlighted the two tests that were used to evaluate the performance of our models on unseen data.

A. MODEL VALIDATION

In the validation process, we split the dataset into training and validation sets, using a 90:10 ratio for the non-mutated points, and a 50:50 ratio for the sparse mutated points. This ensured that the sparse-mutation points were contained in both the training and validation sets, as shown in Fig. 3. The validation process involved training the models with the training set, and testing the models with varying hyperparameter settings.

We focused our search for the models' hyperparameter settings on those reported to yield the best performance in most genes in the literature (i.e., benchmark work). For the ANN model, we tested models with the following parameters:

- Number of hidden layers: 1, 2.
- Number of nodes for 1 hidden layer: 54, 78, 26.
- Number of nodes for each of the 2 hidden layers: 57_37, 54_12, 49_40.
- Dropout rate: 0.1, 0.2, 0.3, 0.4.

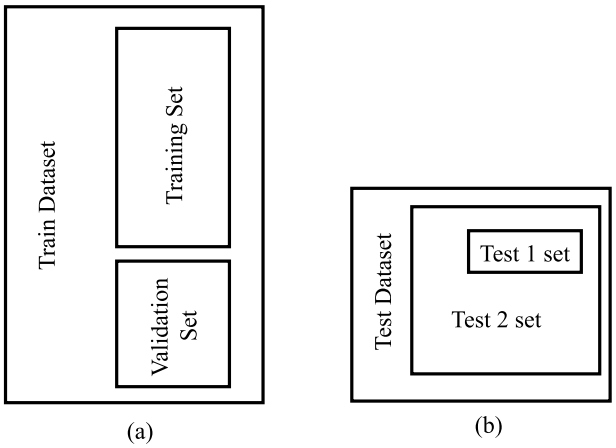

FIGURE 3. Description of training, validation, and test datasets: (a) Training dataset comprised of 178 mutation data (collated in benchmark work) divided into training set and validation set. Training set was used to train models with different hyperparameters, and validation set was used to evaluate performance of each model. (b) Test set is comprised of Test 1 set (one laboratory test in benchmark work) and Test 2 set (all five ALEs extracted from literature).

For the SVC, we tested models with the following parameters:

- Kernels: linear, polynomial, radial basis function, sigmoid.
- Penalty of misclassifications (c): 0.0005,0.005,0.05.
- Kernel coefficient (gamma): 0.1,0.5,1 (varied for all kernels except linear).
- Degree: 2,3 (varied for only polynomial kernel).

The weights by which the two models are combined was also determined via the validation process, using the same testing and validation set. We set the weighting parameter, $\alpha = \{0, 0.1, 0.2, \ldots 1\}$ and identified the best α to use in the combination of the two models.

We performed the validation process at two granular levels: the individual protein gene level and all the protein-coding genes level. At the individual protein gene level we determined the model parameters and weighting factors for each gene. For the second level, we found the parameters and weighting factor that performed best across all of the protein-coding genes. Contrary to our expectation, the parameters obtained at the higher granularity of individual genes did not perform as well as the less-complex procedure of determining parameters with the best performance across the entire protein-coding genes (see supplementary data 1 and 2).

B. MODEL TRAINING

After the validation process, we trained separate ANN and SVC models for each protein-coding gene with the parameters having the best performance across all protein-coding genes using the entire train dataset, i.e, a combination of the training set and the validation set. We used the harmonic mean of both the AUC and AUPRC obtained in the validation test to select the best-performing parameters, in an attempt ensure a balance in optimizing both metrics. The ANN model with

the best performance had a *tanh* activation function for two hidden layers of 54 and 12 nodes, and a *sigmoid* function output layer. To prevent overfitting, we adopted $L1$ regularization and a dropout rate of 0.2, also confirmed to yield the best performance in the validation process. Regularization adds a penalty for model complexity, while dropout randomly ignores some neural nodes in the forward and backward propagations during training. Similarly, the best SVC model utilized a polynomial kernal of degree 2, with a penalty for misclassifications, $c = 0.0005$, and a kernel coefficient of $g = 0.5$. For the ensemble model, we selected $\alpha = 0.5$ in our final model.

For benchmarking purpose, we trained models using the approach in the benchmark purpose, i.e., without either of our feature selections or incorporating the mutation state of the cofunctional genes. In this benchmark approach, the ANN model with the best performance had the same parameters as in our approach, except it had a dropout rate of 0.1. Similarly, the SVC only had a change of $c = 0.005$.

C. TESTING

In our testing, we evaluated the three models: ANN, SVC, and the ensemble model trained using the benchmark approach and our approach. The benchmark approach uses only the strain and exposure conditions as model inputs, while our approach utilizes our feature-selection process to select the most important features from the exposure conditions, and then incorporates the mutation state of the cofunctional genes. These models were trained for each gene, and as such were also tested for each gene.

We ran two tests: the first on the adaptive laboratory evolution experiment reported in the benchmark work, and the final test on five ALE experiments, four from the literature in addition to the benchmark's ALE experiment. For both tests, the strain, exposure conditions (benchmark approach), and the mutation state of cofunctional genes (our approach) were used as input features, and the genes that were mutated were coded as the output features. Details of the input features and mutated genes in the test ALEs can be found in the supplementary data 3.

VI. RESULTS AND DISCUSSION

We begin this section with a look at results of the feature selection and touch upon the biological significance of the selected features. We also highlight the performance metric used in our evaluation. Then, we present the results of the validation process and draw some inferences. Finally, we discuss the test results, and highlight some inferences from the test results and the validation results.

A. BIOLOGICAL DISCUSSION OF SELECTED FEATURES

From the 79 strain and exposure conditions-related features, an average of 20 features were selected by the feature-selection algorithm for the 1, 561 protein-coding genes, with a maximum of 29 and a minimum of 1. Table 2 shows the exposure conditions selected the most for the genes,

TABLE 2. Exposure conditions selected by feature-selection algorithm.

