Yunus Emre IŞIK

A Ph.D. Thesis

AGU 2024

MACHINE LEARNING METHODS FOR DETECTING GENETIC AND INFECTIOUS DISEASES

A THESIS
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF ABDULLAH GUL UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Ph. D.

By
Yunus Emre Işık
March 2024

MACHINE LEARNING METHODS FOR DETECTING GENETIC AND INFECTIOUS DISEASES

A THESIS
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE OF ABDULLAH GUL UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Ph. D.

By
Yunus Emre Işık
March 2024

SCIENTIFIC ETHICS COMPLIANCE

I hereby declare that all information in this document has been obtained in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name-Surname: Yunus Emre Işık
Signature :

REGULATORY COMPLIANCE

Ph.D. thesis titled Machine learning methods for detecting genetic and infectious diseases has been prepared in accordance with the Thesis Writing Guidelines of the Abdullah Gül University, Graduate School of Engineering & Science.

Prepared By	Advisor
Yunus Emre IŞIK	Assoc. Prof. Zafer AYDIN

Head of the Electrical and Computer Engineering Program
Asst. Prof. Samet GÜLER

ACCEPTANCE AND APPROVAL

Ph.D. thesis titled Machine learning methods for detecting genetic and infectious diseases and prepared by Yunus Emre Işık has been accepted by the jury in the Electrical and Computer Engineering Graduate Program at Abdullah Gül University, Graduate School of Engineering & Science.

27 / 03 / 2024

JURY:

Advisor : Assoc. Prof. Zafer AYDIN …….

Member : Asst. Prof. Burcu BAKIR GÜNGÖR …….

Member : Assoc. Prof. Özkan Ufuk NALBANTOĞLU ……

Member : Asst. Prof. Bekir Hakan AKSEBZECİ …….

Member : Asst. Prof. Dinçer GÖKSÜLÜK …….

APPROVAL:

The acceptance of this Ph.D. thesis has been approved by the decision of the Abdullah Gül University, Graduate School of Engineering & Science, Executive Board dated ….. /….. / …….. and numbered ……..……. .

…….. /…….. / ……..

(Date)

Graduate School Dean
Prof. Dr. İrfan ALAN

ABSTRACT

MACHINE LEARNING METHODS FOR DETECTING GENETIC AND INFECTIOUS DISEASES

Yunus Emre IŞIK
Ph.D. in Electrical and Computer Engineering
Advisor: Assoc. Prof. Zafer AYDIN

March 2024

Completion of the whole human genome in the 2003 has led to various advances in many fields, particularly in biology, genetics, health sciences, treatment, and pharmacology. In the following years, spread of faster and cheaper sequencing technologies has enabled us to extract and analyze genetic profiles of individuals digitally. Consequently, individual-specific forecasting and personalized treatment and precision medicine-, what once seemed like science fiction, have become more and more real. In both approaches, one of the crucial steps is identifying the presence of diseases using individual-specific genetic data. This thesis aims to comprehensively and comparatively evaluate the predictive performance of machine learning methods for Behçet's disease and respiratory infections. Additionally, feature selection methods were employed to identify the genetic factors (such as SNPs and genes) associated with disease presence for both diseases. Furthermore, the usability of selected features depending on biological pathway-driven active subnetworks listed in the literature was analyzed for the prediction of Behçet's disease. For the respiratory infection prediction problem, on the other hand, the prediction performance of features calculated by single-sample gene set enrichment analysis (ssGSEA) was evaluated using different machine learning methods. As the data types used in both experiments were different (genome-wide association studies data, gene expression profiles), the performance of machine learning approaches on different data types was also observed. It is hoped that the findings of both experiments will contribute to future machine learning based disease prediction studies.

Keywords: Disease prediction, Machine Learning, Behçet's Disease Prediction, Respiratory Infection Prediction, Feature Selection and Representation

ÖZET

GENETİK VE ENFEKSİYON HASTALIKLARININ TESPİTİ İÇİN MAKİNE ÖĞRENMESİ YÖNTEMLERİ

Yunus Emre IŞIK
Elektrik ve Bilgisayar Mühendisliği Anabilim Dalı Doktora
Tez Yöneticisi: Doç. Dr. Zafer AYDIN

Mart 2024

2003 yılında insan genomunun tamamen dizilenebilmesi, başta biyoloji ve genetik bilimi olmak üzere sağlık bilim, tedavi ve farmakoloji gibi birçok farklı alanda yeni gelişmelerin ortaya çıkmasına neden olmuştur. İlerleyen yıllarda hızlı ve daha ucuz dizileme teknolojilerinin yaygınlaşmasıyla bireylerin genetik profillerinin çıkartılarak dijital ortamda işlenebilmesi mümkün hale gelmiştir. Böylelikle eski zamanlarda bilim-kurgu gibi görünen, bireylere özgü tahmin ve tedavi belirlenmesi, başka bir deyişle kişileştirilmiş ve hassas tıp yaklaşımı hız kazanmıştır. Her iki yaklaşımda da en önemli aşama ise hastalığın bireye özgü genetik veriler kullanılarak belirlenmesidir. Bu tez çalışması Behçet hastalığı ve solunum yolu enfeksiyonu olmak üzere iki farklı türde hastalık için makine öğrenmesi yöntemlerinin tahmin performansını kapsamlı ve karşılaştırmalı olarak değerlendirmeyi amaçlamaktadır. Ayrıca öznitelik seçme yöntemleriyle hastalık tahmininde önemli rol oynayan genetik faktörler (SNP, Gene) her iki hastalık içinde ayrı ayrı belirlenmeye çalışılmıştır. Bunun yanı sıra Behçet hastalığının tahminlenmesinde literatürde yer alan biyolojik yolak temelli aktif-ağlar kullanılarak seçilen özniteliklerin kullanılabilirliği analiz edilmiştir. Öte yandan solunum yolu enfeksiyon tahmin probleminde ise, örneklem bazında uygulanan gen seti zenginleştirme analizi sonrası elde edilen skorların, örneklemlerin temsil edilmesinde ne kadar başarılı olduğu makine öğrenmesi kullanılarak ortaya koyulmuştur. Her iki deneyde kullanılan veri tipleri de farklı olduğu için (genom çapında ilişkilendirme çalışmaları verisi, gen ifadesi profilleri), makine öğrenmesi yaklaşımlarının farklı veri türlerindeki performansları da gözlemlenmiştir. Her iki deney sonucunda elde edilen çıktıların makine öğrenmesi temelli hastalık tahminleme çalışmalarına katkı sağlayacağı umulmaktadır.

Anahtar kelimeler: Hastalık tespiti, Makine Öğrenmesi, Behçet Hastalığı Tahmini, Solunum Yolu Enfeksiyon Tahmini, Öznitelik Seçimi ve Temsili

Acknowledgements

First of all, I would like to thank Assoc. Prof. Zafer AYDIN for accepting me as a Ph.D. student and for having supported me at every stage.

I would like to thank Asst. Prof. Burcu BAKIR GÜNGÖR for introducing me to different aspects of the bioinformatics field, providing support whenever I needed it and taking valuable time to follow my thesis progress.

I am very thankful to Assoc. Prof. Özkan Ufuk NALBANTOĞLU for taking valuable time to follow my progress and giving me advice.

Finally, I would like to gratefully thank my family who waited patiently for me and supported me at every moment of life.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AUPRC	Area Under the Precision and Recall Curve
BD	Behçet's Disease
CFS	Correlation-based Feature Selection
DKSS	Domain Knowledge Based Subset Selection
ES	Enrichment Score
FS	Feature Selection
GEO	Gene Expression Omnibus
GNB	Gaussian Naive Bayes
GO	Gene Ontology
GSEA	Gene Set Enrichment Analysis
GWAS	Genome-Wide Association Studies
HRV	Human Rhinovirus
HTS	High-Throughput Sequencing
KNN	k-Nearest Neighbor
LGB	Light Gradient Boosting Machine
LR	Logistic Regression
MRMR	Minimum Redundancy—Maximum Relevance
MSigDB	Molecular Signatures Database
NGS	Next-Generation Sequencing
NF	Number of Features
ORA	Over Representation Analysis
RF	Random Forest
RSV	Respiratory Syncytial Virus
SNP	Single Nucleotide Polymorphism
ssGSEA	Single-sample Gene Set Enrichment Analysis
SVM	Support Vector Machines
WHO	World Health Organization
XGB	Extreme Boosting Machines

To my family

Chapter 1

Introduction

The sequencing of the entire human genome in 2003 as part of the Human Genome Project has profoundly altered the perspective on biological science and marked the beginning of a new era in genomic research. Although the project lasted more than a decade, involved thousands of scientists from many fields, and reached a cost of 3 billion dollars, it is priceless in the potential opportunities it holds for human health. The human genome is the collection of DNA that exists in every cell of our body. The DNA is arranged into structures called chromosomes, and each chromosome contains numerous genes. Genes are similar to individual chapters in our entire DNA, and they are responsible for everything about us such as height, eye color, and even our susceptibility to diseases. In other words, the genome, and genes' genetic reflection of us makes us unique and deduce how our bodies respond to biological influences. Hence sequencing of all the human genome has paved the way for a lot of opportunities in biology, medicine and healthcare such as understanding biological processes better, determining effective treatment strategies, revealing the underlying grounds for genetic differences. Following the announcement of the completion of the Human Genome Project, the National Human Genome Research Institute (NGHRI) launched a $70 million DNA sequencing technology initiative with the goal of a $1,000 cost per human genome in 10 years, and a flurry of high-throughput sequencing (HTS) technologies emerged [1].

Following years, the rise of next-generation sequencing (NGS) with increasing computing power and storage capacity has significantly reduced the cost of genetic sequencing. It has also brought about a paradigm shift in genomics research, providing unprecedented capabilities for the analysis of DNA and RNA molecules and leading to the emergence of new “omics” data [2]. The “omics” data refers to large-scale data generated from different-level biological processes such as genomics, transcriptomics, proteomics, and metabolomics since these processes provide unique types of information, expressed with specific names. For instance, genomics data relates to the DNA sequence

of organisms, while transcriptomics indicates expression levels of genes in a particular cell or tissue. Proteomics data is associated with proteins produced by an organism, while metabolomics focuses on small molecules or metabolites within a cell, tissue, or organism, such as sugars, amino acids, lipids, etc. In addition, post-genomic advances in biology and medicine in recent years have led to the emerge of new omics type such as pharmacogenomics (effect of drugs on the host and host's response), nutrigenomics (a fast-growing discipline that focuses on identifying the genetic factors that influence the body's response to diet and studies how the bioactive components of food affect gene expression), phylogenomics (analysis using genomic data and evolutionary reconstructions, especially phylogenetics) [3].

As the NGS technologies progressed and attracted the interest of researchers, the amount of genetic data generated by several omics studies continued to grow enormously, and led to the emergence of new ideas in medicine, treatment, biology and diagnosis such as uncovering genetic characteristics of diseases, and forecasting susceptibility to specific diseases. But on the other hand, the increase in data volume also highlighted insufficiency of traditional biological and genetic analysis because handling data generated by high-throughput technologies is a time-consuming and labor-intensive process, especially if the data is complex and large-scale. For instance, predicting the presence of a disease in an individual using traditional methods may be possible if the disease depends on only one or a few genes. But what if hundreds or thousands of expressed genes from both controls and patients need to be evaluated and analyzed together and comparatively to predict the presence of a disease? In such a scenario, traditional approaches, such as simple statistical analysis, would presumably demonstrate low-accuracy prediction about disease presence, due to the fact that it cannot uncover hidden patterns or associations. As a consequence, there is a need for advanced computational approaches that can analyze large amounts of genetic data more efficiently and quickly to yield meaningful results, such as machine learning algorithms [4].

Machine learning (ML) is a branch of computer science and artificial intelligence (AI) that uses algorithms to learn from data and then make decisions or predictions based on that learning. ML algorithms can be broadly divided into two categories: supervised and unsupervised. While supervised learning attempts to map input samples to their respective outputs, unsupervised learning identifies hidden patterns in unlabeled data [5]. Since this approach can take genetic data as input and has the ability to detect patterns by

learning from the input, it is widely used for genetic data in the fields of disease diagnosis, medicine and other medical activities, especially in prediction and classification tasks. In addition, owing to the development of open source ML packages and active research in this field, researchers can easily implement ML models to build predictive models of complex data [6].

The advancement and the easy of application of ML-based models have paved the way for personalized and precision medicine approaches to become widespread. Personalized medicine, also referred to as individualized medicine, simply means the prescription of specific treatments and therapeutics best suited for an individual taking into consideration both genetic and environmental factors that influence response to therapy. Precision medicine, on the other hand, is defined as cutting-edge molecular profiling that helps determine precise diagnostic, prognostic, and therapeutic strategies are precisely tailored to each patient's requirements [7]. Although both terms are used interchangeably personalized medicine is more related to the integration of diagnosis with therapy, screening, prevention, prognosis, and monitoring of treatment as future trends in medicine, while precision medicine helps establish accurate diagnostic, prognostic, and therapeutic approaches [8].

A common step both in personalized and precision medicine is same, the diagnosis of disease, i.e. the prediction of its presence using AI-based methods . This is because, while many diseases show symptoms in early stages, some of them such as cancer, kidney damages remain unidentified in their developing stages. The earlier a disease is predicted, the easier it becomes to cure it and even prevent it. Hence, predictive modeling provides a huge step forward in medical science in preventing diseases [9]. In addition, prediction of disease presence, progression or diagnosis should be based on individual signals such as genetic and clinical profiles, according to both precision and personalized medicine. Hence, AI-based approaches such as machine learning fill the gap in the need for cutting-edge methods that can handle the large amount of data for diagnosing diseases, predicting the presence of diseases, or identifying factors affecting diseases.

This thesis includes comprehensive experiments to investigate the performance of artificial intelligence-based approaches such as machine learning and feature selection in predicting the presence of disease and identifying the factors that have an impact on the prediction of disease, which is an important stage in personalized and precision medicine.

Although this problem has been studied in the literature for many different diseases, most studies have focused on the performance of a particular algorithm. In contrast, our work includes a comprehensive and a comparative analysis of both machine learning and feature selection algorithms in disease prediction problem. Furthermore, our study cover two types of diseases: genetic and infectious. Behçet's disease was chosen as genetic disease. The motivation for choosing this disease is that the studies for predicting Behçet's disease are quite limited in the literature. Additionally, the availability of a large data set for Behçet's disease has provided opportunities to compare different methods. The second study, on the other hand, is related to the prediction of respiratory diseases, which have affected human history in many periods and are still not completely cured. Especially after the COVID-19 pandemic, one of the greatest disasters since the turn of the millennium, the prediction of respiratory diseases has become even more important. An Interesting side of respiratory infection is that while most infections result in mild symptoms such as runny nose, sore throat, and headache, some individuals remain asymptomatic despite exposed to same respiratory viruses. Hence, in the experiments of this thesis, we aimed to develop predictive models that can predict both the presence of infection and whether an individual develops symptoms after exposure to respiratory viruses. For this purpose, a public dataset titled GSE73072 is used, which provides a variety of information, such as different virus types, infection onset/offset and symptom presence. In addition to developing machine learning models that can predict the disease status, in silico experiments have been performed on this dataset to identify genetic factors that influence infection and symptom development. Besides these aims, we used existing community-based information or knowledge for disease prediction by proposing some feature representation and feature selection approaches depending on the experiment type.

The rest of this thesis is structured as follows. The second chapter provides detailed explanations of genetic and infectious diseases separately. It also discusses the factors that affect both diseases and the reasons why artificial intelligence-based models are needed for prediction of both diseases. Chapter 3 explains the materials and methods used in the experiments of the thesis. Since our analyses include several machine learning and feature selection methods, as well as various feature representations and hyperparameter optimization, each of them is described and explained in detail in this chapter. In Chapter 4, machine learning experiments on a genetic data set and on an infectious disease data set are explained and the results are presented. Finally, Chapter 5 includes a conclusion

and a brief summary containing the main findings and experiment-specific outcomes. Future prospects are also included in this chapter.

Chapter 2

Genetic and Infectious Diseases

2.1 Genetic Disease Prediction

The “genome” refers to the complete set of genetic material, i.e. DNA sequence, that contains all the DNA information that contains the instructions for the functions, regulation, development, and growth of the body. Some parts of the genome contain genes, or a unit of our genetic mechanism, that enable proteins to be synthesized, thus ensuring a healthy life. However, mutations or abnormalities may occur in the DNA or genes due to hereditary factors, environmental influences, or other factors. In some cases, when these changes affect the function of the genes, genetic diseases may occur that adversely affect an individual's health.

Genetic diseases can be categorized into 3 different types as monogenic, multifactorial and chromosomal based on hypothetical or known nature of genetic defects underlying diseases [10]. Monogenic diseases are usually caused by a mutation in a single gene or a DNA base pair. For example, a mutation in DNA may cause the production of valine amino acid instead of glutamic acid. When this mutation occurs in the HBB gene, and an abnormal hemoglobin is produced. This inhibits the ability of hemoglobin to carry oxygen [11]. Cystic fibrosis, Huntington's disease, Fragile X syndrome, Tay-Sachs disease are other common monogenic diseases. Multifactorial genetic diseases, on the other hand, result from the joint action of many genes or genetic variants, each have a small or moderate effect, and often interacting with environmental triggers such as diet, lifestyle, exposure to toxins. Some heart diseases, diabetes, some types of cancers, psychiatric disorders are some examples for multifactorial genetic disease.

Due to having complex pathology, a wide range of symptoms, and unique mechanisms that are not yet fully understood, the diagnosis of a genetic disease can be challenging. Nevertheless, thanks to breakthroughs in genetic research and profiling

technologies of human genetics (DNA sequencing, gene expression profiling, etc.) genome information and health-related indicators can be extracted for individuals. However, keeping in view the complex nature of DNA data, the number of features, and the volume of data, manual prediction is laborious, error-prone, and inefficient for diagnosis [12]. Furthermore, the identification of genetic factors having an association with the disease needs advanced techniques that can analyze large genome data.

At this point, AI-based solutions such as machine learning methods, feature selection approaches, etc. can fulfill this need as they have shown great potential for prediction and forecasting in the last decades. Such models are trained on large amounts of genomic and clinical data, identify relationships or patterns between genetic material and disease, and depending on the sensitivity and importance of the task, perform assistive functions for medical experts, which are not by human experts at first sight. This has the potential to have a significant impact in the field of precision medicine, facilitating the development of tailored treatment approaches based on an individual's genetic profile.

One of the most known and extensively studied genetic disease is Alzheimer's Disease (AD). Briefly, AD is a neurological disease that affects memory, thinking, and social behaviors. It typically starts with mild memory loss and causes inability to carry a conversation and respond to the environment. Exact cause of AD is still not fully understood and there is no cure for it. Therefore, researchers have deeply focused the prediction of Alzheimer disease as well as finding key factors that cause this disease. Lee et al. compared the predictive performance of 5 classifiers and 5 feature selection methods on the 3 publicly available AD datasets ADNI, ANM1 and ANM2, all of which contains gene expression profiling samples derived from blood. As feature, differentially expressed genes (DEG), derived by variational autoencoder (VAE), TF-related genes from TRANSFAC database, hub genes associated with gene-gene interactions, and genes selected by Convergent Functional Genomics (CFG) score were used. Prediction results obtained by performing 5-cross validation within each dataset showed that AUC values of 0.65, 0.80 and 0.85 was obtained in the ADNI, ANM2 and ANM1 datasets, respectively. Among five feature selection approaches, on average, the DEG provided the best performance in ADNI and ANM1. Combination of DEG and CFG, on the other hand, was the best feature set for the ANM2 dataset. Additionally, study suggested that gene expression data is useful for the predicting of AD [13]. In another study, gene expression and SNP data were evaluated separately for the prediction of Alzheimer's disease using

the XGBoost classifier. Using gene expression features only, the model achieved an AUC of 0.64, while SNP data only achieved an AUC of 0.56 [14]. A stacked machine learning model consisting of RF, DT, SVM, and LR algorithms was also proposed for prediction of AD. Using a GWAS dataset with 398 samples and 411077 SNPs, the proposed model was able to discriminate AD samples with an accuracy of 93% [15].

Diabetes, especially type 2, is another common genetic disease that threatens public health. It has a complex etiology involving genetic, environmental, and lifestyle factors in the development of clinical conditions and pathology. Therefore, many researchers have endeavored to develop predictive models for diabetes, specially type 2. For example, Kälsch et al. examined associations between liver injury markers and diabetes using an RF classifier-based diabetes prediction model based on HbA1c (blood glucose levels), Adiponectin, and body mass index (BMI). The results of the experiment showed that the model that uses only the HbA1c value reached an AUC of 0.83, while the model that uses all the features achieved an AUC of 0.85 [16]. Shigemizu et al. used the Cochran-Armitage trend test, asymptotic Bayes factor (ABF) and sure independence screening methods to find the most significant SNPs for type 2 diabetes in Japanese individuals. They identified nine significant SNPs and then evaluated them using a Lasso-based prediction model. The results showed that these SNPs were able to classify diabetes samples with an AUC value of 0.806 [17]. Gene expression [18], Metabolome and Proteomics [19] data were also utilized for the diabetes prediction with several machine learning algorithms.

Similarly, machine learning approaches have been proposed for some other genetic diseases such as heart disease [20], obesity [21], Chron's disease, inflammatory bowel disease [22] etc. Although machine learning methods employed in these studies vary, the experimental procedures generally involve predicting whether the samples are at risk, carriers of the disease, etc., using genetic and clinical data as input. In addition, significant factors associated with the disease are evaluated according to their predictive performance. As part of the thesis, we addressed the prediction of Behçet's disease as the genetic disease using machine learning approaches and the identification of genetic factors affecting the disease.

2.1.1 Behçet’s Disease

Behçet's disease (BD), also known as Behçet's syndrome, is a chronic, multisystem inflammatory disorder that affects almost every organ system because it can affect both arteries and veins of any size, resulting in significant organ-threatening morbidity and mortality [23]. The disease was named after a Turkish dermatologist, Hulusi Behçet, who described three cases of patients with recurrent oral-genital ulcers and hypopyon uveitis. Since then, Behçet's disease (BD) has been considered a widespread vasculitis due to the involvement of the central nervous system, large vessels (veins and/or arteries), heart, and rarely the gastrointestinal tract or kidneys [24].

Even though BD occurs in many populations around the world, it is much more common in countries living along the ancient Silk Road, spanning East Asia, the Middle East and the Mediterranean, and is apparently rare in northern Europe. This is why it is known as the “Silk Road Disease”. The geographic distribution of BD can provide important clues to the etiology of BD. This is because it is unlikely that genetic differences between ethnic/racial populations are sufficient to explain variations in the incidence of complex diseases. Therefore, geographic and environmental differences may lead to some of the variations that cause BD [25]. The highest prevalence of BD was observed in Turkey, with an estimated 421 cases per 100,000 inhabitants. The prevalence in the Middle East countries such as Iran, Israel, Jordan, Iraq follows Türkiye with the estimated to be 31.8 cases per 100,000 inhabitants [25-26].

Like other autoimmune and auto inflammatory syndromes, the exact etiology of BD remains to be elucidated. However, the most probable hypothesis is that the viral and bacterial inflammatory reactions triggered by environmental effects and the genetic tendency plays important roles in the development of BD [27]. This hypothesis, i.e. contribution of the genetic and environmental factors to cause the BD, have been supported with many studies. The very first susceptibility genetic factor to be reported as having a strong association with BD was the human leukocyte antigen (HLA) region, particularly the HLA-B51 allele, with its identification dating back to the early 1970s [28]. This factor was also confirmed in different studies derived from multiple populations [29,30].

Outside the HLA region, genome-wide association studies (GWAS) which allowed evaluation and statistical interrogation of additional genetic variants in complex disease

[31], has enabled the researchers to reveal other genetic susceptible factors including the interleukin 23 receptor (IL23R), interleukin 10 (IL10) [32], ERAP1, CCR1, KLRC4, STAT4 genes [33].

Due to the wide-range of symptoms, the confirmation of diagnosis can be difficult for BD. There are no gold standard tests to diagnose BD, and as such, diagnosis is based on clinical criteria. Moreover, the symptoms of BD might vary from person to person. According to the International Criteria for Behçet's Disease (ICBD), patients must have recurrent oral ulcerations at least three times within 12 months. Painful oral ulcers in the tongue, pharynx, buccal and labial mucosal membranes appeared in 98% of cases. Patients should also have evidence of two of the recurrent genital ulcers, eye lesions, skin lesions and pathergy. In addition to the ICBD, several tests such as blood and urine tests, skin biopsy and pathergy may be necessary to make a final decision.

Considering all these difficulties in diagnosing BD, alternative solutions, such as artificial intelligence-based approaches, may be able to predict the presence, severity, and/or progression of BD more swiftly. Delayed diagnosis can lead to irreversible damage to organs such as the eyes, brain, and blood vessels. On the contrary, as early diagnosis can help prevent or reduce the severity of relapses and complications, it is crucial for an effective treatment of BD.

Despite the potential of AI-based solutions for diagnosis, the majority of studies on BD in the literature have focused on identifying genetic material or other disease-influencing factors. Especially, many studies have focused on population-based factors related to being in a particular region such as Türkiye, Iran, Jordan, Japan, etc. Moreover, the association of BD with the nervous system, pregnancy, and even COVID-19 has also been investigated. However, there is a lack of machine learning-related studies to predict the presence or progression of disease. Nevertheless, some machine learning models with different types of input data (e.g. proteomics data, radiology-based images, etc.) have been proposed in the literature for predicting the presence of BD.

Hammam et al. compared XGBoost, Extra Tree Classifier, RF, SVM, and MLP methods for the detection of vision threatening Behçet's disease (VTBD) using 1,049 subjects from the Egyptian College of Rheumatology (ECR) BD cohort. The input values consisted of 26 clinical and demographic features that were routinely and easily

measurable in a clinical setting. The dataset was split into 80% training and 20% test set, and then five different machine learning algorithms, XGBoost, RF, SVM, ANN and MLP were compared on the dataset. The results showed that the XGBoost model performed the best in the test samples with an AUROC value of 0.85, outperforming the other methods. In addition, higher disease activity, thrombocytosis, history of smoking and daily steroid dose were identified as the most important factors associated with the risk of VTBD [34].

Kim et al. developed a deep learning model for the classification of intestinal Behçet's disease (BD), Crohn's disease (CD) and intestinal tuberculosis (ITB) using colonoscopy images. The dataset consisted of 6617 images from 211 CD, 299 BD and 217 ITB patients. Each sample was labelled by two experienced endoscopists and annotated for the presence of a typical pattern. The results showed that the proposed CNN model achieved high performance with an AUROC value of 0.85 in discriminating BD and CD images. On the other hand, an AUROC value of 0.78 was obtained when only CD and ITB images were classified. It was also reported that the CNN model was able to predict the ulcer type with high performance even when it was only partially observed in the image [35].

In another study, a Multi-Layer Perceptron (MLP)-based prediction model for the detection of ocular Behçet's disease was proposed. The experiments used ophthalmic arterial Doppler signals as a dataset derived from 106 subjects, 54 of whom had ocular Behçet's disease, and the rest were healthy subjects. Doppler signals refer to the ultrasound signal frequency measured by ultrasound pulses scattered by red blood cells. These frequency values were then converted to power spectral density using spectral analysis to produce data as input for the prediction model. Results showed that the proposed model is able to predict subjects suffering from ocular Behçet’s disease with a 93.75% accuracy rate. Healthy subject was classified with 96.43% accuracy [36].

The association of gut microbiome composition with BD, as well as its potential role in the development of this disease, was also investigated using metagenomic analysis. A dataset consisting of 32 active BD patients and 74 healthy control samples was analyzed for this study. Metagenomic analysis identified 17,896 microbial genes that were significantly different between patients and controls. Relative abundance profiling then identified 13 metagenomic species (MGS), of which 11 were enriched in the BD group

and 2 in the healthy control group. Using these 13 MGSs as features, a random forest model obtained a mean classification error of 0.18 and an AUC value of 0.811. It is also reported that analysis of a large sample is needed to confirm whether these markers are useful in the diagnosis of BD [37].

Proteomics is a field that examines a large of proteins in biological systems, enabling the identification of an ever-increasing number of proteins and providing useful information about genetic information. Tang et al. conducted an exploratory study on 26 BD patients and 26 healthy control samples using proteomics-based tandem mass tags (TMTs) and parallel reaction monitoring (PRM) analysis to identify potential serum biomarkers for BD. Using differentially expressed proteins (DEPs), and clinical information (age, gender) a random forest model was built to select features. Two proteins TPM4, FLNA and age were selected as independent predictors for the BD by the model. Furthermore, each variable's discriminative power was then evaluated with a logistic regression model on an external validation set comprising of 16 control and 13 BD patients. Finally, single features FLNA, TPM4 and age obtained an AUC value of 0.741, 0.683 and 0.73, respectively. However, when all of the three features were used, the model achieved an increased AUC value of 0.862 [38].

2.2 Infectious Disease Prediction

Diseases caused by infectious agents have a profound impact on human history and biology. Infectious agents are microorganisms or pathogens including bacteria, viruses, fungi and parasites. Although many of them live in and on our bodies and they are normally harmless or even helpful, some organisms may cause disease and infection under certain circumstances such as temperature change [39]. From a demographic perspective, infectious diseases have probably caused more deaths than all wars, non-infectious diseases, and natural disasters combined, including both massive epidemics such as the plague and smallpox, that devastated human populations from ancient to modern times, as well as less dramatic, obscure viral and bacterial infections that cause high infant mortality [40]. For example, the Spanish flu that occurred at the beginning of the 20th century was estimated to have caused almost 500 million infections (a third of the world's population at the time) and at least 50 million deaths.

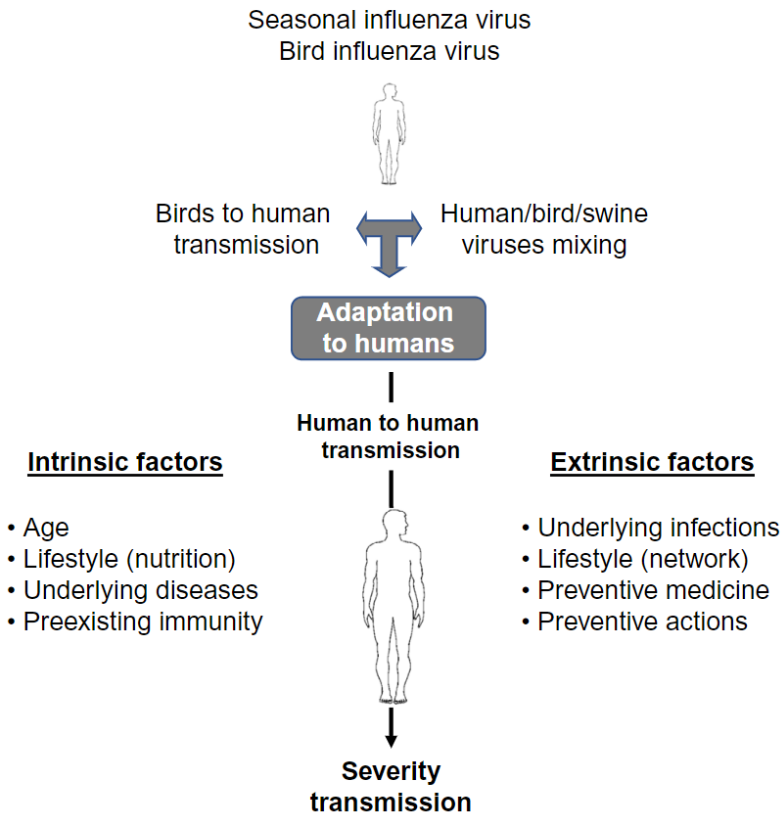

Figure 2.1 An example of how the bird influenza virus is transmitted, the factors influencing it and how it is adopted by humans [41].

