Comparative Analysis of Algorithms for Clinical Variant Prioritization

A Major Qualifying Project
submitted to the Faculty of
Worcester Polytechnic Institute
in partial fulfillment of the requirements for the
Degree of Bachelor of Science

Submitted By: Ilona Regan
Advisor: Dmitry Korkin
Sponsor: The MITRE Corporation

Abstract

As genome sequencing has become cheaper and research into its use in the clinic has advanced, clinicians have increased the use of sequencing results to diagnose and treat genetic diseases. Effective clinical care is enabled by the annotation of genetic variants, including an accurate prediction of any pathogenic effects and the prioritization of these annotations. The goal of this project is to provide clinicians and clinical researchers with a recommendation for an ensemble algorithm that is accurate in its predictions for pathogenicity and deleteriousness, satisfies clinical standards, and is compatible with clinical data. This paper presents a comparative analysis of existing variant prioritization methods for pathogenicity. A dataset of variants annotated for pathogenicity by clinicians is used to compare predictions of pathogenicity from methods and algorithms to determine accuracy and suitability for clinical settings.

Table of Contents

Table of Figures

1 Introduction

Genome sequencing is becoming cheaper, faster, and more accurate as technology improves. These new sequencing platforms are called Next-Generation Sequencing (NGS) and their accuracy enables wider use of genome sequencing (*DNA Sequencing Costs: Data.*, n.d.). Clinicians are now able to use genome sequencing and the technology that interprets genome sequencing to diagnose and treat diseases (Krier et al., 2016). Variants called from genome sequencing are annotated to include information about phenotype, affected gene, associated diseases, frequency, and more. This rich data is important for a clinician to make an informed decision about diagnosis or treatment. However, it is time-consuming for a clinician to examine each bit of this data without prioritization of the variants according to the annotations.

Prioritization of variants enables clinicians and researchers to decide which variants are clinically actionable. Prioritization, for the context of this paper, is a method for determining which genetic variants most likely result in damaged gene function or cause a diseased phenotype (Eilbeck et al., 2017). Prioritization can include processes to identify clinical relevancy of a variant according to pathogenicity or deleteriousness. The American College of Medical Genetics (ACMG) Standards and Guidelines detail terminology to use when describing genetic variants, guidelines about how pathogenicity classification should be applied to variants, and categories for variant pathogenicity classification with requirements variants must meet for each classification (Richards et al., 2015). These guidelines recommend using in silico methods, or predictive algorithms, to predict whether a variant is pathogenic or how deleterious a variant is to supplement the evidence for a final classification of pathogenicity.

Algorithms such as CADD, DANN, MetaLR, MetaSVM, and Eigen produce scores and rank scores that can be translated into a pathogenicity classification (Rentzsch, P. et al., 2019; Quang, D. et al., 2015; Dong et al., 2015; Mccallum, K. et al., 2016). This prediction may be a value on a continuous scale from 0 to 1, with 1 being pathogenic or it may be a rank score where higher scores are more likely to be pathogenic. In some cases, the algorithm will produce a score where it predicts severity or deleteriousness of the variant instead of solely predicting pathogenicity. The result of each of these pathogenicity predictor algorithms can then be interpreted into a classification based on clinical standards and used as prioritization methods. However, the reported accuracies for these pathogenicity algorithms range from 88% to 98%, where less accurate methods could lead to a misdiagnosis of a disease if a clinician fails to find other information to negate the algorithm's prediction (Rentzsch, P. et al., 2019; Quang, D. et al., 2015; González-Pérez, A., & López-Bigas, N., 2011; Mccallum, K. et al., 2016; Ioannidis, N. M. et al., 2016; Dong et al., 2016). Software enables the annotation of these scores of pathogenicity so that variants can be prioritized according to the score and resulting classification of pathogenicity.

The goal of this paper is to provide clinicians with a recommendation for an ensemble algorithm that is accurate in its predictions for pathogenicity, satisfies clinical standards, and works with data that has been validated against two real world clinical data sets. This will be done by first defining the project boundaries within the space of genomic sequencing and annotation practices. Then, several pathogenicity algorithms will be detailed with information about how each is currently used, recommended, and benchmarked. Next, data will be found that can be used to benchmark each of the chosen pathogenicity algorithms on the same dataset so that their accuracies can be compared closely. Finally, the benchmarking results will be considered for a thorough review of the pathogenicity algorithms so that a final recommendation can be made about the pathogenicity algorithms as they relate to the clinical workplace. By the

end of this paper, the reader will understand pathogenicity algorithms in the clinical setting and which of the chosen and reviewed algorithms from this paper are recommended for use in a clinical setting.

2 Background

Genomic sequencing has improved recently to become more cost effective and faster. With NGS technology becoming widely available due to lower cost, the technology to read, process, and utilize that sequencing data has also improved. Genomic sequencing can be used for the diagnostic sequencing of Mendelian disease, preconception carrier screening for genetic disease, and genetic predisposition screening (Krier et al., 2016). Sequences must be annotated based on known and predicted variants that cause disease or have a damaging effect. These annotations give clinicians additional information to thoroughly review and classify a variant as it relates to pathogenicity, or the likelihood that a variant will be damaging and cause disease. This chapter will explore genetic variants, variant annotation, and the standards recommended for genetic variant annotation, especially as it relates to pathogenicity.

2.1 Genetic Variants

Not all genetic variants are damaging. Variants can be classified as benign, pathogenic, or as a "variant of uncertain significance" (Eilbeck et al., 2017). Benign variants will ultimately not cause a change in the phenotype or the benign variant will not affect the function of the sequence or resulting protein despite causing a change in phenotype. Other variants are known to be pathogenic, or deleterious, and cause a diseased phenotype (Richards et al., 2015). Alternatively, a variant may result in a damaged protein but is not called "pathogenic" if it is not implicated in a disease (Richards et al., 2015). Variants of uncertain significance may cause damage to a gene but may ultimately not damage a phenotype. It is also possible that not enough research has been done on a variant to form a conclusive decision about whether it is pathogenic, leading to a classification of uncertain significance. Each variant can ultimately be described by one of these three categories, with varied confidence in that classification.

Some variants that are typically not considered by pathogenicity prediction algorithms are non-coding transcript variants and synonymous variants. Non-coding transcript variants are variants that occur in the non-protein encoding portions of the genome (Dhamija & Menon, 2018). Researchers are often unsure of how this type of variant affects overall phenotype and assume most of these are "silent" mutations and do not cause deleterious effects (Dhamija & Menon, 2018). Synonymous variants are a product of the degeneracy of genetic code (Zeng & Bromberg, 2019). These variants occur in the coding region of the genome but are ultimately assumed to have no effect on the resulting protein sequence. This is due to redundancies such that multiple codons can code for the same amino acid, resulting in some variants coding for the same amino acid (Zeng & Bromberg, 2019). The target of most annotation tools and algorithms to predict pathogenicity are non-synonymous single nucleotide variants or nsSNVs (Zeng & Bromberg, 2019). Non-synonymous variants result in a protein that may experience more deleterious effects, such as a structural change. In general, single nucleotide variants (SNVs) are the most common type of variant in the human genome, making them the target of most functional annotation software.

2.2 Genetic Variant Annotation

Genetic variant annotation software requires the ability to handle large amounts of data. Genomic sequencing files from whole genome sequencing, whole exome sequencing, and other sequencing processes can contain millions of variants, all of which must be annotated with additional information (Wang et al., 2010). Some annotation pipelines use a database to compare called variants against known variants (Wang et al., 2010). These databases are highly

specified, so the use of many databases may be necessary, especially if multiple types of annotations are desired. Archives like ClinVar enable the annotation of clinically relevant data and panel reviews, dbSNP databases enable functional annotation of a variant, and still more information is necessary to allow in silico methods to predict pathogenicity for a variant (Landrum et al., 2020; Wang et al., 2010). Annotation software can annotate to specify which gene a variant impacts, the regional genomic information like conserved genomic regions or predicted transcription factor binding sites, the allele frequency of a variant, and the mutation prediction of that variant (Wang et al., 2010). These scores are often precomputed, and the use of a pipeline and database query must be used to properly annotate called variants. Annotation software allows clinicians to annotate variants with a few commands rather than curating and querying the databases manually.

One annotation tool is ANNOVAR, a free and open source tool that enables annotation of variants with rich data that can come from varied sources. ANNOVAR tool can be utilized to annotate SNVs and insertion or deletion (indel) variants (Wang et al., 2010). ANNOVAR can annotate data with information such as the functional consequence of a variant on genes, the functional importance score of a variant, and whether the variant occurs in a conserved region of the genome (Wang et al., 2010). The tool can also annotate using any database that follows the Generic Feature Format version 3, meaning any available databases for a clinician can be downloaded and used with ANNOVAR to annotate a set of variants (Wang et al., 2010). ANNOVAR allows for flexible and quick annotation of genomic sequencing files.

2.2.1 Variant Prioritization

Variant prioritization is "the process of determining which variants… are most likely to damage gene function and underlie the disease phenotype (Eilbeck et al., 2017)." This process can be completed using algorithms and predictive modeling. Prioritization algorithms consider a multitude of features created from variant annotations to create a new prediction or recommendation about which variants are most likely to be pathogenic or important to researchers and clinicians. Population allele frequency and population stratification should be included as features, since these can yield insight into how the variant occurs in the general population and if there were confounding factors in the population sampling (Eilbeck et al., 2017). Additionally, general gene information can yield insight into how a variant affects phenotype because genes can vary in size, mode of expression, and regulation (Eilbeck et al., 2017). Alongside variant prioritization is determining clinical relevancy, which can be predicted from three major factors. Penetrance, or the probability that the pathogenic variant results in disease, and prevalence, or the frequency of the disease occurring, yield great insight into clinical relevancy for a variant (Eilbeck et al., 2017*)*. Pathogenic variants are not expected to be seen in populations at a high frequency, allowing some variants to be cast as poor candidates for variant prioritization due to high prevalence in a population (Eilbeck et al., 2017). Disease prevalence calculations depend on penetrance and so both features should be considered in variant prioritization processes. Mode of inheritance can also tell if a genetic variant will cause disease in progeny, which can be important information in preconception screening (Eilbeck et al., 2017*)*. There are numerous features that can be considered by an algorithm when predicting the likelihood of pathogenicity for a variant.

Number of features and model type can affect the algorithm's ability to classify or predict pathogenicity of variants. Greater accuracy in variant prioritization algorithms is desired because this means that fewer variants are misclassified. However, more accurate predictions can result

from greater complexity in the model and cause difficulty interpreting results or a model that is overfitted to the training data, resulting in poor performance on other data later on (Eilbeck et al., 2017). Model complexity and loss of interpretability can especially occur when machine learning models are non-linear and extract or create features for classification and prediction that the human mind cannot easily understand and relate to real-world cases. When being used in a clinical setting, the need for model interpretability and model accuracy are both important, and one should not necessarily be sacrificed for the other.

2.2.2 Current Variant Prioritization Methods

Variant prioritization methods can vary widely in the type of model they are built on. Ensemble algorithms consider other pathogenicity predictions that are based on features that are annotated using databases that do not necessarily predict pathogenicity. Features may include population frequency, possible resulting amino acids, gene information, and more. These ensemble methods are complex because they consider a multitude of features and are often non-linear. As mentioned before, this can lead to interpretability issues. This can be mitigated by choosing a model that produces a direct result that is still interpretable and accurate despite the intricacies of the underlying model.