Feature	Selection Frequency	Genes with Highest $Feature\ Importance$
Generation_30001∼400000	684	insB-1, rseP, yagU, yhhI, cbeA
Stress_None	510	evgS, malX, rep, yhiL, ydaG
Generation_0∼500	386	emrD, lpxC, ppiA, mntP, hokE
Medium_M9	376	gcd, waaU, yeaR, sthA, folA
Medium_Davis	285	topA, glpE, fadL, mdfA, rhsA
Carbon_Glucose	279	yfgO, fucO, nadB, ykfH, ydeT,
Generation_1001∼5000	249	ydfI, evgA, dapF, allD, lacI
Temperature_37	206	ygiC, sgbU, yiaA, bcsG, dppC
Temperature_30	185	acrB, cyoA, sbmA, pntB, yaiT
Medium_LB	164	pncB, acrA, yjjA, hscC, entF
Generation_501∼1000	145	pgl, asnS, xapA, pyrB, argR
Generation_5001∼10000	135	gadC, nirC, manY, yjgR, treA
Carbon_None	125	yiaW, mppA, ydaG, yaeF, yobD
Temperature_41∼45	116	abgB, cbpA, ybiO, nagC, dusB
Carbon_Glycerol	104	acrD, pck , glgC, cstA, glcF
Medium_MS-Minimal	92	atpD, argS, cpxR, ypjC, yffS
Stress_Butanol	43	acrA, puuP, cydB, atpF, tsgA
Temperature_46∼50	35	treB, mlc, idi, wecF, tktB
Generation_10001∼20000	29	nrdF, ygbN, nlpD, pcm, rpsD
Stress_Tobramycin	21	potA, yaiY, yaiZ, cpxA, trkH
Stress_Ciprofloxacin	9	sseA, hypF, yfgI, recE, wecB
Stress_Doxycycline	1	rhlB

TABLE 3. Features not selected for any gene.

Features	
Carbon_D-Gluconate	Stress_H_2O_2
Carbon_L-1_2-PDO	Stress_Acid
Carbon_L-Lactate	Stress_Amikacin
Carbon_Maltose	Stress_Ampicillin
Carbon_Xylose	Stress_Cefoxitin
Medium_Gut In vivo	Stress_Chloramphenicol
Medium_LBK	Stress_Chloranphenicol
Medium_M63 Mod	Stress_Clindamycin
Medium_M9-Biotin	Stress_Decreasing Gly_Increasing PDO
Medium_MMA	Stress_Erythromycin
Medium_RPMI	Stress_Fusidic acid
Medium_T-salts Minimal	Stress_Kanamycin
Generation_20001∼30000	Stress_Lomefloxacin
Oxygen_Anaerobic	Stress_Sulfamethaxozole
Oxygen_Aerobic	Stress_Sulfamonomethoxine
Temperature_25	Stress_Nitrofurantoin
Stress_Osmotic	Stress_Murine_monocytic_cells
Stress_Nalidixic acid	Stress_N-Methyl-N9-nitro-N-nitrosoguanidine
Stress_Piperacillin	Stress_Spectinomycin
Stress_Spiramycin	Stress_Streptomycin
Stress_Tetracycline	Stress_Trimethoprim

the frequency of selection, and the genes that selected the feature with the highest *Feature Importance*. The features not selected for any of the genes are found in Table 3. Details of the features selected for each of the protein-coding genes is presented in supplementary data 4.

The most-selected feature was the *Generation*_30001 ∼ 400000 feature, which indicates if the generations of exposure lies withing the $30,001-400,000$ range. This is the highest number of exposure generations available in the feature set. It was observed that five out of the entire selected features were related to the number of generations

of exposure. This seems to be in line with the understanding that *E. coli* has an approximate spontaneous mutation rate of 1.0×10^{-3} mutations per genome per generation [33]. Hence, the generation of exposure would most likely be relevant to the mutation of genes.

The next set of highly selected features was the *Stress_None* feature, which represents when no stressing conditions such as ethanol and antibiotics are present. In addition to this feature, only three others related to stress were selected, and only by less than a total of 35 genes. A majority of the stress conditions, a total of 26 other such conditions, were not selected by any of the genes. This appears to support the position that the emergence of antibiotic resistance mutants in *E. coli* is a complex phenomenon [34]. This may be responsible for the stressing condition, which in isolation is considered to be less relevant to the genes that are mutated.

Finally, out of the media considered, four were selected and seven were not selected. Also, normal room temperature (25°C) appears to not be relevant because it was not selected by any gene, while the other higher temperatures were relevant, with 37°C being the most selected, in line with the findings of [35] that there is an increase in the mutation rate and a bias for protein-coding genes mutation at this temperature. The presence or absence of glucose and glycerol carbon sources appears to be relevant, while the other five carbon sources were not selected by any of the genes.

B. PERFORMANCE METRICS

In our analysis, we used the area under the precision-recall curve and also the more optimistic area under the receiver operating characteristic curve [14], [36], both threshold-free metrics. The AUC indicates the performance over all false-positive rates and is a measure of how well the model distinguishes the mutated points from the non-mutated points. The AUPRC is a measure of how well the model is performing on the positive class, that is, the mutated genes [37]. The baseline AUPRC value, which shows how a random predictor will perform, is dependent on the ratio of the number of positives to the total number of samples [14]. The baseline AUC is fixed at 0.5. It is also worth noting that algorithms that optimize the AUC do not generally optimize the AUPRC [38].