The key feature that makes infectious diseases so effective, spreadable, and destructive is the dynamics of propagation capability. Therefore, these diseases are also called as “communicable diseases”. Their causative agents can be transmitted from person to person (or between animals and people, or animals and animals), leading to sustained transmission. Moreover, infectious agents can be transmitted through a variety of routes, including water, air, food, insect bites and sexual intimacy, depending on the type of disease. For example, HIV virus is only transmitted through close physical contacts such as sexual or blood transmission, while influenza is spread through airborne droplets after a person sneezes, coughs, or talks. The transmissible nature of infections also means that, for most infectious diseases, the impact of any single case and its public health and economic consequences may go beyond those attributable to the loss of quality of life and risk of death to that individual [42]. The dramatic increase in travelling also makes it easier for infections to spread around the world in much less time than ever before, leading to global pandemics and epidemics. The most devastating example of this situation is the COVID-19 outbreak, which we have recently overcome but which continues to have far-reaching effects. According to the WHO, the number of COVID-19

cases worldwide has reached 775 million. Although originated from China, it had been spread rapidly around the world via air travel and caused millions of deaths.

In addition to the human toll, infectious diseases affect human well-being, social-life and global economy on a global scale. According to another report by the World Health Organization and the World Bank, the economic impact of a pandemic could be as high as 4.8% of global GDP [43]. Routine infectious diseases that occur outside of pandemics or epidemics can lead to a reduction in labor supply, increased treatment and healthcare costs, and a decline in tourism and trade. This devastating impact of infectious diseases is a forceful reminder of the vulnerability of infectious diseases to governments, policymakers and scientists, and prompt them to find some health strategies such as prevention, mitigation, or containment of disease transmission. Even before these strategies, the most effective solution is early detection. This is because even if the signal of disease propagation or spread is weak, early detection enables to figure out trends before they become significant and important. However, early detection is not an easy task as infectious diseases usually have multiple factors that affect the process of infection and transmission. Therefore, there is a need for cutting-edge AI-based solutions, such as machine learning, to analyze, interpret and predict signals. This is because AI models allow machines to act or react to inputs in a way similar to humans, by performing cognitive functions and making decisions about inputs [41].

There are many studies in the literature where machine learning models have been utilized for infectious disease problems, including diagnosis, drug resistance, transmission and risk prediction. For example, Sun et al. proposed a system to predict the risk of influenza infection in patients by analyzing their vital signs, such as respiratory rate, heart rate, and facial temperature. The proposed system consists of a neural network and a fuzzy clustering method (FCM) with a self-organizing map (SOM) and classify individuals into three groups (higher-risk, lower-risk, and non-influenza). The experimental results indicated that the proposed system achieved high sensitivity (97.1%) and high negative predictive value (97.5%) in identifying high-risk influenza patients within 10 seconds, by outperforming traditional thermography-based screening methods. The authors also stated that such systems serve as potential tools for rapid screening of infectious diseases and can be used as a first step for screening [44]. Another infectious disease, tuberculosis (TB), was predicted perfectly with an AIRS model developed by Saybani et al. Artificial Immune Recognition Systems (AIRS) are specialized machine

learning methods that originally use some pre-processing steps and the kNN algorithm as the background classifier. However, the authors have improved the standard AIRS model by using SVM instead of kNN as the classifier. Experimental results showed that the proposed method, RAIRS2, achieved a perfect performance in classifying TB samples, with an accuracy of 100%, sensitivity of 100%, and specificity of 100% [45]. [45]. In another study, several machine learning algorithms were compared to predict COVID-19 mortality. As a feature set, clinical features such as smoking, oxygen therapy, etc. were used as input to the models. Experimental results showed that the random forest algorithm outperformed the others with an accuracy of 95.03%. In addition, dyspnea, ICU admission, and oxygen therapy were selected as the most discriminative features to predict mortality after COVID-19 infection [46]. DNA-based genetic materials have also been widely used with machine learning models to predict whether individuals are infected with an infectious virus. In the study [47], feature selection techniques and the XGBoost classifier were used to predict hepatitis B (HBV) and hepatitis C (HCV) related hepatocellular carcinoma (HCC) using gene expression profiling data. Using feature selection, 17 genes were selected as significant for discriminating HBV and HCV. In addition, the XGBoost algorithm achieved 97.1% accuracy in predicting HCC samples. Tai et al. investigated an individual's susceptibility to developing malaria using risk scores derived from the cumulative effects of SNPs. To find susceptibility SNPs, Logistic Regression and Recursive Feature Elimination (LR-RFE) based feature selection was applied to the SNPs dataset and the most contributing features to disease development were identified. XGBoost, LightGBM and Ridge Regression algorithms were also compared and LightGBM achieved the highest model performance with an MAE score of 0.0373 in predicting weighted genetic risk scores [48].

Predicting the spread, number of cases and onset of infectious diseases are other beneficial applications of machine learning models to public health issues. Especially since the COVID-19 pandemic, attempts to predict any pandemic or epidemic have received more attention from researchers. For example, Nawi et al. conducted a study to develop a forecasting model for pandemic cases with a hybrid model consisting of ARIMA and SVM algorithms. In their experiment, a daily number of confirmed cases, fatalities, and recoveries of COVID-19 in Malaysia were used as input data for the predictor model. The results show that the proposed hybrid model was 63% more accurate than the standard ARIMA model in predicting new positive cases. This rate was 60.46%

for daily new deaths and 73.12% for daily new recovered cases [49]. In another study, the performance of the autoregressive statistical model, XGBoost, random forest, multi-layer perceptron, and encoder-decoder model were compared in forecasting three different infectious disease incidences across different countries and time intervals. In the short-time-interval forecasting problem (2-5 months) XGBoost model showed the best performance for campylobacteriosis and typhoid diseases [50]. Ajith et al. compared naive Bayes, random forest, and adaptive boosting methods in forecasting the occurrence of The West Nile virus (WNV). Experiments showed that the random forest model achieved nearly 70% accuracy in predicting the possibility of the presence of WNV [51].

Drug-related studies have also been conducted by researchers. For example, machine learning algorithms were used in rapid in silico predictions of tuberculosis drug resistance, with the goal of a reliable and cost-effective alternative to in vitro assays [52]. Rajput et al. take advantage of machine learning algorithms such as artificial neural network, support vector machine and random forest algorithms to predict small molecule inhibitors of Ebola virus, i.e. identify potential anti-Ebola compounds. As a result of their experiment, the ANN model outperformed others with a Pearson correlation coefficient of 0.65 in the validation dataset [53].

In addition to these diseases, a large number of machine learning-based prediction studies on other infectious diseases are available in the literature. Moreover, some studies include the detection of both environmental and genetic factors, known as “biomarkers”, which have an impact on disease prediction and can be identified using feature extraction methods. However, among infectious diseases, respiratory viral infections are the most prominent in terms of disease severity, contributing to significant morbidity, mortality and economic losses worldwide.

2.2.1 Respiratory Infection Prediction

Respiratory infections (or respiratory tract infections) are the leading cause of acute illnesses and deaths worldwide in both adults and children from past to present. According to a report by the World Health Organization [54], respiratory-related infections cause nearly four million deaths per year. It is also one of the major diseases that threaten human health with high morbidity, severity, and medical costs [55]. Estimated cost of respiratory infections are estimated to be responsible for approximately

$15 billion in direct treatment costs in United States [56]. The numbers are even higher especially in undeveloped and developing countries due to inadequate healthcare systems. Geographic differences and socioeconomic factors of the populations also affect the variation in viral etiology and the number of cases across countries (Liu et al., 2015). It's been reported that the case-fatality rate for respiratory infections is significantly higher in temperate regions of the world, especially among impoverished populations in tropical regions [57]. For example, respiratory illnesses are responsible for an approximate death rate of 1 in every 5 children according to a study conducted in the Rwanda [58].

Similar to other infectious diseases, numerous pathogens such as bacteria, fungi, mycoplasma, etc. can cause a respiratory-related disease. However, a large proportion of respiratory infections are caused by viruses. There is a wide range of respiratory viruses that have been identified to date including Human Rhinovirus (HRV), Respiratory Syncytial Virus (RSV), Influenza A and B, Adenoviruses, Coronaviruses, Parainfluenza virus and Parvovirus. Among these viruses, HRV has been identified as the virus most commonly associated with respiratory diseases, accounting for about 40% of infections. Influenza viruses, RSV, and Coronavirus follow HRV in terms of frequency [59].

Clinical manifestations of respiratory infections are familiar and well-known. Most infections result in mild symptoms such as runny nose, sore throat, and headache. Although clinical symptoms depend on the type of virus, the site of infection (e.g. sinusitis, bronchitis, rhinitis, etc.), the patient's age, general health, comorbidities, immunity and whether the infection is primary or secondary, most of the symptoms of respiratory viruses overlap [60]. However, different virus infections might require completely different treatments. Otherwise, severe pneumonia may develop, which can cause mortality or some complications.

Another noteworthy aspect of respiratory infections is that some people remain asymptomatic despite exposure to respiratory viruses, while others become symptomatic [61,62]. It is interesting because the penetration of viruses may be similar, but the immune response, or the body's defense mechanism, varies from person to person. This was commonly reported by people during the period of COVID. Some COVID patients went through the disease with severe symptoms despite being in the best of health before infection, while some chronically ill elderly people showed no symptoms [63,64]. Notwithstanding, this is contrary to what might be expected, i.e. elderly people being

symptomatic. These variations on being infected evidenced that the host response following exposure is linked to genetic predisposition, disruption of the individual's microbiome [65], being in high-risk group [66] and effective immune surveillance. However, the variation in the physiological responses of people to viral exposure is poorly understood. The lack of understanding about the precise physiological or genetic factors delays the detection of infection, which leads to the spread of virus and thereby increasing the death toll. As mentioned in the previous section, the main solution for preventing unwanted effects of infectious diseases are to forecast or predict them as soon as possible. But forecasting a disease is not an easy job using traditional methods as there are a large number of factors such as environmental, genetic, geographical etc. that have impact on infection. That's why an advanced & intelligent system such as a machine learning system is needed in the field of respiratory infections. Some studies in the literature focused on the forecasting of respiratory infections. For instance, Lim et al. proposed an ensemble Gradient Boosting Machines (GBM) classifier-based model to predict respiratory burden using average daily polyclinic attendances [67].

However, most of the other studies focused on the idea of using both statistical and in silico methods to find out predictors of respiratory infection and make forecasting for individuals, including our thesis. After all, if an individual's susceptibility and resistance to infection can be predicted, then a predictive model can be developed and the biomarkers responsible for infection can be found. Thus, several studies to predict infection or the development of symptoms in individuals have been carried out. Barlacchi et al. conducted a study to predict the future presence of flu-like and cold symptoms in individuals using past mobility activities such as total distance travelled, total displacement, number of different places visited, etc. In their experiment, these mobility activities were used as input for three machine learning models including logistic regression, random forest and gradient boosted trees (GBT), and then the presence of symptoms was predicted. The best result was obtained by an AUROC value of 0.62 by the GBT algorithm [68]. Elbasi et al. compared several machine learning algorithms on the classification of influenza H1N1 and COVID-19 patients. Different types of data information such as patient age, gender, blood or tissue sample results, and risk factors were used to represent patients for ML models. Results showed that multilayer perception (MLP) algorithm achieved the highest accuracy with an accuracy of 99.31% [69].

Bongen et al. [70] applied a meta-analysis to several data sets and observed that the expression of the KLRD1 gene in blood decreased after influenza virus infection. They were also able to predict the symptomatic and asymptomatic samples with an area under the receiver operating characteristic (AUROC) value of 0.91 in a validation set of H3N2 influenza samples. Barral-Arca et al. [71] found 17 characteristic genes for RSV by applying logistic regression to 296 infected and 266 healthy samples from different datasets. ORA of these genes showed that immunological pathways such as the innate immune system and the adaptive immune system are closely associated with RSV. In another study by Xu et al. [72], the OTOF and SOCS1 genes were identified as discriminators of HRV infections in machine learning experiments on gene expression profiles.

In a comprehensive study, different machine learning and feature selection methods were compared using three different datasets containing RSV-, HRV-, and influenza-infected samples [73]. The proposed model included a modified minimum Redundancy—Maximum Relevance (mRMR) method and a naïve Bayes classifier that achieved an average accuracy of 91% when the number of gene expression features is 40. The authors also applied an ORA on the top-50 genes selected by the best feature selection method and reported that all genes are related to the immune response to viral infection.

Hung et al. developed and compared machine learning algorithms to predict influenza infection using clinical features. The dataset in their experiment included 2189 patients, of whom 1104 tested positive for influenza. Their results show that the XGBoost algorithm outperformed other known ML algorithms as well as some deep learning models with two, three and four layers, with an AUROC value of 0.82 [74].

Verma et al. analyzed gene expression levels in both infected and uninfected individuals to identify potentially effective genes and to predict the state of respiratory virus infection. As a dataset GSE73072 containing H1N1, H3N2, RSV, and HRV samples, also identical to our thesis experiment, was used in 10-fold cross-validation settings. In addition, this dataset contains time points of samples before and after the exposure. Experiments showed that the SVM algorithm using the RBF kernel achieved an accuracy of 82.83% in 10-fold settings. In addition, the genes IFIT1, DDX58, and PLSCR1 were selected as the most significant biomarkers according to the random forest importance score [75].

Recently, the prediction of respiratory virus infection has become popular using models based on deep learning. To predict whether an individual will develop symptom prior to exposure to influenza A virus, Zan et al. [76] proposed a six-layer deep neural network (DNN) model. The model outperformed the SVM, the RF and the convolutional neural network, achieving a cross-validated AUPRC of 0.758 for the DEE3 H1N1 experiments and an AUPRC of 0.901 for the DEE2 H3N2 experiments, respectively.

Chapter 3

Materials and Methods

3.1 Data Resources for Disease Prediction

Bioinformatics studies still face many challenges related to the collection, protection, and integration of domain-related data. One of these challenges is the difficulty of sequencing DNA or RNA data from individuals, which is a standard data source in a wide range of applications in genetics, proteomics, and personalized medicine studies [77]. The high cost of sequencing two decades before could have been the main reason. But today, the sequencing cost of DNA has fallen to $300, which is an affordable price. However, device cost for genome sequencing is still high, especially in undeveloped and developing countries. In addition, finding volunteers to participate in genetic studies to have their DNA or gene-related material extracted is particularly challenging, as it requires extensive engagement efforts, concerns about privacy and data security [78]. The collection of genetic data poses significant challenges because of these issues. On the contrary, one of the most important factors in artificial intelligence-based learning algorithm is data quantity and quality. The predictive ability of AI models is directly related to the volume and discriminative power of the data. This has necessitated quality data, particularly for local institutes and researchers, including biostatisticians, computer scientists, etc., who cannot collect genetic data. Nevertheless, thanks to the connected world and the decreasing cost of both internet technologies and storage, online data repositories which contains genetic data derived for several diseases are fully open to researchers. These repositories that are frequently used by the community and that contain a large amount of data can be listed as follows:

Gene Expression Omnibus (GEO): GEO is a public repository of high-throughput gene expression and other functional genomics data submitted by scientists. It was launched by the National Center for Biotechnology Information (NCBI) in 2000 for

expression data, but with the rapidly changing needs of the bioinformatics field, it now accepts other types of data, including genome methylation, chromatin structure, and genome-protein interactions [79]. As a report published in 2024, GEO resource contains over 200000 studies and 6.5 million samples, all of which are indexed, searchable and downloadable [80].

- **Single Nucleotide Polymorphism Database (dbSNP):** dbSNP is a free public genetic variation resource for single nucleotide variations, microsatellites, and small-scale insertions and deletions. Similar to GEO, dbSNP is maintained by NCBI to satisfy the need for a comprehensive resource of genomic variation for large-scale sampling designs required by association studies, gene mapping, and evolutionary biology [81].

- **The European Genome-Phenome Archive (EGA)**: EGA is another genetic repository that stores and distributes genetic, phenotypic and clinical data derived from biomedical studies with the mission to re-use data, enable reproducibility and accelerate biomedical and translational research. Unlike other repositories, the EGA has several strict protocols for data security, and therefore data is consented for specific uses that require approval but is not fully open. According to a 2021 study, the EGA has archived over 4500 studies with 6800 datasets, totaling almost 15 PB of sensitive human data [82].

- **The Cancer Genome Atlas (TCGA)**: The Cancer Genome Atlas (TCGA) is a large-scale cancer genomics collaboration maintained by National Cancer Institute (NCI) and the National Human Genome Research Institute (NHGRI) to map the genomic and epigenomic changes in several types of human cancer [83]. Data on the TCGA are centralized and stored in databases when they are available, making them quickly accessible to researchers. It is estimated that TCGA repository keeps nearly 2.5 petabytes of genomic, epigenomic, transcriptomic, and proteomic data derived from 20000 cancer samples.

In addition to these databases, there are several other less popular repositories available to researchers, such as ArrayExpress [84], miRbase [85] and MGnify [86], which contain different levels of biological and genetic data. These data sources fill a critical need for genetic, bioinformatic, biostatistical and other researchers who struggle

to find data. Our secondary study, the prediction of infectious disease, was also an analysis of data from GEO, which is also one of these sources.

3.2 Machine Learning Algorithms

Through the use of statistical techniques, mathematical algorithms, and computational power, machine learning algorithms can analyze given features of data in order to identify patterns and predict an output with remarkable accuracy [87]. Although the output(s) varies depending on the nature of the problem being addressed, bioinformatics studies can typically infer information about individuals or living organisms, including the presence of disease, disease severity, disease prediction, and factors contributing to disease (e.g. bacteria, virus, genes, SNPs, etc.)

One of the main objectives of our thesis is to predict whether an individual has a particular disease or not using genetic profile information. However, prediction performance can vary significantly depending on the machine learning algorithms employed. Therefore, instead of using a specific machine learning method, we have comprehensively examined a total of eight different algorithms to reveal how algorithms with different statistical characteristics behave for the prediction of genetic and infectious diseases.

3.2.1 Logistic Regression

Linear regression is the fundamental regression algorithm used in mathematics and statistics to figure out the relationship between a dependent and independent variable(s). In the machine learning field, it is used to predict continuous output using independent variables, i.e. feature values [88]. Mathematically, the training step of linear regression tries to estimate coefficients, which are assigned to each independent variable individually, with a minimum of error between predicted and actual dependent values. Logistic regression (LR), on the other hand, is a transformation of linear regression for classification problems using the logistic function (also known as the sigmoid function) [89]. Simply, the output of the linear regression is fed into the logistic function to limit its range between 0 and 1. This allows the logistic regression to perform a binary classification by setting a threshold. For multi-class problems , the SoftMax function is used instead of the sigmoid. Mapping linear combination of input data to a probability

between 0 and 1 allows non-linear relationships to be modeled. One of the main disadvantages of logistic regression is that it tends to overfit as the number of independent variables increases. Regularization techniques have been developed to overcome this problem. Briefly, regularization is a term that expresses the addition of a penalty to the loss function according to a norm of the weight vector to prevent an excessive increase in model weights.

3.2.2 Support Vector Machine

Support Vector Machine (SVM) is a supervised learning algorithm widely used for classification and regression tasks. It was developed by Vapnik et al. [90] and is fundamentally based on statistical learning theory. Conceptually, the basic principle of SVM is to find the optimal separating hyperplanes, which linearly separates the two class labels. Using the determined hyperplane, a new unseen sample is classified into one of the two classes depending on which side of the hyperplane it falls. Suppose we have training samples with a binary output y= {-1,+1} and our data can be separated linearly. The goal is to select the hyperplane H that best separates the classes. Consider the hyperplane H defined by the function $f(x) = 0$ below,

$$f(x) = W \cdot X + b \tag{3.1}$$

where w is the weight vector that is perpendicular to the hyperplane and the sign of $f(x)$ indicates the position (or label) of point x with respect to the hyperplane. The SVM tries to find the hyperplane that maximizes the margin, which is the distance between the hyperplane and the nearest data point of either class, according to formula (3.2):

$$H^* = argmax_H \gamma\mathrm{H} \tag{3.2}$$

where $\gamma\mathrm{H} = \min_{i=1\ldots n} \gamma_i$ corresponds to the margin of the nearest point to H. However, the optimal hyperplane may not always be found at a given dimension, especially if the data is non-linear. For these situations, SVM maps the input data into a high-dimensional feature space using a kernel function such as a radial basis function (RBF), polynomial or sigmoidal function [91].

Although SVMs have built-in noise protection and overfitting control, they can be computationally intensive, especially when dealing with large training datasets. In

addition, SVMs are primarily designed for binary classification, but they can be extended to handle multi-class problems using techniques such as one-vs-one or one-vs-all [92].

3.2.3 K-Nearest Neighbors (kNN)

kNN is another simple and intuitive algorithm used for classification tasks [93]. The basic idea of the standard kNN algorithm is predicting the class label of the given sample with the majority class of its most similar training data points in feature space [94]. In other words, kNN assumes that data points that are close together in feature space are likely to be in the same class. Therefore, there is no actual training process to do any generalization. Instead, the method simply stores all training data samples in the memory instead [95]. Therefore, this algorithm is also called a lazy learner. Moreover, it is a type of non-parametric algorithm due to not having any assumption about data distribution. To calculate the similarity between data points, a distance metric is used such as Euclidean or Manhattan. After calculating the distance, the new sample will be assigned to the class label for which the number of neighbors is maximal.

3.2.4 Random Forest

The decision tree (DT) is another non-parametric algorithm that evaluates the importance of features in the dataset and splits it into subsets based on the value of the feature until a certain termination criterion is met, e.g., reaching a maximum depth or when further splitting does not improve prediction accuracy. The idea behind this algorithm is divide and conquer, searching for optimal split points within a tree greedily [96]. However, this greedy search can cause overfitting when a decision tree becomes too complex due to capturing noise or irrelevant patterns, and therefore an ensemble method called Random Forest was developed.

Random Forest (RF) is an ensemble method that includes many decision trees where each tree makes a prediction individually and then the final prediction for a sample is chosen as the class label with the most frequent votes [97]. The idea behind random forest is “the wisdom of crowds” which means multiple random and uncorrelated trees can act as a group to overtake the result of any separate constitute model [98]. This is because uncorrelated multiple trees may lead to correct direction by correcting each other's prediction error.

Random forest algorithm begins with bagging, or bootstrap aggregation, which generates random subsets of data by sampling with replacement where the number of samples in each subset is equal to the number of samples in the training set, and each subset is used as input to a decision tree model. Decision tree uses impurity criteria for deciding how to split data at each node. The impurity measures the quality of a split and is used to determine which feature and split point to choose at each node. Gini index and entropy criteria such as information gain and gain ratio are examples of criteria used in the impurity calculation at this stage. After the training process, prediction for a new sample is computed by averaging the predictions from all trees.

3.2.5 Boosting Algorithms

Boosting is another popular ensemble learning approach that builds a strong classifier by combining several basic learners, similar to the bagging approach mentioned above [99]. However, these approaches have some differences in what they produce and how they handle the training data. Bagging essentially reduces variance and improves stability by using multiple base classifiers simultaneously on a random subset of the dataset, thus increasing prediction performance. Boosting, on the other hand, is an iterative technique that involves multiple weak leaner models like bagging, but the models are trained sequentially by correcting the error made by the previous model to build a stronger classifier [100].

For example, suppose there is a dataset $D = \{s_1, s_2, s_3 \dots s_n\}$ with $s_i = (\boldsymbol{x_i}, y_i)$ where $\boldsymbol{x_i} \in \boldsymbol{X}$ and $y_i \in \{-1, +1\}$. The boosting approach initially generates a subset D_1 by randomly choosing samples from dataset and this subset is uset do train a base learner to form a predictor model H_1. In the next iteration, a subset D_2 is generated the same way but misclassified samples of H_1 have higher probability of being selected and H_2 is formed by training a base learner using D_2. In the next stage, a subset consisting of the higher proportion of misclassified samples from models $H_1$and H_2 are used to train the base learner and the H_3 model is generated. Boosting briefly combines these predictors iteratively to produce the final classifier using majority voting [101]. The strong idea of boosting approaches have lead to have lead improved version of boosting-based algorithms such as gradient boosting.

Unlike the traditional boosting, gradient boosting tries to find an approximation by minimizing the expected value of a given loss function. The algorithm is initialized with a constant α approximation of $H^*(x)$ as formula (3.3), where N is the number of samples.

$$H_0(x) = argmin_{\alpha} \sum_{i=1}^{N} L(y_i, \alpha) \tag{3.3}$$

Subsequent models are expected to minimize loss as in formula (3.4) where ρ_m is the weight of the m^{th}function of $h_m(x)$.

$$argmin_{\rho,h} = \sum_{i=1}^{N} L(y_i, H_{m-1}(x_i) + \rho_m h_m(x_i)) \tag{3.4}$$

The algorithm trains each model h_m instead of solving the optimization problem directly and calculates residuals r_{im} by formula (3.5) and the value of ρ_m is subsequently solved by a line search optimization algorithm [102].

$$r_{im} = -\left[\frac{\partial L(y_i, H(x_i))}{\partial H(x_i)}\right]_{H(x)=H_{m-1}}, i = 1,2, \dots n \tag{3.5}$$

Finally, the new model is added to the ensemble using the following formula (3.6),

$$H_m(x) = H_{m-1}(x) + \nu \rho_m h_m(x) \tag{3.6}$$

The value of ν which is also called the shrinkage parameter enables to control overfitting by shrinking the contribution of gradient descent in each iteration.

3.2.5.1 Ligth Gradient Boosting (LightGBM)

LightGBM is an extended version of traditional gradient boosting that proposes some additional variants, especially in the splitting procedure, to make the models computationally more efficient [103]. Traditional GBM scans all data samples and examines possible splitting features, which is slows down the model running time. LightGBM proposes two new features to prevent these drawbacks: Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB).

GOSS is a subsampling technique that relies on retaining data instances based on their gradients during the down-sampling process. Instances with large gradients contribute more to the information gain criteria. Therefore, it is expected that keeping

instances with large gradients while randomly dropping instances with small gradients can lead to more accurate gain estimation than uniform random sampling. EFB, on the other hand, refers to the reduction of sparse features using a greedy approach that bundles mutually exclusive features. This technique reduces sparse features to a single feature and thus speeds up the model without losing any information.

3.2.5.2 Extreme Gradient Boosting (XGBoost)

XGBoost (eXtremeGradientBoosting) is an improved version of GBM including additional techniques to control overfitting, split finding and handling missing values during training [104]. It also provides a more generalized model to control complexity by adding regularization term as in equations (3.7) and (3.8),

$$\Omega(h_m) = \gamma T + \frac{1}{2}\lambda \|w\|^2) \tag{3.7}$$

$$L_{xgb} = \sum_{i=1}^{N} L(y_i, H(x_i)) + \sum_{m=1}^{M} \Omega(h_m) \tag{3.8}$$

where T is the number of leaf nodes, providing a varying number of leaves instead of a fixed number, w are the output scores of the leaves and γ, λ are coefficients for regularization, which is allow controlling the complexity in each iteration separately. One of the main advantages of XGBoost is that it is designed to be highly scalable. It allows parallel learning of the model , resulting in faster results.

3.2.6 Naïve Bayes

Bayesian classifiers are probabilistic approaches that use Bayes decision theory to make predictions or estimates for a given sample and prior expectations. This technique is based on the assumption that the decision problem is formulated in probabilistic terms and that all relevant probability values are given [105]. Naive Bayes algorithm can be defined as the simplest Bayes classifier. Main characteristic of this algorithm is the very strong (i.e., naïve) assumption that the features are conditionally independent with respect to the class label [106]. This assumption simplifies the calculation of likelihood when estimating their probabilities. Although this assumption may not hold in all cases, or even when the assumption is poor, Naive Bayes classifiers have been shown to be remarkably effective [107]. For a given sample $X = \{x_1, x_2 \dots x_n\}$, the formula of the Bayes classifier is:

$$NB = argmax_{c_i \in C} P(c_i) \prod_i P(x_i \mid c_i) \quad (3.9)$$

where c_i represent class label and $P(c_i)$ is prior probability of c_i. Depending on likelihood and prior probabilities, NB selects the target class label with the highest probability.

The Naive Bayes classifier is known for its robust performance, performing well even with relatively small training datasets. This makes it a valuable method in bioinformatics, where datasets may be limited due to collection difficulties. Given the small number of samples in our secondary experimental study, we also wanted to observe the results of the Naive Bayes algorithm.

3.3 Feature Selection

Measuring the changes, activations, or intensities of genetic materials such as gene expression levels that occur in humans typically generates thousands of data values. For example, with the revolution in next-generation DNA sequencing, some array technologies are capable of genotyping up to 1 million polymorphic variations for each sample in a single experiment [108]. Alternatively, more than ten thousand gene expression intensities can be measured with current profiling microarrays [109]. In these cases, a sample (or individual) was represented with thousands to millions of parameters. Each parameter is called a "feature" in the field of artificial intelligence, which can be measured and expresses the characteristic properties of a sample. These features can be helpful or informative in understanding the underlying molecular mechanism for in vitro experiments. However, data may contain redundant and highly noised features caused due to collection technology, the nature of biological data, etc. [110]. In general, only a small subset of these features is relevant, depending on the problem being addressed. Furthermore, using irrelevant or redundant features causes the curse of dimensionality and makes it difficult to discover potential markers [111]. Conversely, eliminating the redundant features often leads to increased model accuracy and decreased running time [112], which are also evidenced empirically [113]. One way to remove noise and redundancy is dimensionality reduction (DR), where usually the feature space is reconstructed by combining the original features. This technique results in losing the

effectiveness of each feature on the problem outcome. Alternatively, instead of combining existing features to generate new features, a subset of biological features can be considered to identify effective drug response markers associated with cancer [114,115].

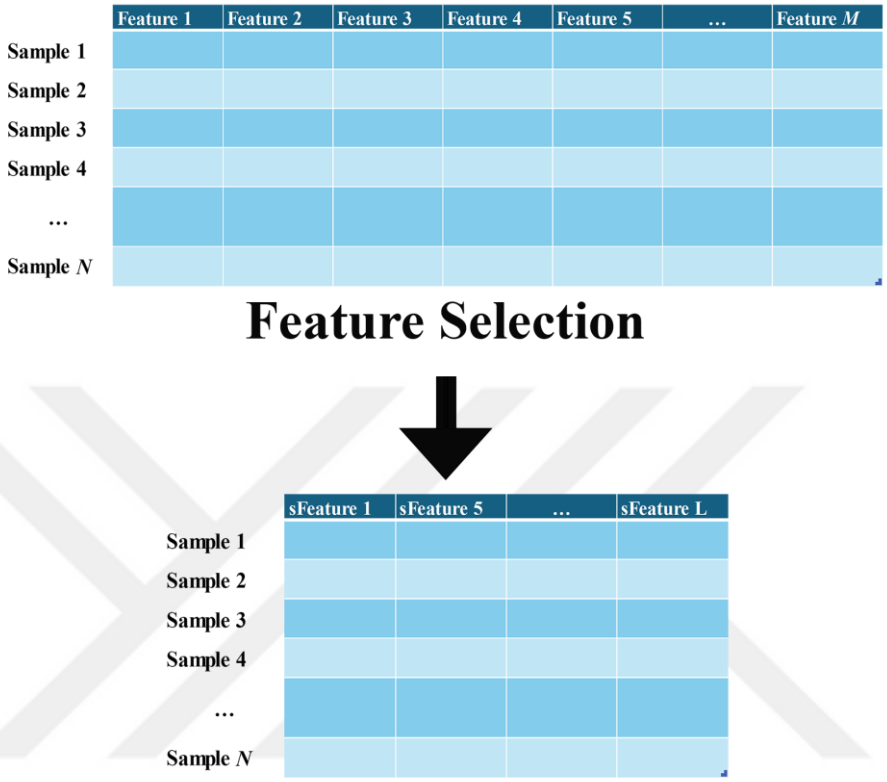

Figure 3.1 Brief illustration of the feature selection process. The original data set may contain a large number of features, most of which may be irrelevant. Feature selection reduces the number of features by keeping the significant ones [116].

But how do we figure out the significant features of genetic input material that have an impact on disease development, viral infection, or other phenotypic conditions ? For a dataset with N features, there is 2^N unique candidate feature subsets. Considering the huge number of features that arise due to the nature of biological data, finding the perfect subset is very costly in terms of computational time. Moreover, the subset should be "necessary" and "sufficient" to describe the target concepts, while representing the patterns of samples [117] . Alternatively, feature selection may also provide an approximation to the optimal subset of features.