Variant prioritization methods of interest for this paper are pathogenicity algorithms. Pathogenicity algorithms are methods that try to predict the likelihood that a variant is pathogenic or deleterious. The initial result is often a numeric result, which can then be transformed into a classification. Pathogenicity algorithms like CADD, DANN, Condel, MetaLR, MetaSVM, and Eigen produce a number that represents a rank score or other type of score that does not directly represent a probability of pathogenicity (Rentzsch, P. et al., 2019; Quang, D. et al., 2015; González-Pérez, A., & López-Bigas, N., 2011; Dong et al., 2015; Mccallum, K. et al., 2016). Instead, these scores represent the relative likelihood that a variant is pathogenic when compared to other variants sampled or stored in a database. Alternatively, pathogenicity algorithms like REVEL and TAPES output a value between 0 and 1 that directly represents the probability that a variant is pathogenic, such that 1 is classified as pathogenic and 0 is classified as benign (Ioannidis, N. M. et al., 2016; Xavier, A. et al.,2019). Variants are thus labeled on a spectrum from either of these types of methods and can be prioritized as such later in clinician review. These classifications should follow the ACMG Standards described in Section 2.3. of the background in this paper. Variant prioritization algorithms that are ensemble algorithms are detailed in Table 1. Ensemble algorithms incorporate other algorithms' scores into their own scoring algorithm. The algorithms that are used as features in ensemble algorithms will be referred to as feature algorithms for the remainder of this paper.

Table 1 - Pathogenicity Ensemble Algorithm Comparison Chart

METHOD NAME	METHOD APPROACH	FEATURES	DATA	MODEL OUTPUT	REPORTED ACCURACY	AVAILABILITY
CADD	Logistic regression	63 features including: - conservation metrics - functional genomics - transcript information - protein-level scores	- Data for training does not include manually curated variants, or variants where they are known pathogenic - Operates on assumptions that variants are mostly benign - Tuning data is manually curated for known pathogens	A rank score based on genome-wide distribution of scores	Overall AUC Score (TPR) = 91.64% AUC Score for Missense Variation (TPR) = 93.05%	Available for non-commercial use via online query or precomputed scores
DANN	Deep neural net	949 features, all Boolean, including: - conservation metrics - functional genomics - transcript information - protein-level scores	- Training dataset consisted of observed and simulated variants	A functional prediction score	Classification Accuracy = 66.1% ROC Curve (TPR) = 72.4% for testing set, 94.59% for ClinVar and ESP	Precomputed scores available for non-commercial use or repository available to run locally
REVEL	Random forest	18 individual pathogenicity prediction scores: - conservation scores - functional prediction scores	- Human Gene Mutation Database (only missense disease variants) - Exome Sequencing Project (missense exome variants) - 1000 Genomes Project (missense exome variants)	REVEL Score (0,1) based on proportion of trees in random forest classifying variant as pathogenic	Overall AUC (TPR) = 0.908 Sensitivity = 0.754 Specificity = 0.891	Precomputed scores available for non-commercial use as a single file
Condel	Weighted Average of the normalized Scores of the individual methods (WAS)	- LogR Pfam E-Value (Logre) - MAPP - Mutation Assessor - Polyphen2 - SIFT	- HumVar & HumDiv from PPH2 website (deleterious and neutral variants) - Cosmic database (deleterious variants) - TP53 mutants	The “consensus deleteriousness” score of missense mutations	88% accuracy for classification	Precomputed scores available as a tsv file with all available scores or scores can be queried online
Eigen	- Unsupervised - Eigen Score: Weighted linear combo of annotations based on estimated accuracies - Eigen-PC Score: eigendecomposition of the annotation covariance matrix	Many annotations that each get a weight that corresponds to their importance, including: - evolutionary conservation scores - individual annotations - individual tool predictions	- Assumption 1: blockwise conditional independence between annotations - Assumption 2: correlation is due to annotation differences	An estimate of accuracy for each functional annotation score An aggregate Eigen score for variants of interest	Eigen-PC has more sensitivity, better for non-coding variants Eigen performs better for coding variants Eigen Score AUC = 0.864	Precomputed scores available as a .tsv and tabix indexing
MetaLR	Logistic regression	- SIFT - PolyPhen-2 - LRT - MutationTaster - MutationAssessor - FATHMM - PhyloP - GERP++ - SiPhy - MMAF	- dbNSFP database (potential human nsSNVs) - 1000 Genomes project (MMAF)	Raw score is a score between 0 and 1 of pathogenicity	Overall AUC (TPR) = 0.92 for dataset 1 Overall AUC (TPR) = 0.94 for dataset 2	Precomputed scores available through ANNOVAR

MetaSVM	Support vector machine	- SIFT - PolyPhen-2 - LRT - MutationTaster - MutationAssessor - FATHMM - PhyloP - GERP++ - SiPhy - MMAF	- dbNSFP database (potential human nsSNVs) - 1000 Genomes project (MMAF)	Raw score is a score between 0 and 1 of pathogenicity Converted score is a score between 0 and 1, where 1 is more deleterious	Overall AUC (TPR) = 0.91 for dataset 1 Overall AUC (TPR) = 0.93 for dataset 2	Precomputed scores available through ANNOVAR
TAPES	Transforms ACMG classification into linear probability of pathogenicity – naïve Bayesian classifier	A file annotated with ANNOVAR or VEP	- Japanese pediatric cancer data for benchmark - ClinGen clinician reviewed and labeled variants for validation	A probability of pathogenicity (0,1)	AUC for Pathogenic = 0.959 AUC for Benign = 0.957	A GitHub repository of software is available online

Algorithms like SIFT, PolyPhen-2, MutationTaster, and FATHMM are used often as features in ensemble algorithms that predict pathogenicity. The algorithms listed in the 'Features' column of Table 1 are used as features because their results often predict pathogenicity or a characteristic of a variant associated with pathogenicity or deleteriousness. Ensemble algorithms leverage multiple scores to create an overall prediction for pathogenicity or deleteriousness. SIFT and FATHMM predict the functional effects of a variant (Ng & Henikoff, 2001; Shihab et al., 2013). MutationTaster predicts the disease potential of a variant (Schwarz et al., 2010). PolyPhen-2 predicts the functional significance of the variant (Adzhubei et al., 2010). SIFT, PolyPhen-2, MutationTaster, and FATHMM are detailed in Table 2 with their method approaches, model outputs, and availability.

Table 2 - Pathogenicity Algorithms Used as Features in Ensemble Algorithms Comparison Chart

METHOD NAME	METHOD APPROACH	FEATURES	MODEL OUTPUT	AVAILABILITY
SIFT	Probabilistic classifier	- Position of amino acid substitution - Type of amino acid substitution	Prediction of functional effects	Through ANNOVAR and SIFT website
PolyPhen-2	Naïve Bayes Classifier	- Homologous sequences - Identity-based scores - Profile-based scores - Pfam domain	Prediction of functional significance	Through ANNOVAR
MutationTaster	Naïve Bayes Classifier	- evolutionary conservation - splite-site changes	Prediction of disease potential	Through website query and through ANNOVAR
FATHMM	Hidden Markov models	- Species-specific pathogenicity weights - Amino acid substitution consequences	Prediction of functional effects	Online web query and through ANNOVAR

SIFT, PolyPhen-2, MutationTaster, and FATHMM are trained on different datasets. Both PolyPhen-2 and MutationTaster use Naïve Bayes Classifiers to compute a prediction score, howerever PolyPhen-2 predicts functional significance and MutationTaster predicts disease potential (Adzhubei et al., 2010; Schwarz et al., 2010). This slight difference in interpretation of the algorithm output is what makes it useful to use multiple of these algorithms as features in ensemble algorithms. The algorithm outputs from SIFT, PolyPhen-2, MutationTaster, and FATHMM are all slightly different, but they also all relate to pathogenicity. Ultimately, this offers diversity and a thorough consideration of a vast array of pathogenicity influences when creating an ensemble algorithm that uses other algorithms as features.

2.3 Standards and Guidelines for Variant Prioritization

Standards allow for replication and can also act as guidelines to inform decisions. The American College of Medical Genetics (ACMG) published standards and guidelines for genetic tests used in clinical laboratories, including genotyping, single genes, panels, exomes, and genomes (Richards et al., 2015). The ACMG standards are applicable to variants in all Mendelian genes, regardless of how that variant was sequenced prior to variant calling (Richards et al., 2015). The ACMG guidelines are meant to offer guidance for classifying the pathogenicity of genetic variants.

When beginning to review the ACMG guidelines and speaking about variants, there are a few key details to keep in mind. The word "variant" is used to be more inclusive than the term "polymorphism", which is a variant that has a frequency above 1% (Richards et al., 2015).

"Variant" is also the preferred term to use instead of "mutation", which can imply pathogenic effects (Richards et al., 2015). Variants can be described using the ACMG five-tier classification standards (Richards et al., 2015). Some descriptions may use the terms "mutation" or "polymorphism", but these are not recommended per ACMG standards. Variant is a more neutral word and additional information about variant nomenclature has been put forth by the Human Genome Variation Society or HGVS (*Sequence Variant Nomenclature*).

In contrast to what HGVS recommends, the ACMG standards do not suggest using the phrase "affects function". Instead, the ACMG recommends the use of the term "pathogenic" because clinical interpretations are "typically directly evaluating pathogenicity". The ACMG classifications are as follows: Pathogenic, Likely Pathogenic, Uncertain Significance, Likely Benign, and Benign (Richards et al., 2015). There is no quantitative significance in the word "likely" amongst these classifications, though the ACMG panel suggests that a "likely" classification may be used when the certainty of that classification is greater than 90% (Richards et al., 2015). These classifications can be assigned when other data about a variant has been reviewed.

A variant classification and description are only as good as the research put into that classification. ACMG guidelines recommend searching for databases that host quality material (Richards et al., 2015). This means that the data should be well-sampled, or be a known sampling, and that diseased individuals are marked or noted as part of sampling. Additionally, data must be quality enough to trust that the variants in the database have been sampled and classified correctly. Those classifications should also relate to the HGVS nomenclature and use a current genome build as the reference genome (Richards et al., 2015). Data validation for analytical accuracy makes a database more reliable too. Databases offer contextual and varied data that can supplement and improve decisions in classification of variants.

Databases and contextual data are necessary to improve classification prediction. Contextual data can include previous research, several databases, gene information, and more. Computational (in silico) predictive programs are not recommended to be the only indicator of pathogenicity and should not be the sole way to make a clinical assertion (Richards et al., 2015). Using additional information to support the prediction from an in silico method more deeply informs a clinician as they use the standards for variant classification set by ACMG. The ACMG classification standards are stringent so that there are fewer variants reported as pathogenic without sufficient supporting evidence (Richards et al., 2015). There is no universal threshold suggested for use in using in silico predictive programs to classify a variant because each method produces a unique score. Pathogenic variant classification can imply that the variant is "actionable" for clinicians, so the accuracy with which this classification is used should be high to avoid medical complications (Richards et al., 2015). Additionally, if all in silico predictions agree on a classification, then the predictions should be used as evidence in support of that classification (Richards et al., 2015). Otherwise, if the in silico predictions classify a variant differently, then these predictions should be disregarded and none should be used as supporting evidence in the final variant classification (Richards et al., 2015). A thorough approach that reviews quality databases, academic articles, and in silico prediction methods to determine a variant classification achieves the best result.

3 Methodology

The goal of this project is to provide clinicians and clinical researchers with a recommendation for an ensemble algorithm that is accurate in its predictions for pathogenicity and clinical relevancy, satisfies clinical standards, and is compatible with realistic data. This goal is achieved completing the following objectives:

1. Define the scope of the project and outline the history of genomic sequencing, variant calling, annotation, and prioritization.
2. Define which ensemble methods are suitable for a case study for clinical prioritization which involves performing a comparative analysis of a variety of available ensemble methods.
3. Find data suitable for testing and benchmarking the methods.
4. Prepare the data for the methods.
    a. Annotate the data if necessary.
    b. Reformat the data if necessary.
5. Test the methods with the case study data and assess the results with the standards for prioritization.

This project begins by selecting pathogenicity algorithms for comparison and comparing the models, the data used, and the result of the algorithm. Then, each suitable algorithm will be implemented to predict pathogenicity with a set of case study data. A comparison of each prediction on a standardized scale against the same data set will be performed. This will enable the final recommendation to be formulated. Each part of this project will work towards satisfying the project goal and are detailed in the following section.