To investigate the statistical significance of changes in both the AUC and AUPRC, we utilized the *t*-test to compare the results of our approach and the benchmark approach over ten simulation runs. A similar approach of comparing across simulation runs was utilized in [39]. To test if the mean performance of our approach exceeds that of the benchmark approach, we formulated an upper tail test where the null hypothesis and the alternate hypothesis are defined, respectively, in equations 4 and 5:

$$H_0 : \mu_o - \mu_b \leq 0, \tag{4}$$

$$H_1 : \mu_o - \mu_b > 0, \tag{5}$$

where μ_o and μ_b are the mean across the ten runs for our approach and the benchmark approach, respectively. We assume the data points are drawn from a normally distributed population with unknown but equal variance, and then we compute the critical statistic, t_c, as

$$t_c = \frac{(\mu_o - \mu_b)}{S_p \sqrt{\left(\frac{1}{n_o} + \frac{1}{n_b}\right)}}, \tag{6}$$

$$S_p = \sqrt{\frac{(n_o - 1)s_o^2 + (n_b - 1)s_b^2}{n_o + n_b - 2}}, \tag{7}$$

where s_o^2 and s_b^2 are the standard deviation of the performance of our approach and the benchmark approach across the n_o and n_b simulation runs, respectively. The procedure to determine significance involves calculating the critical value (t_c), using equation 7, and determine the p-value from tables or using statistical software. We used the software approach to determine the p-value. The p-value is compared with a set threshold of 0.05, a commonly used value [40]–[42], to determine statistical significance. The observed improvement is not by chance if the p-value is below this threshold. As such, all subsequent use of the term statistical significance in this work does not mean a big improvement in values but that the observed improvement is not due to chance. Since only one t-test was run, it was not necessary to apply the false discovery rate correction due to multiple testing to the p-values.

C. VALIDATION RESULTS

The AUC and AUPRC values of the models averaged over ten simulation runs are reported in Table 4 (the results of each of the ten runs are presented in supplementary data 5). This shows that our approach outperformed the benchmark approach for all models, except for the AUC of the SVC model. The p-values obtained confirm that the improvement observed across all models, in both the AUC and AUPRC values, was statistically significant. The ROC and precision-recall (PR) curves for one of the ten simulation runs are shown in Fig. 4. These curves show that although the variation of the true positive rate with the false positive rate was very similar for both approaches, our approach achieved a higher precision rate for a majority of recall values. Also, both approaches performed better than the baseline AUC and AUPRC values, the dotted lines on the plots, which indicate how a model randomly predicting gene mutation would perform.

To explore the impact of more mutations points, we repeated the validation procedure for the 112 protein-coding genes with ten or more mutation points. Results of this test, shown in Fig. 5 and Table 4, indicate an increase in the performance of all models, demonstrated by the higher AUC and AUPRC values, and our approach still reveals a better performance. Also, the p-values confirm that the improvements observed across all models, in both AUCs and AUPRCs, but were statistically significant.

The validation results illustrate the tendency of making better predictions by our approach of using cofunctional genes to select the relevant exposure conditions and the mutation state

TABLE 4. Validation results of ML models tested over entire 1, 561 and selected 112 protein-coding genes.

Model	Metric	Average Performance over Ten Tests (mean ± std)		*P*-value
		Benchmark Approach	Our Approach	
Test over 1,561 Protein-Coding Genes				
ANN	AUC	0.86 ± 0.00	$\mathbf{0.87 \pm 0.00}$	10^{-13}
	AUPRC	0.62 ± 0.00	$\mathbf{0.66 \pm 0.00}$	10^{-21}
SVC	AUC	$\mathbf{0.83 \pm 0.00}$	0.82 ± 0.00	10^{-13}
	AUPRC	0.56 ± 0.00	$\mathbf{0.61 \pm 0.00}$	10^{-21}
Ensemble	AUC	0.86 ± 0.00	$\mathbf{0.87 \pm 0.00}$	10^{-11}
	AUPRC	0.61 ± 0.00	$\mathbf{0.66 \pm 0.00}$	10^{-24}
Test over Selected 112 Protein-Coding Genes				
ANN	AUC	0.94 ± 0.00	$\mathbf{0.95 \pm 0.00}$	10^{-13}
	AUPRC	0.91 ± 0.01	$\mathbf{0.92 \pm 0.00}$	10^{-20}
SVC	AUC	0.91 ± 0.00	$\mathbf{0.92 \pm 0.00}$	10^{-12}
	AUPRC	0.86 ± 0.01	$\mathbf{0.89 \pm 0.00}$	10^{-20}
Ensemble	AUC	0.94 ± 0.00	$\mathbf{0.95 \pm 0.00}$	10^{-11}
	AUPRC	0.90 ± 0.01	$\mathbf{0.92 \pm 0.00}$	10^{-23}

Higher AUC and AUPRC of approaches are written in bold letters, only exponent reported in the *p*-values.

