SMU Data Science Review

Volume 5 | Number 1 Article 10

2021

Analyzing Empirical Quality Metrics of Deep Learning Models for Antimicrobial Resistance

Huy H. Nguyen
Southern Methodist University, hoangnguyen@smu.edu

Sanjay Pillay
Southern Methodist University, spillay@smu.edu

Allison Roderick
Southern Methodist University, allroderick@gmail.com

Hao Wang
Southern Methodist University, wangmichael@smu.edu

John Santerre
SMU, jsanterre@smu.edu

Analyzing Empirical Quality Metrics of Deep Learning Models for Antimicrobial Resistance

H.H. Nguyen, S. Pillay, A. Roderick, H. Wang, J. Santerre

Master of Science in Data Science, Southern Methodist University,
Dallas, TX 75275 USA
{hoangnguyen, spillay, aroderick, wangmichael, jsanterre}@smu.edu

Abstract. Antimicrobial Resistance (AMR) is a growing concern in the medical field. Over-prescription of antibiotics as well as bacterial mutations have caused some once lifesaving drugs to become ineffective against bacteria. However, the problem of AMR might be addressed using Machine Learning (ML) thanks to increased availability of genomic data and large computing resources. The Pathosystems Resource Integration Center (PATRIC) has genomic data of various bacterial genera with sample isolates that are either resistant or susceptible to certain antibiotics. Past research has used this database to use ML algorithms to model AMR with successful results, including accuracies over 80%. To better aid future biologists and healthcare workers who may need a predictive model without the benefit of thousands of bacteria samples, this paper explores quantifying the empirical quality of some machine learning models—that is, quantifying how well a model will perform without prior knowledge of how the model performed on a training dataset. WeightWatcher is a Python package that offers various algorithms to measure model quality. This research uses the empirical quality metrics that WeightWatcher introduces for Deep Neural Network (DNN) models to evaluate AMR models, even on datasets based on small sample sizes of bacterial strains. The use of ML in AMR and pharmacogenetic research can help lead to increased efficacy of antibiotic treatments by predicting whether a strain of bacteria will be resistant or susceptible to an antibiotic.

1 Introduction

Using antibiotics to treat diseases caused by bacteria has become an important and widespread medical tool since the advent of Penicillin in 1928. Antibiotic medications are used in various medical treatments and to prevent infections. However, there are also drawbacks related to the use of antibiotics. The rising use of antibiotics has caused bacteria to develop the ability to mutate and become resistant to antibiotic use, termed antimicrobial resistance or AMR. AMR can occur via an innate resistance in bacteria to antibiotics, genetic mutations, or inter-species resistance acquisition [1]. According to the U.S. Centers of Disease Control and Prevention (CDC), potentially excessive antibiotics are introduced into human environments via medical treatments for humans

or animals; crop pesticides; human, animal, and pharmaceutical manufacturing waste; and animal feed and drinking water [2]. This pervasive presence of antibiotics has accelerated AMR, increasing its threat to the entire globe. In the U.S. alone, over 2.8 million people face antibiotic-resistant diseases each year, resulting in over 35,000 deaths [2].

The CDC describes AMR as a threat to "progress in health care, food production, and life expectancy" [2]. While humans and animals are at risk for AMR-related infections, hospital patients are often at greater risk. Certain procedures such as joint replacements, organ transplants, and cancer therapy have risks of infection that can be caused by antimicrobial-resistant bacteria. Without effective antibiotics to combat these infections, mortality rates could increase. In addition to hospital settings, bacteria can spread easily by multiple methods: close person-to-person contact, airborne transmission, contaminated water or surfaces, animals, or sexual contact [2].

The CDC offers some instructions for combating AMR on the individual level, including washing hands, especially after handling food or animals; getting vaccinated, particularly when traveling to other countries where vaccine recommendations may be different from those in the U.S.; using antibiotics wisely; and preventing STDs. To healthcare providers, the CDC recommends following infection prevention and control protocol, such as screening at-risk patients, asking patients if they have been in risky locations, administering vaccines, monitoring global outbreaks of infections, and so on. Additionally, healthcare providers can take important steps to improving antibiotic prescription by following CDC guidelines [2].

One bacteria genus that is resistant to multiple drugs is *Acinetobacter*. A sample of *Acinetobacter* was first discovered in 1911 by Dutch microbiologist Beijerinck, who described it under the name *Micrococcus calco-aceticus* [3]. Throughout the 20th century, biologists further researched and refined scientific understanding of the genus. Various *Acinetobacter* species can be found in human skin flora, soil, water, and other common bacterial locations. However, the locations of *A. baumannii*, one of the most researched *Acinetobacter* species, are more troubling: hospital equipment, surfaces, food, and staff [4]. These bacteria pose elevated risk for patients in Intensive Care Units (ICU), acute care, or long-term care settings [5, 6]. Additionally, *Acinetobacter* can survive on both dry and wet surfaces for long periods of time [6].

Not only is the prevalence of *Acinetobacter* a cause for concern, but also its resistance to antibiotics is rapidly growing. In its first report on Antibiotic Resistance Threats in 2013, the CDC listed multidrug-resistant *Acinetobacter* as a serious threat. The CDC increased the threat level in 2019, and this updated report focused on Carbapenem-resistant *Acinetobacter* as an urgent concern due to the lack of antibiotics that are either currently efficacious or in development for future use [2]. This research will use create a model to predict antibiotic resistance in samples of *Acinetobacter*, publicly available on the Pathosystems Resource Integration Center (PATRIC), discussed further in Section 3.1.

One method for solving the problem of AMR is by utilizing modern machine learning algorithms to predict whether a strain of bacteria will be resistant or susceptible to an antibiotic. There is much published research surrounding modeling AMR using highly accurate Machine Learning (ML) models [7, 8, 9, 10, 11, 12, 13, 14]. Generally, these research articles will take bacterial genomic data—that is, strands of bacterial

deoxyribonucleic acid (DNA) in the form of reads—and model the processed data using models such as AdaBoost, Random Forest (RF), or Support Vector Machine (SVM). Section 2.1 of this paper presents previous work in the field of machine learning for AMR.