3.1 Defining the Project Scope

The goal of this project is to find a pathogenicity algorithm for genetic variants that can be used in a clinical setting. This means that the algorithm should be reliable, have a high accuracy, and should be suitable for clinician use. There exist many variant annotations that can be used to calculate pathogenicity or infer pathogenicity, but the focus of this paper is pathogenicity algorithms that predict the likelihood and output a numeric result that can be classified according to ACMG standards later. Additionally, the focus will be on recent algorithms, since these have access to more data, new techniques, and more features than their older counterparts.

Prediction algorithms like SIFT, PolyPhen-2, MutationAssessor, and FATHMM predict pathogenicity from features directly relating to variant information. SIFT, created in 2009, is an algorithm that predicts whether an amino acid substitution due to a nsSNV affects the protein's overall function (Kumar, P., Henikoff, S., & Ng, PC., 2009). PolyPhen-2 is a probabilistic classifier that assesses nsSNVs to determine the functional significance of allele replacement based on sequence-based and structure-based features (Adzhubei, I. et al., 2010). MutationAssessor also analyzes functional significance and creates a functional impact score calculated from evolutionary information and is recommended for use with cancer research (Reva, B. et al., 2011). Like SIFT, PolyPhen-2, and MutationAssessor, the algorithm FATHMM uses Markov models to predict the functional effects of nsSNVs (Shihab, H.A. et al., 2013). These algorithms assess functional impact of an nsSNV on protein function, which can imply deleteriousness. However, this paper will not focus on these algorithms and will instead use

them for primarily benchmarking purposes and for understanding how the ensemble algorithms create predictions.

This paper will instead investigate pathogenicity ensemble algorithms that often include SIFT, PolyPhen-2, MutationAssessor, FATHMM, and other algorithm predictions as features to create a new prediction that directly represents the likelihood of pathogenicity. These types of pathogenicity algorithms will be detailed and chosen for benchmarking and comparison in Section 3.2 of this paper.

3.2 Choosing Pathogenicity Algorithms for Comparison

Before comparing pathogenicity scores, the scoring methods must be compared overall to see if their goals and results are similar enough to be compared on a standardized scale. In Table 1, several pathogenicity ensemble algorithms are compared. Six characteristics are compared for each algorithm to later select which algorithms are suitable for a comparative study. These characteristics are the underlying statistical or mathematical model, the features this model trains on, the data that this model was originally trained and tested on, the model output, the reported accuracy of the model in its respective published paper, and the availability of the algorithm.

The availability of the algorithm indicates if the algorithm can be studied in this project. To be able to use it in this comparative analysis, the algorithm tool or scores must be free or open-source and available for download online. All of the algorithms researched in Table 1 either had open-source code or availability of pathogenicity prediction scores for free for non-commercial use.

The next important step in determining if the algorithm would be suitable for this study was the method in which it applied the pathogenicity scores to a given dataset. CADD, DANN, MetaLR, MetaSVM and Eigen made their precomputed pathogenicity scores available through ANNOVAR, a tool which annotates VCFs. Additionally, feature algorithms like SIFT, PolyPhen-2, MutationTaster, and FATHMM are available through ANNOVAR. Since five ensemble algorithms and four feature algorithms researched in this paper are available through ANNOVAR, this tool was implemented first to compare most pathogenicity algorithms detailed.

Not all the ensemble algorithms detailed in Table 1 are available through ANNOVAR. REVEL and Condel make their precomputed pathogenicity scores available through a file download of a tab-separated file. Since these two ensemble algorithms are not available through ANNOVAR, their scores were not benchmarked in this study. TAPES was also not chosen to be benchmarked because, as described in Section 3.3, its benchmarking and validation data was chosen to benchmark the other ensemble and feature algorithms. Therefore, the TAPES algorithm did not need to be benchmarked again and was not compared to the other algorithms because that would have resulted in a biased score for the TAPES algorithm.

3.3 Choosing Benchmarking Data

Knowing that five pathogenicity ensemble algorithms were available through ANNOVAR, the next goal was to find data suitable for a case study and compatible with ANNOVAR. Specifically, the goal was to find a VCF file that included panel decisions from clinicians or genetic experts that adhered to ACMG criteria. The datasets that TAPES benchmarked and validated their algorithm with satisfy these specifications. TAPES used two datasets that are publicly available through their paper, one for benchmarking purposes and the other for validation purposes (Xavier et al., 2019). The benchmarking dataset contains about 880 variants that are labeled according to ACMG standards based on an expert panel and will be

referred to as the "benchmark dataset" for the remainder of this paper (Zhang et al., 2015). These variants are sampled from patients younger than 20 and annotated with information from cancer-specific and locus-specific genetic databases in order to identify germline mutations (Zhang et al., 2015). These panel decisions were based on annotations, medical literature, and additional contextual information to form a final panel decision (Zhang et al., 2015). Due to the dedicated reviews and careful labelling of each variant, the TAPES benchmarking dataset has been selected for benchmarking the pathogenicity algorithms in this project despite it having a highly specialized sample demographic.

TAPES used a second dataset, now referred to as the "validation dataset" in this paper, to validate their algorithm's accuracy and performance. The validation dataset contains panel decisions from the ClinGen repository (Xavier et al., 2019). ClinGen is a public database where researchers, clinical laboratories, expert groups, clinicians, and even patients can submit variants along with supporting evidence of an ACMG classification to be stored in the ClinGen database (Rehm et al., 2015). The TAPES validation dataset contains 530 variants from ClinGen that include various types of genetic diseases (Xavier et al., 2019; Rehm et al., 2015). The validation dataset will be treated differently than the benchmark dataset because the ClinGen data may not have undergone as rigorous of a review that the benchmark dataset received by the Zhang et al. panel and the data was not manually reviewed and curated by this study before use in benchmarking the algorithms.

In addition to the panel decisions in the benchmark dataset and the validation dataset, there are related synthetic VCFs which TAPES used to annotate the variants as they relate to the panel decisions from Zhang et al and to the ClinGen panel decisions. The reference genome for the benchmark synthetic and the validation synthetic VCF was GrCh37, which is a build compatible with all pathogenicity algorithms chosen that are available in ANNOVAR. The synthetic VCFs were able to be annotated with ANNOVAR to get additional pathogenicity prediction scores to compare against the panel decisions provided by the TAPES datasets. The annotated pathogenicity scores were able to be compared to the panel decisions, which are labeled following ACMG standards as the following:

i. P = Pathogenic
ii. PP = Probably/Likely Pathogenic
iii. U = VUS = Variant of Unknown/Uncertain Significance
iv. PB = Probably/Likely Benign
v. B = Benign

The TAPES datasets followed the standards and guidelines set forth by the ACMG, making this dataset with panel decisions and synthetic VCFs suitable for a comparative analysis of the chosen pathogenicity algorithms. Of all 883 entries in the benchmark dataset, 103 variants were classified as P, 24 variants were classified as PP, 440 variants were classified as U, 274 variants were classified as PB, and 42 variants were classified as B. These classifications were assigned by the Zhang et al. panel (Zhang et al., 2015). Despite most variants being classified as uncertain significance, the benchmark dataset has an ample number of variants classified as Pathogenic or Probably Pathogenic to be used as a dataset to calculate accuracy scores. The validation dataset has 530 variants, where 165 variants were classified as P, 104 variants were classified as PP, 109 variants were classified as U, 94 variants were classified as PB, and 58 variants were classified as B. The benchmarking dataset also has an ample number of variants classified as Pathogenic or Likely Pathogenic to be analyzed further.

3.4 Annotating and Preparing the Data

The TAPES synthetic VCF relating to the benchmark dataset was annotated with ANNOVAR. To do this, ANNOVAR was downloaded and installed. Since ANNOVAR is Linux compatible, a virtual machine was used to host this software and all necessary data. Later, files were transferred remotely from the virtual machine to the local machine for benchmarking purposes. Once ANNOVAR was installed, their example scripts were run to ensure that the software had been installed properly. With a validated installation, the database "dbnsfp33a" was downloaded from ANNOVAR for hg19. The hg19 build is the equivalent of GrCh37, which is what the benchmark and validation synthetic VCFs use as a reference genome. This database was selected for download because it included multiple pathogenicity prediction scores, including CADD, DANN, Eigen, SIFT, PolyPhen-2, MutationAssessor, and FATHMM (Wang, K., 2019). In addition to offering this database, ANNOVAR was chosen as the software for annotating the synthetic VCF because it is a free software for non-commercial use, which includes this project.

The benchmark dataset's corresponding synthetic VCF and the validation dataset's corresponding synthetic VCF were annotated with the dbnsfp33a database using ANNOVAR. The resulting file was a tab-separated value file (.tsv) which was downloaded into a local computer from the virtual machine. Since the ANNOVAR results file was a .tsv file and the panel decisions were in a .tsv file, Python was used to analyze the data because the Pandas library allows for quick data manipulation (McKinney, W., 2010). The ANNOVAR file and the file containing panel decisions were merged to create a final file. The final file was a .tsv file that contained information from the panel decisions file and all the information from the ANNOVAR pipeline.

3.5 Benchmarking the Pathogenicity Scores

The pathogenicity algorithms were benchmarked on the dataset against the panel decisions which were treated as a ground truth for classification. To begin, box plots were created to illustrate the distribution of pathogenicity algorithm predictions for each classification of variant pathogenicity. With knowledge of the ranges for outputs by each pathogenicity algorithm, the graphs were able to be analyzed to deem if the algorithm was scores that imply pathogenicity to variants with a ground truth of Pathogenic or Likely Pathogenic and scores that imply benignity to variants with a ground truth of Benign or Likely Benign. Additionally, descriptive statistics were retrieved for relevant columns. The mean, mode, max, and min were obtained and compared against the box plots to ensure the graphs were created correctly. These descriptive statistics also validated whether the algorithms were functioning properly.

The primary method used to benchmark the pathogenicity algorithms' accuracy was Receiver Operating Characteristic (ROC) curves. ROC curves plot true positive rates on the y axis and false positive rates on the x axis by testing different thresholds for the model outputs. In this manner, it illustrates the tradeoff between sensitivity and specificity. The ideal ROC curve is steep and follows the top left-hand border, meaning there is a great deal of accuracy in both specificity and sensitivity in the test. For this project, ROC curves were calculated by using the panel decisions as the ground truth. Panel decisions are the closest to a ground truth because they have been heavily reviewed by multiple experts in a thorough process that involves reviewing databases, academic papers, and other supporting evidence to reach the conclusion and classification. The panel decisions were transformed from ACMG classifications to a binary

classification, with a positive label being the ACMG class of Pathogenic and Likely Pathogenic and the negative label being the ACMG classes of Uncertain Significance, Likely Benign, and Benign. Then, using the transformed binary classifications as ground truth for classification, the true positive rate, false positive rate, and thresholds were calculated for each pathogenicity algorithm using the scikit-learn library (*Scikit-learn: Machine Learning in Python*). Additionally, the area under the curve (AUC) was calculated for each pathogenicity algorithm as a measurement of overall accuracy. The ROC curves and AUC were compared between algorithms that produce similar outputs. Therefore, the algorithms that produce rank scores were compared to one another, the algorithms that produce overall scores were compared to one another, the two Eigen scores were compared to one another because they produce results different from any other algorithm, and the algorithms that are not ensemble learners were compared to one another. Through these comparisons, conclusions about accuracy in the benchmarking dataset were able to be formed.

3.6 Forming a Recommendation

To form a recommendation, all previous data gathered in this comparative analysis was considered. This included how the algorithm was trained, what the algorithm specifically predicts, and how the algorithm performed on the benchmarking dataset. The performance comparison was completed by comparing graphs and AUC metrics between similar algorithms. It was important to include an analysis of each algorithm's characteristics, such as what type of model it uses, the training data developers used, the features developers chose, the availability of the algorithm. This analysis helped to balance any influence from the accuracy and performance of the algorithm in the benchmarking set, since the benchmark dataset itself was biased towards cancer patients and the applicability of these algorithms claim to go beyond cancer research. The final recommendation of this paper was written with clinicians and clinical researchers in mind who may be interested in applications of in silico prediction methods for pathogenicity and deleteriousness.