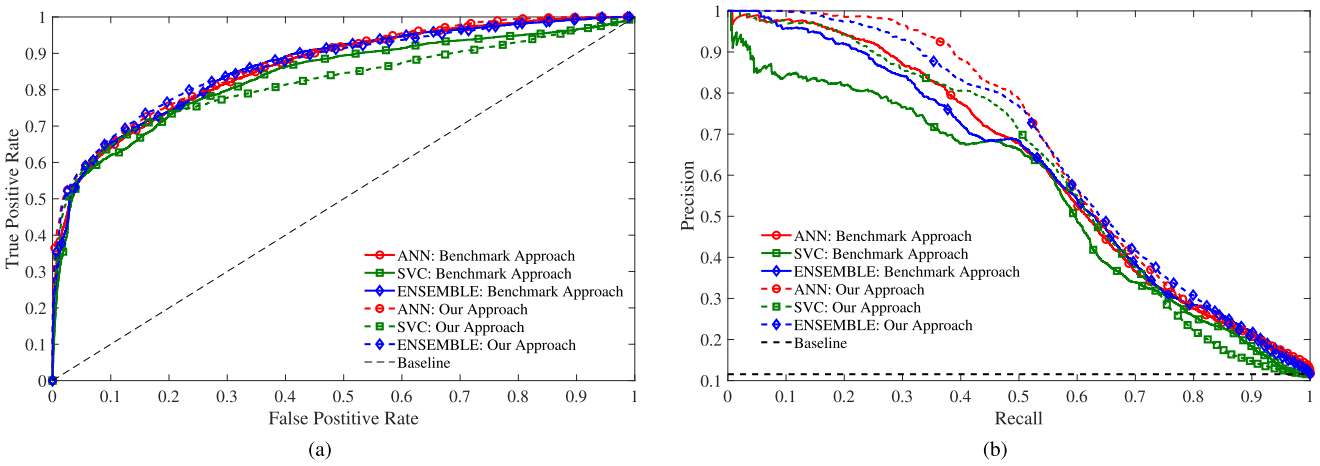

FIGURE 4. Performance curves for validation of models on all protein-coding genes in dataset: (a) ROC curve shows that both approaches achieve similar true positive rate for majority of false positive rates. (b) PR curve also indicates that our approach has better performance and results in higher AUPRC.

added to the selected features. This apparent improvement in the prediction tends to hold for the availability of more data, with an even better performance by both approaches.

D. TEST RESULTS

In our first test, the exposure conditions in the ALE in the benchmark work (Test I in Table 1) is used as the test input, and the reported mutations are used as the test output. The models were trained and tested ten times, and the averaged AUC and AUPRC values are illustrated in Table 5. Results show that our approach outperformed the benchmark approach for all models, except for the AUPRC of the SVC model. The largest improvement was observed for the ensemble model, with an increase of 8.74% for the AUC. The *p*-values reveal a statistical difference between both approaches for all but the AUPRC of the ANN model. The ROC and PR curves for one of the simulation runs are shown in Fig. 6. The ROC shown in Fig. 6a shows that a higher true positive rate can be achieved at lower false positive rates by our approach for most of the decision thresholds available. On the other hand, both approaches shown in Fig. 6b indicate similar low precision values for most values of the recall. It is worth noting that though the precision values are low, they are orders of magnitude greater than the baseline of 0.0083. This applies also to the AUPRC values for the benchmark ALE shown in Table 5.

The final test incorporated four other ALE experiments from the literature. As in the first test, the test inputs were set to the exposure conditions of each of the ALEs, and the test results were set to the observed mutations. The ROC and PR curves of this test, evaluated over the five ALEs, are depicted in Fig. 7, and the corresponding AUC and AUPRC are displayed in Table 5. As in the test on the benchmark ALE, our approach showed a better performance for all models, except for the AUPRC of the SVC model. The largest improvement was still observed for the ensemble model,

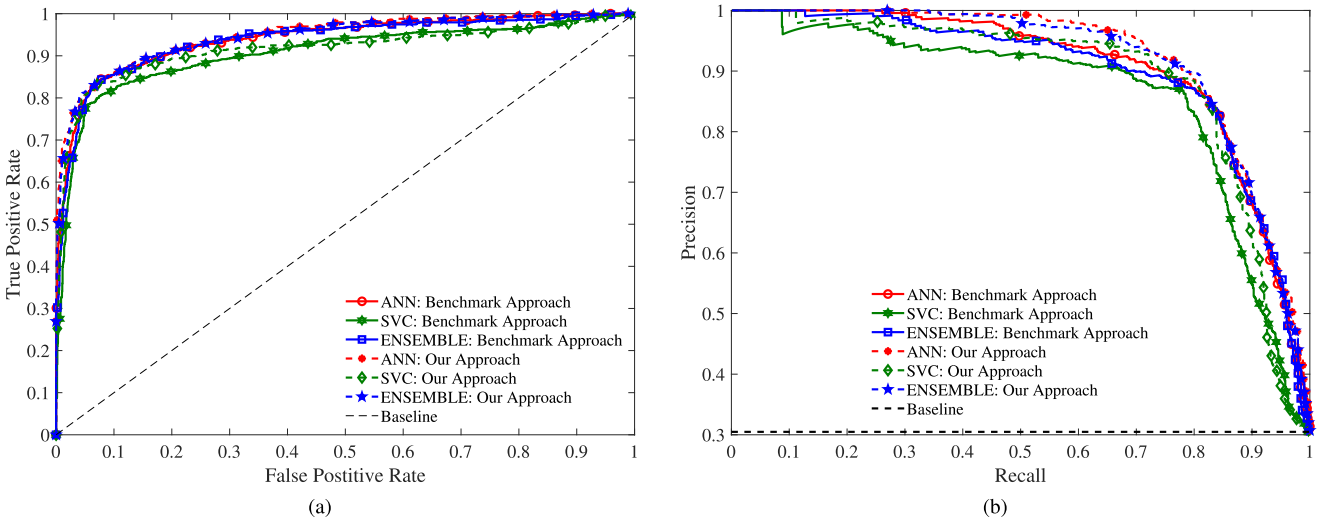

FIGURE 5. Performance curves for validation of models on selected protein-coding genes in dataset: Compared with Figure 4, both (a) ROC curve and (b) PR curve confirm that more mutation data would improve performance, with our approach still showing improvement over benchmark.