Feature selection (FS) is a technique where an optimal subset of features related to phenotype or outcomes is found according to certain relevance evaluation criteria [118]. Unlike other dimensionality reduction techniques that distort the original feature

representation, FS preserves the semantic integrity of the data [119]. Furthermore, the optimal subset following feature selection that is used as input to a predictive model typically assumed to be functionally associated with a disease etiology [116,120]. Hence, feature selection makes it possible to identify significant features, such as genes responsible for a disease, and to interpret them straightforwardly.

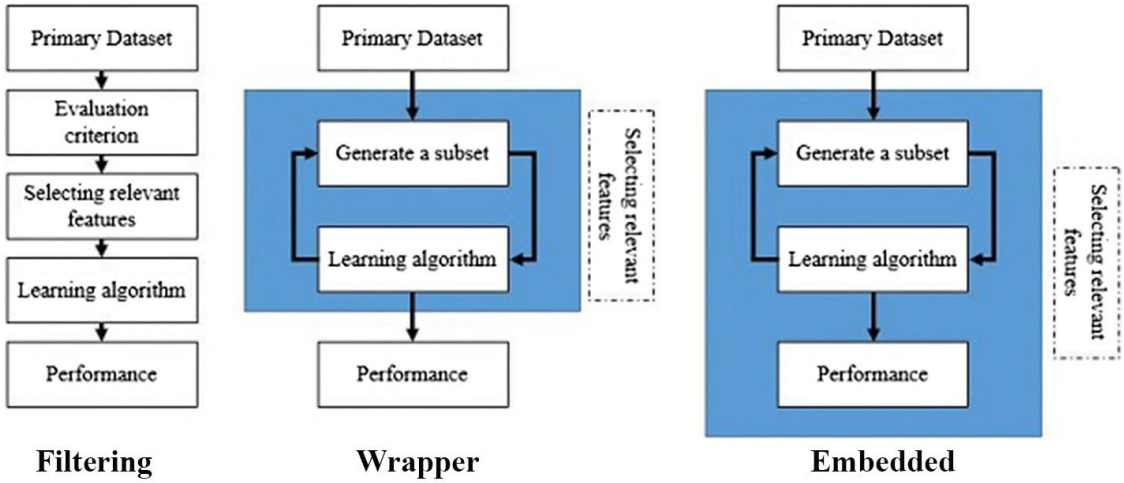

Figure 3.2 An illustration of the three main types and process of feature selection approaches - filtering, wrapper and embedded [121].

In the machine learning environment, feature selection approaches are divided into three main categories, namely filtering, wrapper, and embedded, according to their interaction with data or a learning algorithm (see Figure 3.2). In addition, ensemble and hybrid approaches have also been proposed in the literature, where different methods are combined or cooperated in an appropriate layout to reveal the best-performing feature subset. Moreover, recent studies have used integrative methods that take advantage of pre-defined domain knowledge, particularly in the fields of bioinformatics and computational biology, where domain knowledge plays a crucial role in the interpretation of results and can improve performance. Each of these approaches is described in detail in the next section.

3.3.1 Filtering Approach

Filtering approaches simply filter out features based on a certain performance metric calculated directly from the data, without using a prediction algorithm. The evaluation of the performance metric, which is known as the relevance score, is an

indicator of how relevant the subset is to the problem being addressed. In addition to some algorithmic procedures, the score is usually calculated with an information-theoretic measures that measures the absolute differences of the values depending on the class labels, such as the chi-square, information gain, mutual information or fisher-score [122]. These scores then are used to rank and select best performing features. The independence of these metrics from a prediction algorithm makes filtering more computationally efficient and practical, especially for high-dimensional datasets. Conversely, the lack of interaction with a training model may have the effect of not improving the performance of the machine learning model.

Filter-based methods are divided into univariate and multivariate methods, depending on how they handle feature associations [123]. Univariate approaches usually evaluate informativeness of each feature individually, rank them using the relevance score and final feature subset is determined by establishing a threshold value or specifying the number of features to retain [124]. Therefore, univariate methods assumes that features are independent to each other. Fisher score, Gini index and Chi-square can be counted among the best known and most widely used univariate methods. Multivariate methods, on the contrary, take into account the interdependencies of features to figure out how they effect as a group [125]. The rationale for the multivariate approach is that features which are individually irrelevant may become relevant when used in combination. Relevance scores of a feature are also dependent to other features that are to be selected or not. ReliefF, mRMR and MIFS could be considered as well-known multivariate methods.

Although multivariate approaches appear to be better at selecting features, it is not possible to say that one approach is better than another. This is because feature selection methods are highly dependent on the data. For example, univariate models rank the features with the highest correlation with the output, and it would be assumed that the top N features from the sorted list are the best at discriminating the target or class label. However, it ignores the possibility that related features may be more discriminating when used together. Multivariate, on the other hand, ignores the isolated discriminative power of each feature. However, multivariate methods are also used as “ranker” as they provide a relevance score for each feature, although scores are calculated jointly. Therefore, the choice of the optimal filtering approach should be based on the data set used. Nevertheless, univariate methods are preferable in terms of computational efficiency as they are much faster than multivariate methods.

3.3.1.1 Fisher Score

The Fisher Score algorithm is a widely used univariate feature selection method that searches for optimal features using within-class and between-class distances. The key idea of Fisher Score to find a subset such that the distance within-class points are minimized while different-class points are maximized in data space spanned by selected features [126]. Normally, for a given subset $Z \in R^{m*n}$ includes selected m feature from input dataset $X \in R^{d*n}$ where n is sample size and d is number of all features, Fisher score calculated as follow:

$$fs(Z) = a\frac{tr(S_D)}{tr(S_W)} \tag{3.10}$$

where $tr()$ function denotes trace of a matrix, S_D and S_W are different-class and within-class data point matrices [127]. However, as there are $\binom{d}{m}$ candidate to form Z, calculating fisher score is became a very challenging optimization problem. To address this difficulty, a heuristic strategy is employed in computing score of each feature individually based on some criteria. Thus, it enabled to shrink computational space of each feature j to $x^j \in R^{1*n}$. Then, fisher score of feature j is calculated with:

$$fs(j) = \frac{\sum_{k=1}^{c} n_k\left(\mu_k^j - \mu^j\right)^2}{\sum_{k=1}^{c} n_k\left(\sigma_k^j\right)^2} \tag{3.11}$$

where μ^j , μ_k^j and σ_k^j are mean of the samples, mean and standard deviation of $k.th$ class corresponding to $j.th$ feature, respectively. Given that the numerator and denominator of the fraction in formula (3.11) indicate between-class and within-class data points, respectively, $fs(k)$ represents the discriminant ability of the j.th feature. The higher fisher score, the more discriminative power the feature has. Since the method assumes that each feature with high scores has good discriminative ability separately, it ignores the improvement ability of using them in combination [128].

3.3.1.2 Relief-F Algorithm

The Relief algorithm is a multivariate filtering method that iteratively computes the weights of features, considering the other samples in the dataset [129]. Algorithm starts with defining n-long weight (W) vector of zeros and taking randomly a sample from the dataset. Using the feature vector of the selected sample, "nearest neighboring" belongs

the same class and the “nearest neighboring” belongs to the different class samples are determined by using Euclidean distance.

$$W_i = W_i - (X_i - H_i)^2 + (X_i - M_i)^2 \tag{3.12}$$

Then, weight vector is updated via formula (3.11), where W indicates initialized zero-valued weight vector, X randomly selected sample, H same-class nearest neighbor sample to X, M difference-class nearest neighbor sample to X, and i i.th feature of the vector. This process is repeated to m times iteratively, and finally relevance vector is obtained by dividing each value of the weight vector by m. The number m can be chosen randomly but usually equal to the sample number.

Thus, if a feature value changes while the same feature of another same-class sample does not change similarly, the weight of the feature is reduced on the assumption that the feature has no effect on the class. Conversely, joint changes of both the feature and feature of same-class sample signalize the feature is discriminative to the class label and therefore weight of that feature is increased [130].

Relief-F an extended version of Relief for multi-class problems that is also more robust and can deal with incomplete and noisy data [131]. Instead of just one nearest sample, Relief-F considers k of its nearest neighbors from the same class, and k nearest neighbors from each of the different classes, which may prevent redundant and noisy features to affect the selection of the nearest neighbors. In addition, Relief-F uses the Manhattan distance to find the nearest samples rather than the Euclidean distance. Rest of algorithmic process are almost same.

The main advantage of relief-based algorithms is that they retain the generalized strengths of the filtering approach, such as relatively fast, selected features not dependent on the predictor algorithm, and flexibility in terms of individual feature weighting [132].

3.3.1.3 Minimum Redundancy Maximum Relevance (mRMR)

When performing feature selection on a dataset, the main purpose is obviously to find the most representative features according to their relation to the class label or phenotype [133]. Because it is trusted that selecting the most relevant features is the best way to figure out important parts of the whole dataset and maximize predictive performance. This is right, but it is not enough. For instance, suppose there are two

distinct features each of which has the same and high discriminative power to sample to the class labels. Standard feature selection, such as a univariate method, solve selection problem by selecting both of them. However, since both of them carry the same information choosing just one of them should be sufficient. Therefore, while selecting features based on their relevance score, redundancy should also be avoided which appears when two highly relevant variables are closely associated with each other.

mRMR is a mutual information-based method that considers both relevance and redundancy of features when selecting them [134]. It attempts to form best feature subset by maximizing relevance of feature to the class label while minimizing feature redundancy which measures the similarity between features. Therefore, relevance and redundancy of each feature in candidate subset should be calculated. Measuring relevance is relatively easier, because "maximum dependency" between x_i and class label c can be calculated using mutual information (MI), as shown in formula (3.13) where $p(m)$ and $p(n)$ are marginal probabilities and $p(m,n)$ joint probability of given two variables.

$$I(M;N) = \sum_{m,n} p(m,n) \log \frac{p(m,n)}{p(m)p(n)} \tag{3.13}$$

Since mRMR aims to maximize relevance, maximum value of MI between a feature set S and class should be selected, therefore relevance score of set S calculated using formula (3.14), where $|S|$ is the size of features subset S.

$$\max D(S;c); D = \frac{1}{|S|} \sum_{x_i \in S} I(x_i, c) \tag{3.14}$$

The larger value of MI indicates that the feature has higher discriminative information. In this way, iteratively maximum relevant features can be added to optimal subset S. However, selecting features based only on maximum relevance can lead to high redundancy in the subset [135]. While choosing each next optimal feature (i.e. high relevant), redundancy between features in iteratively formed subset S should be considered.

$$\min R(S); R = \frac{1}{|S|^2} \sum_{x_i, x_j \in S} I(x_i, x_j) \tag{3.15}$$

This need can be solved employing MI criteria because of the fact that it also measures amount of information between features using the formula (3.15), where x_i, x_j

indicates two different features. if the value of MI increases when the next feature is added, it means that the duplicate information increases, and the feature can be discarded.

$$\max_{x_i \notin S} = \left[I(x_i; c) - \frac{1}{|S|} \sum_{x_j \in S} I(x_j, c) \right] \tag{3.16}$$

mRMR feature selection method optimizes these evaluation criteria relevance D and redundancy R simultaneously, as following formula (3.16). Iteratively increment procedure to increase number of features in set S as follows [136]:

1- The optimal feature x_i with the highest mutual information between features is determined using $I(x_i; c)$ and added to the empty feature subset S.

2- Then, the subsequent highest information carried feature x_j is selected which satisfies formula (3.16).

3- Step 2 is repeated until the stopping criteria, such as the number of selected features, is reached.

For example, some coding part of a gene contain multiple probe intensities while measuring with a microarray. Originally, all these probes are associated with the same gene. Therefore, the effect on the class label activity might be similar. mRMR is one of the best methods that perceives the redundancy among them. In fact, the mRMR approach was first proposed as a gene selection method [137]. It is therefore widely used in computational biology, genetics and multi-omics studies.

3.3.2 Wrapper Methods

Discarding the prediction algorithm during feature selection may result in poor prediction performance. Filtering algorithms select features based on their correlation with class labels, without using a classifier or predictor. Therefore, the selected subset may not contain the features that are most important for improving the performance of the predictor algorithm. If the main purpose of performing feature selection is to improve predictive performance, the relevant feature subset will reflect the classifier characteristics should be selected [138]. To meet exactly this need, a wrapper approach has emerged.

Wrapper approach relies on the utilize predictive power of a chosen machine learning algorithm as an evaluation metric to aid optimum feature subset. Technically, prediction score of each iteratively generated subset that evaluated with the algorithm reflects the quality of the subset. Feature dependencies, interactions and redundancies have been considered during the selection of the subset. Thus, wrappers enable to identifying best-performing features in terms of prediction performance even though selected ones not have grounded association with class label. However, assessing the possible feature combination is computationally intractable, especially high-dimensional data like SNPs or gene expression [139]. To keep wrapper method feasible, number of generated subset or iteration should be reduced using a search strategy. The search strategy is a way to find a subset with the highest evaluation score, using a heuristic function to guide it [140]. These methods start with a randomly generated subset and iteratively move one step closer to the best solution [141]. Heuristic approaches are usually categorized as sequential and randomized search. There are 2 flavors of sequential methods: forward selection and backward selection.

Starting with an empty set, forward selection (i.e., sequential forward selection, SFS) algorithms iteratively add each candidate feature that is not already in the subset. Each addition proceeds by evaluating the subset score. This cycle continues until a specified number of features have been selected or no improvement is observed after a specified number of iterations [142]. In contrast, backward selection (i.e. Sequential Backward Selection, SBS) iteratively eliminates the least promising feature from a set consisting of all candidate features. Similar to SFS, the pruning process continues by checking the subset prediction score one by one until a certain number of features remain or the subset score decreases [143].

Although the sequential methods are easy to implement and provide a better way than exhaustive search, they often tend to get trapped in a local optimum [144]. For example, due to 5 times consecutive non-improvement in performance might be stopped the elimination of features in SBS. To avoid trapping in local optima problem and accelerate the running time, randomized search strategy is introduced. As the name implies, randomized search aims to select features at random within a specific logical framework such as an evolutionary algorithm. For each iteration of this approach, a random subset is generated or iteratively modified with the aim of maximizing prediction performance [145]. These methods usually employes or combine an existing heuristic

algorithm such as Genetic algorithm [146], simulated annealing or Artificial Bee Colony (ABC) [147]. Thus, well-developed optimization algorithms are utilized to avoid from getting trapped in local optima. Besides maximizing the prediction performance, the main advantage of wrapper algorithms is that there is no need to specify or determine the threshold for selecting the number of features. The optimal subset is obtained immediately after running a wrapper. However, it is not known which features are relatively more important within the set, as wrappers do not calculate a score for each feature.

3.3.3 Embedded Methods

In most machine learning studies, the main goal is to maximize the predictive performance of the model. Among the feature selection approaches that provide a way to improve predictive performance, filtering is not associated with a predictor, so performance can be poor. Wrapper, on the other hand, directly evaluates each subset separately with an algorithm, but it has a major drawback, which is high runtime [148]. Therefore, there is a need for an approach that both uses an algorithm to enhance predictive performance and runs relatively faster than Wrapper.

Similar to wrapper, embedded methods also incorporate a machine learning algorithm during feature selection process. However, wrappers are actually utilize the predictive performance of a given algorithm, not learning process. In contrast, feature selection is a part of learning algorithms in embedded methods. In other words, features are selecting during the training process of the algorithm. Therefore, embedded methods consider the dependencies amongst features, as well as the relationship between the input feature and the output [149]. Moreover, advantage of interacting with the learning algorithm during training results in simultaneous selecting features and thus reducing computational time compared to wrapper methods. The embedded approaches usually grouped under two categories regularization and tree-based [150].

3.3.3.1 Regularization-based Embedded Methods

Machine learning essentially aims to extract a pattern from input data. When the same data are trained repeatedly, not only features but also noise are learned. In this case, the model will not be able to accurately make predictions for new samples or other data sets. The reason for this is the exact adaptation, or fit, to the data. In other words, the model has become over-fitted. Regularization is a technique that have ability to limiting

complexity and preventing the occurrence of overfitting in a model by suppress learning coefficients [149].

$$Regularized\ Loss = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{m}\beta_j x_{ij}\right)^2 + \lambda \sum_{j=1}^{m}|\beta_j| \quad (3.17)$$

This technique is mostly used in linear models and basically adds a penalty value to the loss function to control over-raising of coefficients, as seen Lasso in formula (3.17) where n, m, β and λ indicates sample size, feature size, coefficient, and regularization parameter respectively. If the model is overfitted by over-weighting or under-weighting, the loss value will also be high because the penalty term is also high. The model will therefore tend to shrink the coefficients as low as possible.

Coefficients are not directly related to the importance of features because they have different scales or ranges. This means that it cannot be concluded from a feature is significant if the coefficient is high. Nevertheless, if a coefficient is 0, this is evidence that the feature has not affect the outcome due to the nature of linear regression. Consequently, features having zero coefficients could be marked as redundant and removed. Hence, regularization techniques can be used as feature selection to remove some features from the model and make it more robust, less complex, and computationally faster.

There are 2 well-known regularization-based embedded techniques, Lasso and Ridge. Both use Linear Regression, but the penalty terms for the loss function differ. While Lasso (i.e. L1 norm) penalized with the absolute value of each feature coefficient, Ridge (i.e. L2 norm) penalized by the squared magnitude of each feature coefficient. Since quadratic form of the Ridge penalty term, a small number of coefficients will be set exactly zero. On the other hand, Lasso penalty term leads many parameters to equal zero rigorously due to linear form, resulting in efficiently discarding redundant features [151].

3.3.3.2 Tree-based Embedded Methods

Tree-based methods, which build trees iteratively by recursively partitioning the data by features, are the most popular and powerful prediction algorithms. In the determination of the features of the partition samples, each feature is calculated in accordance with some specific standards, and the most important feature is selected as the feature of the partition samples each time [152].Thus, tree-based methods have an

inherent feature selection process, as the splitting features are selected based on their discriminative power.

One of the early implementations of the tree algorithm, ID3 proposed by Quinlan [153], used information entropy to measure the importance of each feature. It is expected that the entropy will be lower when a system has a stable processing. Therefore, the decision tree is constructed based on these using information gained from features and the feature of maximum entropy reduction is selected to divide the data. Since the standard ID3 algorithm can only deal with discrete features, an improvement version of it was proposed called as C4.5 algorithm [154]. C4.5 uses the information gain rate. Information gain is the amount of knowledge gained from a particular decision or action. It is calculated by comparing the entropy of the original set with the weighted sum of the entropies of the subsets created by splitting the data on a particular feature. In the following years, Gini Index was proposed as the measurement of the features [155]. The differences among feature measurement techniques have also led to emerge of different feature selection methods depending on measurement during splitting such as information gain based or gain ratio based feature selection.

There are also ensemble tree-like feature selection methods that follow the same procedure but combine multiple decision trees through processes such as bagging or boosting, such as the XGBoost or LightGBM feature selection techniques. However, as there are multiple trees, the importance scores of features are calculated for each tree, then averaged across all trees and finally normalized to sum up 1 [156].

3.3.4 Hybrid Methods

Conventional feature selection methods typically yield different optimal feature subsets. This is because each method has a different statistical or computational underlying theory. For example, the Relief method makes calculations based on sample similarity within and between classes, while the Fisher score focuses on the significance of each feature to the class label. The Wrapper approach, on the other hand, decides the importance of a subset in a completely different way, using the predictive power of a machine learning model. As each approach has its own theoretical background, they offer some advantages or disadvantages in different aspects. A comparison of the strengths and weaknesses of each approach is shown in Table 3.1. Searching a way to use strengths

sides of each approach to obtain best-performing feature subset led to emerge of hybrid perspective.

Table 3.1 Taxonomy of three feature selection approaches with advantages and disadvantages [157].

Approach	Advantages	Disadvantages
Filtering	**Univariate**	
	Fast	Ignores feature dependencies
	Scalable	Ignores classifier interactions
	Independent of the classifier	
	Multivariate	
	Consider feature dependencies	Slower than univariate methods
	Independent of the classifier	Less scalable than univariate methods
	Better in computational compexity than wrapper	Ignores classifier interactions
Wrapper	Simpler than Filtering methods	Risk of over fitting
	Interacts with the classsifier	Classifier dependent selection
	Considers feature dependencies	Computationally intensive
Embedded	Interacts with the classsifier	Classifier dependent selection
	Less intensive complexity than Wrappers	
	Consider feature dependencies	

Hybrid methods are essentially a combination of different feature selection methods in sequential steps, taking advantage of each method [158]. The first step usually starts with applying a filtering approach to the dataset, since filters are simple and run quickly. It also allows features to be sorted by importance and the number of features to be reduced by a threshold. However, filters do not account for the effects of feature correlations on predictive performance. To account for dependencies between features, a wrapper method can be performed on the sorted features. In this way, the most important features are considered, both in terms of individual feature values and prediction performance. Moreover, some heuristic and evolutionary algorithms have also been utilized to determine the best feature subset within hybrid methods. For example, Butler-Yeoman et al. have shown that combining a hybrid approach with particle swarm optimization outperforms filtering while being less computationally complex than wrappers [159]. Because they can be easily combined with a variety of methods and ideas, hybrid methods offer a flexible solution to the challenge of finding the optimal feature subset.

3.3.5 Integrative (Knowledge-Based) Methods

Feature selection methods have been in search of new ways to find the best subset as the number of data and features representing the samples increased. Primitive approaches rely on rules or expert opinion to find important properties of the data, while current approaches like filtering, wrappers and embedded depend on data-driven analysis. Being "data-driven" in feature selection means determining relevant features that contribute to predictive performance based solely on the data presented, without any external source [160].

Besides theoretical and academic contribution, data-driven approaches offer many practical utilities in various fields, such as text mining and image processing, where data is principal. However, due to the complex and multifaceted nature of biological systems, it may not be sufficient adhere to only the input data in biological or genetic experiments to obtain coherent results with the biological domain. This is because some feature selection approaches may identify some important parts, e.g. genes, but these may only have been selected based on prediction performance and have no real association with phenotype. If the experiment is designed solely to improve prediction performance, the biological aspect of the results may not be valuable. Conversely, if a logical connection with the biological domain is sought, integrative methods should be used.

Integrative feature selection refers to the process of combining multiple sources of knowledge in subset selection. Particularly in bioinformatics studies, the incorporation of additional sources and the integration of prior knowledge about the underlying biology allows for more accurate interpretation and improves the reliability of findings [161].

One of the most common ways in integrative approaches is to use the information in knowledgebase repositories derived as a result of various experiments. For example, in study [162], the authors used biological relationships of genes provided in the Gene Ontology (GO) repository to rank genes for a cancer-related microarray dataset. GO Annotations were used both to correct inaccurate measurements of the microarray technology and to detect redundancies between genes. Integrative use of GO annotation led to more accurate results, as reported.

3.3.6 Ensemble (Aggregation) Methods

Ensemble learning, also known as committee-based learning, briefly combines multiple learners to solve a problem being addressed [163] . The rationale behind ensemble learning is to use multiple methods and then combine them or their outputs to achieve better results, treating them as a 'committee' of decision-makers [164]. In this way, the advantages of different models are utilized. The effectiveness and efficiency of that approach has also been proven in many prediction studies through recent years [165,166].

Despite the fact that idea of ensemble learning usually to be associated with classification problems, it can be used to improve other machine-learning disciplines like feature selection [167]. Since each feature selection method has a special statistical or computational background for evaluating subsets, selected features are varying from method to method. In addition, the selected features may also vary depending on the parts of the same dataset. Ensemble selection provides a solution by aggregating the outputs of many selector methods, usually improving performance, and freeing the user from having to choose a single method.

Ensembles for feature selection are usually divided into 2 approaches. If the selection methods are all the same, but the training data varies over several nodes, it is called a “homogeneous” approach. Otherwise, i.e. feature selector methods are changing, it is called as "heterogeneous" approach [168]. A heterogeneous approach takes into account the strengths and weaknesses of each method. Therefore, if a method fails to select important features, it reduces the overall performance of selecting optimal subset. Nevertheless, it is likely to yield a more discriminating subset because it combines selected features from different methods.

3.3.7 Domain Knowledge Based Subset Selection (DKSS)

As mentioned above, integrative feature selection methods have become more popular in recent years in favor of being able to utilize community knowledge in the selection process. For the second experiment of this thesis, about Behçet’s Disease, we have proposed an integrative feature selection method that uses biological networks in SNPs selection.

A biological network is a graphical representation of genes, proteins or other biological molecules that contains physical/functional interactions and helps to understand cellular processes and disease mechanisms. These interactions can be represented in terms of nodes and edges, with the nodes keeps the biological molecules and the edges express the interactions between them. By studying biological networks, researchers gain insight into the complex web of relationships within living organisms. When analyzing a biological network, identifying active sub-networks becomes crucial as it allows researchers to focus on the most relevant genes and their interactions.

An active sub-network is the connected subgraph of a biological network that comprises genes that are significantly associated with disease-predisposing single nucleotide polymorphisms (SNPs), based on genotypic p-values. These SNPs are genetic variations that have been linked to an increased risk of developing a particular disease. Understanding of the mechanisms of disease development can be gained from the genes in an active sub-network and their interactions. Thus, active sub-networks could be used to predict disease [169].

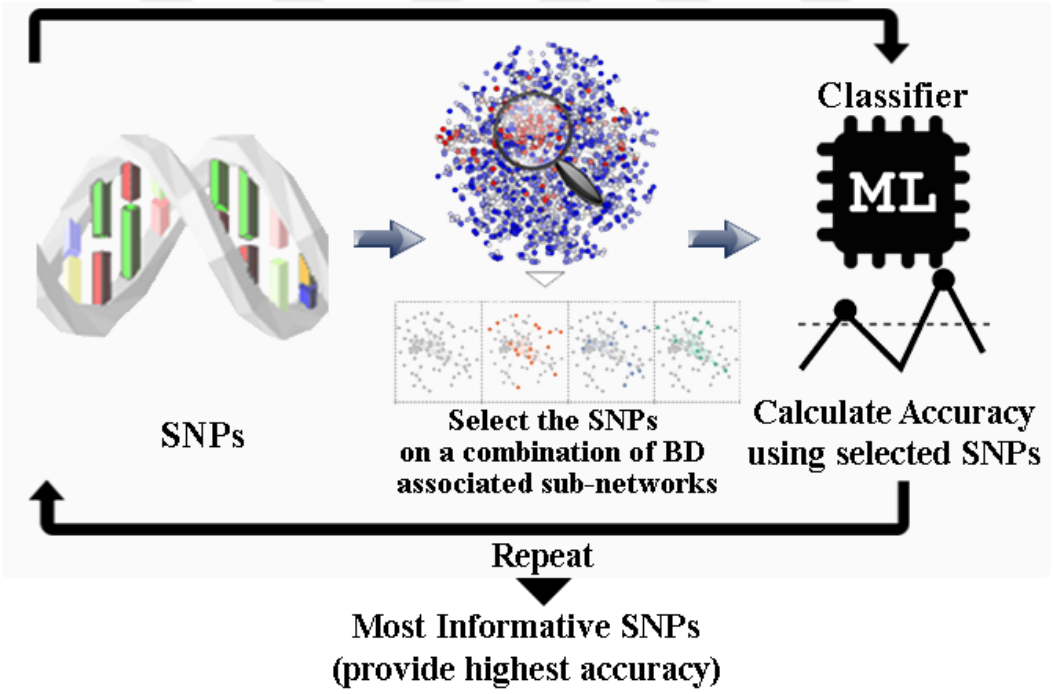

Figure 3.3 Steps of of proposed DKSS methods as a feature selection approach for the prediction of Behçet's Disease problem [170].

DKSS is theoretically based on the idea that selecting SNPs within the same active network would be more biologically meaningful, as these SNPs are already related to

each other. This is because standard feature selection methods either evaluate each feature independently, or a combination of several, but evaluated according to a class label. This can provide a more biologically accurate evaluation of the results obtained, even if it does not improve classification performance.

The idea behind DKSS is that a SNP is selected if the gene associated with that SNP is included in an active subnetwork. Figure 3.3 shows general workflow of DKSS method.

Table 3.2 Domain Knowledge Based Subset Selection Pseudo Code.

Input:	***Subnetworks****:* List contains predefined active sub-networks
	Data*:* GWAS data where features are SNPs.
	Min_SN: Minimum number to be chosen subnetworks
	Max_SN**:** Maximum number to be chosen subnetworks
	Classifier: A base learner
	RepeatCount: Number of repeat times (100 default)
	Min_SN & Max_SN: Numbers to be selected Minimum and Maximum subnetworks
Output:	***List of Selected SNPs***
	Function: getBestSNPList*(Data,Subnetworks,Min_SN,Max_SN,Classifier,RepeatCount)*
1	**for i= Min_SN : Max_SN**
2	SelectedSN = choose **i** active networks from **Subnetworks**)
3	snpList = union SNPs from **SelectedSN**
4	Rearrange Data by filtering on SNPs in **snpList**
5	ACC = Train and Evaluate Accuracy of classifier using Rearranged Data
6	If **ACC** > *prev***ACC**:
7	BestSNPs = **snpList**
8	return **BestSNPs**

The first step in DKSS is to define the minimum and maximum number of subnetworks to be selected. The active sub-networks are then randomly chosen from the pool of sub-networks and the SNPs associated with the genes in the selected sub-network are selected. A base learner is used to calculate the classification score of samples with selected SNPs. This process is repeated 300 times, whereby the number of trials can be specified as a parameter. The SNPs that yield the highest classification score are kept as the final set of SNPs. In this way, the biological information contained in the active subnetworks of BD, the associated proteins, genes, and SNPs are integrated with the statistical methods. Pseudo code of DKSS method is given in Table 3.2.

3.4 Enrichment Analysis

Advanced technologies in information systems have enabled to accelerate studies of computational biology and genetics, as in many other fields. In addition, the falling cost of storage devices has made it possible to store vast amounts of genomic data generated by high-throughput sequencing technologies. This allows researchers to analyze genetic data from thousands of samples across hundreds of diseases and identify biomolecules, such as genes, that are involved in disease development. However, interpreting the outcomes is a challenge for computational studies. For example, although a computational analysis may show that a particular gene is highly expressed after infection, the gene may be completely meaningless when interpreted alone. Furthermore, even though all differentially expressed genes are extracted, it needs to be known relation between them to understand effecting mechanism of the infection. At this point, further analyses are needed to uncover biological themes associated with outcomes and to interpret them appropriate to domain knowledge.

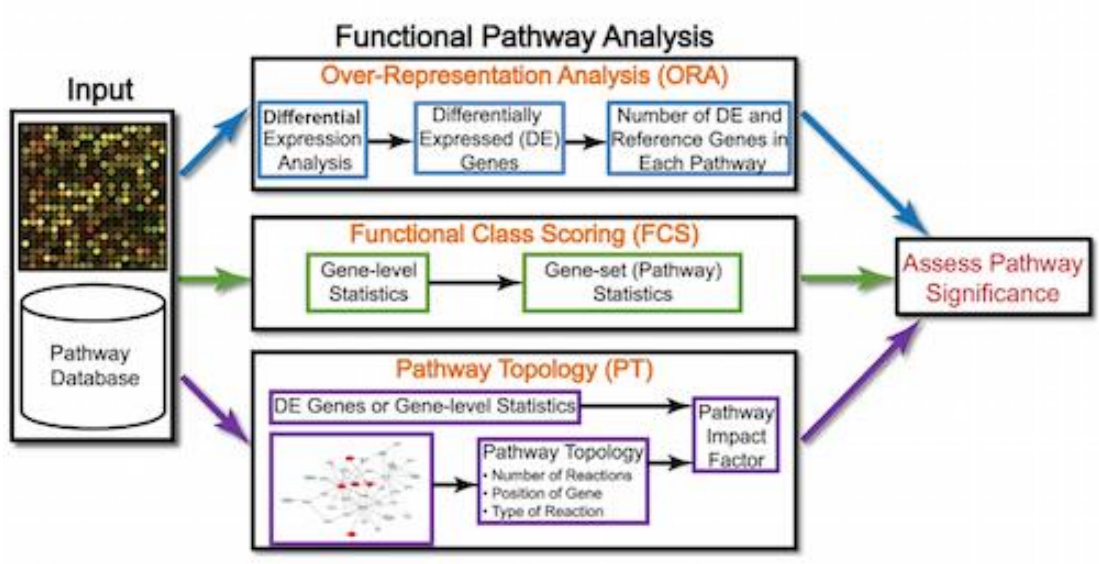

Figure 3.4 Types of Functional Pathway Analyses, ORA, FCS, and PT [172].