4 Results and Analysis

The following results include an analysis of the data sets, an overview of the algorithms predictions once they were applied to each dataset, and the accuracies of each algorithm as it pertains to each dataset. It was found that the validation dataset was insufficient for benchmarking capabilities in the scope of this study, so only the benchmark dataset was used to form the final conclusion.

4.1 Qualitative Assessment of Pathogenicity Algorithms

The pathogenicity algorithms detailed in Table 1 all produce a resulting rank score or raw score that corresponds to a prediction of pathogenicity or deleteriousness. Eigen is the only unsupervised method in this set of algorithms, meaning it was not trained with a labeled dataset. The other algorithms, including CADD, DANN, REVEL, Condel, Eigen, MetaLR, and MetaSVM are trained on dataset(s) labeled with ground truths. This means that the data these algorithms, or even algorithms like SIFT, Poly-Phen2, Mutation Taster, and FATHMM could be trained on mislabeled data if the datasets were pulled from an unreliable database or source. Additionally, labeled training data could bias algorithms towards specific dataset characteristics, especially if all the variants are associated with cancer or occur in a specific region of the genome.

Algorithms that rely on primarily other algorithms can result in potential bias too. If an algorithm only relies on other algorithms, it will be biased towards what data those feature algorithms trained on, what their tuned features are, and the validity of their outputs. The algorithm that behave as features in ensemble algorithms need to be accurate, otherwise the overall ensemble algorithm score can be affected, especially depending on how much weight is assigned to each of the feature algorithms.

CADD, DANN, MetaLR, and MetaSVM provide both at least a raw score and a rank score. Both scores are useful to consider because they imply different things while still both predicting pathogenicity. For each algorithm that produces a rank score, the higher the value, the more likely that variant is to be pathogenic or deleterious versus any other variant in that sample (Rentzsch, P. et al., 2019; Quang, D. et al., 2015; Dong et al., 2015). For each algorithm that produces a score, the higher the value, the more likely that variant is to be pathogenic or deleterious (Rentzsch, P. et al., 2019; Quang, D. et al., 2015; Dong et al., 2015). The threshold at which a score classifies a variant as Pathogenic or Benign is different for each algorithm. Eigen calculates two scores, an Eigen-PC score and an Eigen raw score. The Eigen raw score is calculated as a weighted linear combination of annotations and the Eigen-PC score is an Eigen decomposition of the annotation covariance matrix. Both Eigen-PC and raw Eigen scores will be higher (greater than 0, closer to 1, if not greater than 1) if the variant is pathogenic (Mccallum, K. et al., 2016). The rank scores or raw scores of other algorithms may be easier to interpret than the Eigen algorithm's results.

4.2 Algorithm Score Distributions

The scores from each pathogenicity algorithm were used to create a box plot to visualize the spread of the data. In these plots the y-axis is the algorithm result value, and the x-axis is the variant classification according to the panel decision. P is Pathogenic, PP is Likely/Probably Pathogenic, U is Uncertain Significance, PB is Likely/Probably Benign, and B is Benign. These categories are according to ACMG standards. Figures 1 through 16 depict the distribution of raw scores and rank scores for CADD, DANN, MetaLR, and MetaSVM.

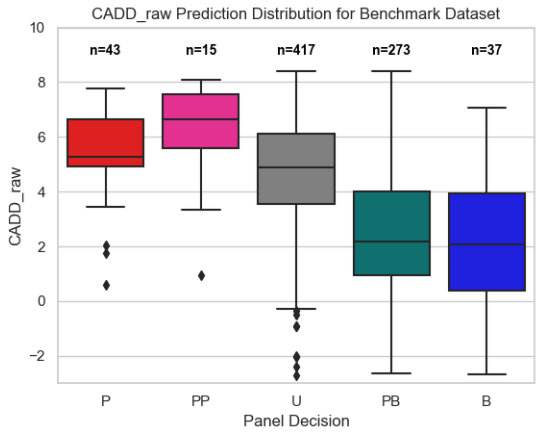

Figure 1: Distribution of CADD raw scores per classification in the benchmark dataset.

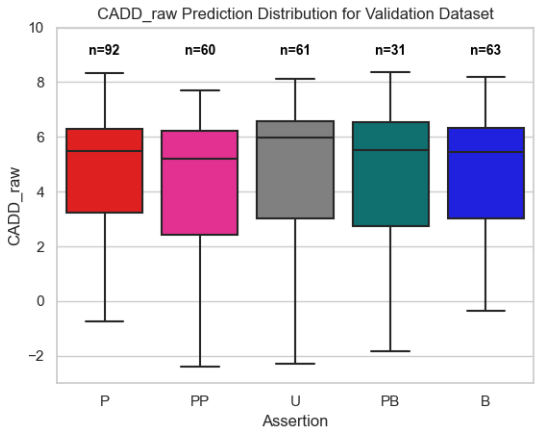

Figure 2: Distribution of CADD raw scores per classification in the validation dataset.

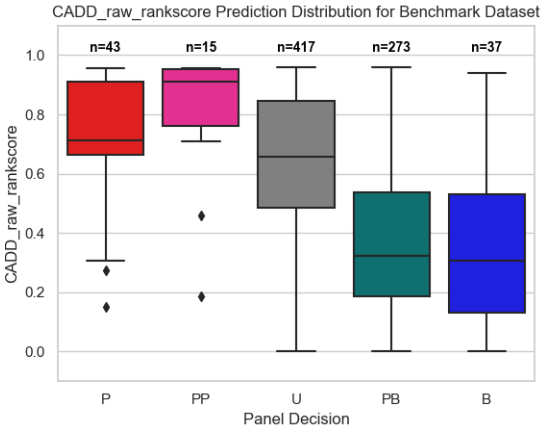

Figure 3: Distribution of CADD raw rank scores per classification in the benchmark dataset.

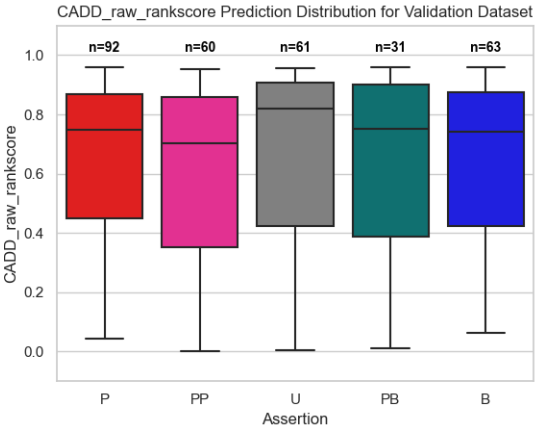

Figure 4: Distribution of CADD raw rank scores per classification in the validation dataset.

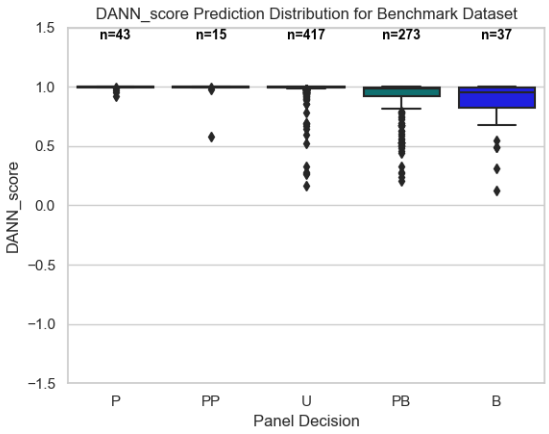

Figure 5: Distribution of DANN raw scores per classification in the benchmark dataset.

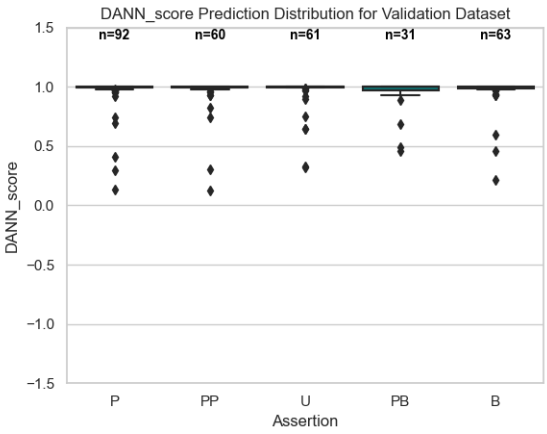

Figure 6: Distribution of DANN raw scores per classification in the validation dataset.

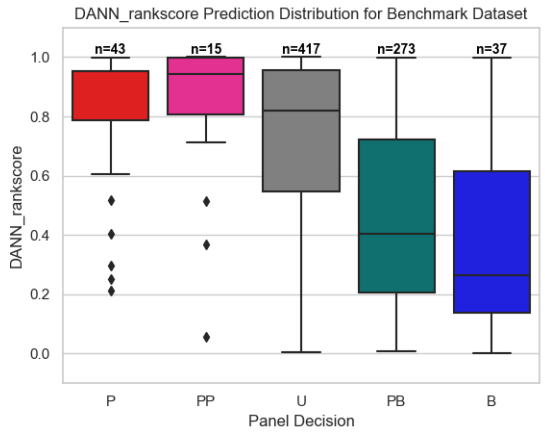

Figure 7: Distribution of DANN rank scores per classification in the benchmark dataset.

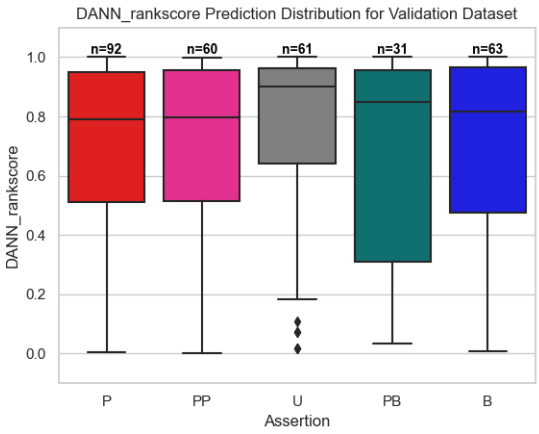

Figure 8: Distribution of DANN rank scores per classification in the validation dataset.

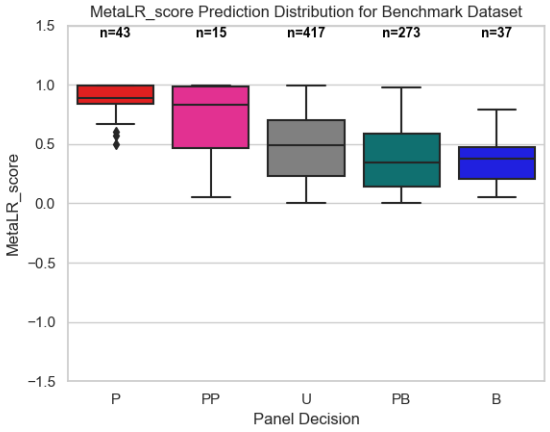

Figure 9: Distribution of MetaLR scores per classification in the benchmark dataset.

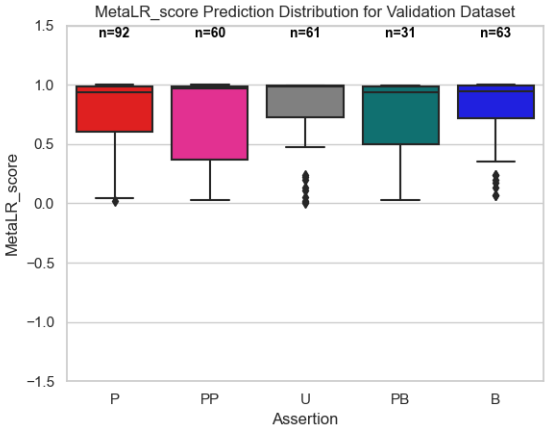

Figure 10: Distribution of MetaLR scores per classification in the validation dataset.