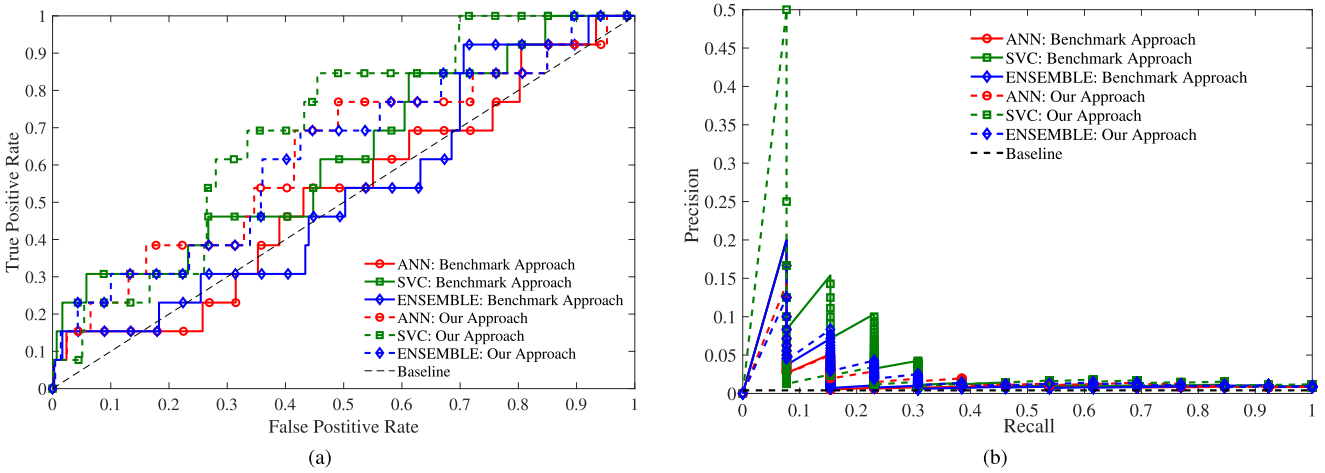

FIGURE 6. Performance curves for Test 1 set. Strain and exposure conditions in laboratory test of benchmark was used as input, and mutations observed in laboratory test were used to evaluate performance of models: (a) ROC curve shows that for majority of false positive rates, our approach achieved higher true positive rates. (b) PR curve reveals drop in precision for values of recall greater than 0.3 in both approaches.

with an increase of 3.24% in the AUC. The p-values showed that both approaches were statistically different in all but both the AUC and AUPRC for the SVC.

From the series of tests, it can be inferred that the cofunctional relationships among the protein-coding genes appear to improve the prediction performance of both the ANN and ensemble models. This improvement was noticed both in the validation results and test results. The SVC was not consistent in performance, with our approach showing improvement in only the AUC and the AUPRC in the validation and test results, respectively.

E. BIOLOGICAL INSIGHTS FROM PERFORMANCE AT GENE LEVEL

All the previously reported AUCs, AUPRCs, ROC curves, and PR curves were evaluated at the entire protein coding gene level. That is, predictions of all the individual genes were concatenated, and then evaluated with a concatenation of the true state of all the genes. To gain some biological insight from our study, we look closely at the impact of incorporating the cofunctional genes on the mutation prediction of each of the protein coding genes of *E. coli*. From the ten validation tests, the average difference between the AUCs and AUPRCs of both approaches was obtained for each gene. We only considered the validation results in this analysis, as the test data did not have mutations for all the genes, making it impossible to satisfy the requirement that the genes have both mutated and non-mutated states in the evaluation of the AUCs and AUPRCs.

We divided the protein-coding genes, based on the impact of incorporating the relationship among the cofunctional genes, into four classes: class 1 genes had an increase in AUC, class 2 genes showed a decrease in AUC, class 3 genes

TABLE 5. Results of test on ALE(s).

Model	Metric	Average Performance over Ten Tests (mean ± std)		*P*-value
		Benchmark Approach	Our Approach	
Test on Benchmark ALE				
ANN	AUC	0.59 ± 0.06	$\mathbf{0.63 \pm 0.02}$	0.028
	AUPRC	0.03 ± 0.00	$\mathbf{0.03 \pm 0.01}$	0.281
SVC	AUC	0.65 ± 0.04	$\mathbf{0.70 \pm 0.01}$	0.001
	AUPRC	$\mathbf{0.05 \pm 0.01}$	0.04 ± 0.00	0.027
Ensemble	AUC	0.62 ± 0.06	$\mathbf{0.67 \pm 0.02}$	0.006
	AUPRC	0.03 ± 0.00	$\mathbf{0.04 \pm 0.01}$	0.009
Test on Five ALEs				
ANN	AUC	0.56 ± 0.03	$\mathbf{0.57 \pm 0.01}$	0.073
	AUPRC	0.00 ± 0.00	$\mathbf{0.01 \pm 0.00}$	0.001
SVC	AUC	0.64 ± 0.02	$\mathbf{0.64 \pm 0.01}$	0.471
	AUPRC	$\mathbf{0.01 \pm 0.00}$	0.01 ± 0.00	0.241
Ensemble	AUC	0.59 ± 0.02	$\mathbf{0.61 \pm 0.01}$	0.019
	AUPRC	0.01 ± 0.00	$\mathbf{0.01 \pm 0.00}$	0.007

Higher AUC and AUPRC of approaches are written in bold letters.