Enrichment analysis is a computational method for determining whether certain functional categories, biological processes, or features are overrepresented in a particular group of genes, genomic regions, or other types of molecular materials. The main principle of enrichment analysis is to compare the observed genes (or genetic material) within a predefined target set with the expected occurrences based on a reference by

identifying statistically significant. It can reveal how these entities collectively function and interact in a biological context by mapping findings to existing knowledge of biological sets (gene sets, pathways, etc.) [171]. Hence, results of enrichment analyses help researchers to gain a deeper understanding the underlying molecular mechanisms involved in disease.

The enrichment approaches are usually split into three categories, over-representation analysis (ORA), functional class scoring-based methods (FCS) and pathway topology-based methods (PT) [173]. While ORA has a simple theoretical background relying only on overlap between the list, FCS methods try to use all the information in gene expression values. On the other hand, PT methods consider the topological importance and relationships of genes in a pathway [174]. In our thesis experiments we have used most-known types, ORA and GSEA.

3.4.1 Over Representation Analysis (ORA)

ORA is a simple, easy-to-implement and statistically well-established gene set analysis method that makes it possible to performing single-gene analysis on a set of genes [175]. It basically evaluates the fraction of given list of genes in pre-defined gene sets using a statistical test. Although binomial distributions or chi-squared tests are also used in the literature, the hypergeometric distribution is most commonly used way to test for over-or under-representation in a given gene list.

$$P(X \geq k) = \sum_{k=0}^{x} \frac{\binom{K}{k}\binom{N-K}{n-k}}{\binom{N}{n}} \tag{3.18}$$

Performing ORA requires only main three inputs, a collection of gene sets (or pathways), an observed gene list of interest, and reference set of genes. Reference set usually microarray platform of collection for gene expression profiles such as Affymetrix HGU133. As an enrichment scoring, the statistic of interest is the probability of observing k or more genes in the set or pathway by chance. Therefore, probability of observing at least k genes in a set by chance is calculated by formula (3.18) where K and k denote size of gene list of interest and number of genes of interest that are also in given pre-defined gene sets, n is size of genes in pre-defined set, and N the size of background reference set. As the hypergeometric is a discrete distribution, a one-tailed statistic is the sum of

probability mass functions calculated at a set of values equal to or more extreme than the value of interest [176].

3.4.2 Gene Set Enrichment Analysis (GSEA)

Despite practicality and widespread usage, ORA has some drawbacks that affect the reliability of results. ORA treats each given gene independently, and thus it evaluates them equally in terms of effect to biological process by ignoring intensities. Additionally, biological studies tend to consider only differentially expressed genes, which are often selected by applying a p-value cut-off. However, biomolecules such as genes and proteins interact together and constitute sets and pathways even if p-values are slightly greater due to complexity of biology. Contrary to ORA, FCS based methods tries to use whole expression information including all genes, gene intensities etc. by solving invalid equality and independence assumption of ORA [172].

Gene Set Enrichment Analysis (GSEA) is one of the most widely used FCS method that is a statistical method to evaluate whether a predefined set of genes (or proteins) derived from a biological, biochemical, or computational analysis can provide information on the differences between two different biological states [177]. The method briefly compares gene expression patterns between two groups and identifies pre-defined gene sets, or pathways, that are significantly enriched in one group compared to the other. These groups, often called phenotypes, represent the different classes or groups of samples such as male-female, green-blue-eyed, or healthy-infected. On the other hand, gene sets often represent specific biological processes, molecular functions, or cellular pathways, providing valuable association information.

GSEA results allow researchers to explore similarities between phenotypes associated with particular disease gene lists or pathways, thereby identifying functional associations to shed light on the underlying biological mechanisms. For example, if gene sets associated with skin cancer are over-enriched in samples belonging to the “green-eyed” phenotype, it can be argued that there is a biological link between “green-eyed” individuals and skin cancer. Hence, GSEA can help researchers prioritize genes and pathways for further investigation, leading to a better interpretation of the experimental results [178]. Furthermore, analyzing the enrichment of gene sets across multiple datasets may allow to reveal of pathways or biological processes that may be involved in

seemingly unrelated events. Nonetheless, it should be noted that these inferences stem solely from data-driven results and necessary confirmation through biological experiments.

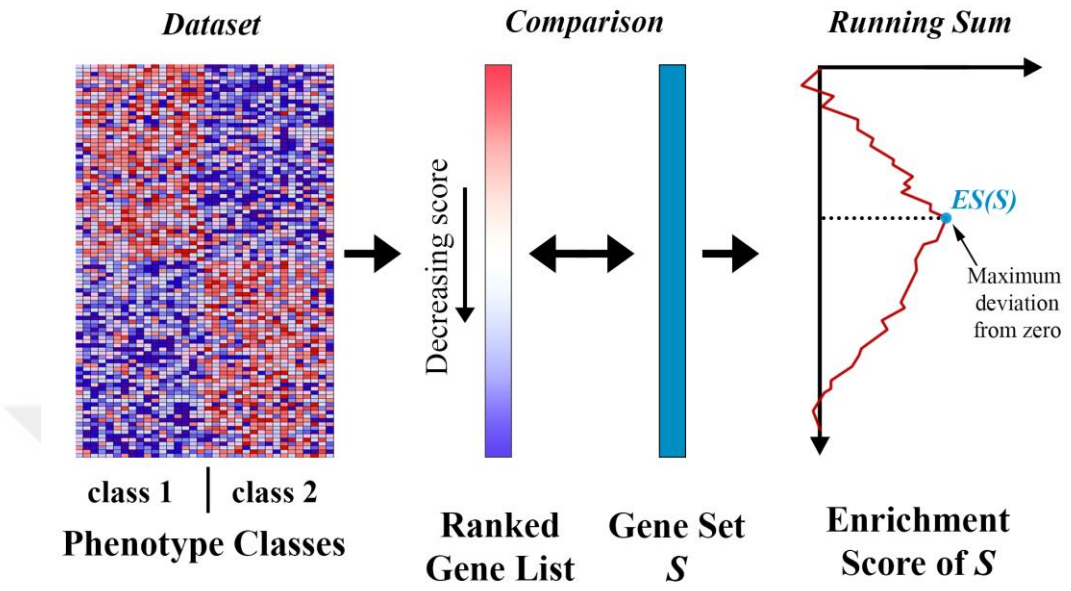

Figure 3.5 Steps of the GSEA. Genet Set S express the pre-defined gene set and ES(S) is the value of representation degree of given set S on the dataset.

GSEA mainly consists of 3 steps. Initially, the mean expression value of each gene is calculated according to each phenotype using a metric such as Wilcoxon test, t-test, or signal-to-noise ratio (default), which represents the association degree between the gene and the phenotype. As a result of the calculations, each gene is assigned a score indicating how much the gene is related to the first or second phenotype. A positive correlation indicates that the gene is more related to the first phenotype, while a negative correlation indicates that is more related to another phenotype. Subsequently, the genes are sorted in descending order based on these scores to form ranked list *L*.

The second step is to calculate the enrichment score (ES) between the ranked gene list L and a predefined gene set *S*. The genes in the set *S* can consist of genes that are associated with a biological pathway, that are curated as a result of an experiment, or that are independently collected genes. If the set S is related to a phenotype, e.g. infected individuals, then genes in the set *S* will tend to have a higher association score than genes in gene sets not related to the phenotype. The proportion of genes in *S* ranked at the top of list *L* is expected to be larger than the proportion of others [179]. Thus, the difference in the cumulative proportions of genes present in set *S* and not present in set *S* can be

used to assess the degree of association or enrichment of set S on list L, while walking down the ranked list L. In other words, If the i.th gene in the L list is also included in the S set, the "hit" score, otherwise the "miss" score increased, and the ratio of these scores indicates the enrichment score up to the i.th gene. Eventual ES of the set S is the maximum deviation from zero observed during this walking down process, as shown in Figure 3.5.

$$P_{hit}(S,i) = \sum_{gene_j \in S, j \le i} \frac{|r_j|^p}{N_R}, N_R = \sum_{gene_j \in S} |r_j|^p \quad (3.19)$$

$$P_{miss}(S,i) = \sum_{gene_j \notin S, j \le i} \frac{1}{(N - N_H)} \quad (3.20)$$

$$ES(S) = \max\left(|P_{hit}(S,i) - P_{miss}(S,i)|\right) \quad (3.21)$$

Mathematically, the enrichment score up to the i.th gene in the ranked list L is equal to the difference of P_{hit} and P_{miss}, shown in formula (3.19) and formula (3.20), respectively, where r_j expresses the correlation score of $gene_j$, N number of genes in the ranked list L, $N - N_H$ number of genes in the list L but not in the set S, and p is a weighting factor that allows to reducing score for gene sets enriched near the middle of the ranked list L. Starting at the top of the list L, running sum is calculated by increasing the P_{hit} if gene is present in set, otherwise P_{miss}. This process is repeated for each element in L. Finally, maximum absolute value of $P_{hit} - P_{miss}$ at a point during running sum gives the enrichment score for set S.

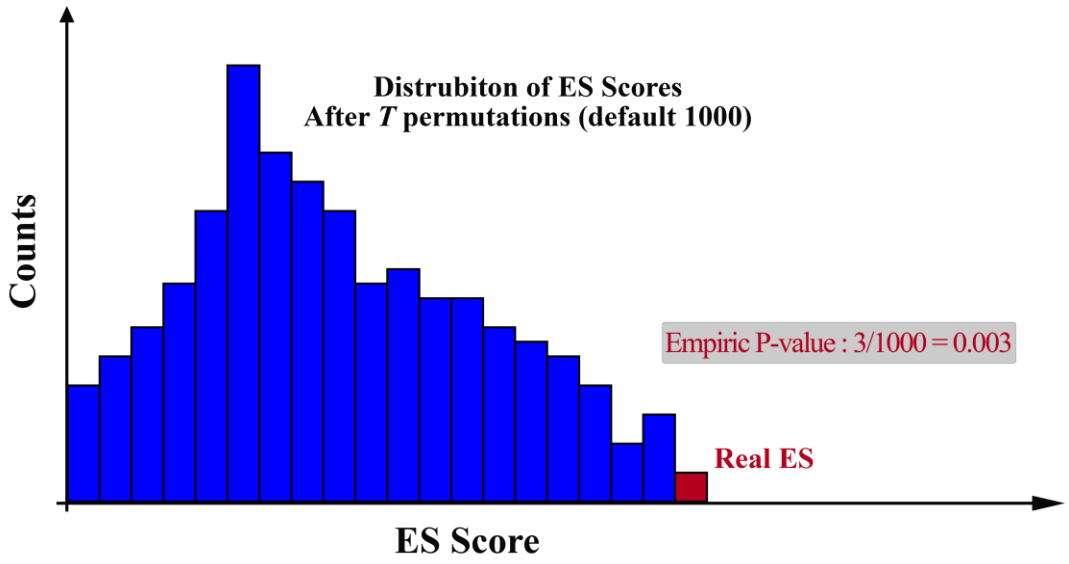

Figure 3.6 Calculation of final ES Score of GSEA method with P-value. Real ES express the obtained enrichment score when original dataset is used. Other ES scores obtained when the dataset permutated.

This score indicates the degree of enrichment of S, but another problem is whether or not this value is statistically significant. Therefore, the final step is to estimate the significance value of the calculated ES. Since each ES can be obtained using a random class distribution, the statistical significance level of the ES for a gene set S is crucial for evaluating the performance of GSEA.

In order to show significance, P-value is calculated through a permutation-based approach. The p-value provides a quantitative measure for the observed enrichment score due to random chance, assuming the generated null hypothesis that there is no association between the gene set and the phenotypes. To do this, the class labels of the dataset are randomly assigned to the samples, thus permutated version of the dataset D_{null} is generated. Then, ES_{null} of D_{null} is calculated as explained above. After repeating this process T times (1000 by default), a histogram is created that includes frequency distribution of ES_{null}, as seen in Figure 3.6. The ratio of the ES frequency obtained from real data to the total number of permutations shows the p-value, which shows significance degree of the ES score. The lower the P value, the less likely the results are to be random. Therefore, gene sets with high P values should not be considered when interpreting the results. In the literature, a 0.05 cut-off is generally accepted as the confidence level for the P value [180].

3.4.3 Single Sample Gene Set Enrichment Analysis (ssGSEA)

The classical GSEA utilize pre-defined sets to find out which sets are more enriched on the given samples and thus it enables to dig underlying associations of genes for phenotypes. In other word, GSEA results can be assessed on phenotype level or about entire dataset given for analysis. Therefore, the GSEA cannot provide an insight about gene associations of only an individual. Because the GSEA cannot be applied directly to a gene expression data of sample due to the fact that it requires at least 2 phenotypes, and each sample must be belonged to only one phenotype or class label. To overcome that problems, Single Sample Gene Set Enrichment Analysis (ssGSEA) is emerged.

Single Sample Gene Set Enrichment Analysis (ssGSEA) is an extension of the GSEA that allows for the estimation of pathway or gene set enrichment for individual samples. It is particularly useful when working with single samples or when the focus is

on characterizing the activity of gene sets within a specific sample rather than comparing gene sets between different conditions.

Unlike standard GSEA, ssGSEA uses gene expression values directly in the gene ranking phase, since each given sample belongs to only one specific phenotype. For a given sample m, sorted gene list L_m is formed using expression values.

$$P_S^w(S,m,i) = \sum_{gene_j \in G, j \le i} \frac{|gene_j|^p}{\sum_{gen_j \in S} |gene_j|^p} \tag{3.22}$$

$$P_{NS}(S,m,i) = \sum_{gene_j \notin S, j \le i} \frac{1}{(N - N_H)} \tag{3.23}$$

$$ES(S,m) = \sum_{i=1}^{N} [P_S^w(S,m,i) - P_{N_S}(S,m,i)] \tag{3.24}$$

For each $gene_i$ in L_m, the "hit" score is calculated according to formula (3.22) if the $gene_i$ is also within S, otherwise the "miss" score is calculated according to formula (3.23), where m is sample, p is the damping factor that controls weight of expression value, N and N_H represent size of L_m and pre-defined gene set S. Similar to standard version, all genes in L_m are processed using the formulas sequentially starting from highest expression-valued gene. Eventually, a histogram that shows running sum enrichment scores is generated. Maximum deviation from zero, i.e. highest point of running sum is marked as $ES(S,m)$ of the given sample m. In addition to being able to apply at the sample level, another major advantage of ssGSEA is that it allows samples to be represented in different feature spaces. For example, each sample in a given dataset can be represented in a vector containing only gene sets or pathways associated with a cancer type. This could improve the prediction performance or forecasting about the individual sample.

3.5 Hyper-Parameter Optimization

One of the factors that lead to high predictive performance for machine learning methods is the proper tuning of hyper-parameters. When the hyper-parameters of an algorithm are tuned properly, the prediction accuracy can be increased. Machine learning algorithms usually have two types of parameters: model parameters and tuning parameters. Model parameters are initialized and updated during the learning process, such as coefficients in linear regression [181]. Algorithms already optimize these

parameters internally. The tuning parameters or hyperparameters, on the other hand, are parts architecture of the statistical background of the algorithm and have to be defined before starting of the learning process. Therefore, building an optimal machine learning model also relies on exploring the range of all possible hyperparameters of the algorithm. The traditional way to tune hyperparameters is manual testing which is composed of defining a set of candidate values and estimating the model utility over the candidate values [182]. However, the hyperparameter settings of the algorithm should be well-known to tune the parameters manually. Moreover, manual testing is practically infeasible due to several factors, including non-linear interaction, a large number of parameters, and the complexity of the model. These factors have led to the emergence of hyperparameter optimization (HPO) techniques as grid search, random search, and Bayesian optimization [183].

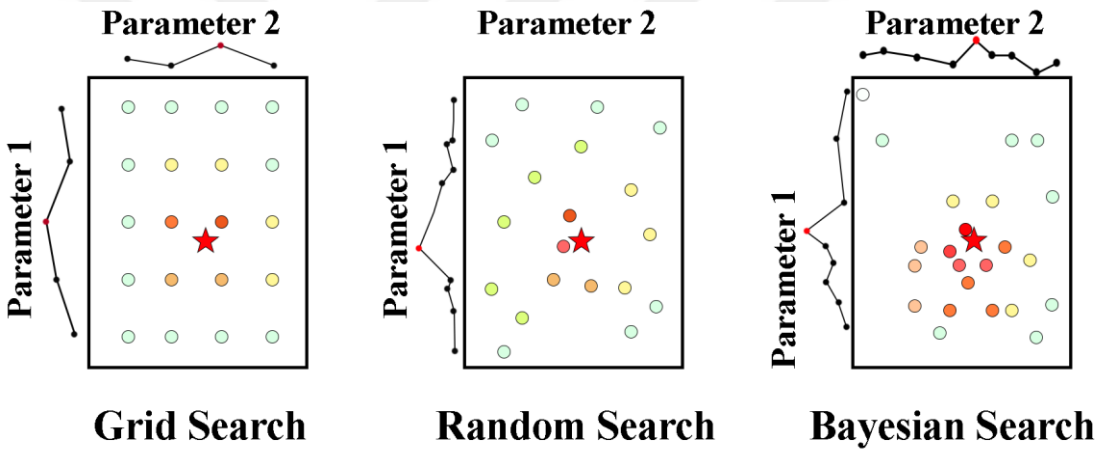

Figure 3.7 Illustration of grid, random and Bayesian search based hyperparameter optimization. The red star indicates the optimal parameter set.

Grid Search (GS) is a systematic search method that exhaustively evaluates all possible combinations given the predefined configuration grid [184]. The range of values of each hyperparameter is discretized and models are evaluated using all possible combinations. Although it has a simple implementation, parallelization, and certain guarantees, the computational cost increases exponentially with the number of hyperparameters. Therefore, it is not efficient in experiments with high-dimensional data.

Random Search (RS) overcomes the limitations of grid search by randomly selecting candidate values between the upper and lower bounds of predefined hyperparameters. The theoretical basis of RS is that given a sufficiently large configuration space, it is possible to find the global optimal solution or at least its closest

approximate solution [185]. Independent evaluation of each randomly selected parameter set allows flexible resource allocation, making random search more feasible in terms of computational cost than grid search. In addition, experimental results show that random search gives better results both theoretically and empirically regarding hyperparameter tuning [181].

Both of RS and GS techniques evaluate the candidate parameter set independently. Candidate parameters do not guide models on how to limit the search space to ensure optimal search, although they provide feedback parameter optimality through performance metrics. Thus, searching on poorly performing parameter bounds leads to massive time wastage. This need is addressed by advanced optimization techniques like Bayesian optimization.

Bayesian optimization is a prominent technique frequently employed in artificial intelligence applications and other disciplines where a function with an unknown analytic form needs to be optimized [186]. Unlike GS and RS, the Bayesian approach considers prior trials to guide the next search for optimal points by integrating a surrogate model and acquisition function. Surrogate model, such as Gaussian process, fits available data points and provides a posterior distribution over the objective function. Then, this model is utilized to the next candidate parameter point to evaluate based on an acquisition function. Acquisition function quantifies the value of sampling a specific point in the search space, considering both the predicted function value at that point (exploitation) and the uncertainty of the prediction (exploration) [186]. Typical acquisition functions include Expected Improvement (EI), Probability of Improvement (PI) and Upper Confidence Bound (UCB). Whereas exploration indicates candidate points within unexplored space where the prediction uncertainty is high, exploitation involves sampling candidate points in the current search region where the global optimum is most likely to occur based on the posterior distribution [185,187] . The key inspiration of BO is to balance the processes of exploration and exploitation. In this way, optimal regions for hyperparameter candidates, including missing better regions, can be discovered. The main drawback of the BO approach is the parallelization of the tuning process since the model depends on previously tested values. However, it is more efficient than other approaches because updating the surrogate model after each candidate point evaluation leads to a search over the optimal region.

3.6 Performance Evaluation Metrics

In the majority of studies on the application of machine learning, the primary objective is to predict true sample outcome(s) as accurately as possible. Accuracy is an indicator of “success” how well the model is able to learn patterns from the data to be used in the prediction of samples. However, the definition of “success” may vary depending on the nature of the problem, the data to which the model is applied, the expected outcomes, and the risks. For instance, consider a dataset in which 90% of the cases are noncancerous and 10% are cancerous. A model that predicts non-cancer cases for each case would achieve 10% accuracy even though it does not identify cancer cases, which is the critical goal in cancer prediction. In this scenario, high accuracy does not necessarily indicate a good cancer prediction model. Therefore, different metrics can be used to measure the 'success' of the model, depending on the problem being addressed.

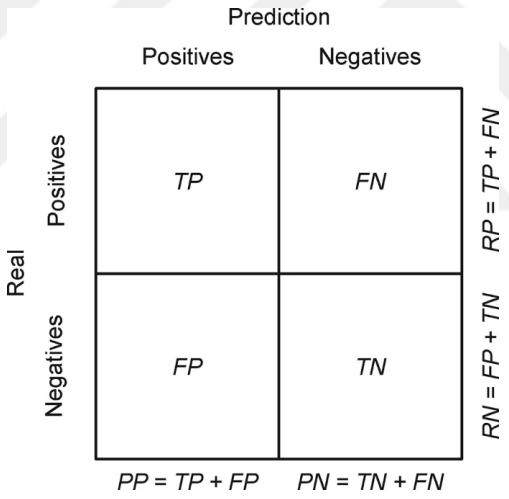

Figure 3.8 The confusion matrix in testing a predictor. All the testing samples are divided into four categories, according to the real labels and the prediction results [188].

There are four main counts that are used to form metrics, true positive (TP), true negative (TN), false positive (FP) and false negative (FN), as shown in Figure 3.8. True positive occurs when both the actual and predicted labels of a sample are positive, while if both are negative, it is called a true negative [188]. These numbers signify the number of truly predicted samples. When the actual label is positive but predicted as negative, this count known as false negative. Conversely, the inverse of this scenario is called as false positive. Based on these counts, basic performance metrics such as Recall (i.e. True

Positive Rate), Precision (i.e. positive predictive rate) and False Positive Rate (FPR) are calculated as following formulas (3.25), (3.26) and (3.27).

$$Recall = \frac{TP}{TP+FN}, \tag{3.25}$$

$$Precision = \frac{TP}{TP+FP} \tag{3.26}$$

$$FPR = \frac{FP}{FP+TN} \tag{3.27}$$

The recall indicates how many of the actual positive samples were correctly predicted to be positive by the model. Thus, a high hit rate means that the model is effective in predicting actual positive samples. On the contrary, The FPR is similar to the recall, but pertains to negatively labelled samples. On the other hand, the precision metric expresses the proportion of samples that the model predicted for the relevant class that were actually relevant. Although both metrics provide useful information about different aspects of outcomes, they are threshold dependent as they depend on the choice of decision threshold. Therefore, the need to simplify them into a single term has led to trade-off curves becoming the preferred method of evaluation for binary classification models [189]. In fact, curves are graphical technique depict trade-offs between metrics such as precision and recall on the different axes across all threshold points. Idea behind use of curves as performance indicator of the model relies on integral of area underneath the entire plotted curve. When the primary goal is to achieve good discrimination so that cases are efficiently classified into binary classes such as infected and healthy, AUROC and AUPRC are the preferred measures [190].

AUROC is one of the most common curves that plots the recall value against the FPR value at different thresholds. If the AUROC value is 0.5, it means that the classifier is not able to discriminate the class label and is predicting a random class or a constant class for all the samples. On the contrary, equality of AUROC value to 1 indicates that classifier predict all samples perfectly. AUPRC, on the other hand, is an alternative curve for assessing model performance that shows the trade-off between precision and recall metrics. Instead of AUROC, AUPRC focuses on the predictive ability of the diseased, i.e. positively labelled, samples by ignoring the correctly classified healthy (TN) samples [191]. Because most bioinformatics-based studies often experiment with genetic data

collected from individuals, and collection is more difficult than in other fields, the datasets tend to have an unbalanced class distribution. Therefore, most bioinformatics studies use curve-based metrics to demonstrate the actual model performance.

Chapter 4

Experiments

4.1 Experiments on Behçet's Disease Prediction

4.1.1 Dataset

The Behçet's disease GWAS dataset consists of 1215 affected and 1278 unaffected (control) samples from the Turkish population. Human CNV370-Duo v1.0 and Human CNV370-Quad v3.0 chips had been used to type DNA samples. Subsequently, SNPs had been refiltered according to strict quality control standards using call rate (>95%), minor allele frequency (>1%) and Hardy-Weinberg equilibrium (>0.00001) criteria, resulting in 311,459 SNPs. For each SNP, the dataset included a genotypic p-value, which indicates the significance of a SNP for disease. These genotypic p-values are calculated by comparing the genotypic frequencies of the SNPs between cases and controls. A chi-squared test had been performed to obtain p-values in the BD GWAS analysis. Detailed information about the dataset and pre-processing during data collection can be found in the original paper [32]. Once the raw data had been obtained, some pre-processing was performed to make it suitable for our analyses.

In the raw version of the data set, each SNP reading can take one of the following four values, i.e. "A_A", "B_B" and "A_B" that represent the type of variant, i.e. homozygous reference, homozygous variant and heterozygous variant, respectively. Furthermore, some SNPs may not be assigned to any zygosity due to bit read errors. These unread SNPs are indicated as "?_?", denoting missing value. In order to prepare the dataset as input for machine learning algorithms, a common strategy is applied to the features in which each of the non-numeric SNP readings is converted to numeric values, such that "A_A" is mapped to 0, "A_B" to 1, "B_B" to 2, and "?_?" to 3. This strategy may capture the additive effect of minor alleles when the codes are used as numerical features [192].

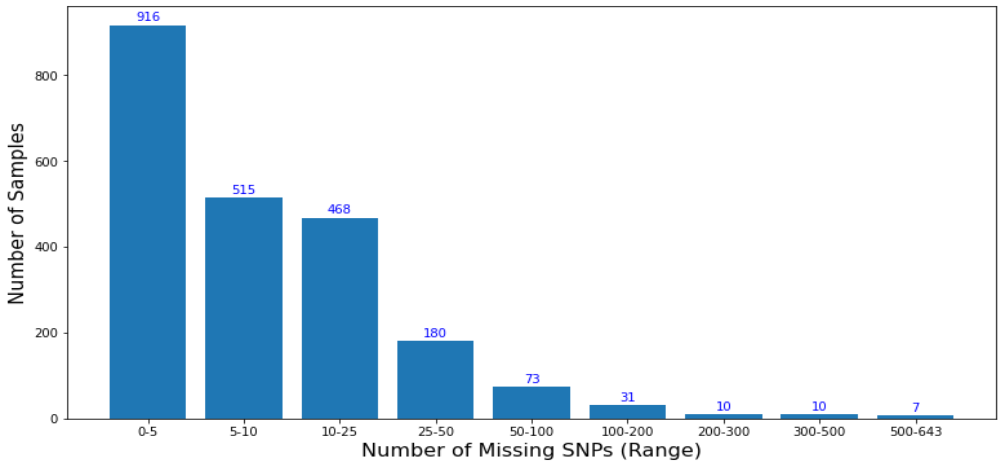

Figure 4.1 Number of missing SNPs with the number of samples, after P-value criteria applied.

As a result of conversation process, count of "A_A" values is 14,082,566 (30.569%), count of "B_B" values is 15,924,781 (34.568%), count of "A_B" values is 16,016,797 (34.768%), and the count of "?_?" values is 44,003 (0.0955%). Note that the proportion of missing values is quite small compared to the others. In addition, Figure 4.1 shows a bar chart of the number of samples with the number of missing SNPs in a given range, for samples after applying the P value <0.05 criterion. Although there are a few samples with more than 200 missing SNPs, the majority of samples have less than 25 missing SNPs. Since the rarity and randomness of the missing values is not expected to bias the prediction models, it is not considered a falsity to convert the "?_?" values to 3.

4.1.2 Experimental Design

The Behçet's disease experiments mainly consist of 2 main phases, where the difference between the phases is number of used SNPs (features). When running the feature selection and machine learning models, all features, i.e. 311459 SNPs, were used in the first phase, whereas in the second phase, SNPs were filtered according to the genotyping p-value. Genotypic p-values (GWAS p-value) represent the significance of the odds ratio for the putative disease-associated variant (a measure of whether it could occur by chance) [193]. Although in GWA studies the traditional strict p-value threshold is $5 * 10^{-8}$, it has been reported in the literature that a p-value of less than 0.05 indicates a mild association between a SNP and disease [194]. Therefore, we set the genotypic p-value threshold at 0.05. As a result of filtering depending on P-value, the dataset of second phase contained 2493 samples and 18479 features to be used as input for our models. The distribution of genotypic p-values of SNPs is shown in Table 4.1.

Table 4.1 Counts of the SNPs in the Behçet's Disease dataset according to P-value ranges.

Lower Bound of P-value	Upper Bound of P-value	Number of SNPs
10^{-46}	10^{-43}	2
10^{-43}	10^{-40}	2
10^{-40}	10^{-25}	2
10^{-25}	10^{-20}	5
10^{-25}	10^{-15}	3
10^{-15}	10^{-10}	34
10^{-10}	10^{-5}	170
10^{-5}	10^{-4}	142
10^{-4}	10^{-3}	506
10^{-3}	10^{-2}	3526
10^{-2}	10^{-1}	14087

The rest of the flow of both phases is practically identical, as can be seen in Figure 4.2, with the exception of some additional feature selection methods and parameter optimization in the second phase. Nonetheless, our experiments mainly focused on the dataset generated after filtering P-value, since the accuracy obtained with the all-feature dataset was quite low. Therefore, the rest of this section explains the experimental flow of the second phase.

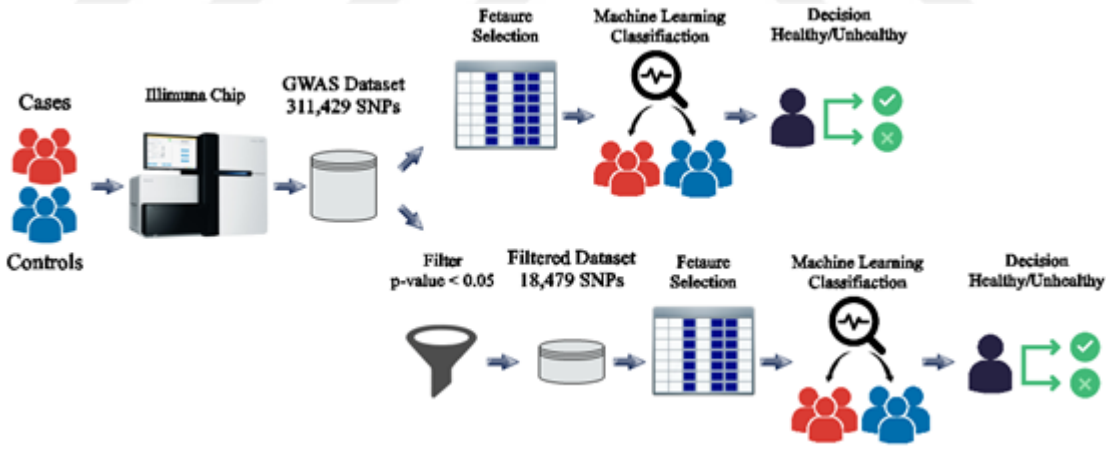

Figure 4.2 General flow and steps for Behçet's Disease prediction experiment.