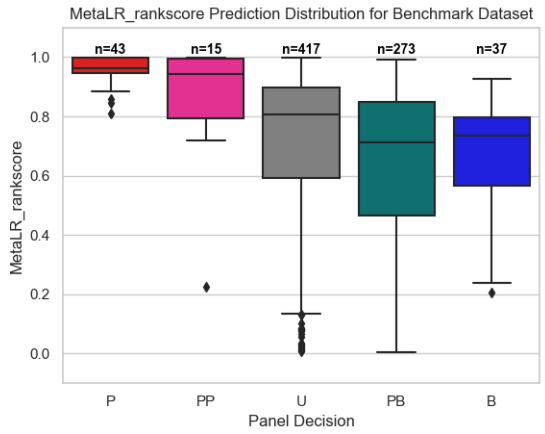

Figure 11: Distribution of MetaLR rank scores per classification in the benchmark dataset.

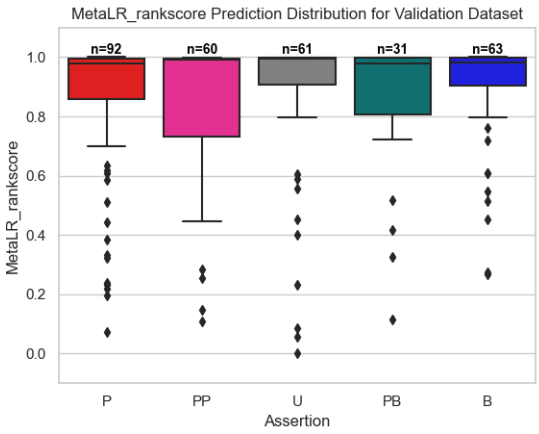

Figure 12: Distribution of MetaLR rank scores per classification in the validation dataset.

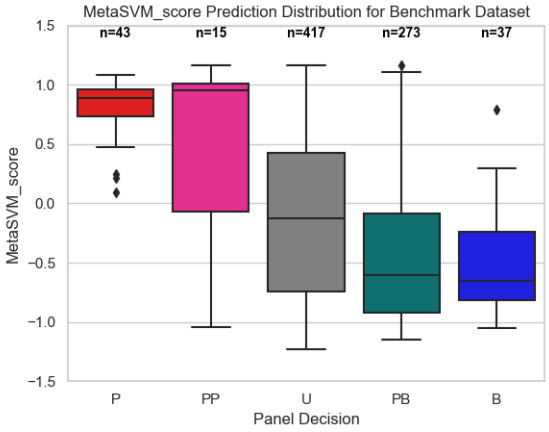

Figure 13: Distribution of MetaSVM scores per classification in the benchmark dataset.

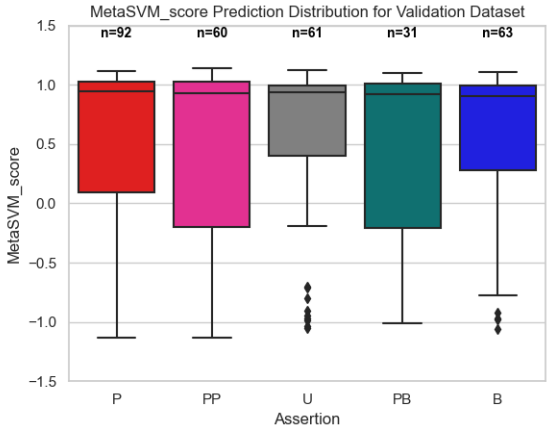

Figure 14: Distribution of MetaSVM scores per classification in the validation dataset.

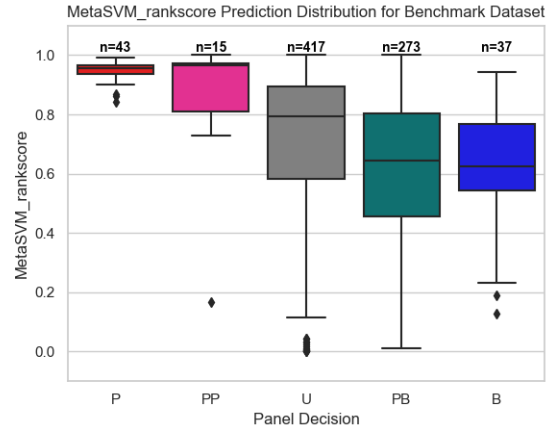

Figure 15: Distribution of MetaSVM rank scores per classification in the benchmark dataset.

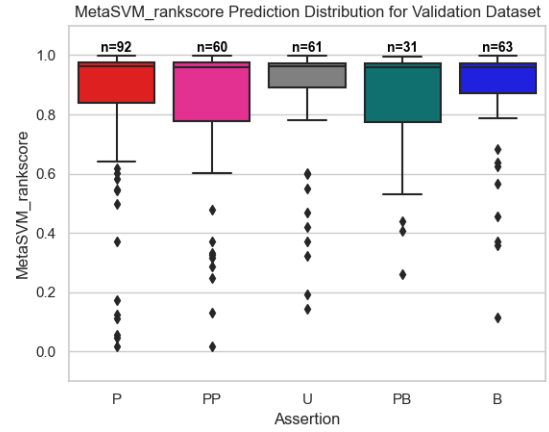

Figure 16: Distribution of MetaSVM rank scores per classification in the validation dataset.

Shown in Figures 1 and 2, the distribution of CADD predicted raw scores differs between the benchmark dataset and the validation dataset. In Figure 1, the predicted values in the Pathogenic and Likely Pathogenic categories have higher means than the predicted values in the Unknown, Likely Benign, and Benign categories. This implies that at a basic level, CADD is producing greater value predictions for variants that have a ground truth of Pathogenic or Likely Pathogenic, which is how the algorithm should behave. However, in Figure 2, the mean predicted score is approximately the same between all categories. This suggests that the CADD algorithm had more difficulty discriminating between benign and pathogenic variants in the validation dataset. Looking at the boxplots of the other scores produced by DANN, MetaLR, MetaSVM, it is evident that this trend is present for each algorithm. DANN raw scores are the only raw score algorithm product that has a similar box plot between the benchmark dataset and the validation dataset. This suggests that DANN discriminated poorly between the categories when the algorithm produced a raw score value, shown in Figures 5 and 6, regardless of the dataset. The DANN algorithm did not experience this same difficulty when asked to produce rank scores for the variants of the benchmark dataset, show in Figure 7.

The Eigen algorithm produces results different than a rank score or a raw score produced by one of the algorithms shown in Figures 1 through 16. Therefore, Eigen results for the benchmark dataset and the validation dataset were plotted and analyzed separately than the rank score and raw score data. Figures 17 through 20 show the distribution of raw Eigen scores and raw Eigen-PC scores for both the benchmark and the validation datasets.

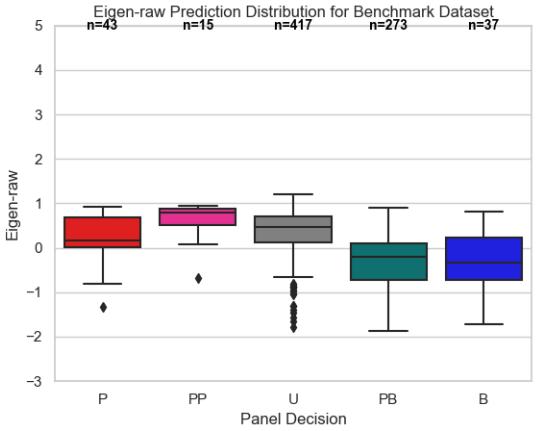

Figure 17: Distribution of Eigen raw scores in the benchmark dataset.

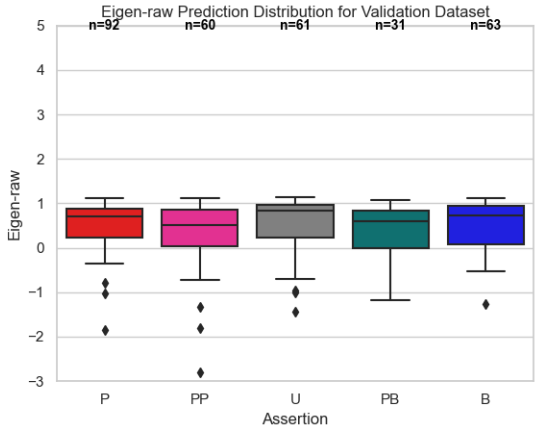

Figure 18: Distribution of Eigen raw scores in the validation dataset.

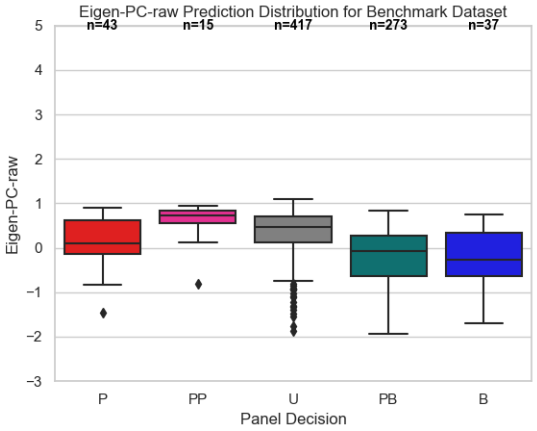

Figure 19: Distribution of Eigen-PC scores in the benchmark dataset.

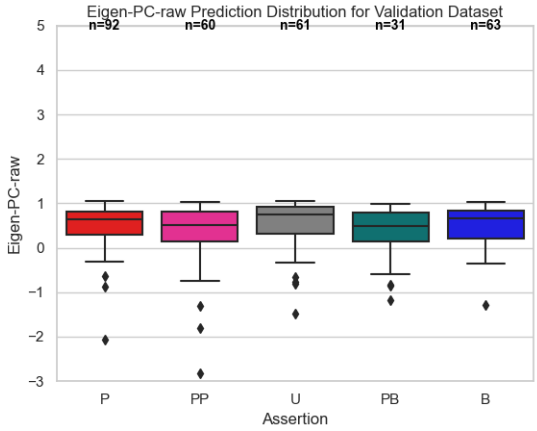

Figure 20: Distribution of Eigen-PC scores in the validation dataset.

Eigen raw scores and Eigen-PC raw scores both discriminated well between pathogenic and non-pathogenic variants in the benchmark dataset. In the benchmark dataset, the Pathogenic category has a lower mean than the Likely Pathogenic category, but both are above a score of 0 for most variants in those categories. The variants of uncertain significance have a relatively neutral mean score of 0.5 with a lot of denoted outliers below the lower quartile. The Likely Benign and Benign categories see scores that are well below 0, signifying more of the annotations the Eigen algorithm considered were negatively scoring the pathogenicity of those variants. Looking at the validation dataset, there is less of a difference in the means between the different categories. Also, the lower quartiles do not predict as low for Benign and Likely

Benign categories. Instead, there are outliers of lower scores in the Benign and Likely Benign categories. This suggests that the discrimination between pathogenicity classes was better in the benchmarking dataset than in the validation dataset.

The distributions of some algorithms that the ensemble algorithms use as features were also plotted. Figures 21 through 28 show the distribution of scores produced by MutationTaster, PolyPhen-2, SIFT, and FATHMM in both the benchmark dataset and the validation dataset.

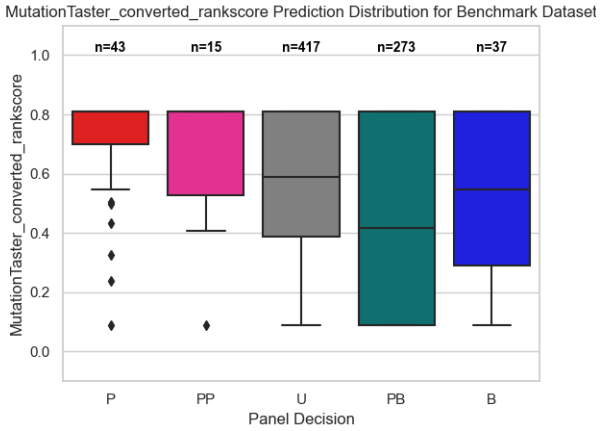

Figure 21: Distribution of MutationTaster converted rank scores in the benchmark dataset.