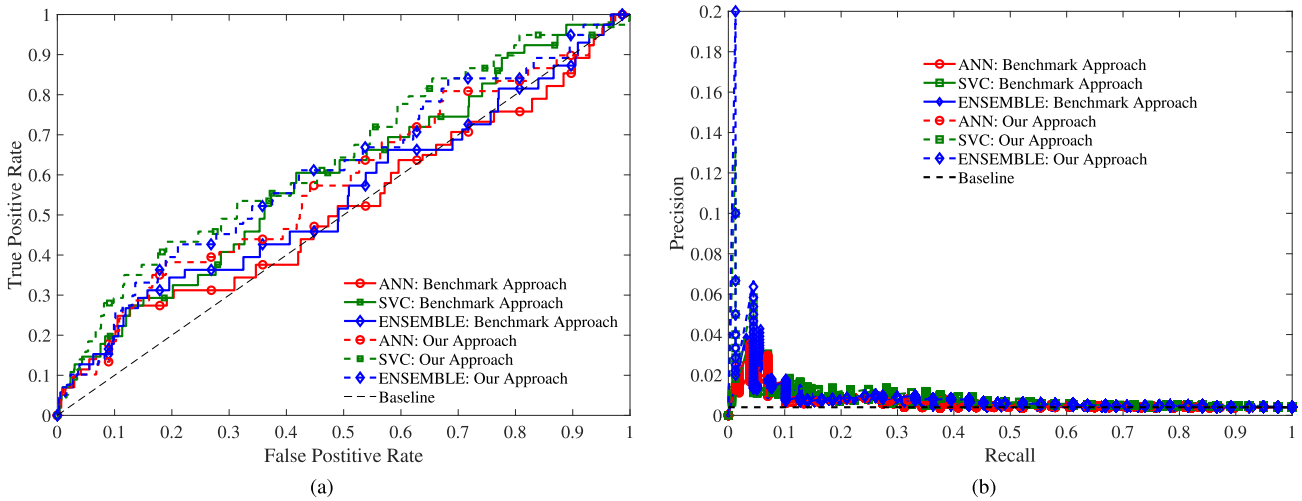

FIGURE 7. Performance curves for Test 2 set. Strain and exposure conditions reported in benchmark and four other identified studies were used as input, and mutations observed in each laboratory test were used to evaluate performance of models. (a) ROC curve shows that for majority of false positive rates, our approach achieved higher true positive rates. (b) PR curve still reveals low precision for both approaches for most values of recall.

exhibited an increase in AUPRC, and class 4 indicated a decrease in AUPRC. A list of all the genes and the values of impact on the prediction performance is provided in supplementary data 6. To investigate the impact of our approach on the predictions of the individual genes, we focused on the 50 genes with the most improved prediction performance (in class 1 and class 3) and the 50 genes with the most degraded performance (in class 2 and class 3). We then performed gene ontology enrichment analysis on the four sets of genes and highlighted the top cellular components, biological processes, and molecular functions enriched in each set of genes. We used the database for annotation, visualization, and integrated discovery (DAVID) [43] for the analysis, and set a *p*-value of less than 0.1 as the cut-off.

The top GO cellular component in class 1 is an ATP-binding cassette transporter complex (5), i.e. five genes identified in the class. The most common GO biological process of class 1 genes was DNA replication (3), compared to RNA modification (2) for class 2 genes. For the GO molecular function, class 1 genes were involved in unfolded protein binding (3) and copper ion binding (3), and class 2 genes were involved in transcription factor activity/sequence-specific DNA binding (7) and RNA binding (4). For class 3, the most common biological process and molecular function were transport related, involving both proton (4) and amino acids (4), and the GO cellular component was primarily the plasma membrane (21). In class 4, the GO biological processes were tRNA processing (3) and ferrous iron transport (2), the cellular component identified was a type II protein secretion system complex (2), and the molecular function was RNA binding (4).

We believe that this GO enrichment analysis is a first step towards identifying cellular components, molecular functions, and biological processes where the genetic mutation of

genes may follow a pattern that can be learned by machine learning models. The application of machine models to predict different biological phenomena may not offer direct biological explanations or knowledge [18], for example, why the sets of identified genes with their properties had such impacts. However, this method holds promise for providing frameworks with the potential for assisting with informed decisions in medicine and biotechnology.

VII. CONCLUSION

In this article, we have investigated the impact of the cofunctional relationship between protein-coding genes on predicting mutations. Specifically, for each of the 1, 561 protein-coding genes, we used their cofunctional relationships in our proposed feature-selection algorithm to choose the most relevant exposure-condition features. Then, we extended the selected features with the mutation state of the cofunctional genes and trained three machine learning models with the base dataset and the extended dataset.

Validation results, averaged over ten runs, show that incorporating the cofunctional relations into the training models improves both the area under the receiver operating characteristic curve and the area under the precision-recall curve. Compared with the performance across all 1, 561 protein-coding genes, there was an improvement in the AUC and AUPRC of both approaches for the validation on 112 protein-coding genes that have ten or more mutation points in the dataset. Also, our approach still slightly outperformed the benchmark approach in the validation on this set of protein-coding genes. This indicates that as more data, and hence mutations points, become available from ongoing research, models can be better trained with better prediction performance.

To observe the out-of-sample performance of our approach, we identified adaptive laboratory evolution experiments from the literature for testing. Results show that our approach achieved a boost in performance for both AUC and AUPRC values of the ANN and ensemble models, but the AUPRC of the SVC was just about constant. The maximum increase in performance was a 8.74% increase in the AUC value of the ensemble model. In general, results confirm that there is evidence that our approach shows some improvement in mutation prediction, and that the availability of more training and test data can go a long way in the development of optimal models.

Finally, we classified the genes based on the impact of the cofunctional genes on the prediction performance, and we carried out GO enrichment analysis on the genes in each class. We highlighted the cellular components, biological processes, and molecular functions that are enriched in the top genes in each class. This provides a starting point in identifying genes for which functions and interactions with similar genes can enhance the prediction of mutation states.

This investigation points to the possibility of better prediction performance by using cofunctional interactions in the selection of features and as additional training features. As future work, in order to overcome the assumption in this work that the mutation state of the cofunctional genes are known a priori, we are looking at ways of learning the state from the training data. We are currently investigating recursive deep neural network models and learned-features distance-based techniques to estimate the mutation states of cofunctional genes.

REFERENCES

[1] O. Tenaillon, J. E. Barrick, N. Ribeck, D. E. Deatherage, J. L. Blanchard, A. Dasgupta, G. C. Wu, S. Wielgoss, S. Cruveiller, C. Médigue, D. Schneider, and R. E. Lenski, "Tempo and mode of genome evolution in a 50,000-generation experiment," *Nature*, vol. 536, no. 7615, pp. 165–170, Aug. 2016.