However, the results of these previous ML models, while very promising, are dependent on the data used to test the models. This paper will present a method of evaluating AMR prediction models independent of test data. This paper presents a Deep Neural Network (DNN) to model AMR and then analyzes the weight layer matrices of the DNN using the Python package WeightWatcher [15, 16, 17, 18, 19, 20, 21]. This package presents methods for determining the empirical quality of models. Section 2.2 presents an overview how WeightWatcher works. Section 3.4 dives deeper into the mathematical basis of WeightWatcher. Section 4.2 discusses the results of WeightWatcher applied to a DNN trained on *Acinetobacter* genomic data.

The best cure for AMR—prevention of bacterial diseases and reduction of the unnecessary prescription of antibiotics—is not always possible, so identifying the genotypes in bacterial DNA that are resistant to various drugs is critical in developing new techniques for fighting AMR. This paper discusses this past research in Section 2 and implements models of its own in Section 3. However, the goal of this research is not to create a new model for predicting AMR, but rather to introduce a method by which biologists, healthcare professionals, and biostatisticians can evaluate the performance of a pre-trained AMR model with limited amount of genomic data samples and evaluate the model for performance using WeightWatcher.

2 Literature Review

2.1 Machine Learning and Antimicrobial Resistance

Machine Learning methods have been used to create models that can predict with high accuracy whether a strain of bacteria will be resistant or susceptible to a specific antibiotic. The section summarizes past research with various models, bacterial genera, and methods of simplifying or extending the problems that occur with this undertaking.

When applying ML to AMR, Santerre et al. discusses different models of four different strains of bacteria [12]. The models showed that performance increased as the length of k-mers increased for some modeling techniques (AdaBoost), but not others (Random Forest, Lasso). Additionally, the authors presented three potential objectives for continued application of machine learning to the problem of AMR: maximize classification accuracy, maximize generalization accuracy, and aggregate feature selection.

Davis et al. considered the results of the AdaBoost algorithm on 31-mers of different samples from *Acinetobacter*, *Staphylococcus*, and *Streptococcus* strains, achieving accuracies from 88% to 99% [7]. *Mycobacterium tuberculosis* was also modeled with

an accuracy of 71% to 88%. AdaBoost was chosen for simplicity, and later research has improved on these results using other machine learning algorithms.

Lingle and Santerre used various ML algorithms—Random Forest, Naïve Bayes, and Support Vector Machine—on *Neisseria gonorrhea* to achieve accuracy above 90% with 10-mers. This high accuracy was particularly notable due to the model features containing smaller k-mers than those used by the AdaBoost algorithm in previous work, showing that highly accurate models could be obtained with smaller values of k [8].

Jha et al. present Deep Learning (DL) models based on genomic k-mers of bacteria from the National Center for Biotechnology Information (NCBI) to predict AMR [22]. They tie each k-mer to the gene in which is appears most frequently. They built their model in Java and RapidMiner, and trained the data using cross-validation (CV).

One barrier to efficient AMR models is the large feature space associated with the models. As this paper discusses more in depth in Section 3.1, k-merization of DNA contigs can result in tens of thousands of k-mers, each a feature in a model. A method of lossless string compression to reduce the size of k-mers, thus reducing the feature space, was presented by Partee, et al [10]. ML algorithms performed on this compressed representation were able to produce similarly high accuracies to Lingle and Santerre [8].

A promising solution to the problem of maximizing the generalization of AMR models was presented by Jacob Durrant and colleagues at the University of Pittsburgh, Pennsylvania [9]. In this article, Marchant reports on the use of Artificial Intelligence (AI) in fighting AMR. The goal of the approach is to create new antibiotics that will be effective against multiple bacteria. Durrant and colleagues used a neural network model to predict how the antibiotic will behave, rather than focusing on the structures of the molecules.

Machine Learning has been applied to gene-related research outside the scope of bacteria and AMR. Sealfon et al. look at kidney transcriptomics, proteomics, metabolomics, genome sequencing, and medical data can be used with ML in order to perform genotype-phenotype analysis [13]. The data used in this research shares similarities to the data this paper uses: the data is high-dimensional but could have the number of features reduced due to correlation among features.

Onate et al. used k-mer analysis for studying the quality control of microbiota metagenomics [23]. The authors used k-mer method for serial dilutions of gut microbiota metagenomic datasets and demonstrate that the k-mer analysis can identify metagenomes of low quality. The k-mer distribution analysis also predicted low gene mapping efficiency. Their method was to collect fecal samples from 30 human donors, process the samples, extract genomic DNA, and then construct a metagenomic library. Finally, they analyzed the k-merized data by using an in-house developed C++ software.

2.2 WeightWatcher

In a very short time, Deep Learning has become one of the hottest topics in the machine learning field. DL is a technique that is widely useful in solving and automating problems in various areas such as biology, healthcare, robotics, and countless others

[24, 25, 26, 27]. A Deep Neural Network, part of the DL domain, is defined as "an Artificial Neural Network (ANN) with multiple layers between the input and output layers" and is built upon hypotheses and models to mimic human brains in order to solve complex machine learning tasks [24, 28].

Martin introduced a package used extensively in this research called WeightWatcher (WW) [17]. It is a tool that supports PyTorch, Keras, and Hugging Face models for computing empirical quality metrics in ML models. One way in which this package has been used has been in improving the quality of Natural Language Processing (NLP) models in fake test studies. NLP is defined as "an interdisciplinary area of research aimed at making machines understand and process human languages," combining "perceptual, behavioral, and communicational techniques" [29, 30]. Martin also suggested the WW tool can also be used to compare models trained using Auto-ML or give insights in choosing the best clusters in unsupervised clustering machine learning models. This research uses WW to evaluate if the sample data is sufficient or more data is needed for a model or sequence of models in AMR research.

The idea proposed by Martin that this paper intends to adapt to the problem of AMR is, in Martin's words, "to predict the generalization accuracy of a DNN solely by looking at the trained weight matrices, without looking at the test data" [16]. In many cases, researchers cannot test problems to get results for evaluation. For instance, in the medical field, physicians or biologists cannot use some experimental treatment on patients to evaluate its effectiveness due to safety and ethical reasons. The mathematical background behind the WW package, focusing on Random Matrix Theory (RMT), is discussed further in Section 3.3 of this paper and in past literature [15, 16, 17, 19, 20, 21, 31, 32, 33]. In one such work, Martin, Peng, and Mahoney found that Power Law (PL) based metrics—implemented in WeightWatcher—can distinguish between not only between good and bad models, but also between good and better or best models [20]. In the following sections, their research is applied to the problem of AMR.