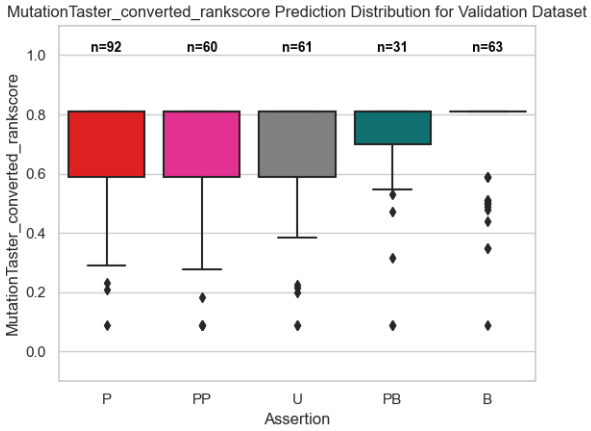

Figure 22: Distribution of MutationTaster converted rank scores in the validation dataset.

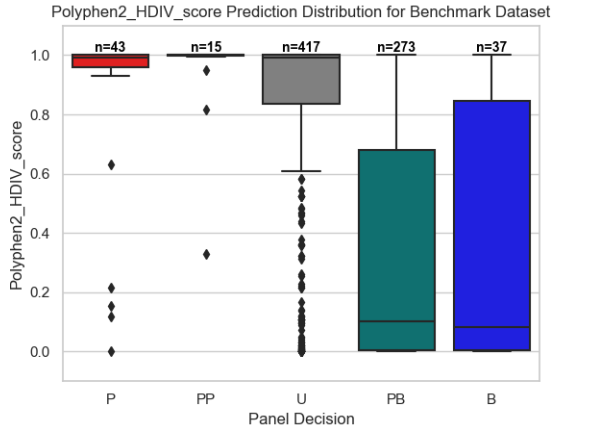

Figure 23: Distribution of PolyPhen-2 HDIV scores in the benchmark dataset.

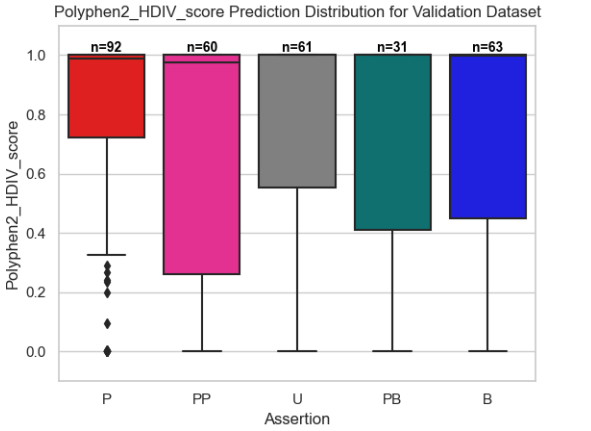

Figure 24: Distribution of PolyPhen-2 HDIV scores in the validation dataset.

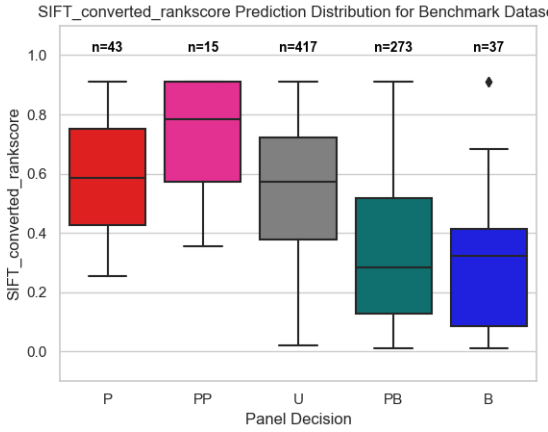

Figure 25: Distribution of SIFT converted rank scores in the benchmark dataset.

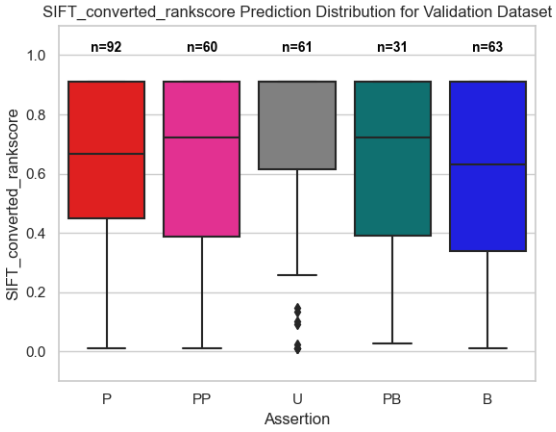

Figure 26: Distribution of SIFT converted rank scores in the validation dataset.

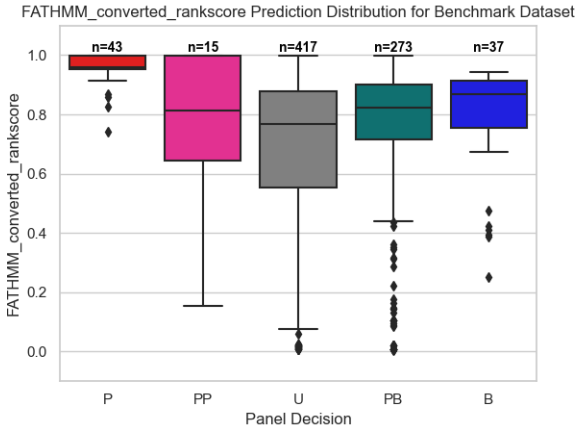

Figure 27: Distribution of FATHMM converted rank scores and in the benchmark dataset.

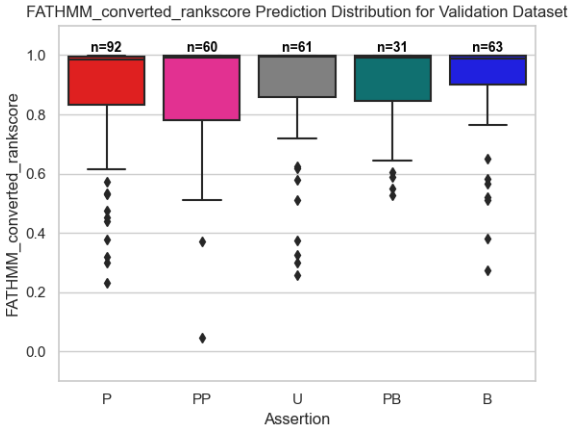

Figure 28: Distribution of FATHMM converted rank scores and in the validation dataset.

MutationTaster, PolyPhen-2, SIFT, and FATHMM all performed poorly on the validation dataset. The highest score in the Pathogenic category and the Benign category were approximately the same compared within each algorithm's predictions for the validation dataset. This suggests very little discrimination between Pathogenic and Benign categories when the distributions that are all high or widespread across the possible scores are also considered. MutationTaster did not show better discrimination between Pathogenic and Benign categories in the benchmarking dataset. On the contrary, PolyPhen-2, SIFT, and FATHMM were able to discriminate variants better in the benchmarking dataset than in the validation dataset.

None of the algorithms, whether they be an ensemble algorithm or a feature algorithm, performed better at discriminating pathogenicity classifications in the validation dataset than in the benchmark dataset. Using the box plots to see how the algorithms scored variants per classification later informed why the corresponding ROC curve and AUC measurement was poor or positive.

4.3 Benchmarking

ROC curves were plotted for each pathogenicity algorithm and the corresponding AUC was calculated. ROC curves were also plotted for each feature algorithm. These ROC curves and corresponding AUC measurement were compared to each other within groups. The algorithms were separated into a group of raw scores, a group of rank scores, a group of Eigen algorithm scores, a group of pathogenicity predictors that were used as features in the other ensemble algorithms. This was done so that each algorithm being compared was producing a similar output so that their accuracies were measuring the same goal. Then, each of these groups' ROC curves and AUC measurements were compared to one another on two plots. The ROC curves corresponding to the benchmark dataset were plotted on one graph and the ROC curves corresponding to the validation dataset were plotted on a second graph. Figures 29 and 30 show the ROC curves for CADD, DANN, MetaLR, and MetaSVM raw scores for the benchmark dataset and the validation dataset, respectively. Figures 31 and 32 show the ROC curves for CADD, DANN, MetaLR, and MetaSVM rank scores for the benchmark dataset and the validation dataset, respectively. Figures 33 and 34 are the Eigen group of algorithms, so the Eigen-PC raw scores and the Eigen raw scores, corresponding ROC curves for the benchmark dataset and the validation dataset, respectively. Finally, Figures 35 and 36 depict the ROC curves for the feature algorithms, including SIFT, PolyPhen-2, MutationTaster, and FATHMM, for the benchmark dataset and the validation dataset, respectively. The red dashed line in Figures 29 through 36 represent what a random classification would look like as a ROC curve.

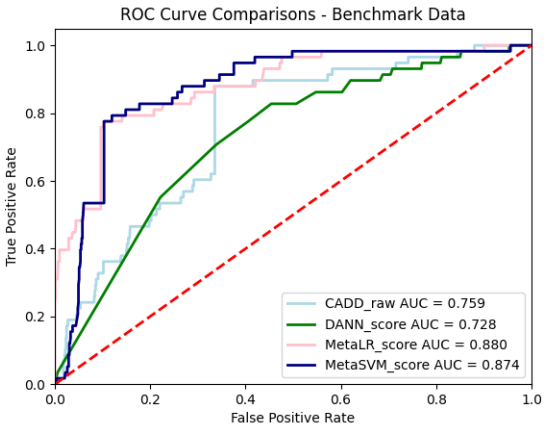

Figure 29: ROC curves for CADD, DANN, MetaLR, and MetaSVM raw scores with benchmark dataset, with corresponding AUC measurements of greater than 0.720.

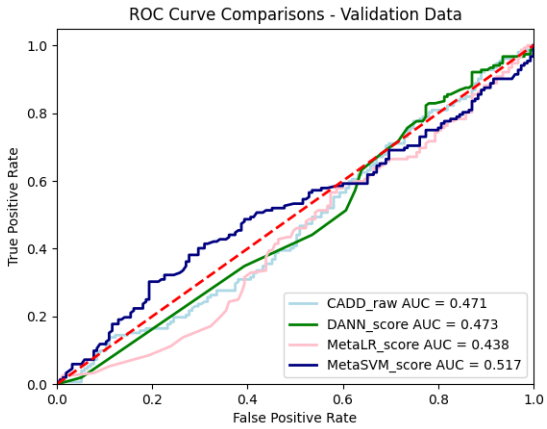

Figure 30: ROC curves for CADD, DANN, MetaLR, and MetaSVM raw scores with validation dataset.

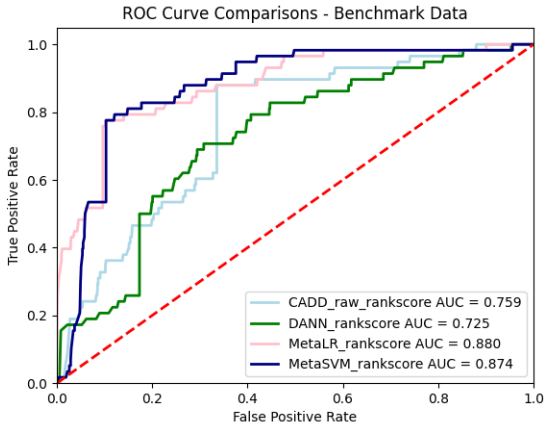

Figure 31: ROC curve for CADD, DANN, MetaLR, and MetaSVM rank scores with benchmark dataset, , with corresponding AUC measurements of greater than 0.720.