In the next step, the data set was divided into 10 subsets using the k-Fold Cross Validation Python package from Scikit-Learn [195]. One subset was stored as the test set for the final models and the rest was used as the training set. Feature selection methods were then applied separately to kept only significant features of each fold subset.

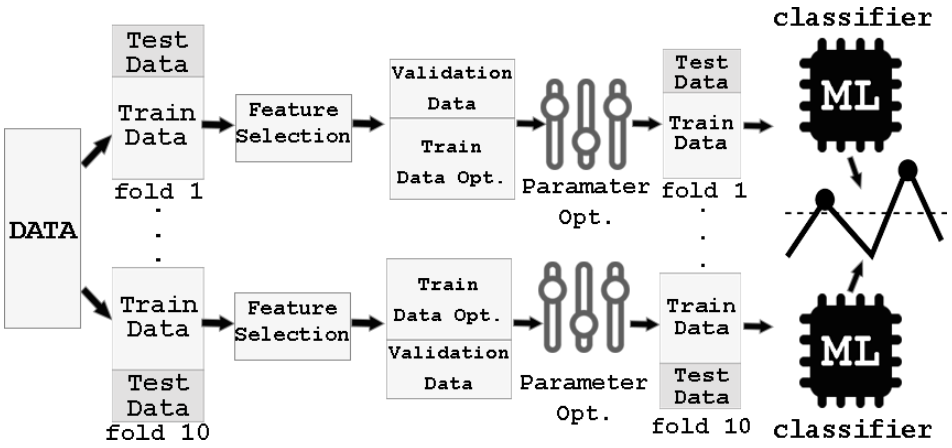

Figure 4.3 10-Fold cross validation settings with feature selection step used during the Behçet's Disease Prediction experiment.

In order to compare the effectiveness of different feature selection approaches in predicting Behçet's disease, we employed several methods: Fisher Score, ReliefF, F-Score, Trace Ratio, T-Score, Gini Index, Information Gain, Gain Ratio, Robust Feature Selection (RFS), Chi Square, LR Lasso and Extra Decision Tree. Since the CFS also requires a search algorithm, best-first, genetic and greedy search approaches were combined with CFS. WEKA software [196] was used to implement CFS, gain ratio and information gain as attribute evaluators and best-first, genetic and greedy as search methods.

Table 4.2 Number of SNPs selected by feature selection methods for each fold of cross-validation.

Fold	CFS Genetic	CFS BestFirst	CFS Greedy	DKSS	Extra DT	LR Lasso	Wrapper Greedy
1	3691	188	190	7942	7754	1975	27
2	4109	215	217	8062	7704	1976	27
3	3680	199	200	8121	7888	2008	26
4	5118	214	217	7936	7769	1965	36
5	3689	207	208	8027	7726	1997	31
6	3205	209	213	8028	7797	1945	21
7	4444	203	206	8192	7718	2020	24
8	4445	174	176	7976	7772	1992	32
9	3681	192	195	8233	7714	1998	14
10	3205	189	192	7473	7759	1960	32
AVG	3926.7	199	201.4	7999	7760.1	1983.6	27

The embedding methods (i.e. Lasso and Extra Trees) are implemented with feature selection package from the Scikit-Learn Python library. Remaining feature selection methods were implemented through Python Scikit-Feature library [197]. In addition to these known methods, our proposed "DKSS" approach was also applied to the subset of

each fold in this step. As DKSS methods require a list of active subnetworks as input, we used networks identified for Behçet's disease in study [198]. Using the proposed DKSS method, an average of nearly 8,000 SNPs were selected within the chosen subnetworks in the 10 cross-validation folds. Therefore, the number of selected features in the ranking methods was set to 8,000 to ensure a fair comparison. However, it is worth noting that the embedded and wrapper approaches resulted in a different number of features due to their inherent selection processes. The number of features for each fold following the selection process performed by wrapper and embedded can be found in Table 4.2.

Once the significant features were identified, the training and test sets were rearranged, and then optimization phase was launched. As detailed in Section 3.5, the high predictive performance of machine learning methods may also be related to the setting of the hyperparameters. Therefore, each machine learning model was optimized before the final classification. For this purpose, 40% and 20% of the training subset is randomly chosen to create the validation training and validation test sets, respectively (see Figure 4.3). Subsequently, the hyperparameters were tuned using Bayesian optimization techniques considering the parameter ranges given in Table 4.3. Finally, the last classifier with tuned parameters was trained on the first training set and evaluated on the test set.

Table 4.3 Optimized hyper-parameters of each classifier with lower and upper bounds for Behçet's Disease prediction.

Classifier	Parameter	Lower Bound	Upper Bound
XGB	Learning Rate	0.005	1
	Number of Estimators	10	800
	Gamma	0.01	2
LR	Regularization (C)	0.0001	2^{15}
SVM	Regularization (C)	0.0001	2^{15}
kNN	k – Number of Neighbors	10	800
RF	Number of Estimators	10	800

In this experiment we preferred 5 well-known machine learning algorithms LR, SVM, kNN, RF and XGB. Additionally, voting of these algorithms was also used as an ensemble method. The following performance metrics are used to evaluate the predictor performance: overall accuracy (Acc), area under the ROC curve (AUC) and area under the precision and recall curve (AUPRC). For these metrics, the averages over the 10 folds of cross-validation were calculated.

4.1.3 Results

Table 4.4 shows the prediction performance of 5 machine learning and 5 feature selection methods applied to the full dataset of 311459 SNPs. The column “Number of features” indicates how many features were selected after applying the respective feature selection method. For the ranking approaches, this number was set to 18479. This is because make an appropriate comparison with the values resulting from the “P-value filtering” results in subsequent tables.

Table 4.4 Results of machine learning methods when feature selection methods were applied to all features (i.e. 311459 SNPs). Dash ("-") represents models in which no feature selection was performed.

Classifier	Feature Selection	Number of Features	ACC	AUC	AUPRC
LR	-	311459	0.6210	0.6730	0.6551
	DKSS	8076.4	**0.9699**	**0.9957**	**0.9960**
	Fisher Score	18479	0.6218	0.6646	0.6353
	ReliefF	18479	0.6129	0.6499	0.6316
	LR Lasso	4685.1	0.6446	0.7029	0.6837
	Extra DT	32188.9	0.6266	0.6869	0.6680
SVM	-	311459	0.6274	0.6763	0.6583
	DKSS	8076.4	**0.9699**	**0.9958**	**0.9960**
	Fisher Score	18479	0.6210	0.6704	0.6426
	ReliefF	18479	0.6286	0.6742	0.6630
	LR Lasso	4685.1	0.6414	0.7029	0.6840
	Extra DT	32188.9	0.6507	0.7099	0.6908
KNN	-	311459	0.5271	0.5327	0.5218
	DKSS	8076.4	0.7068	0.7897	0.7905
	Fisher Score	18479	0.5680	0.6002	0.5875
	ReliefF	18479	0.5351	0.5624	0.5514
	LR Lasso	4685.1	0.5945	0.6249	0.6127
	Extra DT	32188.9	0.5536	0.5754	0.5631
RF	-	311459	0.5680	0.6058	0.5896
	DKSS	8076.4	0.6948	0.7497	0.7660
	Fisher Score	18479	0.6342	0.6934	0.6876
	ReliefF	18479	0.6141	0.6570	0.6385
	LR Lasso	4685.1	0.6599	0.7050	0.6985
	Extra DT	32188.9	0.6302	0.6748	0.6616
XGB	-	311459	0.6374	0.6840	0.6896
	DKSS	8076.4	0.6948	0.7575	0.7680
	Fisher Score	18479	0.6579	0.6981	0.7014
	ReliefF	18479	0.6298	0.6761	0.6692
	LR Lasso	4685.1	0.6474	0.7039	0.7030
	Extra DT	32188.9	0.6334	0.6942	0.6933

In the scenario where all SNPs were used, even the best model (XGB) could only correctly predict 63% of the samples. When existing feature selection was applied, a slight improvement was achieved, and the accuracy increased to 65%. Nevertheless, it can be concluded that using all SNPs or not performing any initial filtering (such as P-value filtering) hinders the improvement of model performance, with the exception of specialized techniques like DKSS. The DKSS approach achieved an accuracy of more than 96% and an AUPRC of 0.996 using nearly 8076 SNPs, meaning that almost all positive samples were correctly classified. Consequently, it is observed that the proposed DKSS method is better at improving the prediction performance for Behçet's disease, especially in contrast to conventional feature selection techniques. Due to the large number of features, only fast selection methods were preferred for the full dataset. Other traditional feature selection methods were not tried further due to the poor initial results and the experiments were continued in the second phase.

Table 4.5 Results of LR and SVM classifiers with different feature selection methods on dataset generated after P-value criteria was applied.

Feature Selection	Logistic Regression					Support Vector Machines				
	ACC	AUC	AUPRC	Train Time	Test Time	ACC	AUC	AUPRC	Train Time	Test Time
CFS BestFirst	0.6458	0.7083	0.6847	64.306	0.090	0.6462	0.7148	0.6962	83.238	0.949
CFS Genetic	0.9687	0.9919	0.9894	186.925	1.305	0.9683	0.9959	0.9955	187.877	3.786
CFS Greedy	0.6510	0.7075	0.6819	73.354	0.091	0.6558	0.7180	0.7090	84.269	1.114
Chi Square	0.8841	0.9583	0.9578	236.401	2.406	0.8877	0.9581	0.9573	173.679	4.820
DKSS	0.9691	0.9930	0.9910	279.953	2.689	0.9683	0.9954	0.9950	316.216	7.614
Extra DT	0.9743	0.9946	0.9939	274.086	2.540	0.9739	0.9971	0.9969	305.412	9.777
F Score	0.7666	0.8230	0.8074	145.748	1.092	0.7674	0.8481	0.8412	328.877	11.771
Fisher Score	0.7641	0.8319	0.8167	149.273	1.034	0.7662	0.8433	0.8357	293.585	11.333
P-value Filter	**0.9956**	**0.9999**	**0.9998**	554.026	5.770	**0.9956**	**0.9999**	**0.9999**	556.862	21.810
Gain Ratio	0.9767	0.9951	0.9946	361.083	2.752	0.9795	0.9975	0.9973	357.367	12.082
Gini Index	0.8472	0.9175	0.9106	327.397	2.802	0.8524	0.9221	0.9206	264.927	8.238
Info. Gain	0.9847	0.9963	0.9962	396.885	4.123	0.9787	0.9973	0.9970	293.447	9.244
LR Lasso	0.7654	0.8167	0.7990	103.278	0.445	0.7625	0.8452	0.8382	209.474	7.839
ReliefF	0.9727	0.9943	0.9938	236.141	1.739	0.9735	0.9963	0.9959	316.088	10.145
RFS	0.9382	0.9743	0.9725	137.530	0.865	0.9370	0.9906	0.9904	295.368	10.834
T Score	0.7690	0.8262	0.8126	195.870	1.234	0.7706	0.8462	0.8380	289.787	10.110
Trace Ratio	0.7634	0.8302	0.8090	154.848	1.053	0.7650	0.8460	0.8374	331.891	11.504
Wrapper Greedy	0.6458	0.6927	0.6741	66.283	0.056	0.6478	0.6951	0.6757	64.931	0.314

Table 4.5 to Table 4.7 shows the performance metrics of the classifiers for the data set consisting of filtered SNPs according to the P-value threshold.

Table 4.6 Results of kNN and RF classifiers with different feature selection methods on dataset generated after P-value criteria was applied.

Feature Selection	k-Nearest Neighbors					Random Forest				
	ACC	AUC	AUPRC	Train Time	Test Time	ACC	AUC	AUPRC	Train Time	Test Time
CFS BestFirst	0.6595	0.7156	0.6973	0.257	72.971	0.6619	0.7218	0.6987	242.851	2.054
CFS Genetic	0.8568	0.9443	0.9352	5.003	546.888	0.7421	0.8299	0.8186	997.010	12.234
CFS Greedy	0.6522	0.7122	0.6916	0.259	80.582	0.6687	0.7259	0.7018	258.676	2.144
Chi Square	0.7633	0.8678	0.8489	10.976	1147.815	0.6960	0.7693	0.7576	1443.386	16.277
DKSS	0.8688	0.9677	0.9656	9.567	988.006	0.7369	0.8220	0.8196	1407.542	19.635
Extra DT	0.8744	0.9517	0.9469	9.487	991.586	0.7136	0.8056	0.7946	1432.645	16.974
F Score	0.6626	0.7652	0.7453	10.767	1114.953	0.6755	0.7372	0.7290	1491.833	15.910
Fisher Score	0.6803	0.7845	0.7599	10.037	1015.376	0.6695	0.7337	0.7135	1308.551	15.021
P-value Filter	**0.9326**	**0.9914**	**0.9906**	21.522	2189.197	**0.7485**	**0.8449**	**0.8390**	2266.199	30.690
Gain Ratio	0.8291	0.9291	0.9065	11.394	1161.505	0.7332	0.8255	0.8160	1556.403	21.644
Gini Index	0.7200	0.8332	0.8078	10.712	1111.711	0.6871	0.7633	0.7452	1546.853	21.355
Info. Gain	0.9322	0.9342	0.9188	20.061	2013.611	0.7561	0.8235	0.8139	2193.064	32.227
LR Lasso	0.7469	0.8222	0.8045	2.295	268.013	0.6855	0.7451	0.7306	622.732	6.572
ReliefF	0.8849	0.9517	0.9460	11.182	1105.847	0.7112	0.8012	0.7943	1574.588	15.852
RFS	0.8881	0.9702	0.9682	11.125	1082.181	0.7316	0.8084	0.8094	1609.311	23.488
T Score	0.6626	0.7705	0.7470	10.274	1033.935	0.6787	0.7336	0.7161	1460.893	16.150
Trace Ratio	0.6815	0.7890	0.7752	11.150	1134.333	0.6723	0.7323	0.7120	1580.726	20.729
Wrapper Greedy	0.6358	0.6713	0.6472	0.092	68.499	0.6266	0.6570	0.6343	205.979	1.070

Table 4.7 Results of XGB classifier and Ensemble Voting method with different feature selection methods on dataset generated after P-value criteria was applied.

Feature Selection	XGBoost					Ensemble Voting (LR + SVM)				
	ACC	AUC	AUPRC	Train Time	Test Time	ACC	AUC	AUPRC	Train Time	Test Time
CFS BestFirst	0.6395	0.7089	0.6926	126.594	0.431	0.6470	0.7134	0.6936	NA	NA
CFS Genetic	0.9544	0.9837	0.9818	185.490	0.463	0.9683	0.9952	0.9937	NA	NA
CFS Greedy	0.6482	0.7052	0.6887	455.752	2.208	0.6550	0.7159	0.7023	NA	NA
Chi Square	0.9231	0.9700	0.9705	876.370	7.832	0.8845	0.9588	0.9581	NA	NA
DKSS	0.9615	0.9881	0.9879	1487.282	13.540	0.9687	0.9947	0.9937	NA	NA
Extra DT	0.9632	0.9886	0.9884	759.222	8.781	0.9759	0.9963	0.9960	NA	NA
F Score	0.7430	0.8167	0.8083	941.374	10.675	0.7629	0.8396	0.8286	NA	NA
Fisher Score	0.7387	0.7986	0.7876	949.713	13.009	0.7653	0.8425	0.8345	NA	NA
P-value Filter	**0.9912**	**0.9992**	**0.9992**	1716.624	13.000	**0.9956**	**0.9999**	**0.9999**	NA	NA
Gain Ratio	0.9664	0.9924	0.9910	700.043	3.513	0.9783	0.9967	0.9963	NA	NA
Gini Index	0.8228	0.8892	0.8808	828.508	11.133	0.8452	0.9224	0.9185	NA	NA
Info. Gain	0.9632	0.9878	0.9868	821.819	10.293	0.9795	0.9974	0.9972	NA	NA
LR Lasso	0.7437	0.8324	0.8336	317.972	2.031	0.7669	0.8316	0.8149	NA	NA
ReliefF	0.9566	0.9909	0.9914	849.870	7.929	0.9727	0.9963	0.9961	NA	NA
RFS	0.9140	0.9711	0.9700	944.679	6.570	0.9434	0.9870	0.9868	NA	NA
T Score	0.7496	0.8112	0.8061	738.165	13.171	0.7694	0.8393	0.8277	NA	NA
Trace Ratio	0.7560	0.8102	0.8024	1097.323	9.731	0.7649	0.8441	0.8365	NA	NA
Wrapper Greedy	0.6466	0.6932	0.6734	215.749	0.144	0.6474	0.6941	0.6752	NA	NA

Unlike to the first phase, a much more comprehensive analysis was performed in the second phase by testing 17 feature selection methods. Moreover, not only ranking and embedded methods but also wrapper feature selection techniques were experimented. Running times of both train and test process were also calculated and included in tables.

A general overview of the results shows that 6 feature selection methods have achieved an accuracy rate exceeding 95% with LR, SVM, XGB, and Ensemble Voting algorithms. In the voting ensemble method, two distinct approaches were used to combine the predictions of the base learners: Hard Voting and Soft Voting. Hard voting determines the class by selecting the majority of classes predicted by the base learners. In contrast, Soft Voting calculates the average of the prediction probabilities for each class as predicted by the base learners. To minimize misclassification in the ensemble, we preferred the soft voting approach and combined only the LR and SVM methods as base learners. This was because the RF and k-NN methods had lower accuracy, which could potentially have decreased the overall accuracy of the voting ensemble. Despite the results being quite close, the voting ensemble approach did not outperform the LR and SVM methods used individually. On the other hand, the XGB ensemble method achieved comparable results for all feature selection methods.

In the evaluation of feature selection methods, decision tree-based techniques, including Extra DT, Information Gain, and Gain Ratio, emerged as the top performers. Subsequently, the ReliefF method, followed by Genetic Search-based CFS and proposed DKSS methods, demonstrated competitive results. On the contrary, methods employing Best First/Greedy Search-based CFS and Wrapper strategies obtained lowest prediction accuracies among the models, which might be due to the selection of very few SNPs.

Among the filtering methods, ReliefF achieved significantly higher prediction scores than the others, such as Fisher Score, F Score, Trace Ratio and T Score. One of the reasons for this result could be related to the fact that the ReliefF algorithm is a multivariate approach. Analyses using the DKSS method have already demonstrate that considering groups of SNPs with known biological interactions improves predictions of Behçet's disease susceptibility compared to considering SNPs independently. Hence, multivariate selection approaches may be able to capture the relationship of these interactions between SNPs.

Another remarkable observation is an astronomical increase in prediction accuracy when the SNPs are filtered by P-value criteria. For example, the logistic regression model was able to achieve a prediction accuracy of 62% for the full data set, while this percentage increases to 99% when only SNPs with a P-value of less than 0.05 are considered. This result provides evidence that the P-value indicates the importance of SNPs. Although the second phase, unlike the first phase, includes the hyperparameter optimization step for machine learning algorithms, the reason for the improvement observed in the second phase is closely related to P-value filtering.

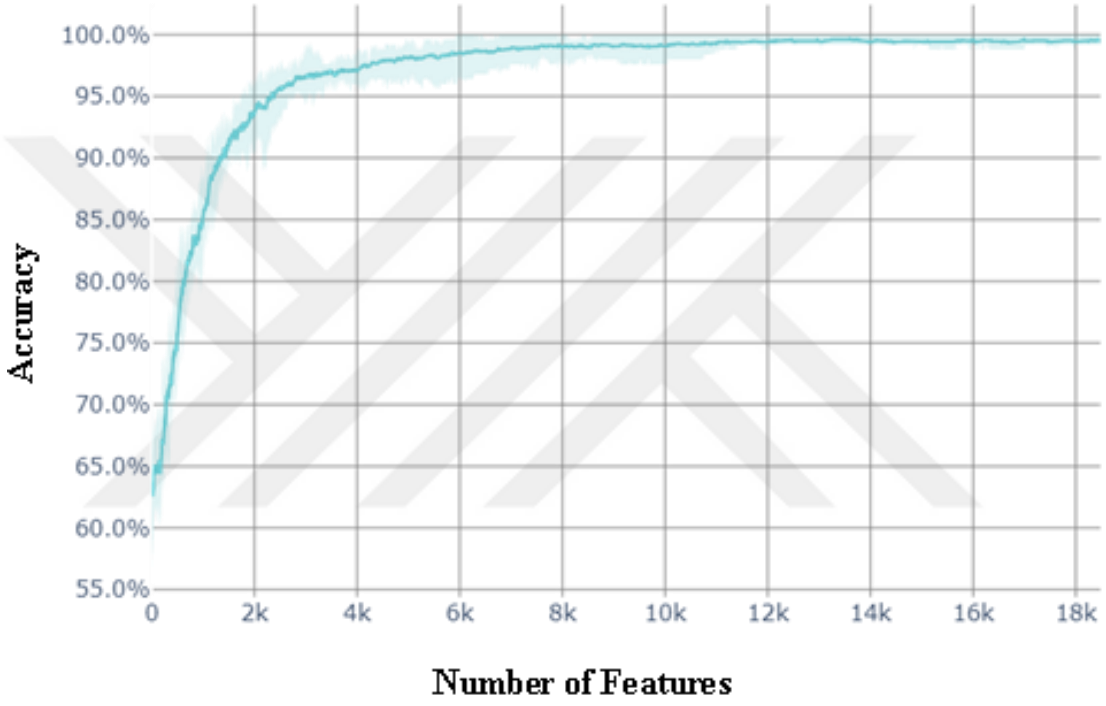

Figure 4.4 Accuracy of logistic regression with respect to number of features according to P-value criteria. Each feature was added one by one, and performance was evaluated using logistic regression algorithm.

The fact that the best predictive accuracy is obtained by the P-value filtered method compared to others may indicate that all 18,479 SNPs contribute to disease prediction. To test this hypothesis, we performed the following experiment. We ranked the features by p-value and performed a forward feature selection strategy (increasing the number of features by 1 at each step) in the first feature selection step. After each addition, the accuracy of these feature subsets was evaluated using logistic regression, as shown in Figure 4.4. Based on this figure, the maximum accuracy of 99.64% is achieved when the number of features is equal to 13,611. This accuracy is also obtained for some of the higher number of features. This shows that it is not necessary to use all 18,479 features

to achieve the best classification accuracy, and that FS methods using CFS as a subset evaluator and embedding methods are conservative in the number of features selected. For the ranker-based methods, the number of features was set at 8000 to allow a fair comparison with the DKSS method, which uses biological subnetwork information. If the number of features selected by the ranker-based methods were allowed to vary, it may be possible to find a feature subset containing a smaller number of features.

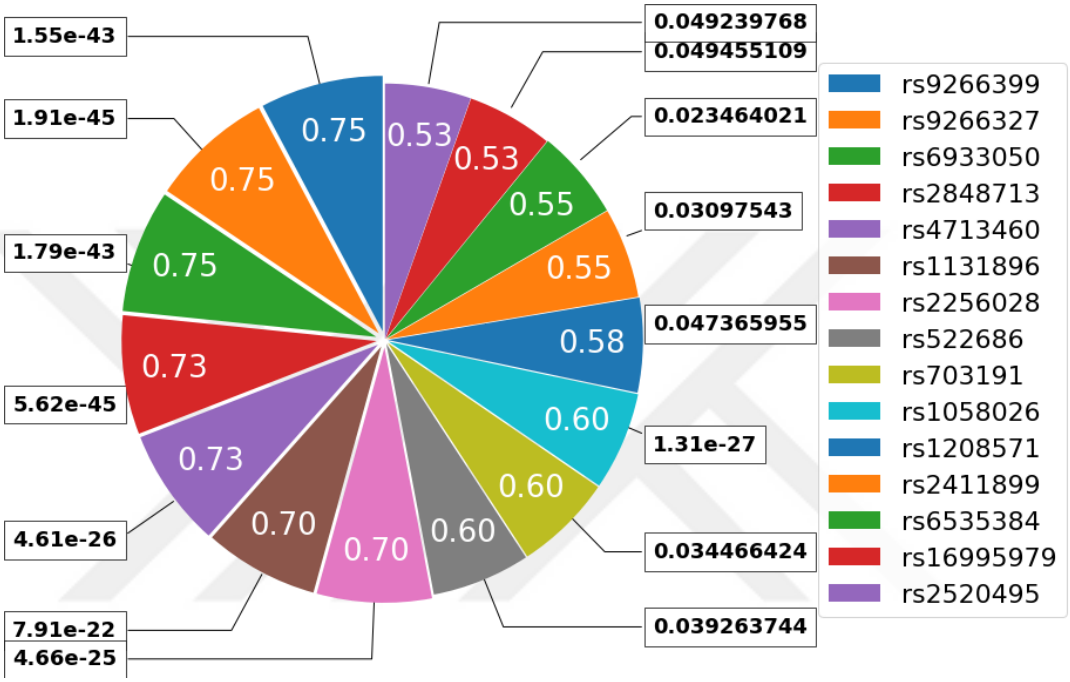

Figure 4.5 Most representative SNPs that are identified by feature selection methods. Their genotypic p-values are shown in boxes. Numbers on the slices of the pie chart represent occurrence rates.

As part of the experiments, selected, i.e. significant, features were further analyzed using a biological investigation instrument. In the Behçet experiment, we first identified the most significant SNPs by the selection frequency of highly performing feature selection methods. Initially, SNP groups were determined by selecting the top 25 SNPs ranked as highly relevant for prediction by each of the four best feature selection methods used in this study. These best feature selection methods are Extra Decision Tree, Gain Ratio, Information Gain and Relief. This strategy was applied for each cross-validation fold and 40 SNP lists were generated, with each list containing 25 SNPs. The next step is to determine the frequency of each SNP's occurrence across the 40 lists. If a SNP appears in a list, then it gets 1 point, otherwise 0. Once frequency counting is done, occurrence rate for each SNP is calculated by dividing its frequency of occurrence by 40. Figure 4.5

shows the occurrence rate and the genotypic p-values of the top 15 SNPs selected by the best four feature selection methods. Values on the slices of the pie chart indicates occurrence rates. According to the figure, SNPs rs9266399, rs9266327 and rs6933050 are top-selected SNPs among best 4 feature selection method, with 0.75 occurrence rate. These are followed by rs2848713 with an occurrence rate of 0.73, rs4713460, rs1131896, rs2256028 with 0.70 and rs522686, rs703191 and rs1058026 SNPs with 0.60.

Table 4.8 Detailed Information about Significant SNPs and Genes according to occurrence rate.

SNP ID	Overlapped Gene	Type	Nearest Upstream Gene	Type of Nearest Upstream Gene	Nearest Downstream Gene	Type of Nearest Downstream Gene	Feature Type Class
rs9266327	None	None	AL671883.2	unprocessed pseudogene	DHFRP2	processed pseudogene	
rs2848713	HCP5	sense_overlapping	None	None	None	None	
rs9266399	AL671883.3	lincRNA	None	None	None	None	
rs6933050	AL671883.3	lincRNA	None	None	None	None	Transcription Factor
rs1058026	HLA-B	protein coding	None	None	None	None	Transcription Factor
rs4713460	None	None	FGFR3P1	processed pseudogene	ZDHHC20P2	processed pseudogene	
rs2256028	MICA	protein_coding	None	None	None	None	Transcription Factor
rs1131896	HCP5	sense_overlapping	None	None	None	None	
rs6535384	SCD5	protein_coding	None	None	None	None	
rs2411899	None	None	B4GALNT2	protein coding	GNGT2	protein coding	Transcription Factor
rs703191	None	None	SGK1	protein coding	CHCHD2P4	processed pseudogene	
rs522686	KIRREL3	protein_coding	None	None	None	None	Transcription Factor
rs1208571	None	None	ZNF33CP	processed pseudogene	ZNF25	protein_coding	Transcription Factor
rs2520495	None	None	RNU7-197P	snRNA	AC108156.1	lincRNA	
rs16995979	LAMP5-AS1	antisense	None	None	None	None	Transcription Factor

A further analysis was performed on these most represented SNPs to reveal the information behind the SNP ids using the SPOT tool [199], which is also used to map the SNPs to genes and then find associated active subnetworks of our dataset. As a result, we found that 7 of the top 15 SNPs are associated with 6 different genes. These genes are HLA-B (rs1058026), HCP5 (rs1131896, rs2848713), KIRREL3 (rs522686), LAMP5-

AS1 (rs16995979), MICA (rs2256028) and SCD5 (rs6535384). Note that, the HLA-B gene has already been mentioned as having a strong association with Behçet's disease in the literature [32,200]. Associated Pathways and GO terms with these genes were also checked. There is no any meaningful association biological pathways, but there are some findings on antigen processing and presentation, defense response, regulation of immune response GO Biological Process terms; integral component of membrane, cell surface GO Cellular Component terms and antigen binding GO Molecular Function terms. Detailed Information about Significant SNPs and Genes according to occurrence rate presented in the Table 4.8.

4.2 Experiments on Respiratory Infection Prediction

4.2.1 Dataset

To evaluate the performance of machine learning models in predicting respiratory infections, we used a comprehensive respiratory dataset publicly available on the Gene Expression Omnibus (GEO) with accession number GSE73072. This dataset actually consists of seven distinct datasets, each derived from different challenges conducted by Duke University under a contract awarded by the DARPA Predicting Health and Disease program. These seven challenge experiments are denoted to as RSV DEE1, H3N2 DEE2, H1N1 DEE3, H1N1 DEE4, H3N2 DEE5, HRV UVA and HRV DUKE. Each dataset contains a varying number of samples from one of four different respiratory viruses: H1N1, H3N2, HRV, or RSV. Throughout each challenge study, peripheral blood was collected from healthy volunteers a day prior to inoculation (i.e., T.-24 or T.-30 hours), immediately before inoculation (T.0), and at predetermined intervals following the challenge. Each volunteer was exposed to only one of the four live respiratory viruses by inoculating at time T.0 in a controlled environment. Data sampling of the microarray began 1 day (24 hours or 30 hours) before inoculation and continued at various intervals up to 7 days later. To extract microarray data, Human Genome U133A 2.0 array was used. Further details about dataset can be found in [201]. Infection of a volunteer was detected by analyzing nasal lavage particles in a clinical setting. If the particles had a viral indicator, the infection status of the individual was labeled 1, otherwise 0. Subjects were also asked to periodically rate the severity of 8 different symptoms, including runny nose, headache, malaise, myalgia, sneezing, sore throat, and nasal congestion, from 0-4.

Afterward, rates were used to calculate the Jackson score [202], which is known to be the best method for measuring symptom severity. The individuals who scored above 6 were labeled 1 to indicate symptomatic, otherwise labeled 1. Hence, the dataset contains 2 different label values, infection and post-exposure symptom development.

One of the objectives of this thesis is to integrate existing knowledge into the prediction model to improve the predictive performance. In the respiratory infection prediction experiment, different feature representation approaches were comparatively analyzed. Feature representation is the transformation of raw data into another format or feature space that may contain more or fewer features. This process can improve the performance of machine learning models as the performance is highly dependent on how the data is represented. This is because a proper feature representation can lead to the hidden patterns of the raw data being captured. Hence, we have used 3 feature types to represent samples, gene-level, probe-level and a pathway-centric approach named ssGSEA-based representation. In order to generate these types of features it should be performed some pre-processing on the raw microarray data.

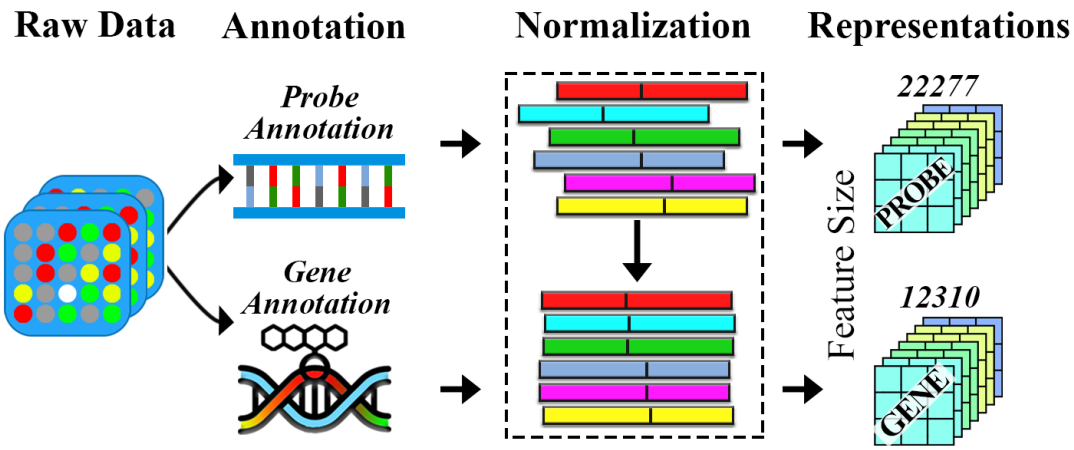

Figure 4.6 Generation of probe- and gene-based expression values using CDF files and the normalization steps. Due to variations in the number of CDF file mappings, each representation type has a different number of expression values.