3 Methods

This research created multiple models on antibiotic-resistant *Acinetobacter* and then analyzed the empirical quality of the DNN model using WeightWatcher. However, before discussing the results of that analysis, this section first presents the data, processes, and tools used to achieve the results, which are presented in Section 4.

3.1 Data and K-merization

The data used in this research comes from the Pathosystems Resource Integration Center or PATRIC. PATRIC is the Bacterial Bioinformatics Resource Center funded by the National Institute of Allergy and Infectious Diseases (NIAID). PATRIC contains genomic data, including annotations created by Rapid Annotation using Subsystems Technology (RAST), for a variety of bacteria, including those on the NIAID priority watch list [1].

A variety of data, analysis tools, and studies on genetic determinants of AMR can be found at the online[1]. AMR phenotype data are collected from published studies and PATRIC collaborators using antimicrobial susceptibility testing methods (AST) and predicted AMR phenotypes also provided by PATRIC using ML classifiers [1].

This section of PATRIC contains FASTA files with DNA isolates of 12 different bacteria genera. The DNA is represented as contigs. Contigs are assembled, non-overlapping DNA sequences. Each isolate is labeled either "Resistant" or "Susceptible" to a certain antibiotic for that bacteria.

The bacteria genus this research uses to model AMR is *Acinetobacter.* This bacterium had a sufficient and balanced number of samples of Susceptible and Resistant strains. The genomic data then underwent a k-merization into a Python NumPy array to be used as features for classifying the genetic strain either as susceptible or resistant using DNN models and WeightWatcher to analyze the models.

In order to use the genomic information as features for a ML model, the DNA is transformed into k-mers, which are substrings of the contigs of length k. The features of the potential models will consist of all unique k-mers for antibiotic-resistant and susceptible isolates for the bacteria of interest and the number of times each k-mer appeared in the isolate. These k-mers are overlapping and can have length k as large as 50. However, as k increases, the total number of unique k-mers increases, leading to prohibitively large datasets [12]. This analysis focuses on k-mers of length k=10, also called 10-mers, examples of which can be found in Table 1.

Table 1. Examples of k-mers of length k=10 from sample DNA sequence CCAGCTCATTATTCTA.

Example	k-mer bolded in sequence	k-mer (10-mer)
First unique k-mer	**TCAGCTCATT**ATTCTA	TCAGCTCATT
Second unique k-mer	T**CAGCTCATTA**TTCTA	CAGCTCATTA
Third unique k-mer	TC**AGCTCATTAT**TCTA	AGCTCATTAT

Table 2. Example stages of overall volume reduction

Stage	Reduction	Susceptible Samples	Resistant Samples
0	0%	640	724
1	20%	512	579
2	40%	384	434
3	60%	256	289
4	80%	128	144

The antibiotic of *Acinetobacter* used in this research was Imipenem, which has 640 strains of susceptible and 724 strains of resistant bacteria for a baseline. When k-merized, the samples have over 1 million possible 10-mers, thus the feature space of this model will have over 1 million features. A PyTorch DNN model will be used to

[1] ftp://ftp.patricbrc.org/AMR_genome_sets/

evaluate and baseline the model's accuracy and weights. To test the quality of the model, the sample sizes are gradually reduced and then re-evaluated on the same model, as shown in Table 2. After every stage of reduction, WeightWatcher is used to measure and report on the model accuracy and the loss parameter values.

3.2 ManeFrame II

Due to the large size of the genomic data, all computations were run on ManeFrame II (M2), located at Southern Methodist University (SMU). M2 is a computational cluster used by faculty and researchers at SMU for high-performance computing (HPC). M2 offers various resources depending on job size and computational requirements.

Running on M2, the k-mers generation process used 36 cores and 1024 GB memory. This code used the ray package to multithread the process of k-merization. The k-mer process generated a NumPy array of approximately 1.4 GB when exported to a pickle file. The same process was followed to generate other k-mer files for all antibiotics for *Acinetobacter* contained in the PATRIC database.

3.3 WeightWatcher Methodology

As discussed in Section 2, WeightWatcher is a Python package used to evaluate models based on metrics that do not rely solely on a traditional training or test set. This package's utility as a method for measuring the empirical quality of models are at the core of this research. This section presents a brief overview of the mathematical basis of how WeightWatcher analyzes DNNs and evaluates quality and performance. Following that discussion, Section 4 will present these metrics for the DNN modeled on *Acinetobacter* genomic data.

3.3.1 Self-Regularization

A typical DNN with L layers and with activation functions $h_l()$ can be written in the form of equation (1) where W_L and b_L are layer weight matrices and biases respectively.

$$E_{DNN} = h_L(W_L \times h_{L-1}(W_{L-1} \times h_{L-2}(\dots) + b_{L-1}) + b_L) \quad (1)$$

There are different activation function types that can be considered in the models such as linear activation function, sigmoid function, tanh function, and ReLU. ReLU, or Rectified Linear Unit, is one of the most widely used activated functions, especially in hidden layers of NNs because it is faster to train than sigmoid and tanh functions [36].

The overview of this approach is to train the model on labeled data $\{d_i, y_i\} \in \mathcal{D}$, using the backpropagation approach, a widely used algorithm in training NNs for

supervised learning efficiently and by minimizing loss $\mathcal{L}$, as shown in equation (2) [37].

$$\min_{W_l,b_l} \mathcal{L}\left(\sum_i E_{DNN}(d_i) - y_i\right) \quad (2)$$

Per Martin and Mahoney, the energy landscape (E_{DNN}) changes at each epoch [19]. Some models with large numbers of weight layers can have multitudes of local minima or the possibility of overfitting. However, the solution WeightWatcher presents is adding more capacity and letting the model self-regularize. In other words, over-parameterization applied to avoid overtraining [38].

One approach is to modify (2) by adding an explicit regularization control parameter α in the following formula (3)

$$\min_{W_l,b_l} \mathcal{L}\left(\sum_i E_{DNN}(d_i) - y_i\right) + \alpha \sum_l \left|\left|W_l\right|\right| \quad (3)$$

where $||.||$ is some norm. There are different norm types for regularization problem depending on objective situations leaving different empirical signatures on the layer weight matrices W_L. In Ridge Regression or Tikhonov-Phillips regularization, the familiar optimization problem in L^2-norm is as follows

$$\min_{W_{ij}} \left|\left|\widehat{W}x - y\right|\right|_2^2 + \alpha \left|\left|\widehat{W}\right|\right|_2^2 \quad (4)$$

for $\widehat{W}x = y$ [39, 40]. Remark that it can be rewritten as

$$x = \left(\hat{X} + \alpha I\right)^{-1} \widehat{W}^T y \quad (5)$$

where $\hat{X} = \widehat{W}^T \widehat{W}$.