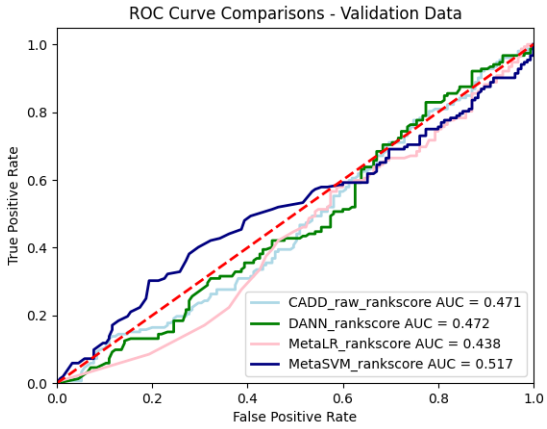

Figure 32: ROC curve for CADD, DANN, MetaLR, and MetaSVM rank scores with validation dataset

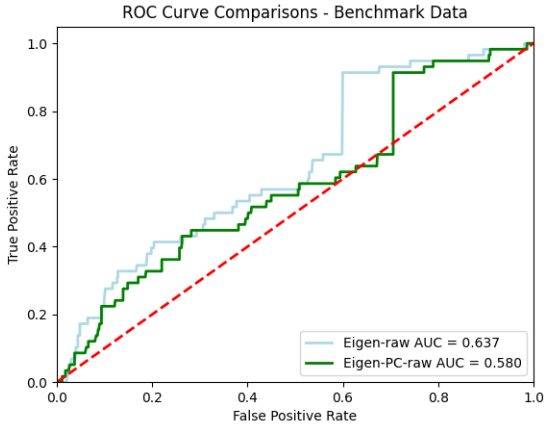

Figure 33: ROC curves for Eigen-raw and Eigen-PC-raw scores with benchmark dataset.

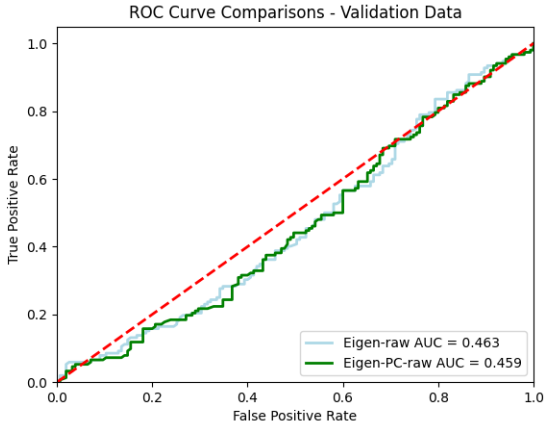

Figure 34: ROC curves for Eigen-raw and Eigen-PC-raw scores with validation dataset

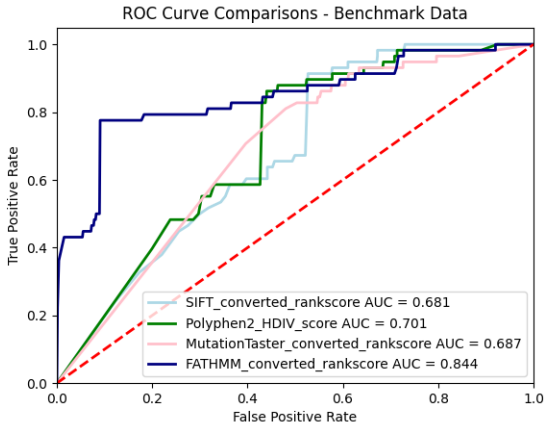

Figure 35: ROC curves for SIFT, PolyPhen-2, MutationTaster, and FATHMM scores with benchmark dataset.

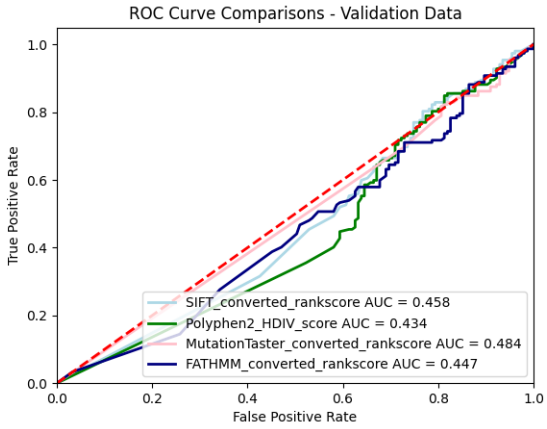

Figure 36: ROC curves for SIFT, PolyPhen-2, MutationTaster, and FATHMM scores with validation dataset.

Like how the box plots behaved, the ROC curves and AUC measurements were better for the benchmark dataset than for the validation dataset. The ROC curves for each of the algorithms corresponding to the validation had a range of accuracy from 0.434 to 0.517. This is poor accuracy and suggests that the algorithms are approximately as good as a random guess for classification. The ROC curves for each algorithm is extremely close to the red dashed line that symbolizes a random guess and sometimes the ROC curves dip below the random guess line. This can be seen in Figure 36, where the feature algorithms all had an AUC measurement below 0.500, meaning these algorithms performed worse than a random guess.

Each algorithm performed better on the benchmark dataset than on the validation dataset. In Figure 29, MetaLR and MetaSVM performed the best at classifying variants as Pathogenic and non-Pathogenic, with AUC measurements of 0.880 and 0.874, respectively. CADD and DANN also performed well, scoring AUC measurements of 0.759 and 0.728, respectively. These AUC scores are approximately the same as how these algorithms have performed on datasets other than their own training and testing datasets. Similarly, in Figure 31, the MetaLR and MetaSVM rank scores score the exact same AUC measurement, 0.880 and 0.874, respectively again. Similarly, CADD and DANN produce AUC measurements of 0.759 and 0.725 respectively using rank scores as classifiers. For all four of these algorithms, the AUC score between the raw score that the algorithm produces and the rank score that the algorithm produces was nearly identical or differed by 0.003. For this dataset, the discrimination between Pathogenic variants and non-Pathogenic variants did not improve when considering rank score versus raw score.

In Figure 33, the Eigen raw score and the Eigen-PC raw score produce AUC measurements of only 0.637 and 0.580 respectively when classifying variants in the benchmark dataset. Despite Eigen being an unsupervised algorithm, it did not perform better than MetaLR, MetaSVM, CADD, or DANN rank scores or raw scores at discriminating between Pathogenic and non-Pathogenic variants.

In Figure 35, the feature algorithms SIFT, PolyPhen-2, MutationTaster, and FATHMM all perform well at classifying Pathogenic and non-Pathogenic variants. The converted rank score produced by FATHMM performed the best at discriminating between Pathogenic and non-Pathogenic variants, with an AUC measurement of 0.844. The converted rank score produced by

the SIFT algorithm, the HDIV score produced by PolyPhen2, and the converted rank score produced by MutationTaster scored similar AUCs at 0.681, 0.701, and 0.687, respectively. MetaLR and MetaSVM use FATHMM scores as a feature in their ensemble algorithm, which may imply that MetaLR and MetaSVM produced the best AUC measurements out of all the ensemble algorithms because FATHMM produced the best AUC measurements out of all the feature algorithms on this specific dataset. On a similar note, since all the feature algorithms performed poorly at classifying variants in the validation dataset, this could contribute to why the ensemble algorithms also had difficulty and poor scores classifying the variants in the validation dataset. Overall, the ROC curves and AUC measurements from testing the algorithms with the benchmark dataset were better than the ROC curves and AUC measurements from testing the algorithms with the validation dataset.

5 Conclusion and Recommendation

In silico methods, or prediction algorithms, are a useful tool in reviewing a variant's pathogenicity classification. In this study, ensemble pathogenicity prediction algorithms classify pathogenic variants more accurately than algorithms that do not utilize other existing pathogenicity prediction algorithms as features. However, using other algorithms as features creates a cascading effect of dependencies on training data, features used, and tuned parameters for the ensemble algorithms. Ultimately, as with any machine learning problem, the original training and testing dataset can cause overfitting and bias in a model, leading to poor accuracy when used to test more diverse datasets.

Based on how the ensemble algorithms performed on the benchmark dataset compared to the validation dataset, there are two things that could have occurred in this study. First, the validation dataset may be incorrectly labeled for some variants or have missing information that could not be annotated based on the original given information. This means that the validation dataset chosen to benchmark these algorithms was the cause of the poor accuracy, not the algorithms themselves. Despite ClinGen data being reviewed before acceptance into the public database, it is nonetheless a public resource, and by not manually curating a dataset from their repository like this study failed to do, error could be introduced. Code was reviewed to ensure that the annotation pipeline and graph creation was not the cause for poor ROC curves with regard to the validation dataset. Should this study be repeated in testing the ClinGen data, variants should be curated from the ClinGen database that have ample amounts of evidence to inform each assertion. This was not performed in the current methodology explained in this paper.

The other reason for poor performance on the validation dataset could be that the algorithms performed poorly due to the underlying algorithms used as features (*i.e.*, SIFT, PolyPhen-2, etc.) performing poorly first. This poor performance in the feature algorithms could be due to their original training datasets having been mislabeled, though this is unlikely due to no news of this occurring. It is more likely the training data was carefully curated, and the validation dataset was different characteristically. This would suggest a possible overfitting of the original model. Additionally, with some of these algorithms nearly twenty years old now, it could be that the data they trained on would be classified differently now should it be under review again. This methodology did not delve deeply into the algorithms' original training datasets to examine how closely those variants compared to the benchmark dataset variants and the validation dataset variants used in this study. Should this study be repeated, more time should be spent on manually curating datasets for benchmarking purposes and taking careful note of how the dataset is similar to or different than each algorithm's original training and testing datasets.

Pathogenicity algorithms are only a piece of the puzzle. They are a useful tool and with the help of a pipeline that quickly annotates data with multiple algorithms, multiple pathogenicity algorithms can be used to inform a final pathogenicity classification for variants. According to ACMG standards, pathogenicity algorithms should not be the sole influencer of a classification but should guide clinicians and researchers towards a possible classification that can further be supported by academic literature, expert opinion, and database information. This is especially important if the pathogenicity algorithm was not trained on data specific to the dataset that a clinician or clinical researcher is dealing with. Poor performance of an algorithm, such as the performance of all algorithms in this study on the validation dataset, could influence

a classification to be incorrect if the person reviewing the classification does not check other outside sources.

From this comparative analysis, MetaLR and MetaSVM perform the best. They both also produce a rank score and a raw score for a variants' pathogenicity, meaning either could be used to inform a formal classification. CADD and DANN performed similarly, and both also produce a rank score and a raw score. It is recommended that CADD, DANN, MetaLR, and MetaSVM be considered when investigating variant pathogenicity. Eigen did not perform as well as CADD, DANN, MetaLR, or MetaSVM, but did not perform poorly enough to be completely discounted. Instead, it is recommended that Eigen be used to validate other algorithm predictions, or to be used on diverse datasets, since it is an unsupervised algorithm. Additionally, if the algorithms used as features in CADD, DANN, MetaLR, MetaSVM, or other algorithms are available to be annotated, it is recommended that they are. As it was seen in the Results section of this paper, knowing what the underlying algorithm scores are for a variant can support and suggest why an ensemble algorithm is behaving a certain way. Using a diverse set of algorithmic predictions to inform a pathogenicity classification alongside academic literature, quality databases, and expert opinion will lead to the most accurate classification possible according to ACMG standards.