[2] X. Wang, V. Zorraquino, M. Kim, A. Tsoukalas, and I. Tagkopoulos, "Predicting the evolution of escherichia coli by a data-driven approach," *Nature Commun.*, vol. 9, no. 1, p. 3562, Dec. 2018.

[3] M. Lässig, V. Mustonen, and A. M. Walczak, "Predicting evolution," *Nature Ecol. Evol.*, vol. 1, no. 3, pp. 1–9, 2017.

[4] Y. Wang, S. Yang, J. Zhao, W. Du, Y. Liang, C. Wang, F. Zhou, Y. Tian, and Q. Ma, "Using machine learning to measure relatedness between genes: A multi-features model," *Sci. Rep.*, vol. 9, no. 1, pp. 1–15, Dec. 2019.

[5] M.-N. Wang, Z.-H. You, L.-P. Li, L. Wong, Z.-H. Chen, and C.-Z. Gan, "GNMFLMI: Graph regularized nonnegative matrix factorization for predicting LncRNA-MiRNA interactions," *IEEE Access*, vol. 8, pp. 37578–37588, 2020.

[6] M. Kim, V. Zorraquino, and I. Tagkopoulos, "Microbial forensics: Predicting phenotypic characteristics and environmental conditions from large-scale gene expression profiles," *PLoS Comput. Biol.*, vol. 11, no. 3, p. e1004127, pp. 1–21, 2015.

[7] M. O. Sommer, C. Munck, R. V. Toft-Kehler, and D. I. Andersson, "Prediction of antibiotic resistance: Time for a new preclinical paradigm?" *Nature Rev. Microbiol.*, vol. 15, no. 11, pp. 689–697, 2017.

[8] J. T. Kwak and S. M. Hewitt, "Nuclear architecture analysis of prostate cancer via convolutional neural networks," *IEEE Access*, vol. 5, pp. 18526–18533, 2017.

[9] A. Ghulam, X. Lei, M. Guo, and C. Bian, "Disease-pathway association prediction based on random walks with restart and PageRank," *IEEE Access*, vol. 8, pp. 72021–72038, 2020.

[10] A. Chaddad, C. Desrosiers, B. Abdulkarim, and T. Niazi, "Predicting the gene status and survival outcome of lower grade glioma patients with multimodal MRI features," *IEEE Access*, vol. 7, pp. 75976–75984, 2019.

[11] R. Qin, Z. Wang, K. Qiao, J. Hai, L. Jiang, J. Chen, X. Pei, D. Shi, and B. Yan, "Multi-type interdependent feature analysis based on hybrid neural networks for computer-aided diagnosis of epidermal growth factor receptor mutations," *IEEE Access*, vol. 8, pp. 38517–38527, 2020.

[12] B. Alipanahi, A. Delong, M. T. Weirauch, and B. J. Frey, "Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning," *Nature Biotechnol.*, vol. 33, no. 8, pp. 831–839, 2015.

[13] H. Zeng, M. D. Edwards, G. Liu, and D. K. Gifford, "Convolutional neural network architectures for predicting DNA–protein binding," *Bioinformatics*, vol. 32, no. 12, pp. i121–i127, Jun. 2016.

[14] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PLoS ONE*, vol. 10, no. 3, pp. 1–21, 2015, Art. no. e0118432.

[15] J.-J. Tu, L. Ou-Yang, X. Hu, and X.-F. Zhang, "Inferring gene network rewiring by combining gene expression and gene mutation data," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 16, no. 3, pp. 1042–1048, May 2019.

[16] M. Bersanelli, E. Mosca, D. Remondini, E. Giampieri, C. Sala, G. Castellani, and L. Milanesi, "Methods for the integration of multi-omics data: Mathematical aspects," *BMC Bioinf.*, vol. 17, no. S2, pp. 167–177, Dec. 2016.

[17] P. V. Phaneuf, D. Gosting, B. O. Palsson, and A. M. Feist, "ALEdb 1.0: A database of mutations from adaptive laboratory evolution experimentation," *Nucleic Acids Res.*, vol. 47, no. D1, pp. D1164–D1171, Jan. 2019.

[18] B. Papp, R. A. Notebaart, and C. Pál, "Systems-biology approaches for predicting genomic evolution," *Nature Rev. Genet.*, vol. 12, no. 9, pp. 591–602, Sep. 2011.