3.3.2 Empirical Spectral Density

The empirical spectral density (ESD) describes the distribution of eigenvalues of the correlation matrix $X = W^T W$ of a weight layer matrix W [19]. In general, less sophisticated models have weight layers with ESDs described by the Marchenko-Pastur (MP) distribution associated with Random Matrix Theory (RMT). However, considering that the goal of this research is to present a robust solution to predicting AMR with few to test samples, a more sophisticated DNN may exhibit heavy-tailed behavior in the ESDs. Heavy-tailed behavior occurs in both weight layers and the correlation matrices of those weight layers when the weight layers are elementwise highly correlated. This heavy-tailed distribution is described by the Pareto or Power Law (PL). Two different forms of the PL equation are shown in equations (6) and (7).

$$\rho(\lambda) = \lambda^{-\alpha} \quad (6)$$

$$\rho(\lambda) = \lambda^{-(\frac{1}{2}\mu+1)} \quad (7)$$

The equation (6), the distribution of eigenvalues λ of X, is given in a generic form. For this paper's purposes, the parameter α of this equation is an important indicator of model behavior. In general, well-trained DNNs have $2 < \alpha < 6$ [17]. Similarly, when PL is written as equation (7) if $0 < \mu < 2$, then the distribution is very heavy-tailed; if $2 < \mu < 4$, then the distribution is moderately heavy-tailed; and if $\mu > 4$, then the distribution is weakly heavy-tailed [19]. A more in-depth explanation of this behavior is presented in Table 3.

Table 3. Summary of MP, Spiked-Covariance, and heavy-tailed distributions.

Distribution	Finite *N* global shape of $\rho_N(\lambda)$	Bulk edges behavior	Maximum eigenvalue behavior $\lambda \approx \lambda_{max}$
Basic MP (Gaussian)	MP	Tracy Widom Law	No tail
Spiked-Covariance (Gaussian with low-rank perturbations)	MP with Gaussian spikes	Tracy Widom Law	Gaussian
Weakly heavy-tailed	MP with PL tail	Heavy-tailed	Heavy-tailed
Moderately heavy-tailed	PL$\sim \lambda^{-(a\mu+b)}$	No edge	Fréchet
Very heavy-tailed	PL$\sim \lambda^{-\left(\frac{1}{2}\mu+1\right)}$	No edge	Fréchet

A more heavy-tailed distribution may indicate a more sophisticated model. Generally, Martin has found that α is correlated with test accuracy when tuning a model or changing the number of layers [41]. Martin explains the reason for why the PL is a good model for the weight matrices of DNNs by likening it to complex systems in physics that have been known to display PLs.

3.3.3 Kolmogorov-Smirnov Distance

The Kolmogorov-Smirnov (KS) test is a nonparametric test used to compare one-dimensional probability distributions, generally a theoretical or reference cumulative distribution function (CDF) with an empirical cumulative distribution function [42]. The KS statistic quantifies the distance between the reference CDF and the empirical CDF.

WeightWatcher provides the KS statistic (or KS distance) to show the goodness of fit for the PL fit of the ESD to a given weight layer correlation matrix $\hat{X} = \widehat{W}^T\widehat{W}$. in a DNN. In general, a small KS distance indicates a good fit [17].

4 Results

4.1 Model Performance

To model resistance of *Acinetobacter* exposed to Imipenem, this researched created PyTorch DNNs from 2 to 10 hidden layers. However, this paper will focus on the better-performing 2-layer and 3-layer NN models that were created. Each of the hidden layers of the two models used the ReLU activation function. Because the goal result was binary classification, the output layers for the two models used the sigmoid activation function. Various other hyperparameters were used for both models, as shown in Table 4. The models were run using the method described in Section 3.1 and for a total of 50-runs each. The results of each model were then averaged across these 50 runs to smooth out the effects of randomization.

Table 4. Model hyperparameters used for the DNN.

Hyperparameter	Value
Number of hidden layers	2
Batch size	100
Number of epochs	150
Learning rate	0.0001
Train/test split	80%/20%

This research used *Acinetobacter* samples at greater amounts of reduction in sample size. The summary of the model performance at these levels of reduction can be found in Table 5 and in Figure 1. Figure 1 shows that the 2-layer NN consistently outperformed the 3-layer NN. For this 2-layer NN, accuracy is maintained near 79-80% up to a sample size reduction of 40%.

So far, this section has presented traditional, accuracy-focused results of DNNs. Section 4.2 discusses the implications of the sample reductions on model performance without referencing traditional metrics. Instead, it uses WeightWatcher to give a training-agnostic view of model quality and performance at increasing levels of reduction.

Table 5. Accuracy results from the 2-layer NN, averaged across the 50 training runs.

Sample Reduction	Sample Size	Accuracy (50-run mean)	Alpha (50-run mean)
0%	1,364	80.2%	0.80
20%	1,091	79.5%	0.80
40%	818	79.3%	0.77
60%	545	77.3%	0.80
80%	272	73.1%	0.64

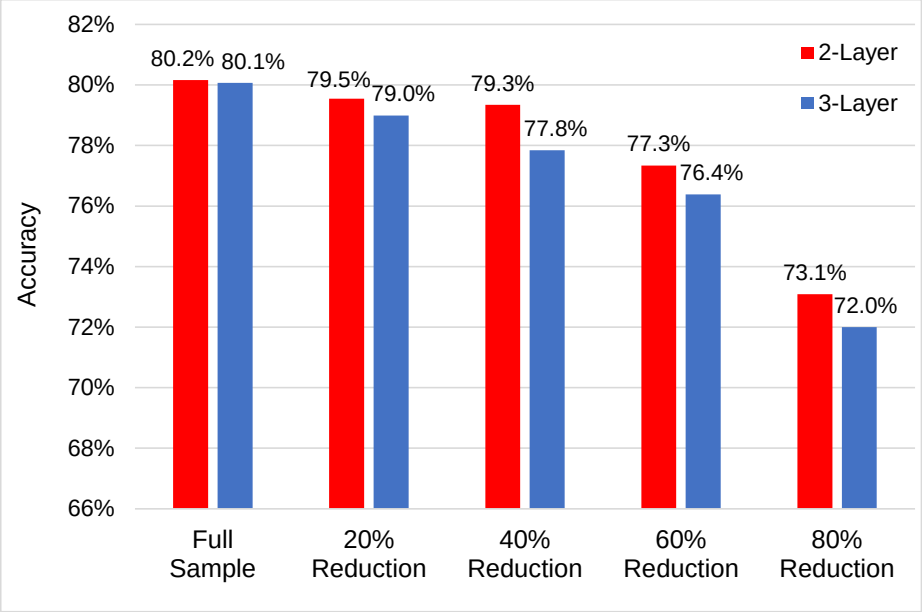

Figure 1. Accuracy of the 2-layer and 3-layer NNs averaged across the 50 training runs are shown for each reduction amount.