Bibliography

Adzhubei, I. A., Schmidt, S., Peshkin, L., Ramensky, V. E., Gerasimova, A., Bork, P., Kondrashov, A. S., & Sunyaev, S. R. (2010). A method and server for predicting damaging missense mutations. *Nature methods*, *7*(4), 248–249. https://doi.org/10.1038/nmeth0410-248

Dhamija, S., & Menon, M. B. (2018). Non-coding transcript variants of protein-coding genes–what are they good for? In *RNA Biology* (Vol. 15, Issue 8, pp. 1025–1031). Taylor and Francis Inc. https://doi.org/10.1080/15476286.2018.1511675

DNA Sequencing Costs: Data. (n.d.). Retrieved September 1, 2020, from https://www.genome.gov/about-genomics/fact-sheets/DNA-Sequencing-Costs-Data

DNA Sequencing Fact Sheet. (n.d.). Retrieved September 1, 2020, from https://www.genome.gov/about-genomics/fact-sheets/DNA-Sequencing-Fact-Sheet

Dong, C., Wei, P., Jian, X., Gibbs, R., Boerwinkle, E., Wang, K., & Liu, X. (2015). Comparison and integration of deleteriousness prediction methods for nonsynonymous SNVs in whole exome sequencing studies. Human Molecular Genetics, 24(8), 2125–2137. https://doi.org/10.1093/hmg/ddu733

Ebiki, M., Okazaki, T., Kai, M., Adachi, K., & Nanba, E. (2019). *Comparison of Causative Variant Prioritization Tools Using Next-generation Sequencing Data in Japanese Patients with Mendelian Disorders*. https://doi.org/10.33160/yam.2019.09.001

Eilbeck, K., Quinlan, A., & Yandell, M. (2017). Settling the score: Variant prioritization and Mendelian disease. In *Nature Reviews Genetics* (Vol. 18, Issue 10, pp. 599–612). Nature Publishing Group. https://doi.org/10.1038/nrg.2017.52

G. C. C. L. Cardenas, R., D. Linhares, N., L. Ferreira, R., & Pena, S. D. J. (2017). Mendel,MD: A user-friendly open-source web tool for analyzing WES and WGS in the diagnosis of patients with Mendelian disorders. *PLOS Computational Biology*, *13*(6), e1005520. https://doi.org/10.1371/journal.pcbi.1005520

González-Pérez, A., & López-Bigas, N. (2011). Improving the assessment of the outcome of nonsynonymous SNVs with a consensus deleteriousness score, Condel. *American Journal of Human Genetics*, *88*(4), 440–449. https://doi.org/10.1016/j.ajhg.2011.03.004

Holt, J. M., Wilk, B., Birch, C. L., Brown, D. M., Gajapathy, M., Moss, A. C., Sosonkina, N., Wilk, M. A., Anderson, J. A., Harris, J. M., Kelly, J. M., Shaterferdosian, F., Uno-Antonison, A. E., Weborg, A., Acosta, M. T., Adam, M., Adams, D. R., Agrawal, P. B., Alejandro, M. E., … Worthey, E. A. (2019). VarSight: Prioritizing clinically reported variants with binary classification algorithms. *BMC Bioinformatics*, *20*(1). https://doi.org/10.1186/s12859-019-3026-8

Ioannidis, N. M., Rothstein, J. H., Pejaver, V., Middha, S., McDonnell, S. K., Baheti, S., Musolf, A., Li, Q., Holzinger, E., Karyadi, D., Cannon-Albright, L. A., Teerlink, C. C., Stanford, J. L., Isaacs, W. B., Xu, J., Cooney, K. A., Lange, E. M., Schleutker, J., Carpten, J. D., … Sieh, W. (2016). REVEL: An Ensemble Method for Predicting the Pathogenicity of Rare Missense Variants. *American Journal of Human Genetics*, *99*(4), 877–885. https://doi.org/10.1016/j.ajhg.2016.08.016

Kim, S., Jhong, J.-H., Lee, J., & Koo, J.-Y. (2017). Meta-analytic support vector machine for integrating multiple omics data. *BioData Mining*, *10*(2). https://doi.org/10.1186/s13040-017-0126-8

Krier, J. B., Kalia, S. S., & Green, R. C. (2016). Genomic sequencing in clinical practice: Applications, challenges, and opportunities. *Dialogues in Clinical Neuroscience*, *18*(3), 299–312. www.dialogues-cns.org

Kumar P, Henikoff S, Ng PC. Predicting the effects of coding non-synonymous variants on protein function using the SIFT algorithm. Nat Protoc. 2009;4(7):1073-81. doi: 10.1038/nprot.2009.86. Epub 2009 Jun 25. PMID: 19561590.

Landrum, M. J., Chitipiralla, S., Brown, G. R., Chen, C., Gu, B., Hart, J., Hoffman, D., Jang, W., Kaur, K., Liu, C., Lyoshin, V., Maddipatla, Z., Maiti, R., Mitchell, J., O'Leary, N., Riley, G. R., Shi, W., Zhou, G., Schneider, V., Maglott, D., Holmes, J.B., Kattman, B. L. ClinVar: improvements to accessing data. Nucleic Acids Res. 2020;48(D1):D835-D844. doi: 10.1093/nar/gkz972.

Li, M. X., Gui, H. S., Kwan, J. S. H., Bao, S. Y., & Sham, P. C. (2012). A comprehensive framework for prioritizing variants in exome sequencing studies of Mendelian diseases. *Nucleic Acids Research*, *40*(7), e53–e53. https://doi.org/10.1093/nar/gkr1257

Mccallum, K., Buxbaum, J., & Genet Author manuscript, N. (2016). A SPECTRAL APPROACH INTEGRATING FUNCTIONAL GENOMIC ANNOTATIONS FOR CODING AND NONCODING VARIANTS IULIANA IONITA-LAZA HHS Public Access Author manuscript. *Nat Genet*, *48*(2), 214–220. https://doi.org/10.1038/ng.3477

McKinney, Wes. (2010) Data Structures for Statistical Computing in Python. *Proceedings of the 9th Python in Science Conference*, 51-56.

Ng, P. C., & Henikoff, S. (2001). Predicting deleterious amino acid substitutions. *Genome Research*, *11*(5), 863–874. https://doi.org/10

Quang, D., Chen, Y., & Xie, X. (2015). DANN: A deep learning approach for annotating the pathogenicity of genetic variants. *Bioinformatics*, *31*(5), 761–763. https://doi.org/10.1093/bioinformatics/btu703

Rehm, H. L., Berg, J. S., Brooks, L. D., Bustamante, C. D., Evans, J. P., Landrum, M. J., Ledbetter, D. H., Maglott, D. R., Martin, C. L., Nussbaum, R. L., Plon, S. E., Ramos, E. M., Sherry, S. T., & Watson, M. S. (2015). ClinGen - The clinical genome resource. *New England Journal of Medicine*, *372*(23), 2235–2242. https://doi.org/10.1056/NEJMsr1406261

Rentzsch, P., Witten, D., Cooper, G. M., Shendure, J., & Kircher, M. (2019). CADD: Predicting the deleteriousness of variants throughout the human genome. *Nucleic Acids Research*, *47*(D1), D886–D894. https://doi.org/10.1093/nar/gky1016

Reva, B., Antipin, Y., & Sander, C. (2011). Predicting the functional impact of protein mutations: application to cancer genomics. *Nucleic acids research*, *39*(17), e118. https://doi.org/10.1093/nar/gkr407

Richards, S., Aziz, N., Bale, S. *et al.* (2015). Standards and guidelines for the interpretation of sequence variants: a joint consensus recommendation of the American College of Medical Genetics and Genomics and the Association for Molecular Pathology. *Genet Med* **17,** 405–423. https://doi.org/10.1038/gim.2015.30

Schwarz, J. M., Rödelsperger, C., Schuelke, M., & Seelow, D. (2010). MutationTaster evaluates disease-causing potential of sequence alterations. In *Nature Methods* (Vol. 7, Issue 8, pp. 575–576). Nature Publishing Group. https://doi.org/10.1038/nmeth0810-575

Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.

Sequence Variant Nomenclature. (2020, May 1). Retrieved October 02, 2020, from http://varnomen.hgvs.org/

Shihab, H. A., Gough, J., Cooper, D. N., Stenson, P. D., Barker, G. L., Edwards, K. J., Day, I. N., & Gaunt, T. R. (2013). Predicting the functional, molecular, and phenotypic consequences of amino acid substitutions using hidden Markov models. *Human mutation*, *34*(1), 57–65. https://doi.org/10.1002/humu.22225

Wang, K. (2019). ANNOVAR Main Package. Retrieved October 13, 2020, from http://annovar.openbioinformatics.org/en/latest/user-guide/download/

Wang, K., Li, M., & Hakonarson, H. (2010). ANNOVAR: Functional annotation of genetic variants from high-throughput sequencing data. *Nucleic Acids Research*, *38*(16), e164. https://doi.org/10.1093/nar/gkq603

Xavier, A., Scott, R. J., & Talseth-Palmer, B. A. (2019). TAPES: A tool for assessment and prioritisation in exome studies. *PLOS Computational Biology*, *15*(10), e1007453. https://doi.org/10.1371/journal.pcbi.1007453

Yandell, M., & Ence, D. (2012). A beginner's guide to eukaryotic genome annotation. In *Nature Reviews Genetics* (Vol. 13, Issue 5, pp. 329–342). Nature Publishing Group. https://doi.org/10.1038/nrg3174

Zeng, Z., & Bromberg, Y. (2019). Predicting Functional Effects of Synonymous Variants: A Systematic Review and Perspectives. In *Frontiers in Genetics* (Vol. 10, p. 914). Frontiers Media S.A. https://doi.org/10.3389/fgene.2019.00914

Zhang J, Walsh MF, Wu G, Edmonson MN, Gruber TA, Easton J, et al. Germline Mutations in Predisposition Genes in Pediatric Cancer. N Engl J Med. 2015;373(24):2336–46. Epub 2015/11/19. pmid:26580448

Appendix A – Online Sources for Ensemble Learners

METHOD	SOFTWARE LINK	CITATION FOR RELATED PAPER
CADD	https://cadd.gs.washington.edu/download	Rentzsch, P., Witten, D., Cooper, G. M., Shendure, J., & Kircher, M. (2019). CADD: Predicting the deleteriousness of variants throughout the human genome. *Nucleic Acids Research*, *47*(D1), D886–D894. https://doi.org/10.1093/nar/gky1016
DANN	https://cbcl.ics.uci.edu/public_data/DANN/	Quang, D., Chen, Y., & Xie, X. (2015). DANN: A deep learning approach for annotating the pathogenicity of genetic variants. *Bioinformatics*, *31*(5), 761–763. https://doi.org/10.1093/bioinformatics/btu703
REVEL	https://sites.google.com/site/revelgenomics/downloads	Ioannidis, N. M., Rothstein, J. H., Pejaver, V., Middha, S., McDonnell, S. K., Baheti, S., Musolf, A., Li, Q., Holzinger, E., Karyadi, D., Cannon-Albright, L. A., Teerlink, C. C., Stanford, J. L., Isaacs, W. B., Xu, J., Cooney, K. A., Lange, E. M., Schleutker, J., Carpten, J. D., … Sieh, W. (2016). REVEL: An Ensemble Method for Predicting the Pathogenicity of Rare Missense Variants. *American Journal of Human Genetics*, *99*(4), 877–885. https://doi.org/10.1016/j.ajhg.2016.08.016
Condel	http://bbglab.irbbarcelona.org/fannsdb/home	González-Pérez, A., & López-Bigas, N. (2011). Improving the assessment of the outcome of nonsynonymous SNVs with a consensus deleteriousness score, Condel. *American Journal of Human Genetics*, *88*(4), 440–449. https://doi.org/10.1016/j.ajhg.2011.03.004
Eigen	http://www.columbia.edu/~ii2135/information_eigen.html	Mccallum, K., Buxbaum, J., & Genet Author manuscript, N. (2016). A SPECTRAL APPROACH INTEGRATING FUNCTIONAL GENOMIC ANNOTATIONS FOR CODING AND NONCODING VARIANTS IULIANA IONITA-LAZA HHS Public Access Author manuscript. *Nat Genet*, *48*(2), 214–220. https://doi.org/10.1038/ng.3477
MetaLR & MetaSVM	http://annovar.openbioinformatics.org/en/latest/user-guide/filter/#-metalr-annotation http://annovar.openbioinformatics.org/en/latest/user-guide/filter/#-metasvm-annotation	Dong, C., Wei, P., Jian, X., Gibbs, R., Boerwinkle, E., Wang, K., & Liu, X. (2015). Comparison and integration of deleteriousness prediction methods for nonsynonymous SNVs in whole exome sequencing studies. Human Molecular Genetics, 24(8), 2125–2137. https://doi.org/10.1093/hmg/ddu733
TAPES	https://github.com/a-xavier/tapes	Xavier, A., Scott, R. J., & Talseth-Palmer, B. A. (2019). TAPES: A tool for assessment and prioritisation in exome studies. *PLOS Computational Biology*, *15*(10), e1007453. https://doi.org/10.1371/journal.pcbi.1007453