The GSE73072 repository contains the raw "CEL" data files for the samples, which are generated by array scanner software and include measured probe intensities. These intensities should be mapped to probes or genes using proper annotation i.e. Chip Description File (CDF). The CDF provides details about the identity and location of each probe on the chip. To derive probe- or gene-level expression values from this data, we developed a script in R, employing the affy package from the Bioconductor library [203].

The latest CDF files published by Bioconducter named “hgu133a2cdf” and “hgu133a2hsentrezgcdf” were used to read data from raw files for probe and gene-level representation, respectively. After executing the script, we acquired expression values for each sample at both the probe and gene levels and then the normalization process was launched.

Microarray experiments, which typically involve numerous arrays, inherently contain some non-biological variations. These variations stem from differences in various stages of the experimental process, such as sample preparation (for instance labeling differences), the manufacturing of the arrays, and the processing steps (such as variations in scanner equipment) [204]. Normalization process is one of the way for reducing this variation. Although there are many proposed normalization techniques in the literature, the most popular method is Robust Multi Array (RMA). RMA is a multi-step process that sequentially performs background correction, quantile normalization, and summarization to estimate the true gene expression values while reducing noise and technical variability [205]. During dataset creation processing, we have applied RMA normalization to expression values just after extraction them from microarray data. Subsequently, batch effect removal step was performed. Similar to variations corrected by RMA normalization, batch effect refers to technical variation stemming from the generation of data in multiple batches [206]. While normalization primarily focuses on correcting the biases within each sequencing experiment, batch effect removal helps to reduce the bias generated across batches. These batch effects may arise from the dates of sequencing, the people who performed the sequencing, the protocol, or the type of sequencing machine [207]. Since our dataset is aggregation of 7 challenge datasets which are collected at different times, there might be a batch effect that results in variations in expression levels. Therefore, we have also applied ComBat [208] to expression values using the pyComBat library, which is a framework to adjust batch effects [209]. Following the normalization process, probe-level and gene-level representations of expression values for the samples were obtained. As previously stated, the GSE73072 was derived from an aggregation of 7 different challenge datasets, each comprising samples exposed with four respiratory viruses. Therefore, obtained expression values from raw data were divided into 7 parts according to their challenges (e.g., DEE1, DEE2, DEE3, etc.), after the normalization process. Consequently, two different type of dataset were generated with the probe-level representation consisting of 22,277 features and the

gene-level representation consisting of 12,310 features for each of 7 sub-dataset. Note that, the feature dimension of gene-level representation is fewer than probe-level. This is because a series of probe pairs known as a probe set can only represent to a single gene in some cases [210]. Flow of the pre-processing to obtain different represented datasets are given as Figure 4.6.

Another approach to represent samples that we have proposed in our respiratory infection prediction experiments is the ssGSEA-based representation, where features consist of enrichment scores. The ssGSEA-based representation, known in the literature as pathway-centric, essentially uses enrichment scores as features that express how representative the predefined gene sets or pathways are of the expression values of the samples, which is explained in detail in Section 3.4.2. Due to the fact that each pre-defined gene set is used to calculate one enrichment score, a large number of gene sets are necessary to represent samples in high discriminative dimensions. Hence, we have used Molecular Signatures Database (MSigDB) collections in our experiments. The MSigDB is a comprehensive and widely-used repository for the analysis of gene expression data [211]. It explicitly designed to provide gene sets for enrichment analyses and covers various gene set sources. More than 36800 different human gene sets from 9 major and 30 sub-collections are available in the database. Each collection is derived for a specific purpose in gene expression analysis to investigate gene function and associations, biological pathways, and disease mechanisms.

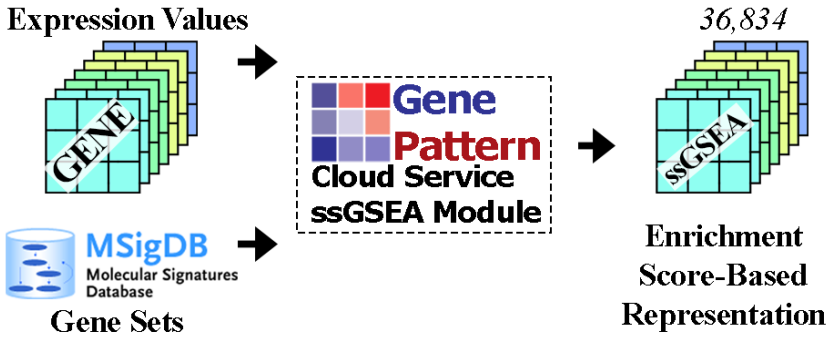

Figure 4.7 Illustration of ssGSEA based feature representation generation using MSigDB repository Gene Pattern cloud service.

Using the MSigDB and gene expression values in our 7 datasets, enrichment scores were calculated using the web-based tool called GenePattern service [212]. GenePattern is a cloud-based computational biology platform that provides free access to complex bioinformatics tools and resources such as ssGSEA, GSEA, TCGA modules, mainly in

the field of genomics. ssGSEA module requires pre-defined gene sets, expression values of samples with gene symbols, and algorithmic parameters including weighting exponent, minimum gene size, sample normalization etc. As we wanted to use as many gene sets as possible as features, the minimum gene size was set to 2, which excluded gene sets with less than this number of overlaps. Since the input expression values must contain gene symbols rather than probe IDs, gene symbol annotated expression values described in the dataset section were used as input to the module. Because matching is carried out between gene sets and inputs with gene symbols, the module cannot calculate scores when probe IDs are uploaded. Utilizing gene-annotated expression data files and all human-related gene sets sourced from MSigDB, the ssGSEA module was executed. As a result, a total of 36,834 unique enrichment scores were calculated for each sample based on the corresponding gene sets. Then, these scores are spliced together as a vector to represent samples. In addition to these 3 types of feature representation, we also generated extended versions of these representations by splicing them end-to-end. For example, using 12,310 features from the gene level and 36,834 features from the ssGSEA, a sample was represented with 49,144 features. This is because some expression values and enrichment scores can be more discriminative when used together. Eventually, 5 different feature types, gene level, gene + ssGSEA combination, probe level, probe + ssGSEA combination, and ssGSEA were generated.

4.2.2 Experimental Design

In this section, the implementations of the models and experiments are presented in detail. The in-depth analysis of the respiratory dataset aims to develop predictive models of resilience or susceptibility to symptoms and infection using various machine learning models. In addition, factors, i.e. expressed genes, responsible for mediating the response to respiratory virus exposure were investigated using significant genes selected by different feature selection methods. Furthermore, the use of enrichment scores as a feature to predict infected and symptomatic individuals exposed to a respiratory virus is another focus of these analyses. Every sample in the dataset has two distinct class labels: Viral Shedding and Symptomatic Response to Exposure. The Viral Shedding label denotes the infection status of the individual, whereas the Symptomatic Response reveals if the individual exhibited severe symptoms following exposure. Therefore, we have 2 different tasks during experiments prediction of being infected and prediction of symptom presence. Both tasks were predicted using separate models. Furthermore, we also

examined the ability to predict disease in the post-exposure period by focusing on the time-point dimension of data sampling.

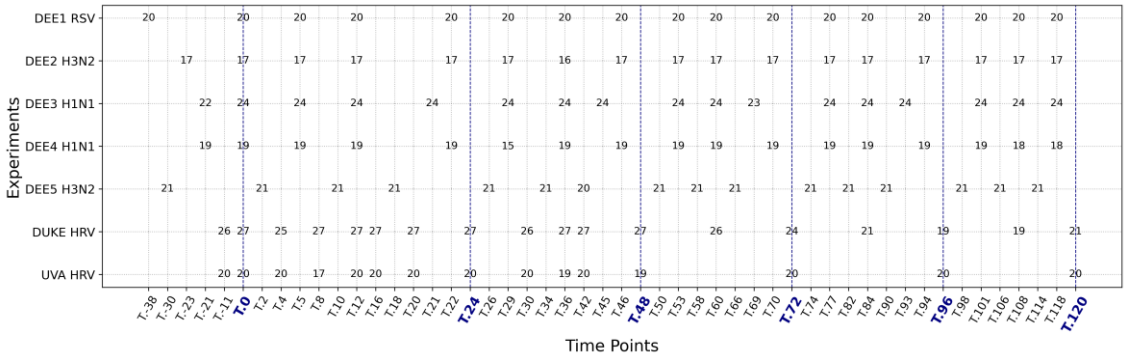

Figure 4.8 Number of samples collected from the subject for each sub-experiment dataset with sampling time points. T.0 indicates the time of inoculation of subjects with related viruses.

The first step in our experiments is the definition of the time points at which the samples Following inoculation of the subject with a virus, probe intensities were sampled periodically, aiming to observe differentially expressed genes and the development of symptoms resulting from virus exposure over time. This enabled to evaluation and examination expressed genes before, just after, a time period after the exposure. However, due to variations in experiments, the sampling time points and intervals varied across experiments.

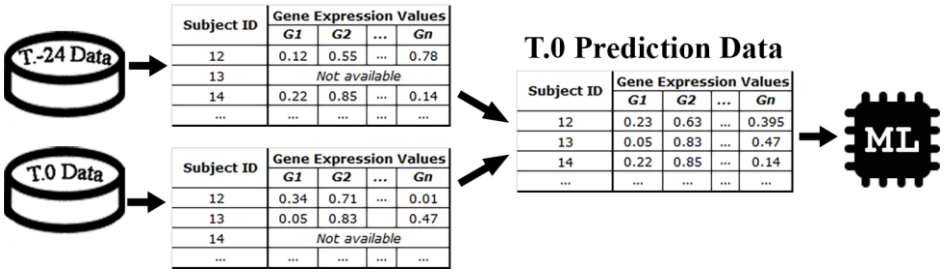

Figure 4.9 Average gene expression value calculation of each samples up to predefined time points **[213]**.

For instance, a sampling was performed 24 hours after the inoculation in HRV experiments, while sampled 21 hours after in H1N1 experiments. In addition, 22 subjects were sampled at timepoint T.-21 in DEE3, while this number is 24 at timepoint T.0. This means that some subjects were not sampled at all time points. Therefore, inconsistency in sample size appeared that caused difficulty perform analysis on datasets collectively. Detailed information on the number of samples and experiment-wise collection time points is given in Figure 4.8. To overcome this problem, we defined 6 different time points

and calculated the average of the expression values for each subject up to these predefined time points. These time points are time point 0 (T.0), time point 24 (T.24), time point 48 (T.48), time point 72 (T.72), time point 96 (T.96), time point 120 (T.120). For example, if a subject had gene expression samples at the time points T-24, T0, T8, and T16, only average of T-24 and T0 was considered for the T0 prediction, as seen Figure 4.9. Although ignoring each time points can be seen as a disadvantage, it can also be seen as an advantage, because the timing of symptoms varies from person to person and even from virus to virus. For example, in the HRV DUKE experiment, there are eight time points up to the T.24 prediction. While some subjects may become symptomatic between time points T.4 and T.12, others may become symptomatic after T.12. As machine learning models cannot be trained/tested individually for each subject depending on the time point, symptom signals from all subjects should be captured in a generalized model. This is because we assume that the changes in gene expression also begin with the onset of symptoms, making it easier to capture the changed signals (i.e. gene expression) by machine learning. In this way, even though subjects' gene expression signals may be weak or strong at different times, this approach can capture distinctive signals for all subjects, which also facilitates the identification of key gene expressions that affect disease prediction [213]. After determining the time points to be predicted and expression values were averaged, the training and test subjects for the sub-datasets DEE1, DEE2, DEE3 and HRV UVA were split using the Scikit-Learn library [195], ensuring a balanced distribution based on class labels. For the other sub-datasets, namely DEE4, DEE5 and HRV DUKE, the same training and test subjects of the Viral DREAM Challenge [214] were used to ensure a fair comparison.

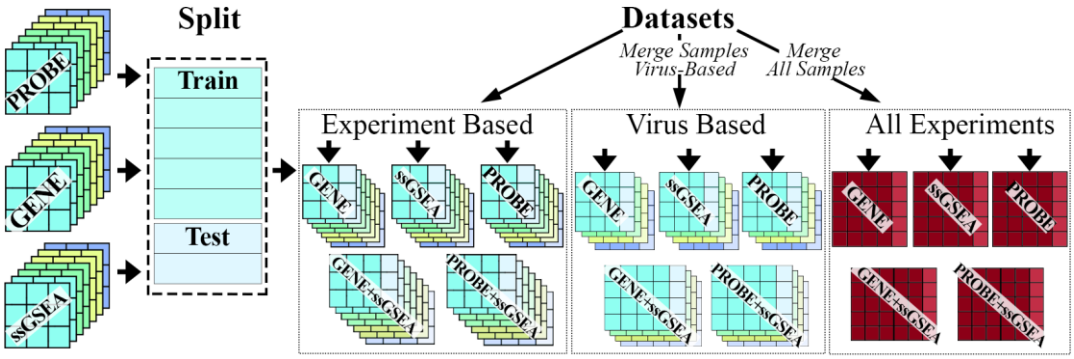

Figure 4.10 Three experimental groups derived from the GSE73072 dataset by merging samples related to the same virus (Virus-Based) and all samples (ALL). For each group, training and test samples were kept equal to ensure a fair comparison.

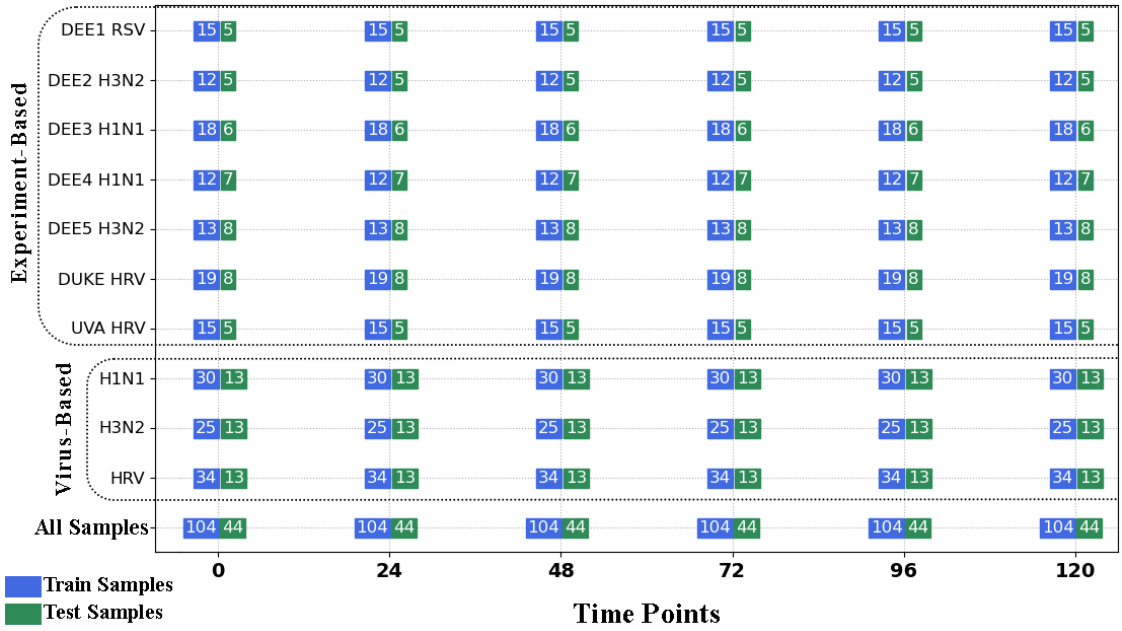

Figure 4.11 Number of training and test samples according to gene expression values averaged up to predefined time points. Virus-based datasets were generated by merging samples belonging to the same virus family.

Addition to the different representation types, we have generated 3 dataset group according to their sample combination. The first one is "Experiment-Based" in which each of the 7 sub-data sets is analyzed separately. Secondary group is based on the merging of sub-sets of data in which similar viruses are being studied. For example, the two sub-sets DEE3 and DEE4 contain samples injected with the H1N1 virus. On the other hand, DEE2 and DEE5 contain samples injected with the H3N2 virus. As can be seen in Figure 4.8, the number of data collected in each challenge is quite limited. The main reason for this is that the data is genetic, and it is necessary to inject the virus directly into individuals to understand the infection. It is therefore difficult to find enough volunteers for such a large and risky study. On the other hand, sample size plays a crucial role in model learning and machine learning results [215]. Large data sets allow algorithms to be trained on broader and more diverse samples, making the model more generalizable to real-world scenarios. Insufficient or biased data can cause the model to produce misleading results. Therefore, increasing the amount of data in machine learning should be balanced with the quality and representativeness of the data. Therefore, samples exposed to identical viruses were combined, thereby increasing the total sample size. Consequently, sub-datasets from experiments DEE2 - DEE5, DEE3 - DEE4, HRV UVA - HRV DUKE were merged. This resulted in the creation of H1N1, H3N2, and HRV sub-datasets, with each one has more samples than experiment-based. The last group contains

all samples merged into a single dataset called “All”. This dataset allows the performance of a general model to be assessed across a wide range of viral infections caused by different viruses.

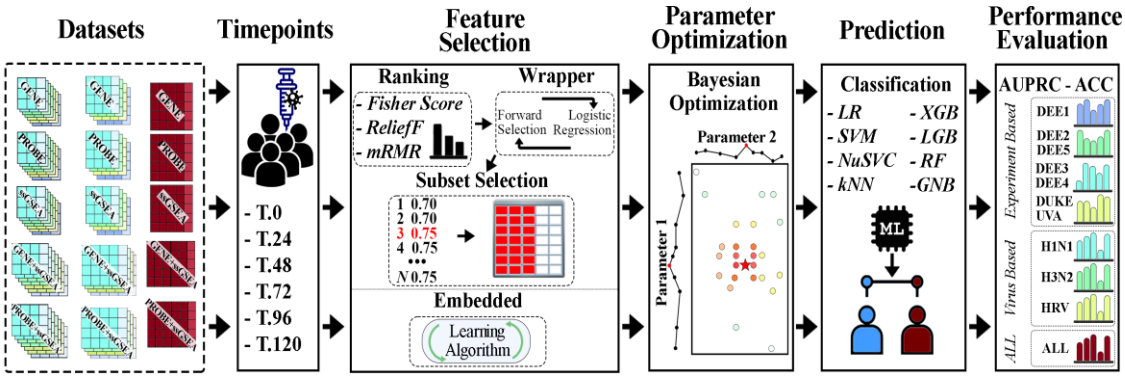

Figure 4.12 Experimental flow of the respiratory infection and symptom development prediction problems.

In addition, further analysis can reveal common genes for all the respiratory viruses we have studied. Number of train and test sample for each group and sub-dataset are shown in Figure 4.11. As seen in the figure, we have 11 separate experiments, each with 6 different time points. Taking into account that each experiment/time point was represented by 5 different feature sets, we prepared 330 unique sub-datasets as input for the machine learning models. After the creation of the data set, the experimental process comprises 4 further steps, which are shown in Figure 4.12. These are the feature selection, parameter optimization, prediction and performance evaluation steps, where all steps are applied to each of the sub-datasets explained above. However, we can shrink these 4 steps to 2 main phases as feature selection and classification. In feature selection phase, we performed two approaches: hybrid and embedded.

The hybrid approach takes the advantages of both ranking and wrapper. It has 2 steps inside, ranking features using a filtering method and selecting the most relevant features using a wrapper method. During the filtering step, the correlation value of each feature was calculated using the only training data samples. Subsequently, training set sorted according to this new feature ordering using correlation values. To determine the optimal filtering method, we compared 1 univariate and 2 multivariate filtering methods during this step: Fisher Score, ReliefF and mRMR. The implementation of these methods was done on Python language with the library [197]. The second step of hybrid approach is determining the number of most significant features from re-ordered dataset. Starting

from the most correlated feature, a subset was then formed by adding the next feature at each iteration. We preferred the logistic regression algorithm as the wrapper method because it is simple, runs fast, and is efficient even with small data sets. After each iteration, number of features in the subset was stored along with the performance score. Since one of the goals of feature selection was to minimize the number of dimensions while maintaining or improving prediction accuracy, the least number of features that achieved the highest predictive performance was marked as the optimal subset of features. For example, if both the top 3 and top 50 features achieve the same maximum accuracy of 75%, the set of top 3 features is selected as optimal. On the other hand, L1 regularized logistic regression model, known as Lasso, and XGBoost backed tree-based feature selection methods were used as embedded approaches. Embedded methods inherently select features during the learning process, we didn't apply any further wrapper like selection strategy to them. Once the best set of features was figured out through FS method from the training set, the test set is rearranged using these features.

Table 4.9 Optimized hyper-parameters of each classifier with lower and upper bounds for Respiratory Infection prediction problem.

Classifier	Parameter	Lower Bound	Upper Bound
XGBoost	Learning Rate	0.0001	1
	Number of Estimators	1	1000
	Maximum Depth of Tree	2	1000
	L1 Regularization Coefficient	0	1
	L2 Regularization Coefficient	0	1
LightBoost	Learning Rate	0.0001	1
	Number of Estimators	1	1000
	Maximum Depth of Tree	2	1000
	L1 Regularization Coefficient	0	1
	L2 Regularization Coefficient	0	1
LR	Regularization (C)	0.0001	1000
SVM	Regularization (C)	0.0001	1000
NuSVC	nu	0.0001	25
kNN	k – Number of Neighbors	2	Number of Class
RF	Number of Estimators	10	1000

In next main step is about to optimize hyper-parameters of the algorithms and make predictions on the test samples. During experiments, 8 different classification algorithms were used. These are logistic regression (LR), support vector machine (SVM), random forest (RF), k nearest neighbors (kNN), Nu-Support Vector Classification (NuSVC), Gaussian Naive Bayes (GNB), XGBoost (XGB), and LightGBM (LGB). In hyper-parameter step, parameters of each these algorithms were tuned between lower and upper

boundaries given in Table 4.9, on the only training samples to avoid overfitting in the results.

Although some datasets have a bit more samples due to merging experiments, the sample size is still not very large. When the sample size in a data set is small, Leave-One-Out Cross Validation (LOOCV) is suggested to achieve reliable prediction performance for a classification algorithm [216]; hence it was used with Bayesian optimization to tune parameters of algorithms. LOOCV is a specific implementation of k-fold cross-validation wherein the “k” value equals the number of samples within the dataset. In each iteration, one sample is marked as validation, and the rest is used to train the model with the specific parameters set determined by Bayesian acquisition function. Trained model then used to predict class probabilities of the validation sample. Once all samples have been predicted, the performance of the candidate parameter set is calculated with a metric according to a prediction metric. Since the results of the models in our study were compared according to their AUPRC scores, we tried to maximize the AUPRC score during optimization. The configuration of the parameter set that achieved the highest score was stored for the final model. The SKOPT library in Python software was used to implement the Bayesian optimization [217]. For each sub-dataset mentioned above, 250 trials were performed using the “gp_minimize” function. Once the optimal parameters were found, the classification algorithms were trained, and test samples were predicted. Finally, metrics were calculated using the prediction probability distribution of each sample, and results tables were formed.

4.2.3 Results and Discussions

In the second experiment of our study, based on the sample combinations mentioned in the previous section, 3 different sub-sections were treated. Each subsection contains corresponding results tables. The "number of features" used during the training / testing process is indicated in the “NF” column. The column “Clf.” expresses the algorithm of the classifier. As there are a large number of tables, the results have been fitted into the pages by shrinking the column names. All results are descending order on AUPRC values.

4.2.3.1 Results for Experiment-Based Groups

Table 4.10 The results of the best-performing models according to feature representation type for each experiment at time points T.0 and T.24 without feature selection on the infection prediction task. An asterisk (*) indicates that the hyper-parameters were not optimized. NF column shows the number of used features after the feature selection methods.

Exp.	TimePoint 0					TimePoint 24				
	NF	Feature	Clf.	AUPRC	ACC	NF	Feature	Clf.	AUPRC	ACC
RSV DEE1	49144	G.+GSEA	XGB	0.867	0.600	36834	GSEA	XGB	0.903	0.800
	36834	GSEA	SVM	0.850	0.600	49144	G.+GSEA	XGB	0.903	0.600
	59111	P.+GSEA	SVM	0.850	0.600	12310	Gene	XGB	0.903	0.600
	22277	Probe	NuSVC	0.850	0.600	22277	Probe	KNN	0.822	0.600
	12310	Gene	XGB	0.822	0.600	59111	P.+GSEA	LGB	0.800	0.600
H3N2 DEE2	22277	Probe	KNN	1.000	1.000	49144	G.+GSEA	RF	1.000	1.000
	49144	G.+GSEA	RF	1.000	0.600	12310	Gene	KNN*	1.000	1.000
	12310	Gene	KNN*	0.933	0.600	36834	GSEA	KNN*	1.000	1.000
	36834	GSEA	KNN	0.933	0.600	59111	P.+GSEA	KNN*	1.000	1.000
	59111	P.+GSEA	KNN	0.933	0.600	22277	Probe	KNN*	1.000	1.000
H3N2 DEE5	36834	GSEA	XGB	0.943	0.750	22277	Probe	RF	0.971	0.625
	49144	G.+GSEA	KNN	0.925	0.375	49144	G.+GSEA	LR	0.938	0.750
	59111	P.+GSEA	KNN	0.925	0.375	36834	GSEA	LR	0.938	0.750
	12310	Gene	XGB	0.850	0.500	59111	P.+GSEA	LR	0.938	0.750
	22277	Probe	GNB	0.850	0.500	12310	Gene	LR	0.938	0.625
H1N1 DEE3	49144	G.+GSEA	RF*	1.000	0.667	12310	Gene	NuSVC	0.903	0.500
	36834	GSEA	RF	1.000	0.667	49144	G.+GSEA	KNN*	0.867	0.500
	12310	Gene	NuSVC	1.000	0.500	36834	GSEA	KNN*	0.867	0.500
	59111	P.+GSEA	NuSVC	1.000	0.500	59111	P.+GSEA	KNN*	0.867	0.500
	22277	Probe	NuSVC	1.000	0.500	22277	Probe	NuSVC	0.850	0.500
H1N1 DEE4	36834	GSEA	XGB	1.000	0.857	12310	Gene	RF	1.000	0.857
	12310	Gene	RF	0.960	0.857	36834	GSEA	XGB*	1.000	0.857
	49144	G.+GSEA	SVM*	0.944	0.857	49144	G.+GSEA	NuSVC	0.944	0.857
	59111	P.+GSEA	SVM*	0.944	0.857	59111	P.+GSEA	NuSVC	0.944	0.857
	22277	Probe	XGB	0.929	0.857	22277	Probe	LGB	0.929	0.857
HRV DUKE	22277	Probe	LR	1.000	1.000	12310	Gene	RF	1.000	1.000
	49144	G.+GSEA	LR	0.974	0.875	36834	GSEA	RF*	1.000	1.000
	12310	Gene	LR	0.974	0.875	59111	P.+GSEA	XGB*	1.000	1.000
	36834	GSEA	LR	0.974	0.875	22277	Probe	RF	1.000	0.875
	59111	P.+GSEA	LR	0.974	0.875	49144	G.+GSEA	KNN	0.979	0.750
HRV UVA	36834	GSEA	XGB	1.000	1.000	49144	G.+GSEA	SVM*	1.000	0.600
	49144	G.+GSEA	SVM*	1.000	0.600	12310	Gene	SVM*	1.000	0.600
	12310	Gene	SVM*	1.000	0.600	36834	GSEA	SVM*	1.000	0.600
	59111	P.+GSEA	SVM*	1.000	0.600	59111	P.+GSEA	SVM*	1.000	0.600
	22277	Probe	SVM*	1.000	0.600	22277	Probe	SVM*	1.000	0.600

Table 4.11 The results of the best-performing models according to feature representation type for each experiment at time points T.48 and T.72 without feature selection on the infection prediction task. An asterisk (*) indicates that the hyper-parameters were not optimized. NF column shows the number of used features after the feature selection methods.

Exp.	TimePoint 48					TimePoint 72				
	NF	Feature	Clf.	AUPRC	ACC	NF	Feature	Clf.	AUPRC	ACC
RSV DEE1	49144	G.+GSEA	KNN*	0.917	0.800	36834	GSEA	XGB	0.903	0.600
	36834	GSEA	KNN*	0.917	0.800	12310	Gene	NuSVC	0.850	0.400
	59111	P.+GSEA	KNN*	0.917	0.800	49144	G.+GSEA	KNN*	0.822	0.600
	12310	Gene	KNN*	0.822	0.600	59111	P.+GSEA	KNN*	0.822	0.600
	22277	Probe	KNN*	0.822	0.600	22277	Probe	LGB	0.800	0.600
H3N2 DEE2	49144	G.+GSEA	KNN*	1.000	1.000	49144	G.+GSEA	GNB	1.000	1.000
	36834	GSEA	KNN*	1.000	1.000	36834	GSEA	NuSVC*	1.000	1.000
	59111	P.+GSEA	KNN*	1.000	1.000	59111	P.+GSEA	GNB	1.000	1.000
	22277	Probe	KNN	1.000	1.000	22277	Probe	GNB	1.000	1.000
	12310	Gene	LR	1.000	0.600	12310	Gene	LR	1.000	0.800
H3N2 DEE5	49144	G.+GSEA	LR*	1.000	0.750	49144	G.+GSEA	LR	1.000	1.000
	12310	Gene	LR	1.000	0.750	12310	Gene	LR	1.000	1.000
	36834	GSEA	LR*	1.000	0.750	36834	GSEA	LR	1.000	1.000
	59111	P.+GSEA	LR*	1.000	0.750	59111	P.+GSEA	LR	1.000	1.000
	22277	Probe	LR*	1.000	0.625	22277	Probe	LR	1.000	1.000
H1N1 DEE3	12310	Gene	KNN	0.933	0.833	49144	G.+GSEA	GNB	0.800	0.667
	59111	P.+GSEA	NuSVC	0.903	0.667	12310	Gene	GNB	0.800	0.667
	49144	G.+GSEA	RF*	0.850	0.500	36834	GSEA	XGB	0.800	0.667
	36834	GSEA	LR	0.800	0.667	59111	P.+GSEA	XGB	0.800	0.667
	22277	Probe	GNB	0.800	0.667	22277	Probe	GNB	0.800	0.667
H1N1 DEE4	49144	G.+GSEA	NuSVC	1.000	0.857	59111	P.+GSEA	XGB	0.974	0.857
	12310	Gene	NuSVC	1.000	0.857	22277	Probe	XGB	0.974	0.857
	22277	Probe	NuSVC	1.000	0.857	36834	GSEA	RF	0.960	0.714
	59111	P.+GSEA	XGB	0.988	0.857	49144	G.+GSEA	NuSVC	0.944	0.857
	36834	GSEA	SVM	0.944	0.857	12310	Gene	LGB	0.929	0.857
HRV DUKE	49144	G.+GSEA	KNN*	1.000	1.000	49144	G.+GSEA	RF	1.000	0.750
	36834	GSEA	KNN*	1.000	1.000	36834	GSEA	RF*	1.000	0.750
	59111	P.+GSEA	KNN*	1.000	1.000	22277	Probe	LR	1.000	0.750
	12310	Gene	RF*	1.000	0.875	59111	P.+GSEA	KNN	0.988	0.875
	22277	Probe	KNN*	0.979	0.875	12310	Gene	RF*	0.974	0.750
HRV UVA	49144	G.+GSEA	SVM*	1.000	0.600	12310	Gene	SVM*	1.000	0.600
	12310	Gene	SVM*	1.000	0.600	22277	Probe	SVM*	1.000	0.600
	36834	GSEA	SVM*	1.000	0.600	49144	G.+GSEA	SVM*	0.903	0.600
	59111	P.+GSEA	SVM*	1.000	0.600	36834	GSEA	SVM*	0.903	0.600
	22277	Probe	SVM*	1.000	0.600	59111	P.+GSEA	SVM*	0.903	0.600

Tables 4.10, 4.11, and 4.12 present the results of the models obtained on the data sets in which feature selection has not been performed for the time points between T0 and T120.