4.2 WeightWatcher Analysis

The WeightWatcher package provides graphics that help interpret the quality of DNNs. WeightWatcher analyzes the eigenvalues of the layers of the NNs. Based on those eigenvalues, Martin has derived empirical quality measurements of the model [17]. This research used NNs with 2- and 3-layers. For each NN, the analysis in this section shows example WeightWatcher graphs for the first inner layer for one model of the 50 training runs.

The empirical spectral densities of the first inner weight layer eigenvalues from two example models (2-layer and 3-layer) are shown in Figures 2a and 2b. Figure 2a, from the 2-layer NN, shows that the weight layer has an alpha α=3.463 and displays some distinguishing characteristics of a moderately heavy-tailed distribution. The distribution has a convex shape, but the "bulk" of eigenvalues end near 1. The max eigenvalue λ_{max} is a clear outlier beyond 7.

Figure 2b, from the 3-layer NN, shows α=4.604 displays more weakly-heavy tailed behavior. There is more "bleeding out" of the eigenvalues from the bulk near 0, and the oultier at near 2 is less extreme than the outlier of the 2-layer NN weight layer.

Because the 2-layer NN in general shows weight layer eigenvalues with a heavier-tailed distribution than the 3-layer NN, the 2-layer NN is considered the better model.

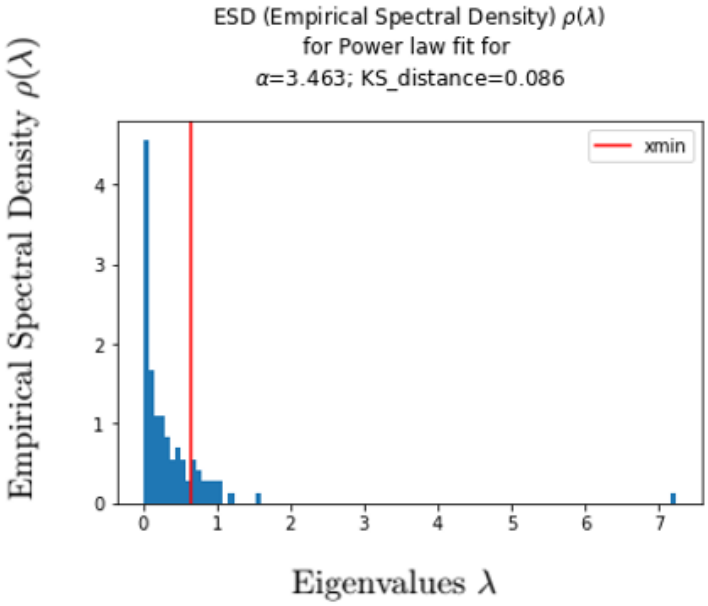

Figure 2a. The empirical spectral density (ESD) for the first inner layer of one iteration of the 2-layer NN model training on the full data sample.

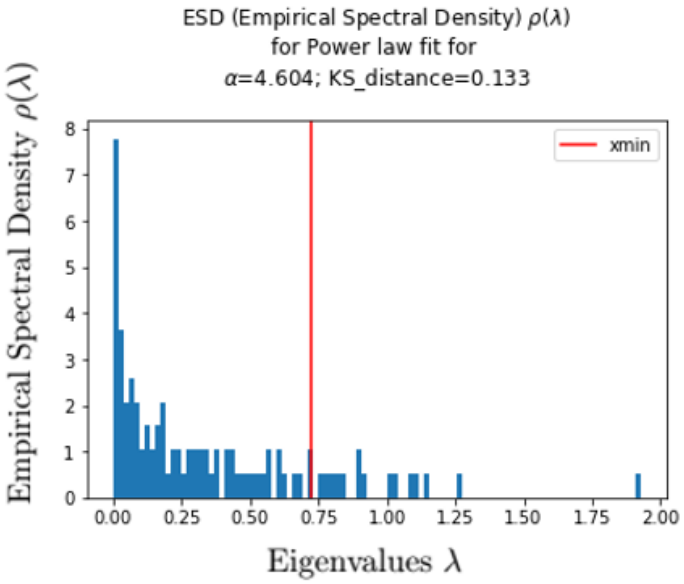

Figure 2b. The ESD for the first inner layer of one iteration of the 3-layer model training on the full data sample.

Figures 3a and 3b show the PL fits for the same inner weight layers as were discussed for Figures 2a and 2b. The figures show the log-log plot of the PL probability density function (blue) and complementary cumulative distribution function (red). The dashed lines are the actual data, while the solid lines are the fits. Good-fitting models are indicated by nearly overlapping dashed and solid lines for both red and blue. The 2-layer NN PL fit overlaps better than the 3-layer NN PL fit.

Additionally, these figures report the KS distance for each inner weight layer. Figure 3a shows a KS distance of 0.086, which indicates that the 2-layer NN is a well-fit model. Figure 3b shows a KS distance of over 0.1, which indicates that the 3-layer NN is not as well-fit.

Figure 4 shows the boxplots of the α for the 2-layer and 3-layer NNs run at each stage of sample size reduction. The 2-layer NN maintains a good α up until 60% of the original sample size, or 40% reduction, at which point the median α increases and the spread of possible values increases as well. The 3-layer NN does not show as good of results at any stage of sample size reduction, showing again that the 2-layer NN is a better model than the 3-layer NN.