[19] C. H. Rodrigues, D. E. Pires, and D. B. Ascher, "DynaMut: Predicting the impact of mutations on protein conformation, flexibility and stability," *Nucleic Acids Res.*, vol. 46, no. W1, pp. W350–W355, Jul. 2018.
[20] D. E. V. Pires, D. B. Ascher, and T. L. Blundell, "MCSM: Predicting the effects of mutations in proteins using graph-based signatures," *Bioinformatics*, vol. 30, no. 3, pp. 335–342, Feb. 2014.
[21] L. Sundaram, H. Gao, S. R. Padigepati, J. F. McRae, Y. Li, J. A. Kosmicki, N. Fritzilas, J. Hakenberg, A. Dutta, J. Shon, J. Xu, S. Batzoglou, X. Li, and K. K.-H. Farh, "Predicting the clinical impact of human mutation with deep neural networks," *Nature Genet.*, vol. 50, no. 8, pp. 1161–1170, Aug. 2018.
[22] A. P. Pandurangan, B. Ochoa-Montaño, D. B. Ascher, and T. L. Blundell, "SDM: A server for predicting effects of mutations on protein stability," *Nucleic Acids Res.*, vol. 45, no. W1, pp. W229–W235, Jul. 2017.
[23] R. M. Bush, "Predicting adaptive evolution," *Nature Rev. Genet.*, vol. 2, no. 5, pp. 387–392, May 2001.
[24] P. A. Lind, E. Libby, J. Herzog, and P. B. Rainey, "Predicting mutational routes to new adaptive phenotypes," *eLife*, vol. 8, pp. 1–31, Jan. 2019, Art. no. e38822.
[25] H. Kim, J. E. Shim, J. Shin, and I. Lee, "EcoliNet: A database of cofunctional gene network for escherichia coli," *Database*, vol. 2015, pp. 1–8, Jan. 2015, Art. no. bav001.
[26] T. E. Sandberg, C. J. Lloyd, B. O. Palsson, and A. M. Feist, "Laboratory evolution to alternating substrate environments yields distinct phenotypic and genetic adaptive strategies," *Appl. Environ. Microbiology*, vol. 83, no. 13, Jul. 2017, Art. no. e00410.
[27] D. E. Deatherage, J. L. Kepner, A. F. Bennett, R. E. Lenski, and J. E. Barrick, "Specificity of genome evolution in experimental populations of Escherichia coli evolved at different temperatures," *Proc. Nat. Acad. Sci. USA*, vol. 114, no. 10, pp. E1904–E1912, 2017.
[28] B. Du, C. A. Olson, A. V. Sastry, X. Fang, P. V. Phaneuf, K. Chen, M. Wu, R. Szubin, S. Xu, Y. Gao, Y. Hefner, A. M. Feist, and B. O. Palsson, "Adaptive laboratory evolution of escherichia coli under acid stress," *Microbiology*, vol. 166, no. 2, pp. 141–148, Feb. 2020.
[29] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794.
[30] F. Chollet. (2015). *Keras, GitHub Repository*. Accessed: Oct. 20, 2019. [Online]. Available: https://github.com/fchollet/keras
[31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.
[32] T. G. Dietterich, "Ensemble learning," in *The Handbook of Brain Theory and Neural Networks*, vol. 2. 2002, pp. 110–125.
[33] H. Lee, E. Popodi, H. Tang, and P. L. Foster, "Rate and molecular spectrum of spontaneous mutations in the bacterium escherichia coli as determined by whole-genome sequencing," *Proc. Nat. Acad. Sci. USA*, vol. 109, no. 41, pp. E2774–E2783, Oct. 2012.
[34] J. L. Martinez and F. Baquero, "Mutation frequencies and antibiotic resistance," *Antimicrobial Agents Chemotherapy*, vol. 44, no. 7, pp. 1771–1777, Jul. 2000.
[35] X.-L. Chu, B.-W. Zhang, Q.-G. Zhang, B.-R. Zhu, K. Lin, and D.-Y. Zhang, "Temperature responses of mutation rate and mutational spectrum in an escherichia coli strain and the correlation with metabolic rate," *BMC Evol. Biol.*, vol. 18, no. 1, pp. 1–8, Dec. 2018.
[36] S. Wang, J. Ma, and J. Xu, "AUCpreD: Proteome-level protein disorder prediction by AUC-maximized deep convolutional neural fields," *Bioinformatics*, vol. 32, no. 17, pp. i672–i679, Sep. 2016.
[37] S. Euh, H. Lee, D. Kim, and D. Hwang, "Comparative analysis of low-dimensional features and tree-based ensembles for malware detection systems," *IEEE Access*, vol. 8, pp. 76796–76808, 2020.
[38] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves," in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 233–240.
[39] Y. Long, R. Xiang, Q. Lu, D. Xiong, C.-R. Huang, C. Bi, and M. Li, "Learning heterogeneous network embedding from text and links," *IEEE Access*, vol. 6, pp. 55850–55860, 2018.
[40] A. Calabria, S. Beretta, I. Merelli, G. Spinozzi, S. Brasca, Y. Pirola, F. Benedicenti, E. Tenderini, P. Bonizzoni, L. Milanesi, and E. Montini, "Γ-TRIS: A graph-algorithm for comprehensive identification of vector genomic insertion sites," *Bioinformatics*, vol. 36, no. 5, pp. 1622–1624, 2020.
[41] M. Ojala and G. C. Garriga, "Permutation tests for studying classifier performance," *J. Mach. Learn. Res.*, vol. 11, pp. 1833–1863, Jun. 2010.
[42] C. Georgescu and J. D. Wren, "Algorithmic identification of discrepancies between published ratios and their reported confidence intervals and P-values," *Bioinformatics*, vol. 34, no. 10, pp. 1758–1766, May 2018.
[43] D. W. Huang, B. T. Sherman, and R. A. Lempicki, "Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources," *Nature Protocols*, vol. 4, no. 1, pp. 44–57, Jan. 2009.

MICHAEL OKWORI (Graduate Student Member, IEEE) received the B.Eng. degree in electrical and computer engineering and the M.Eng. degree in communication engineering from the Federal University of Technology Minna (FUT Minna), Nigeria, in 2007 and 2014, respectively. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering and Computer Science, Wichita State University, KS, USA. He has been a Lecturer with FUT Minna, since 2010. His research interests include mobility management in IP networks, microscale sensor networks, applications of machine and deep learning, and bioinformatics. In 2017, he was a recipient of the Petroleum Technology Development Fund (PTDF) Ph.D. Scholarship Award, Nigeria.

ALI ESLAMI (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Massachusetts, Amherst, in 2013. He was a Postdoctoral Research Fellow with Texas A&M University, College Station, TX, USA, from March 2013 to April 2015. From August 2014 to June 2015, he was a Visiting Research Scholar of the information initiative at Duke (iiD). He is currently an Assistant Professor of electrical engineering and computer science with Wichita State University, Wichita, KS, USA. His current research interests include nanocommunications, applications of coding theory in biology, resilient design of cyber-physical systems, fault-tolerant quantum computing, and big-data storage systems. He is a member of the IEEE Communications and Computer Societies. From 2016 to 2017, he was a recipient of the Wichita State's Young Faculty Risk Taker Award. He has served as the Session Chair for several IEEE conferences and workshops, as well as a Reviewer for numerous IEEE journals. He has also served on several NSF review panels.