Table 4.12 The results of the best-performing models according to feature representation type for each experiment at time points T.96 and T.120 without feature selection on the infection prediction task. An asterisk (*) indicates that the hyper-parameters were not optimized. NF column shows the number of used features after the feature selection methods.

Exp.	TimePoint 96					TimePoint 120				
	NF	Feature	Clf.	AUPRC	ACC	NF	Feature	Clf.	AUPRC	ACC
RSV DEE1	12310	Gene	RF	0.903	0.600	49144	G.+GSEA	XGB*	0.903	0.800
	49144	G.+GSEA	KNN	0.875	0.800	12310	Gene	XGB*	0.903	0.800
	36834	GSEA	KNN	0.875	0.800	59111	P.+GSEA	RF*	0.903	0.600
	59111	P.+GSEA	KNN	0.875	0.800	36834	GSEA	KNN	0.875	0.800
	22277	Probe	LGB	0.800	0.600	22277	Probe	LR	0.850	0.600
H3N2 DEE2	36834	GSEA	XGB	1.000	1.000	36834	GSEA	XGB	1.000	1.000
	49144	G.+GSEA	LR	1.000	0.800	49144	G.+GSEA	LR	1.000	0.800
	12310	Gene	LR	1.000	0.800	12310	Gene	LR	1.000	0.800
	59111	P.+GSEA	LR	1.000	0.800	59111	P.+GSEA	LR	1.000	0.800
	22277	Probe	LR	1.000	0.800	22277	Probe	LR	1.000	0.800
H3N2 DEE5	49144	G.+GSEA	SVM	1.000	1.000	49144	G.+GSEA	LR*	1.000	1.000
	12310	Gene	LR	1.000	1.000	12310	Gene	LR	1.000	1.000
	36834	GSEA	SVM	1.000	1.000	36834	GSEA	LR*	1.000	1.000
	59111	P.+GSEA	LR	1.000	1.000	59111	P.+GSEA	SVM*	1.000	1.000
	22277	Probe	LR	1.000	1.000	22277	Probe	NuSVC*	1.000	1.000
H1N1 DEE3	49144	G.+GSEA	XGB	0.800	0.667	59111	P.+GSEA	RF	0.933	0.667
	12310	Gene	GNB	0.800	0.667	36834	GSEA	RF*	0.850	0.667
	36834	GSEA	XGB	0.800	0.667	49144	G.+GSEA	XGB	0.800	0.667
	59111	P.+GSEA	GNB	0.800	0.667	12310	Gene	GNB	0.800	0.667
	22277	Probe	GNB	0.800	0.667	22277	Probe	GNB	0.800	0.667
H1N1 DEE4	59111	P.+GSEA	RF	0.988	0.857	36834	GSEA	XGB	1.000	0.857
	49144	G.+GSEA	RF	0.944	0.857	12310	Gene	NuSVC	0.944	0.857
	12310	Gene	NuSVC	0.944	0.857	49144	G.+GSEA	LGB	0.929	0.857
	36834	GSEA	KNN*	0.933	0.714	59111	P.+GSEA	LGB	0.929	0.857
	22277	Probe	LGB	0.929	0.857	22277	Probe	LGB	0.929	0.857
HRV DUKE	12310	Gene	RF*	1.000	0.750	22277	Probe	RF*	1.000	0.750
	49144	G.+GSEA	RF*	0.965	0.750	49144	G.+GSEA	XGB	0.988	0.875
	36834	GSEA	RF	0.965	0.750	36834	GSEA	XGB	0.979	0.875
	22277	Probe	RF*	0.943	0.750	12310	Gene	KNN	0.979	0.750
	59111	P.+GSEA	LR	0.929	0.875	59111	P.+GSEA	LR	0.929	0.875
HRV UVA	12310	Gene	SVM*	1.000	0.600	49144	G.+GSEA	XGB*	1.000	0.600
	22277	Probe	SVM*	1.000	0.600	12310	Gene	XGB	1.000	0.600
	49144	G.+GSEA	SVM*	0.903	0.600	22277	Probe	SVM*	1.000	0.600
	36834	GSEA	SVM*	0.903	0.600	36834	GSEA	SVM*	0.850	0.600
	59111	P.+GSEA	SVM*	0.903	0.600	59111	P.+GSEA	SVM*	0.850	0.600

When evaluating the results for the pre-infection period T.0, it can be observed that the Gene+GSEA representation, in which the gene expression values, and the enrichment

values are concatenated end-to-end, is one of the leading models in almost all sub-experiments. Furthermore, the Gene+GSEA representation approach achieved an AUPRC value of 1 in the DEE2, DEE3, and UVA sub-experiments, correctly predicting all infected individuals. In addition, the best performance in T0 prediction was achieved by combinations of XGB classifier with GSEA-based representation (Probe+GSEA, Gene+GSEA or GSEA only) in DEE1, DEE4, DEE5 and UVA sub-experiments.

Another notable finding is 100% accuracy for the HRV UVA sub-experiment was only obtained by of GSEA-based representation at time point T.0. No such accuracy is achieved at other time points, even though it would be expected that post-exposure may result in easy prediction due to changes in genetic expression values.

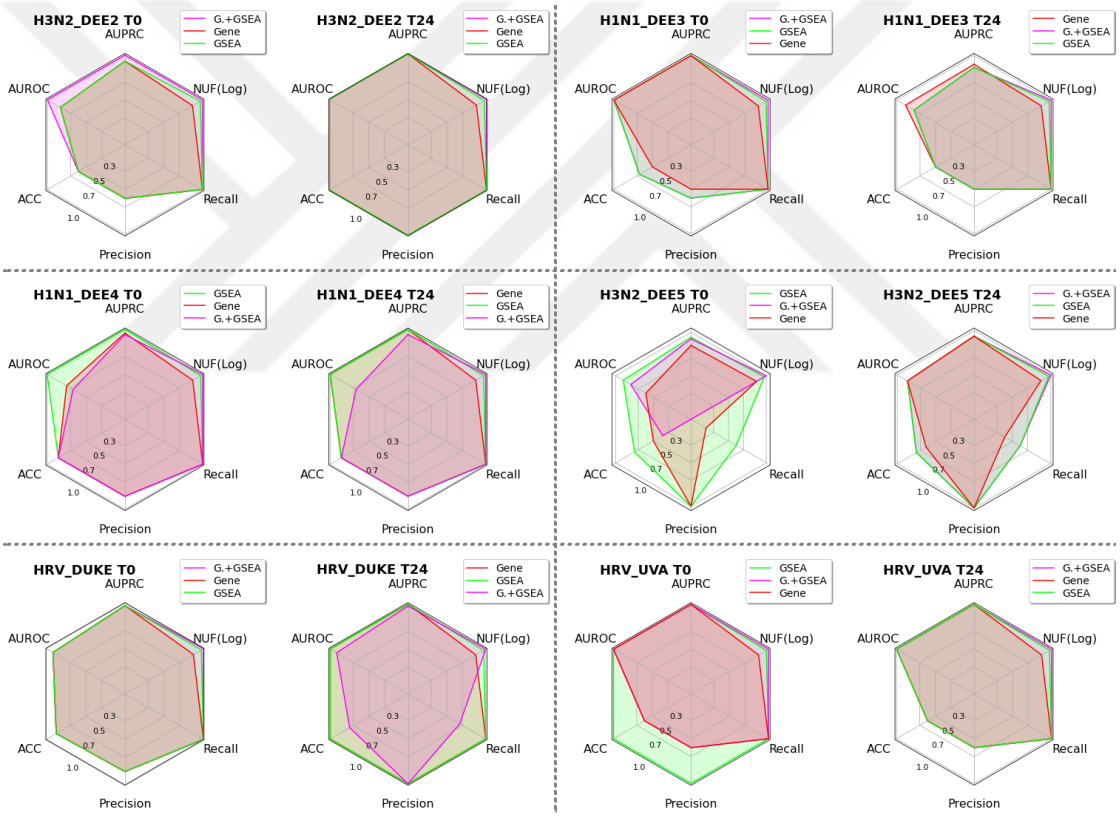

Figure 4.13 Comparison of multiple sub-experiments and time points for infection prediction problem using radar plots. Combining gene expression with GSEA features (i.e. “G+GSEA”) achieved almost the best results in experiment-based group analyses.

Moreover, the comparative radar plots shown in Figure 4.13 indicate that GSEA-based representations generated using gene expression allow for better performance with respect to multiple metrics in the pre- and early post-infection periods. These results may therefore be evidence that the use of GSEA-based representations in experiments with a

limited number of samples can better predict whether an individual becomes infected or not.

Table 4.13 The results of the best-performing models according to feature representation type for each experiment at time points T.0 and T.24 with feature selection on the infection prediction task. An asterisk (*) indicates that the hyper-parameters were not optimized. NF column shows the number of used features after the feature selection methods.

Exp.	TimePoint 0						TimePoint 24					
	FS	NF	Feature	Clf.	AUPRC	ACC	FS	NF	Feature	Clf.	AUPRC	ACC
RSV DEE1	Tree B.	4	G.+GSEA	XGB	0.867	0.600	Tree B.	4	G.+GSEA	NuSVC	1.000	1.000
	Tree B.	4	Gene	XGB	0.867	0.600	Tree B.	4	Gene	NuSVC	1.000	1.000
	Tree B.	4	P.+GSEA	XGB	0.867	0.600	Lasso	2261	GSEA	RF	1.000	1.000
	Tree B.	4	Probe.	XGB	0.867	0.600	Fisher S.	62	Probe.	XGB*	0.958	0.600
	Lasso	1211	GSEA	LR	0.850	0.400	Lasso	2094	P.+GSEA	XGB	0.903	0.800
H3N2 DEE2	Fisher S.	5	G.+GSEA	XGB	1.000	1.000	ReliefF	13	Gene	XGB	1.000	1.000
	Fisher S.	23	Gene	XGB	1.000	1.000	Tree B.	5	GSEA	SVM*	1.000	0.800
	Fisher S.	5	P.+GSEA	XGB	1.000	1.000	Fisher S.	13	Probe.	LR	1.000	0.800
	Fisher S.	17	Probe.	XGB	1.000	1.000	Tree B.	6	G.+GSEA	KNN*	0.958	0.800
	Lasso	1503	GSEA	XGB	0.903	0.800	ReliefF	7	P.+GSEA	XGB	0.933	0.800
H3N2 DEE5	ReliefF	13	G.+GSEA	RF*	1.000	0.500	mRMR	2	GSEA	SVM*	1.000	0.875
	ReliefF	13	GSEA	RF*	1.000	0.500	Fisher S.	4	P.+GSEA	LR	1.000	0.875
	ReliefF	13	P.+GSEA	RF*	1.000	0.500	Lasso	17	Probe.	GNB	1.000	0.875
	Fisher S.	12	Gene	XGB*	0.967	0.625	mRMR	4	G.+GSEA	NuSVC	1.000	0.750
	ReliefF	308	Probe.	XGB	0.938	0.750	ReliefF	25	Gene	LR	1.000	0.625
H1N1 DEE3	ReliefF	16	G.+GSEA	LR	1.000	0.833	ReliefF	7	G.+GSEA	RF	1.000	0.833
	Fisher S.	21	Gene	XGB	1.000	0.833	ReliefF	7	GSEA	RF	1.000	0.833
	ReliefF	16	GSEA	LR	1.000	0.833	ReliefF	7	P.+GSEA	RF	1.000	0.833
	ReliefF	16	P.+GSEA	LR	1.000	0.833	ReliefF	34	Probe.	SVM*	1.000	0.500
	Lasso	59	Probe.	XGB*	0.903	0.833	Lasso	39	Gene	LR	0.903	0.667
H1N1 DEE4	Tree B.	6	G.+GSEA	LR	1.000	0.857	Tree B.	6	GSEA	KNN	1.000	1.000
	Tree B.	6	Gene	LR	1.000	0.857	Fisher S.	31	Probe.	XGB*	1.000	1.000
	Tree B.	6	GSEA	LR	1.000	0.857	Lasso	993	G.+GSEA	RF	0.974	0.857
	ReliefF	20	P.+GSEA	XGB*	0.960	0.857	ReliefF	59	Gene	XGB*	0.974	0.857
	Tree B.	6	Probe.	GNB	0.944	0.714	ReliefF	1	P.+GSEA	LR	0.974	0.714
HRV DUKE	Lasso	49	Gene	XGB*	1.000	0.875	Fisher S.	5	G.+GSEA	XGB	1.000	1.000
	Tree B.	9	GSEA	XGB	1.000	0.875	Fisher S.	6	Gene	XGB*	1.000	1.000
	ReliefF	49	Probe.	GNB	1.000	0.750	Fisher S.	4	GSEA	RF	1.000	1.000
	Lasso	2990	G.+GSEA	LR*	0.974	0.750	Lasso	2878	P.+GSEA	RF	1.000	1.000
	Lasso	2850	P.+GSEA	KNN*	0.955	0.875	ReliefF	39	Probe.	XGB	1.000	0.875
HRV UVA	Fisher S.	7	G.+GSEA	XGB	1.000	1.000	Fisher S.	3	G.+GSEA	XGB	1.000	1.000
	Lasso	20	Gene	XGB	1.000	1.000	Fisher S.	3	GSEA	XGB	1.000	1.000
	Tree B.	3	GSEA	XGB	1.000	1.000	Fisher S.	3	P.+GSEA	XGB	1.000	1.000
	Fisher S.	5	P.+GSEA	XGB	1.000	1.000	Lasso	22	Probe.	RF	1.000	1.000
	Lasso	28	Probe.	LR	1.000	0.800	Fisher S.	21	Gene	SVM	1.000	0.600

Tables 4.13, 4.14, 4.15 show the results of the different models for predicting infection depending on the prediction time after the feature selection was performed.

Table 4.14 The results of the best-performing models according to feature representation type for each experiment at time points T.48 and T.72 with feature selection on the infection prediction task. An asterisk (*) indicates that the hyper-parameters were not optimized. NF column shows the number of used features after the feature selection methods.

Exp.	TimePoint 48						TimePoint 72					
	FS	NF	Feature	Clf.	AUPRC	ACC	FS	NF	Feature	Clf.	AUPRC	ACC
RSV DEE1	Fisher S.	7	P.+GSEA	LR	1.000	0.800	Lasso	1388	GSEA	XGB	1.000	0.800
	mRMR	36	Probe.	SVM	1.000	0.800	ReliefF	1811	G.+GSEA	XGB	1.000	0.600
	Fisher S.	1	G.+GSEA	LR	1.000	0.600	ReliefF	1811	P.+GSEA	XGB	1.000	0.600
	Lasso	38	Gene	RF	1.000	0.600	ReliefF	114	Gene	KNN*	0.917	0.600
	Fisher S.	1	GSEA	LR	1.000	0.600	Lasso	62	Probe.	XGB*	0.903	0.600
H3N2 DEE2	Fisher S.	3	G.+GSEA	XGB	1.000	1.000	ReliefF	29	G.+GSEA	LR	1.000	1.000
	Fisher S.	11	Gene	SVM	1.000	1.000	Fisher S.	96	Gene	NuSVC	1.000	1.000
	Fisher S.	3	GSEA	XGB	1.000	1.000	Tree B.	5	GSEA	GNB	1.000	1.000
	Fisher S.	4	P.+GSEA	XGB	1.000	1.000	ReliefF	29	P.+GSEA	LR	1.000	1.000
	Tree B.	5	Probe.	RF	1.000	0.800	Fisher S.	26	Probe.	LR	1.000	0.800
H3N2 DEE5	Lasso	16	Gene	LR	1.000	1.000	Fisher S.	2	G.+GSEA	GNB	1.000	1.000
	Lasso	872	GSEA	XGB	1.000	1.000	Fisher S.	13	Gene	XGB	1.000	1.000
	Fisher S.	8	Probe.	LR	1.000	0.875	Fisher S.	2	GSEA	GNB	1.000	1.000
	ReliefF	305	G.+GSEA	GNB	1.000	0.750	Tree B.	2	P.+GSEA	GNB	1.000	1.000
	ReliefF	305	P.+GSEA	GNB	1.000	0.750	Tree B.	2	Probe.	GNB	1.000	1.000
H1N1 DEE3	mRMR	6	GSEA	LR	1.000	0.667	ReliefF	2	Gene	SVM*	1.000	0.500
	mRMR	13	P.+GSEA	LR*	1.000	0.667	ReliefF	2	Probe.	LR	1.000	0.500
	ReliefF	116	Probe.	RF	1.000	0.667	Tree B.	1	G.+GSEA	GNB	0.850	0.667
	Lasso	1238	G.+GSEA	RF	1.000	0.500	Tree B.	1	GSEA	GNB	0.850	0.667
	ReliefF	52	Gene	LR	1.000	0.500	Tree B.	1	P.+GSEA	GNB	0.850	0.667
H1N1 DEE4	Lasso	1086	G.+GSEA	NuSVC	1.000	1.000	Fisher S.	2	P.+GSEA	XGB	1.000	1.000
	ReliefF	83	GSEA	NuSVC	1.000	0.857	mRMR	210	Probe.	XGB*	1.000	1.000
	ReliefF	83	P.+GSEA	NuSVC	1.000	0.857	Fisher S.	3	G.+GSEA	KNN	0.988	0.857
	Tree B.	6	Probe.	XGB	0.988	0.857	Fisher S.	1	GSEA	KNN	0.988	0.857
	Lasso	25	Gene	LR	0.974	0.714	Tree B.	6	Gene	KNN	0.976	0.857
HRV DUKE	Tree B.	6	GSEA	GNB	1.000	1.000	Lasso	50	Probe.	RF	1.000	1.000
	Lasso	3269	G.+GSEA	RF	1.000	0.750	Fisher S.	15	G.+GSEA	RF*	1.000	0.875
	ReliefF	4	Gene	RF	1.000	0.750	Lasso	875	GSEA	XGB	1.000	0.875
	ReliefF	72	Probe.	SVM	0.974	0.750	Lasso	827	P.+GSEA	RF*	1.000	0.750
	ReliefF	64	P.+GSEA	KNN*	0.965	0.750	ReliefF	61	Gene	RF*	0.988	0.875
HRV UVA	ReliefF	5	Probe.	RF	1.000	1.000	ReliefF	5	Probe.	RF	1.000	1.000
	ReliefF	4	Gene	SVM	1.000	0.800	ReliefF	4	Gene	SVM	1.000	0.800
	mRMR	4	P.+GSEA	SVM*	1.000	0.800	Tree B.	2	G.+GSEA	SVM*	1.000	0.600
	Tree B.	2	G.+GSEA	SVM*	0.903	0.600	Fisher S.	2	GSEA	SVM*	1.000	0.600
	Lasso	2424	GSEA	SVM*	0.850	0.600	Fisher S.	3	P.+GSEA	SVM*	1.000	0.600

Table 4.15 The results of the best-performing models according to feature representation type for each experiment at time points T.96 and T.120 with feature selection on the infection prediction task. An asterisk (*) indicates that the hyper-parameters were not optimized. NF column shows the number of used features after the feature selection methods.

Exp	TimePoint 96						TimePoint 120					
	FS	NF	Feature	Clf.	AUPRC	ACC	FS	NF	Feature	Clf.	AUPRC	ACC
RSV DEE1	ReliefF	2918	G.+GSEA	XGB*	1.000	0.800	Tree B.	4	P.+GSEA	NuSVC*	1.000	0.800
	ReliefF	3026	GSEA	XGB*	1.000	0.800	Tree B.	4	Probe.	NuSVC*	1.000	0.800
	ReliefF	2918	P.+GSEA	XGB*	1.000	0.800	mRMR	5	GSEA	XGB	0.958	0.800
	ReliefF	273	Probe.	XGB*	1.000	0.800	ReliefF	274	Gene	KNN	0.933	0.600
	ReliefF	179	Gene	KNN	0.958	0.600	Tree B.	3	G.+GSEA	XGB	0.903	0.800
H3N2 DEE2	ReliefF	2602	G.+GSEA	NuSVC	1.000	1.000	Tree B.	5	G.+GSEA	RF	1.000	1.000
	Fisher S.	13	Gene	GNB	1.000	1.000	Tree B.	5	Gene	RF	1.000	1.000
	ReliefF	2602	GSEA	NuSVC	1.000	1.000	Tree B.	5	GSEA	XGB	1.000	1.000
	Tree B.	6	P.+GSEA	SVM	1.000	1.000	ReliefF	6202	P.+GSEA	NuSVC	1.000	1.000
	Tree B.	6	Probe.	SVM	1.000	1.000	Fisher S.	16	Probe.	XGB	1.000	1.000
H3N2 DEE5	Lasso	1888	G.+GSEA	RF	1.000	1.000	Lasso	1921	G.+GSEA	GNB	1.000	1.000
	Fisher S.	23	Gene	RF	1.000	1.000	Fisher S.	11	Gene	RF	1.000	1.000
	Lasso	1692	GSEA	RF	1.000	1.000	Tree B.	2	P.+GSEA	GNB	1.000	1.000
	Tree B.	2	P.+GSEA	XGB	1.000	1.000	Tree B.	2	Probe.	GNB	1.000	1.000
	Tree B.	2	Probe.	XGB	1.000	1.000	Lasso	1650	GSEA	LR	1.000	0.875
H1N1 DEE3	ReliefF	2	Probe.	LR	1.000	0.667	Lasso	51	Probe.	XGB*	1.000	0.667
	mRMR	7	G.+GSEA	KNN	1.000	0.500	Lasso	22	Gene	RF*	1.000	0.500
	ReliefF	1	Gene	LR	1.000	0.500	Tree B.	1	P.+GSEA	LR	1.000	0.500
	Tree B.	1	P.+GSEA	LR	1.000	0.500	ReliefF	439	GSEA	NuSVC	0.903	0.500
	ReliefF	7	GSEA	XGB	0.958	0.500	Tree B.	1	G.+GSEA	GNB	0.850	0.667
H1N1 DEE4	Tree B.	6	G.+GSEA	KNN	1.000	1.000	Tree B.	6	GSEA	NuSVC	1.000	0.857
	Tree B.	6	Gene	KNN	1.000	1.000	Lasso	851	G.+GSEA	XGB	0.974	0.857
	Tree B.	6	GSEA	GNB	1.000	0.857	Lasso	34	Probe.	XGB	0.974	0.857
	Fisher S.	1	P.+GSEA	XGB*	0.976	0.857	Lasso	21	Gene	RF	0.974	0.714
	Lasso	35	Probe.	XGB*	0.974	0.857	Fisher S.	1	P.+GSEA	XGB	0.964	0.571
HRV DUKE	Lasso	41	Gene	RF	1.000	0.875	Lasso	806	GSEA	XGB	1.000	1.000
	Lasso	816	P.+GSEA	XGB*	1.000	0.875	Tree B.	2	G.+GSEA	GNB	1.000	0.875
	Lasso	46	Probe.	RF	1.000	0.875	ReliefF	56	Gene	SVM	1.000	0.875
	Lasso	799	GSEA	XGB	0.974	0.875	Lasso	26	Probe.	RF*	1.000	0.875
	Tree B.	7	G.+GSEA	SVM	0.974	0.750	Lasso	808	P.+GSEA	XGB	0.979	0.875
HRV UVA	ReliefF	3	Gene	RF	1.000	1.000	Tree B.	3	G.+GSEA	GNB	1.000	1.000
	mRMR	5	P.+GSEA	SVM*	1.000	1.000	Tree B.	3	Gene	GNB	1.000	1.000
	ReliefF	5	Probe.	RF	1.000	1.000	Fisher S.	3	GSEA	SVM*	1.000	1.000
	mRMR	5	G.+GSEA	SVM*	1.000	0.800	Fisher S.	6	P.+GSEA	SVM*	1.000	1.000
	Fisher S.	2	GSEA	SVM*	1.000	0.800	ReliefF	3	Probe.	RF	1.000	1.000

Results of the tables broadly indicate that the application of feature selection boosted results of both AURPC and Accuracy across all sub-experiments, with the

exception of the T.0 RSV sub-experiment and the DEE2 sub-experiment at the T24 point. This improvement is not confined to a specific representation approach or classifier but is observed in nearly all combinations despite the small number of features. For instance, an AUPRC value of 1 is obtained using 36,834 features with the XGB algorithm and GSEA feature type without feature selection at the T0 time point prediction in the HRV UVA sub-experiment. However, the same performance is achieved using only 3 features after applying tree-based feature selection with the exact same representation-classifier combination. Similar results are also observable in other sub-experiments. It is also observed that all subjects are correctly predicted in the sub-experiments except DEE3. For example, in the DEE2, DEE5 and UVA sub-experiments, subjects were predicted with 100% accuracy at all time points, whereas in the DEE1, DEE4 and DUKE sub-experiments, subjects were predicted with 100% accuracy specifically at T.24. Considering the number of used features, it can be stated that the application of feature selection has a positive impact on the prediction performance of the sub-experiment. When comparing feature selection methods, Fisher Score, ReliefF and the embedded tree-based approach consistently outperform others at all time points and in all sub-experiments, except for HRV DUKE at T.96 and T.120. Additionally, tree-based method is most efficient approach as uses least number of features while keeping high predictivity.

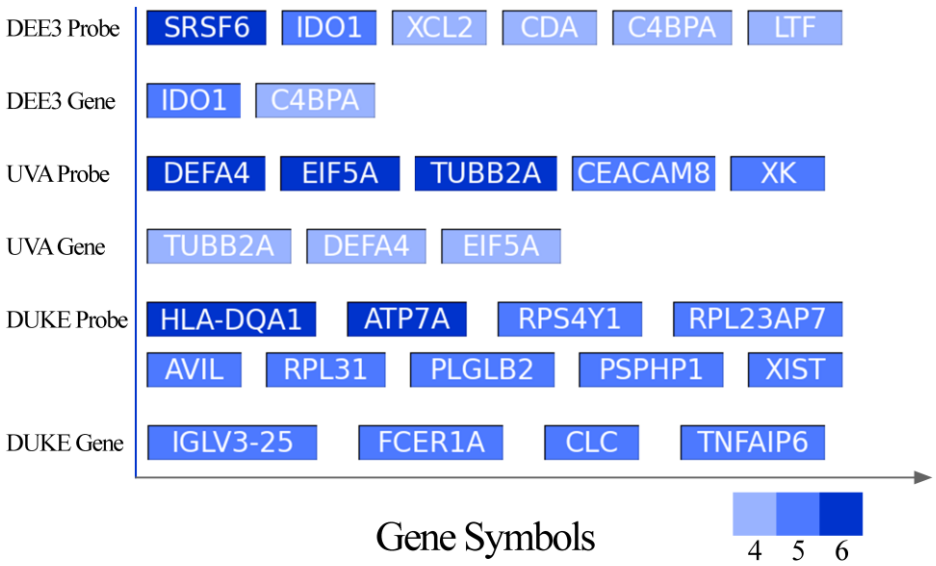

Figure 4.14 Mostly selected genes among the different time points for each sub-experiment according to Probe- and Gene-level representations in infection prediction problem. Genes of the experiment “HRV DUKE” are restricted due to large number of 4 times occurred genes.

In many cases, feature selection is utilized not only to reduce dimensionality but also to detect key-parts that affect the phenotype class, such as infection status. In this way, the expression of selected genes or probes can also be interpreted as being significant in terms of infection. In our analysis, we expect that a gene or probe selected during more than 4 time points is likely significant in terms of prediction of that sub-experiment. Because that mean these genes played an important role throughout the post-exposure period while the biological system responded to the virus. Therefore, we extracted the number of genes and probes meeting this criterion as shown in Figure 4.14. However, some sub-experiments such as DEE1 or DEE2 are not included because no gene or probe was selected for them at more than 4 time points. Gene- and probe-expression based significant parts are figured out separately since the selected parts also vary. Probe ids then mapped to genes using an appropriate annotation file.

When the most frequently selected genes are examined in detail, they are often found in the literature to be related either to the virus involved in the experiment or to the immune system. For example, let's consider IDO1 and C4BPA, genes selected at both gene and probe levels in the DEE3 H1N1 sub-experiment. According to available literature, the IDO1 gene is responsible for encoding the indoleamine-2,3-dioxygenase enzyme which plays a significant role in immune response against influenza A (H1N1) virus [218]. C4BPA, on the other hand, belongs to the group of C4b-binding proteins that suppress the complement system and one of the body's immune systems. This specific C4BP group acts as an entry inhibitor for H1N1 while at the same time promoting an immune response [219]. Similarly, a study has reported an interaction between SRSF group proteins and the Influenza A virus that regulates viral RNA splicing and replication [220].

In other sub-experiments, there are also many literature studies showing that there are relationship between the selected genes and viral infection, a respiratory virus or the immune system [221-223]. For example, the Gene DEFA4, which is selected as significant in UVA sub-experiments, is one of the most commonly overexpressed genes associated with neutrophil function in rhinovirus (HRV) infection [224]. The DEFA4 encodes for a protein called human neutrophil activation (HNP-4), and HNP-4 has been shown to have antiviral activity against HRV, as well as other respiratory viruses. Genes linked to neutrophil activation were also found to be effective in predicting outcomes of respiratory infections through further pathway analysis, explained below.

Examining genes individually is crucial for understanding the function of each gene and its potential role in diseases. However, this approach might ignore the interactions between genes and how they function as a whole. Therefore, Over-representation Analysis was performed on concatenation of all selected genes to identify their collective impact and determine their association with biological pathways or functional groups.

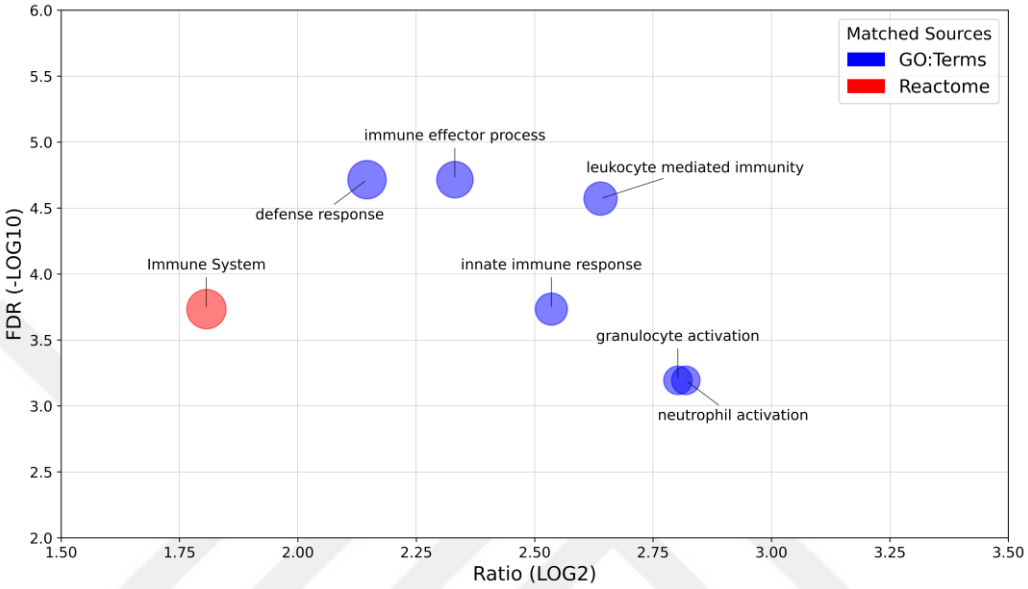

Figure 4.15 Overrepresented Pathways and GO Terms on the mostly selected genes in infection prediction problem presented in the Figure 4.14.