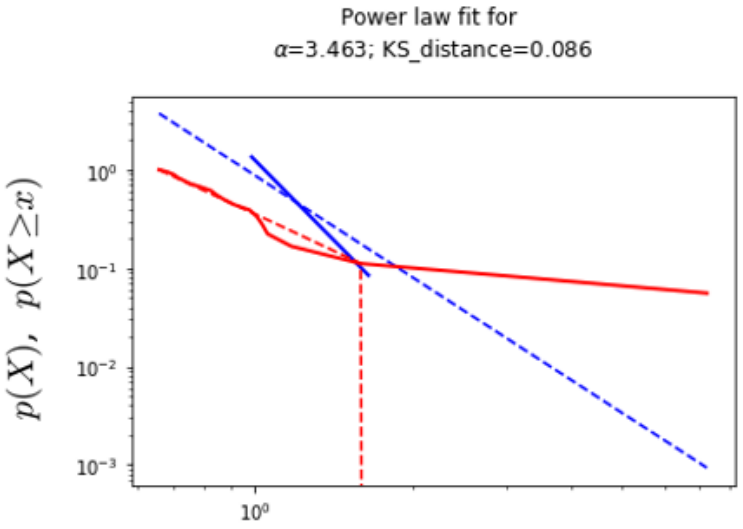

Figure 3a. Log-log plot of the PL probability density function (blue) and complementary cumulative distribution function (red) for the first inner layer of one iteration of the 2-layer NN model training on the full data sample. The dashed lines are the actual data while the solid lines are the fits.

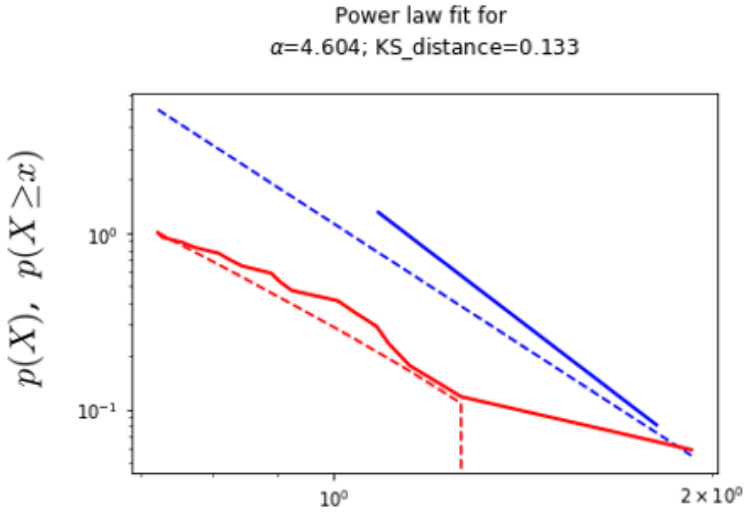

Figure 3b. Log-log plot of the PL probability density function (blue) and complementary cumulative distribution function (red) for the first inner layer of one iteration of the 3-layer NN model training on the full data sample. The dashed lines are the actual data while the solid lines are the fits.

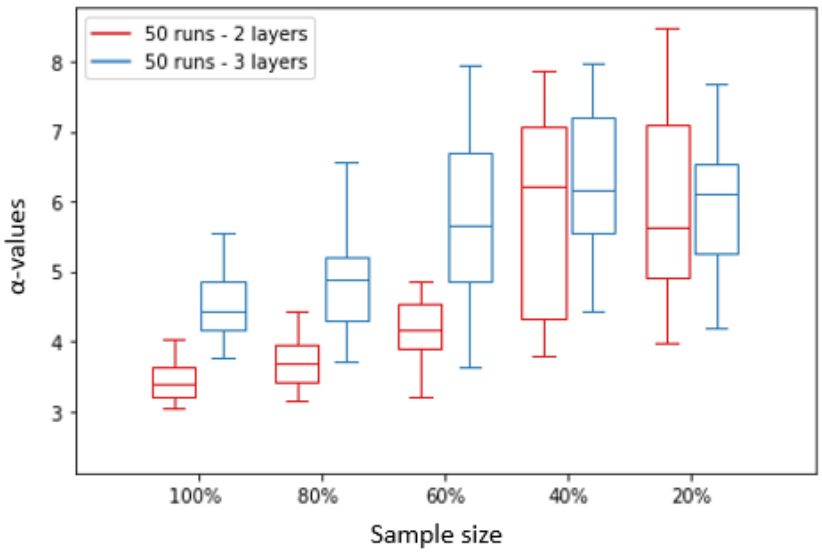

Figure 4. Box plots of the distribution of α for the 50-run training for the 2-layer and 3-layer NNs.

5 Discussion

While the explanation behind Deep Learning and Neural Networks can seem abstruse, DNNs continue to be one of the best performing ML algorithms available. Additionally, researchers Charles Martin and Michael Mahoney have found evidence of implicit self-regularization in DNNs. Martin and Mahoney moved beyond VC theory, which was unable to sufficiently describe NNs. Their research shows that there are mathematical explanations of the quality of pre-trained DNNs [19, 21].

From their research, this paper used the generalization metric α, the PL exponent of a heavy-tailed distribution. This metric has been shown to correlate well with test accuracies from various pre-trained DNNs. In one example using WeightWatcher, Martin showed that the mean α, averaged over all weight layers of a DNN, correlated with the test accuracy even while changing the hyperparameters such as batch size, learning rate, and momentum. Additionally, WeightWatcher can be used to compare and evaluate multiple DNNs without looking at the training data [21].

This paper used WeightWatcher to compare DNNs created to model AMR while reducing the number of genomic samples available to train the DNNs. Hence, this paper was able to provide insight into the quality of the DNNs using WeightWatcher metrics in addition to showing how well those models performed with smaller and smaller sample sizes. With further research into WeightWatcher's applications to AMR, DL

models can be created that are empirically excellent—weight layer matrices whose eigenvalue distributions are strongly heavy-tailed. These models can then be applied on new data samples, and because these models have been empirically evaluated to be excellent, the new test data need not be restricted by scarcity of samples or inability to gather or generate new samples. These empirically evaluated models can benefit future biomedical researchers allowing them to predict whether a strain of bacteria will be resistant to an antibiotic with minimal samples of that bacterial strain.

5.1 Ethical Implications

As stated by the CDC, AMR is "one of the greatest global public health challenges" today [2]. Because the goal of this research is to improve upon predictive models for AMR by providing a framework for empirically evaluating those models, this research has an ethical imperative to ensure that that framework is trustworthy. Additionally, this research is concerned with a divisive topic among biologists and healthcare providers, which is the ethical administration of antibiotics. This section presents ethical concerns that may arise from this research and its continuance.

While Machine Learning, Deep Learning, and Artificial Intelligence (AI) can advance biology and healthcare, they are not always considered trustworthy tools for diagnoses or treatment. Surveys have shown that patients and providers alike think that AI can produce life-threatening errors in medical treatment. Vayena et al. present three categories of ethical and regulatory concerns for Machine Learning in medicine (MLm): data sourcing, product development, and clinical deployment [43]. These are categories that this research must address.

The genomic data used in this research, originating from PATRIC, contains no personally identifiable information (PII) that could trace the data to any specific person. These data are publicly available. PATRIC is a project through the University of Chicago, funded by the National Institutes of Health/National Institute of Allergy and Infectious Diseases (NIH/NIAID), and subcontracted to the Biocomplexity Institute and Initiative of the University of Virginia [44].

Vayena et al. discuss two specific obstacles to ethical MLm product development: biased training data and post-deployment maintenance [43]. The discussion of whether antibiotic strains unequally affect certain demographics is beyond the scope of this paper. The training data used in this research used the entirety of available data for *Acinetobacter* for the antibiotic Imipenem available in PATRIC. One of the goals of this research was to present a model that performed well regardless of training data. However, training the model on data from other sources would broaden the purview of the model's results. The issue of post-deployment maintenance asks the question: How can healthcare providers ensure that the model maintains safety, efficacy, and performance as updates to the model are required? As WeightWatcher is a Python package and thus is being continually updated to newer versions, this poses real problems. As this research is continued in the future, adjustments will be made for updated versions of WeightWatcher.

According to Vayena et al., the issue of clinical deployment manifests as potential lack of transparency and interpretability of MLm [43]. ML algorithms can appear as

"black-box" models to healthcare providers and patients, leading to a lack of trust and understanding of the purpose of MLm. This research attempts to fully explain the underlying mathematics of WeightWatcher tools in Section 3.3. Further explanation may be required were this methodology to be adopted in the future.

Finally, one important consideration for any use of ML for the issue of AMR is the ethical administration by healthcare providers of antibiotics as treatments. One concern is the use of empirical antibiotics, which are antibiotics used to treat a patient's infectious disease without knowing which specific bacterial strain or strains is causing the disease. While this prophylactic use of antibiotics may help the immediate patient, it can cause harm to future generations of patients by contributing to that bacterial strain's resistance to antibiotics [45, 46]. As this research is geared toward assisting healthcare providers and biologists in making decisions about which antibiotics to prescribe for certain bacterial diseases, the authors of this research have an ethical responsibility to ensure that they do not unintentionally promote AMR. Further discussion of this topic can be found in the research of Parsonage et al., in which they discuss the many responsible parties in the ethical use of antibiotics and how those parties can help prevent AMR [46].

7 Conclusion

This research paper modeled resistance of the bacterial strain *Acinetobacter* to the antibiotic Imipenem using multiple ML models, including RF and DNNs. It analyzed the DNN using WeightWatcher, which provides measures of empirical quality of the DNN without looking at the training or test data. This paper provided a brief overview of how WeightWatcher works in Section 3.4 and how the DNN performed in Section 4. Comparing the results from the accuracy of the 2-layer NN presented in Section 4.1 and from the WeightWatcher metrics presented in Section 4.2, there is evidence of Martin's assertions that WeightWatcher results correlate well with accuracy [17]. For the 2-layer NN, both the mean accuracy and the median α maintained good values at up to 40% reduction of sample sizes.

This research can be used as a springboard for exploring the empirical quality of AMR models beyond *Acinetobacter*. Additionally, it presented evidence that large numbers of genomic samples may not be necessary to produce an empirically good model of AMR. In the future, further research into modeling AMR can be paired with WeightWatcher to investigate other bacterial strains and antibiotics. WeightWatcher will allow such research to assess the quality of DNNs without regard to the training data.

Acknowledgments.

Thanks to Dr. Jacquelyn Cheun, Dr. Charles H. Martin, John Partee, and David Josephs for discussions.

References

1. PATRIC Team. (2020). Antimicrobial Resistance (AMR), PATRICBRC.org. https://docs.patricbrc.org/user_guides/data/data_types/antimicrobial_resistance.html
2. CDC. (2019). Antibiotic Resistance Threats in the United States, *U.S. Department of Health and Human Services, CDC*. http://dx.doi.org/10.15620/cdc:82532.
3. Peleg, A. Y., Seifert, H., & Paterson, D. L. (2008). Acinetobacter baumannii: emergence of a successful pathogen. *Clinical microbiology reviews*, *21*(3), 538-582.
4. Tamura, M. & D'haeseleer, P. (2008). Microbial genotype-phenotype mapping by class association rule mining. *Bioinformatics*, *24*(13), 1523–1529.
5. Eliopoulos, G. M., Maragakis, L. L., & Perl, T. M. (2008). Acinetobacter baumannii: epidemiology, antimicrobial resistance, and treatment options. *Clinical infectious diseases*, *46*(8), 1254-1263.
6. Manchanda, V., Sanchaita, S., & Singh, N. P. (2010). Multidrug resistant acinetobacter. *Journal of global infectious diseases*, *2*(3), 291.
7. Davis, J., Boisvert, S., *et al*. (2016). Antimicrobial Resistance Prediction in PATRIC and RAST. *Sci Rep* 6, 27930.
8. Lingle, J., & Santerre, J. (2019). Using Machine Learning for Antimicrobial Resistant DNA Identification, *SMU Data Science Review,* 2(2) (12). https://scholar.smu.edu/datasciencereview/vol2/iss2/12.
9. Marchant, J. (2020). Powerful antibiotics discovered using AI, *Nature,* doi: 10.1038/d41586-020-00018-3.
10. Partee, J., Hazell, R., Solsi, A., & Santerre, J. (2020). Compressed DNA Representation for Efficient AMR Classification, *SMU Data Science Review,* 3(2)(5). https://scholar.smu.edu/datasciencereview/vol3/iss2/5.
11. Santerre, J. W. (2018). *Machine Learning for the Genotype-to-Phenotype Problem,* PhD thesis.
12. Santerre, J., David, J., Xia, F. & Stevens, R. (2016). Machine learning for antimicrobial resistance, *arXiv:1607.01224 [stat.ML]*.
13. Sealfon, R., Mariani, L., Kretzler, M. & Troyanskaya, O. (2020). Machine learning, the kidney, and genotype-phenotype analysis, *Kigney International,* 97(6), 1141-1149.
14. Lau, H.J., Lim, C.H., Foo, S.C. *et al.* The role of artificial intelligence in the battle against antimicrobial-resistant bacteria. *Curr Genet* (2021). https://doi.org/10.1007/s00294-021-01156-5
15. Martin, C. H. (2018). Rethinking-or Remembering-Generalization in Neural Networks, *CIC*. https://calculatedcontent.com/2018/04/01/rethinking-or-remembering-generalization-in-neural-networks/.
16. Martin, C. H. (2019). Towards a New Theory of Learning: Statistical Mechanics of Deep Neural Networks, *CIC*. https://calculatedcontent.com/2019/12/03/towards-a-new-theory-of-learning-statistical-mechanics-of-deep-neural-networks/.
17. Martin, C. H. (2020). WeightWatcher: Empirical Quality Metrics for Deep Neural Networks, *CIC.* https://calculatedcontent.com/2020/02/16/weightwatcher-empirical-quality-metrics-for-deep-neural-networks/.
18. Martin, C. H. (2020). WeightWatcher. *Github*. https://github.com/CalculatedContent/WeightWatcher.

19. Martin, C. H., & Mahoney, M. W. (2018). Implicit Self-Regularization in Deep Neural Networks: Evidence from Random Matrix Theory and Implications for Learning, *arXiv: 1810.01075 [cs.LG]*.
20. Martin, C. H., Peng, T., & Mahoney, M. W. (2020). Predicting trends in the quality of state-of-the-art neural networks without access to training or testing data, *arXiv: 2002.06716v1 [cs.LG]*.
21. Martin, C. H. (2020). NeurIPS Predicting Generalization Contest Post-Mortem. Retrieved December 05, 2020, from https://charlesmartin14.github.io/contest-postmortem/WeightWatcher.html
22. Jha, M., *et al*. (2017). Interpretable Model for Antibiotic Resistance Prediction in Bacteria using Deep Learning, *Biomedical and Pharmacology Journal,* 10(4).
23. Onate, F., Batto, J-M., *et al*. (2015). Quality control of microbiota metagenomics by k-mer analysis, *BMC Genomics,* 16(1), 183.
24. Howard, J., Gugger, S., & Chintala, S. (2020). *Deep learning for coders with fastai and PyTorch: AI applications without a PhD*. Sebastopol, CA: O'Reilly Media.
25. Raschka, S., & Mirjalili, V. (2019). *Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2*. Birmingham, Mumbai: Packt.
26. Calin, O. (2020). *Deep Learning Architectures A Mathematical Approach*. Cham, MI: Springer International Publishing.
27. Ramsundar, B., Eastman, P., Walters, P., & Pande, V. (2019). *Deep learning for the life sciences: Applying deep learning to genomics, microscopy, drug discovery and more*. Sebastopol, CA: O'Reilly Media.
28. Deep learning. (2020, October 23). Retrieved October 29, 2020, from https://en.wikipedia.org/wiki/Deep_learning
29. Kedia, A., & Rasu, M. (2020). *Hands-On Python Natural Language Processing: Explore tools and techniques to analyze and process text with a view to building real-world NLP applications*. Birmingham, Mumbai: Packt.
30. Kandola, A. (2017, December 20). Neuro-linguistic programming (NLP): Does it work? Retrieved October 28, 2020, from https://www.medicalnewstoday.com/articles/320368
31. Tao, T. (2012). *Topics in random matrix theory*. Providence, RI: American Mathematical Society.
32. Akemann, G., Baik, J., & Francesco, P. D. (2015). *The Oxford handbook of random matrix theory*. Oxford, England: Oxford University Press.
33. Erdős, L., & Yau, H. (2017). *A dynamical approach to random matrix theory*. New York City, NY: Courant Institute of Mathematical Sciences, New York University.
34. Gillespie, J. J., *et al*. (2011). PATRIC: the comprehensive bacterial bioinformatics resource with a focus on human pathogenic species. *Infection and immunity* **79**, 4286–4298.
35. Velez, R. & Sloand, E. (2016). Combating antibiotic resistance, mitigating future threats and ongoing initiatives, *Journal of Clinical Nursing*, doi: 10.1111/jocn.13246.
36. Tiwari, S. (2020, October 08). Activation functions in Neural Networks. Retrieved October 28, 2020, from https://www.geeksforgeeks.org/activation-functions-neural-networks/
37. Goodfellow, I., Bengio, Y., & Courville, A. (2017). *Deep learning*. Cambridge, MA: MIT Press.
38. Duda, R. O., Stork, D. G., & Hart, P. E. (2000). *Pattern classification and scene analysis*. New York, NY: Wiley.

39. Tikhonov regularization. (2020, September 28). Retrieved October 29, 2020, from https://en.wikipedia.org/wiki/Tikhonov_regularization
40. Hoerl, A. E., & Kennard, R. W. (1970). Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics, 12*(1), 55-67. doi:https://www.math.arizona.edu/~hzhang/math574m/Read/RidgeRegressionBiasedEstimationForNonorthogonalProblems.pdf
41. Martin, C. (2020, October 09). Why WeightWatcher Works. Retrieved October 29, 2020, from https://calculatedcontent.com/2020/09/14/why-weightwatcher-works/
42. Kolmogorov–Smirnov test. (2020, October 05). Retrieved October 29, 2020, from https://en.wikipedia.org/wiki/Kolmogorov%E2%80%93Smirnov_test
43. Vayena, E., Blasimme, A., & Cohen, I. G. (2018). Machine learning in medicine: Addressing ethical challenges. *PLoS medicine*, *15*(11), e1002689. https://doi.org/10.1371/journal.pmed.1002689
44. PATRIC Team. (2020). About Us, PATRICBRC.org. https://docs.patricbrc.org/about.html
45. Leibovici, L., Paul, M., & Ezra, O. (2011, October 06). Ethical dilemmas in antibiotic treatment. Retrieved January 24, 2021, from https://academic.oup.com/jac/article/67/1/12/726463
46. Parsonage, B., Hagglund, P. K., Keogh, L., Wheelhouse, N., Brown, R. E., & Dancer, S. J. (2017). Control of Antimicrobial Resistance Requires an Ethical Approach. *Frontiers in microbiology*, *8*, 2124. https://doi.org/10.3389/fmicb.2017.02124