Figure 4.15 depicts the over-represented pathways and gene ontology (GO) terms on the selected genes shown in Figure 4.14. The majority of over-represented pathways are either directly or secondarily associated with the immune system, similar to the linkage of individual genes to the immune system and immune response. Additionally, GO terms “Neutrophil activation” and “Granulocyte Activation” have also associations with immunity as they are being activated in response to infection or inflammation as part of the innate immune response [225]. These findings indicate a strong correlation between respiratory virus infection and the immune system being infected after exposure.

The other task we made prediction in our experiments was the severe symptoms presence following exposure to viruses. Tables 4.16 to 4.18 and 4.19 to 4.21 show the prediction scores of symptomatic individuals using the full features and feature selected datasets, respectively. In particular, at time point T.24, it can be observed on the top results of the table that all symptomatic subjects were accurately predicted by achieving an AUPRC value of 1. Similar to the infection prediction task, high classification

performance was achieved across all approaches in predicting whether subjects would develop symptoms.

Table 4.16 The results of the best-performing models according to feature representation type for each experiment at time points T.0 and T.24 without feature selection on the symptomatic prediction task. An asterisk (*) indicates that the hyper-parameters were not optimized. NF column shows the number of used features after the feature selection methods.

Exp.	TimePoint 0					TimePoint 24				
	NF	Feature	Clf.	AUPRC	ACC	NF	Feature	Clf.	AUPRC	ACC
RSV DEE1	49144	G.+GSEA	SVM*	0.850	0.600	49144	G.+GSEA	XGB	1.000	1.000
	12310	Gene	SVM*	0.850	0.600	12310	Gene	XGB	1.000	1.000
	36834	GSEA	SVM*	0.850	0.600	59111	P.+GSEA	XGB	0.875	0.800
	59111	P.+GSEA	SVM*	0.850	0.600	22277	Probe	XGB	0.875	0.800
	22277	Probe	SVM*	0.850	0.600	36834	GSEA	SVM*	0.850	0.600
H3N2 DEE2	49144	G.+GSEA	LR	1.000	1.000	49144	G.+GSEA	RF	1.000	1.000
	12310	Gene	LR	1.000	1.000	12310	Gene	RF	1.000	1.000
	36834	GSEA	LR	1.000	1.000	36834	GSEA	NuSVC	1.000	1.000
	59111	P.+GSEA	LR	1.000	1.000	59111	P.+GSEA	RF	1.000	1.000
	22277	Probe	LR	1.000	1.000	22277	Probe	RF	1.000	1.000
H3N2 DEE5	12310	Gene	NuSVC	0.963	0.625	22277	Probe	NuSVC*	1.000	0.625
	49144	G.+GSEA	KNN	0.950	0.625	49144	G.+GSEA	KNN*	0.971	0.750
	36834	GSEA	KNN	0.950	0.625	12310	Gene	KNN*	0.971	0.750
	59111	P.+GSEA	KNN	0.950	0.625	36834	GSEA	KNN*	0.971	0.750
	22277	Probe	NuSVC	0.938	0.625	59111	P.+GSEA	KNN*	0.971	0.750
H1N1 DEE3	49144	G.+GSEA	RF	1.000	1.000	12310	Gene	RF*	1.000	0.833
	12310	Gene	NuSVC	1.000	0.667	59111	P.+GSEA	RF*	0.975	0.833
	36834	GSEA	RF*	1.000	0.667	22277	Probe	XGB	0.958	0.500
	59111	P.+GSEA	NuSVC	1.000	0.333	49144	G.+GSEA	XGB	0.917	0.667
	22277	Probe	KNN*	0.975	0.833	36834	GSEA	XGB	0.917	0.667
H1N1 DEE4	49144	G.+GSEA	NuSVC	1.000	0.714	49144	G.+GSEA	RF	1.000	1.000
	12310	Gene	NuSVC	1.000	0.714	36834	GSEA	RF	1.000	1.000
	36834	GSEA	NuSVC	1.000	0.714	22277	Probe	NuSVC*	1.000	0.714
	59111	P.+GSEA	NuSVC	1.000	0.714	59111	P.+GSEA	KNN	0.917	0.857
	22277	Probe	LGB	0.643	0.714	12310	Gene	KNN	0.833	0.714
HRV DUKE	49144	G.+GSEA	LR*	1.000	0.750	36834	GSEA	LR	1.000	0.875
	36834	GSEA	LR*	1.000	0.750	49144	G.+GSEA	LR	0.944	0.875
	59111	P.+GSEA	LR*	1.000	0.750	59111	P.+GSEA	LR	0.944	0.875
	12310	Gene	LR	0.871	0.625	12310	Gene	XGB	0.927	0.750
	22277	Probe	LR	0.835	0.625	22277	Probe	SVM	0.908	0.750
HRV UVA	49144	G.+GSEA	RF	1.000	1.000	49144	G.+GSEA	XGB	1.000	0.600
	12310	Gene	RF*	1.000	0.800	12310	Gene	NuSVC	1.000	0.600
	36834	GSEA	KNN	0.958	0.800	22277	Probe	NuSVC	1.000	0.600
	59111	P.+GSEA	KNN	0.958	0.800	36834	GSEA	NuSVC*	0.903	0.800
	22277	Probe	LR*	0.903	0.800	59111	P.+GSEA	NuSVC*	0.903	0.800

Table 4.17 The results of the best-performing models according to feature representation type for each experiment at time points T.48 and T.72 without feature selection on the symptomatic prediction task. An asterisk (*) indicates that the hyper-parameters were not optimized. NF column shows the number of used features after the feature selection methods.

Exp.	TimePoint 48					TimePoint 72				
	NF	Feature	Clf.	AUPRC	ACC	NF	Feature	Clf.	AUPRC	ACC
RSV DEE1	49144	G.+GSEA	XGB	0.875	0.800	49144	G.+GSEA	XGB	0.867	0.600
	12310	Gene	XGB	0.875	0.800	12310	Gene	XGB	0.867	0.600
	36834	GSEA	LGB	0.800	0.400	36834	GSEA	LGB	0.800	0.400
	59111	P.+GSEA	LGB	0.800	0.400	59111	P.+GSEA	LGB	0.800	0.400
	22277	Probe	LGB	0.800	0.400	22277	Probe	LGB	0.800	0.400
H3N2 DEE2	22277	Probe	GNB	1.000	1.000	49144	G.+GSEA	LR	1.000	1.000
	49144	G.+GSEA	LR	1.000	0.800	12310	Gene	LR	1.000	1.000
	12310	Gene	RF	1.000	0.800	36834	GSEA	LR	1.000	1.000
	36834	GSEA	LR	1.000	0.800	59111	P.+GSEA	LR	1.000	1.000
	59111	P.+GSEA	LR	1.000	0.800	22277	Probe	LR	1.000	1.000
H3N2 DEE5	22277	Probe	KNN*	0.983	0.875	22277	Probe	KNN	0.943	0.750
	12310	Gene	KNN	0.950	0.875	36834	GSEA	XGB	0.938	0.750
	49144	G.+GSEA	KNN*	0.943	0.750	49144	G.+GSEA	KNN*	0.923	0.750
	36834	GSEA	KNN*	0.943	0.750	59111	P.+GSEA	KNN*	0.923	0.750
	59111	P.+GSEA	KNN*	0.943	0.750	12310	Gene	KNN	0.875	0.625
H1N1 DEE3	49144	G.+GSEA	RF	1.000	0.833	12310	Gene	KNN*	1.000	1.000
	12310	Gene	RF	0.975	0.833	36834	GSEA	RF	1.000	1.000
	36834	GSEA	RF	0.944	0.833	49144	G.+GSEA	RF	1.000	0.833
	22277	Probe	RF*	0.944	0.667	59111	P.+GSEA	KNN*	0.975	0.833
	59111	P.+GSEA	KNN	0.917	0.667	22277	Probe	LR	0.944	0.667
H1N1 DEE4	49144	G.+GSEA	KNN	0.833	0.714	36834	GSEA	XGB	1.000	1.000
	12310	Gene	KNN	0.833	0.714	49144	G.+GSEA	KNN*	0.875	0.714
	36834	GSEA	KNN	0.833	0.714	59111	P.+GSEA	KNN*	0.875	0.714
	59111	P.+GSEA	KNN	0.833	0.714	22277	Probe	KNN*	0.833	0.714
	22277	Probe	LGB	0.643	0.714	12310	Gene	NuSVC	0.792	0.714
HRV DUKE	49144	G.+GSEA	LR*	0.944	0.875	49144	G.+GSEA	LR	1.000	0.875
	12310	Gene	NuSVC	0.908	0.750	36834	GSEA	LR	1.000	0.875
	36834	GSEA	LR	0.908	0.750	59111	P.+GSEA	LR	1.000	0.875
	59111	P.+GSEA	LR	0.908	0.750	22277	Probe	SVM	0.908	0.750
	22277	Probe	LR	0.908	0.625	12310	Gene	LR	0.908	0.625
HRV UVA	12310	Gene	NuSVC*	0.903	0.600	12310	Gene	KNN*	0.917	0.800
	22277	Probe	NuSVC*	0.903	0.600	22277	Probe	KNN*	0.917	0.800
	36834	GSEA	XGB	0.875	0.800	49144	G.+GSEA	NuSVC*	0.903	0.800
	49144	G.+GSEA	NuSVC	0.850	0.600	36834	GSEA	NuSVC*	0.903	0.800
	59111	P.+GSEA	NuSVC	0.850	0.600	59111	P.+GSEA	XGB	0.903	0.800

During the pre-infection and early post-infection periods (T.0, T.24 and T.48), non-feature selection applied results generally indicate that the Gene, Gene+GSEA, and Probe

representation approaches yield the best results. Furthermore, AUPRC values of Gene+GSEA and Gene are mostly similar. For instance, in the T.0 results of DEE1, DEE2

Table 4.18 The results of the best-performing models according to feature representation type for each experiment at time points T.96 and T.120 without feature selection on the symptomatic prediction task. An asterisk (*) indicates that the hyper-parameters were not optimized. NF column shows the number of used features after the feature selection methods.

Exp.	TimePoint 96					TimePoint 120				
	NF	Feature	Clf.	AUPRC	ACC	NF	Feature	Clf.	AUPRC	ACC
RSV DEE1	49144	G.+GSEA	XGB	0.933	0.800	49144	G.+GSEA	XGB	0.933	0.800
	12310	Gene	XGB	0.933	0.800	12310	Gene	XGB	0.933	0.800
	22277	Probe	KNN	0.917	0.600	36834	GSEA	GNB	0.933	0.800
	36834	GSEA	GNB	0.867	0.600	59111	P.+GSEA	GNB	0.933	0.800
	59111	P.+GSEA	XGB	0.867	0.600	22277	Probe	GNB	0.933	0.800
H3N2 DEE2	49144	G.+GSEA	LR	1.000	1.000	49144	G.+GSEA	LR	1.000	1.000
	12310	Gene	LR	1.000	1.000	12310	Gene	LR	1.000	1.000
	36834	GSEA	LR	1.000	1.000	36834	GSEA	LR	1.000	1.000
	59111	P.+GSEA	LR	1.000	1.000	59111	P.+GSEA	LR	1.000	1.000
	22277	Probe	LR	1.000	1.000	22277	Probe	LR	1.000	1.000
H3N2 DEE5	22277	Probe	KNN	0.925	0.750	12310	Gene	SVM	0.963	0.875
	49144	G.+GSEA	RF	0.918	0.750	36834	GSEA	RF	0.920	0.875
	36834	GSEA	XGB	0.918	0.750	22277	Probe	RF	0.920	0.750
	12310	Gene	LR	0.865	0.625	49144	G.+GSEA	RF*	0.918	0.625
	59111	P.+GSEA	RF	0.865	0.625	59111	P.+GSEA	RF*	0.918	0.750
H1N1 DEE3	49144	G.+GSEA	KNN	1.000	1.000	49144	G.+GSEA	RF	1.000	1.000
	12310	Gene	RF	1.000	1.000	12310	Gene	GNB	1.000	1.000
	36834	GSEA	KNN	1.000	1.000	36834	GSEA	RF	1.000	1.000
	59111	P.+GSEA	RF	1.000	1.000	59111	P.+GSEA	RF	1.000	1.000
	22277	Probe	GNB	1.000	1.000	22277	Probe	KNN	1.000	1.000
H1N1 DEE4	36834	GSEA	XGB	1.000	1.000	36834	GSEA	XGB	0.833	0.857
	49144	G.+GSEA	NuSVC	0.792	0.714	49144	G.+GSEA	KNN	0.792	0.857
	12310	Gene	LR	0.792	0.714	59111	P.+GSEA	KNN	0.792	0.857
	59111	P.+GSEA	NuSVC	0.792	0.714	12310	Gene	LR*	0.792	0.714
	22277	Probe	LR	0.792	0.714	22277	Probe	LR	0.792	0.714
HRV DUKE	36834	GSEA	XGB	1.000	1.000	59111	P.+GSEA	XGB	1.000	1.000
	49144	G.+GSEA	KNN	0.946	0.625	22277	Probe	XGB	1.000	1.000
	59111	P.+GSEA	KNN	0.946	0.625	49144	G.+GSEA	KNN	0.946	0.875
	12310	Gene	SVM	0.908	0.750	36834	GSEA	KNN	0.946	0.875
	22277	Probe	LR	0.908	0.625	12310	Gene	LR	0.944	0.875
HRV UVA	36834	GSEA	NuSVC*	1.000	1.000	12310	Gene	NuSVC	1.000	1.000
	59111	P.+GSEA	NuSVC*	1.000	1.000	49144	G.+GSEA	NuSVC*	1.000	0.800
	12310	Gene	NuSVC*	1.000	0.600	36834	GSEA	NuSVC*	1.000	0.800
	22277	Probe	NuSVC*	1.000	0.600	59111	P.+GSEA	NuSVC*	1.000	0.800
	49144	G.+GSEA	NuSVC*	0.903	0.600	22277	Probe	NuSVC*	1.000	0.600

Table 4.19 The results of the best-performing models according to feature representation type for each experiment at time points T.0 and T.24 with feature selection on the symptomatic prediction task. An asterisk (*) indicates that the hyper-parameters were not optimized. NF column shows the number of used features after the feature selection methods.

Exp.	TimePoint 0						TimePoint 24					
	FS	NF	Feature	Clf.	AUPRC	ACC	FS	NF	Feature	Clf.	AUPRC	ACC
RSV DEE1	Tree B.	1	GSEA	SVM	0.903	0.600	Tree B.	1	G.+GSEA	LR	1.000	1.000
	Tree B.	1	G.+GSEA	SVM	0.903	0.400	Tree B.	1	Gene	LR	1.000	1.000
	Tree B.	1	Gene	SVM	0.903	0.400	Fisher S.	5	Probe.	RF	1.000	1.000
	Lasso	34	Probe.	KNN*	0.878	0.600	Tree B.	1	P.+GSEA	RF	0.875	0.800
	Lasso	732	P.+GSEA	SVM*	0.850	0.600	mRMR	2	GSEA	LR*	0.850	0.800
H3N2 DEE2	Fisher S.	2	G.+GSEA	LR	1.000	1.000	Tree B.	1	G.+GSEA	LR	1.000	1.000
	Fisher S.	2	GSEA	LR	1.000	1.000	Tree B.	1	Gene	LR	1.000	1.000
	Tree B.	2	P.+GSEA	LR	1.000	1.000	Lasso	373	GSEA	RF	1.000	1.000
	Tree B.	2	Probe.	LR	1.000	1.000	Lasso	1527	P.+GSEA	RF	1.000	1.000
	Fisher S.	4	Gene	RF	1.000	0.600	Fisher S.	9	Probe.	XGB	1.000	1.000
H3N2 DEE5	Lasso	1769	GSEA	RF	0.963	0.625	Lasso	27	Gene	XGB	1.000	0.875
	Lasso	1917	P.+GSEA	XGB	0.963	0.625	ReliefF	28	G.+GSEA	LR*	0.963	0.875
	Lasso	2024	G.+GSEA	KNN	0.943	0.750	ReliefF	28	GSEA	LR*	0.963	0.875
	Tree B.	3	Gene	RF	0.920	0.500	ReliefF	28	P.+GSEA	LR*	0.963	0.875
	Tree B.	3	Probe.	XGB	0.857	0.750	Fisher S.	14	Probe.	XGB	0.963	0.750
H1N1 DEE3	Fisher S.	49	G.+GSEA	RF	1.000	0.833	mRMR	18	Gene	LR	1.000	1.000
	mRMR	28	GSEA	RF	1.000	0.833	Fisher S.	8	P.+GSEA	GNB	1.000	1.000
	mRMR	8	Probe.	GNB	1.000	0.833	ReliefF	101	Probe.	KNN	1.000	1.000
	Fisher S.	9	Gene	GNB	1.000	0.667	Fisher S.	6	G.+GSEA	GNB	1.000	0.833
	Fisher S.	54	P.+GSEA	RF	1.000	0.667	Fisher S.	6	GSEA	GNB	1.000	0.833
H1N1 DEE4	ReliefF	64	G.+GSEA	KNN	1.000	1.000	ReliefF	6	G.+GSEA	LR	1.000	1.000
	Lasso	7	Gene	GNB	1.000	1.000	ReliefF	12	Gene	RF	1.000	1.000
	ReliefF	64	GSEA	KNN	1.000	1.000	ReliefF	6	GSEA	LR	1.000	1.000
	ReliefF	64	P.+GSEA	KNN	1.000	1.000	ReliefF	6	P.+GSEA	LR	1.000	1.000
	ReliefF	16	Probe.	LR*	1.000	1.000	ReliefF	22	Probe.	XGB	1.000	1.000
HRV DUKE	ReliefF	3	G.+GSEA	XGB*	1.000	0.875	Lasso	1625	P.+GSEA	LR*	1.000	0.875
	ReliefF	3	GSEA	XGB*	1.000	0.875	ReliefF	2506	G.+GSEA	NuSVC	1.000	0.750
	ReliefF	3	P.+GSEA	XGB*	1.000	0.875	ReliefF	2506	GSEA	NuSVC	1.000	0.750
	Lasso	37	Gene	RF	1.000	0.625	ReliefF	1350	Gene	XGB	0.958	0.750
	Lasso	54	Probe.	GNB	0.866	0.375	ReliefF	2	Probe.	KNN*	0.950	0.750
HRV UVA	Fisher S.	15	Gene	GNB	1.000	1.000	Tree B.	3	G.+GSEA	KNN	1.000	1.000
	Fisher S.	11	P.+GSEA	RF	1.000	1.000	Tree B.	3	Gene	KNN	1.000	1.000
	Fisher S.	12	Probe.	RF	1.000	1.000	Lasso	35	Probe.	XGB	1.000	0.800
	Tree B.	3	G.+GSEA	LR	1.000	0.800	Tree B.	3	GSEA	SVM*	1.000	0.600
	mRMR	4	GSEA	GNB	1.000	0.800	mRMR	8	P.+GSEA	SVM*	1.000	0.600

DEE4 sub-experiments; both the Gene and Gene + GSEA approaches achieved identical highest predictivity. In this case, it can be stated that the extending gene expression values

with enrichment scores didn't improve the ability to classify the samples according to actual classes.

Table 4.20 The results of the best-performing models according to feature representation type for each experiment at time points T.48 and T.72 with feature selection on the symptomatic prediction task. An asterisk (*) indicates that the hyper-parameters were not optimized. NF column shows the number of used features after the feature selection methods.

Exp.	TimePoint 48						TimePoint 72					
	FS	NF	Feature	Clf.	AUPRC	ACC	FS	NF	Feature	Clf.	AUPRC	ACC
RSV DEE1	Tree B.	1	G.+GSEA	LR	1.000	0.800	ReliefF	9	Gene	KNN*	0.958	0.800
	Tree B.	1	Gene	LR	1.000	0.800	mRMR	3	GSEA	RF	0.958	0.800
	Lasso	12	Probe.	XGB	0.933	0.600	mRMR	3	P.+GSEA	RF	0.958	0.800
	Tree B.	1	GSEA	LGB	0.800	0.400	ReliefF	42	Probe.	KNN	0.903	0.600
	Tree B.	1	P.+GSEA	LGB	0.800	0.400	Fisher S.	4	G.+GSEA	KNN*	0.878	0.800
H3N2 DEE2	Lasso	1623	G.+GSEA	LR	1.000	1.000	Tree B.	2	G.+GSEA	LR	1.000	1.000
	Fisher S.	9	Gene	XGB	1.000	1.000	Tree B.	2	Gene	LR	1.000	1.000
	Lasso	432	GSEA	LR	1.000	1.000	Tree B.	2	GSEA	LR	1.000	1.000
	Fisher S.	6	P.+GSEA	XGB	1.000	1.000	Fisher S.	1	P.+GSEA	LR	1.000	1.000
	Fisher S.	6	Probe.	XGB	1.000	1.000	Fisher S.	6	Probe.	LR	1.000	1.000
H3N2 DEE5	Lasso	2203	G.+GSEA	XGB	1.000	1.000	Lasso	2303	G.+GSEA	XGB*	0.951	0.750
	Lasso	3410	GSEA	XGB	1.000	1.000	Lasso	25	Gene	RF*	0.938	0.750
	Lasso	1880	P.+GSEA	XGB	1.000	1.000	ReliefF	68	GSEA	XGB*	0.923	0.750
	mRMR	28	Gene	RF*	1.000	0.625	ReliefF	68	P.+GSEA	XGB*	0.923	0.750
	Tree B.	3	Probe.	GNB	0.860	0.875	Lasso	30	Probe.	RF*	0.906	0.750
H1N1 DEE3	ReliefF	20	Probe.	KNN	1.000	1.000	Lasso	2681	P.+GSEA	XGB	1.000	1.000
	Lasso	1525	G.+GSEA	RF	1.000	0.833	Lasso	2669	G.+GSEA	RF	1.000	0.833
	Lasso	25	Gene	SVM*	1.000	0.667	Tree B.	8	Gene	LR	1.000	0.833
	Lasso	1577	GSEA	RF	1.000	0.667	ReliefF	15	Probe.	RF	1.000	0.833
	Lasso	2730	P.+GSEA	RF	0.975	0.833	Lasso	1609	GSEA	KNN	0.958	0.667
H1N1 DEE4	ReliefF	4	G.+GSEA	RF	1.000	1.000	Tree B.	2	GSEA	RF	1.000	1.000
	ReliefF	4	GSEA	RF	1.000	1.000	Fisher S.	3	G.+GSEA	RF*	1.000	0.857
	ReliefF	4	P.+GSEA	RF	1.000	1.000	Fisher S.	3	P.+GSEA	RF*	1.000	0.857
	Fisher S.	11	Probe.	SVM*	1.000	0.714	ReliefF	106	Gene	KNN	0.917	0.857
	ReliefF	158	Gene	KNN*	0.833	0.857	ReliefF	33	Probe.	KNN*	0.917	0.857
HRV DUKE	Fisher S.	7	G.+GSEA	XGB*	0.944	0.875	ReliefF	13	G.+GSEA	LR	1.000	1.000
	Lasso	36	Gene	XGB	0.944	0.750	ReliefF	13	GSEA	LR	1.000	1.000
	Lasso	1695	P.+GSEA	NuSVC	0.944	0.750	ReliefF	13	P.+GSEA	LR	1.000	1.000
	Lasso	57	Probe.	KNN*	0.917	0.750	Lasso	35	Gene	KNN	1.000	0.500
	Tree B.	5	GSEA	KNN	0.896	0.625	mRMR	67	Probe.	RF	0.884	0.500
HRV UVA	ReliefF	1	G.+GSEA	SVM	1.000	0.800	Lasso	19	Gene	RF*	1.000	0.800
	ReliefF	1	GSEA	SVM	1.000	0.800	Tree B.	2	P.+GSEA	NuSVC	1.000	0.800
	ReliefF	1	P.+GSEA	SVM	1.000	0.800	Tree B.	2	Probe.	NuSVC	1.000	0.800
	Lasso	42	Probe.	RF	1.000	0.800	Fisher S.	63	GSEA	XGB	0.933	0.800
	Tree B.	3	Gene	SVM*	1.000	0.600	Fisher S.	93	G.+GSEA	RF	0.903	0.600

Table 4.21 The results of the best-performing models according to feature representation type for each experiment at time points T.96 and T.120 with feature selection on the symptomatic prediction task. An asterisk (*) indicates that the hyper-parameters were not optimized. NF column shows the number of used features after the feature selection methods.

Exp.	TimePoint 96						TimePoint 120					
	FS	NF	Feature	Clf.	AUPRC	ACC	FS	NF	Feature	Clf.	AUPRC	ACC
RSV DEE1	Lasso	33	Probe.	XGB*	0.958	0.600	Tree B.	1	G.+GSEA	SVM	1.000	0.800
	Tree B.	1	G.+GSEA	RF	0.933	0.800	Tree B.	1	Gene	SVM	1.000	0.800
	Tree B.	1	Gene	RF	0.933	0.800	ReliefF	36	Probe.	KNN*	0.933	0.800
	Tree B.	1	P.+GSEA	SVM	0.903	0.600	Tree B.	1	P.+GSEA	RF	0.933	0.600
	Tree B.	1	GSEA	LGB	0.800	0.400	Lasso	714	GSEA	GNB	0.878	0.800
H3N2 DEE2	Tree B.	2	G.+GSEA	LR	1.000	1.000	Tree B.	2	G.+GSEA	LR	1.000	1.000
	Tree B.	2	Gene	LR	1.000	1.000	Tree B.	2	Gene	LR	1.000	1.000
	Tree B.	2	GSEA	LR	1.000	1.000	Tree B.	2	GSEA	LR	1.000	1.000
	ReliefF	4	P.+GSEA	LR	1.000	1.000	Tree B.	2	P.+GSEA	RF	1.000	1.000
	Fisher S.	2	Probe.	LR	1.000	1.000	Tree B.	2	Probe.	RF	1.000	1.000
H3N2 DEE5	Lasso	51	Probe.	RF	1.000	1.000	ReliefF	29	Gene	GNB	0.938	0.750
	Lasso	970	G.+GSEA	RF	0.951	0.750	Lasso	2563	P.+GSEA	RF	0.938	0.750
	Tree B.	3	Gene	KNN*	0.943	0.875	Lasso	55	Probe.	RF*	0.938	0.625
	ReliefF	78	GSEA	XGB*	0.943	0.750	Lasso	994	G.+GSEA	KNN*	0.925	0.625
	ReliefF	78	P.+GSEA	XGB*	0.943	0.750	Lasso	2656	GSEA	RF*	0.893	0.625
H1N1 DEE3	mRMR	35	G.+GSEA	KNN	1.000	1.000	Fisher S.	7	G.+GSEA	RF	1.000	1.000
	ReliefF	28	Gene	RF	1.000	1.000	Tree B.	1	GSEA	SVM	1.000	1.000
	ReliefF	15	GSEA	XGB	1.000	1.000	Fisher S.	7	P.+GSEA	RF	1.000	1.000
	ReliefF	15	P.+GSEA	XGB	1.000	1.000	Fisher S.	224	Probe.	GNB	1.000	1.000
	Fisher S.	438	Probe.	LR	1.000	1.000	Fisher S.	16	Gene	NuSVC*	1.000	0.833
H1N1 DEE4	ReliefF	4	Gene	LR	1.000	1.000	ReliefF	4	Gene	GNB	1.000	0.857
	Tree B.	2	GSEA	RF	1.000	1.000	Tree B.	2	GSEA	NuSVC	1.000	0.857
	ReliefF	66	G.+GSEA	KNN*	0.917	0.857	ReliefF	5083	G.+GSEA	KNN	0.917	0.857
	ReliefF	66	P.+GSEA	KNN*	0.917	0.857	ReliefF	4874	P.+GSEA	KNN	0.917	0.857
	ReliefF	88	Probe.	XGB	0.917	0.857	ReliefF	215	Probe.	XGB	0.833	0.857
HRV DUKE	Tree B.	6	GSEA	XGB	1.000	1.000	Tree B.	4	GSEA	SVM*	1.000	1.000
	Lasso	39	Gene	LR	1.000	0.875	Tree B.	5	P.+GSEA	XGB	1.000	1.000
	Lasso	1573	P.+GSEA	LR	1.000	0.875	Tree B.	5	Probe.	XGB	1.000	1.000
	Lasso	1551	G.+GSEA	LR	0.944	0.875	Tree B.	4	G.+GSEA	SVM*	1.000	0.625
	Tree B.	4	Probe.	LR	0.944	0.875	Tree B.	4	Gene	SVM*	1.000	0.625
HRV UVA	Tree B.	3	GSEA	NuSVC	1.000	1.000	Lasso	3237	P.+GSEA	XGB*	1.000	1.000
	Lasso	41	Probe.	RF	1.000	1.000	mRMR	363	Probe.	KNN*	1.000	1.000
	Lasso	27	Gene	RF	0.958	0.800	Tree B.	3	GSEA	RF*	1.000	0.800
	Tree B.	3	P.+GSEA	RF	0.903	0.800	ReliefF	5799	G.+GSEA	XGB	1.000	0.600
	Lasso	3244	G.+GSEA	XGB	0.903	0.600	Lasso	27	Gene	RF	0.903	0.800

In other words, proposed GSEA based representation was not made any improvement in predicting the presence of symptoms, especially in the pre-infection and

early post-infection periods. On the other hand, when feature selection was applied, GSEA-based approaches showed improvement. For example, at T.0, features derived from enrichment scores achieved the highest performance in all sub-experiments of DEE1, DEE2, DEE3, DEE5 and DUKE. This result suggests that feature selection, i.e. enrichment scores of predefined gene sets, is more useful in predicting symptomatic subjects for the sub-experimental problems.

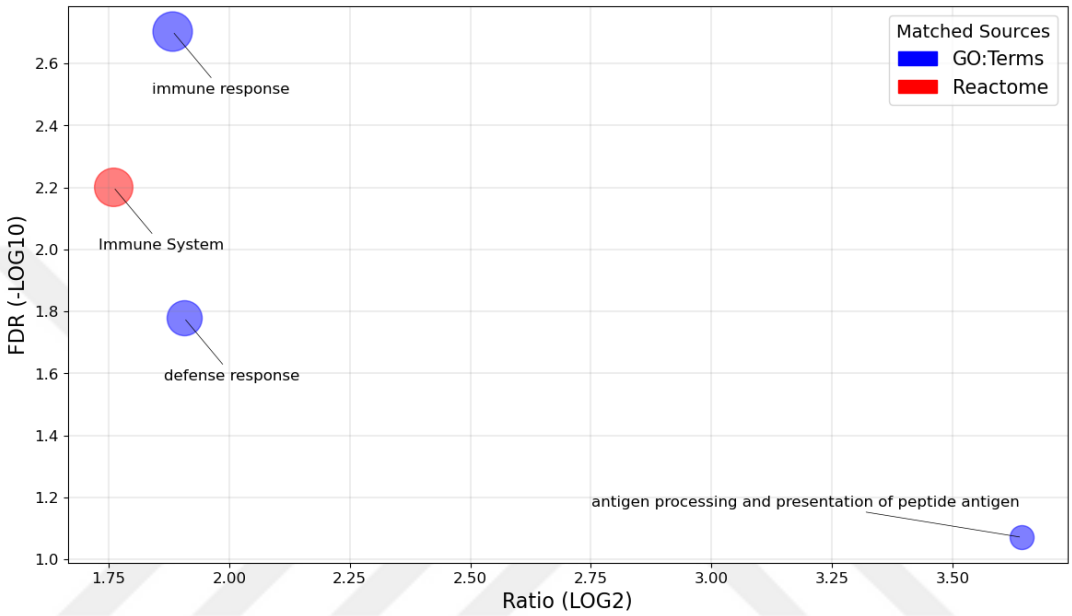

Figure 4.16 Overrepresented Pathways and GO Terms on the mostly selected genes in symptom development prediction problem.

In feature selection applied results, same as the infection prediction problem, Tree-Based, ReliefF and Lasso methods have emerged as top-performing approaches. In particular, the tree-based approach has achieved 100% accuracy using only 3-4 features at some time points/sub-experiments (e.g. DEE1 T.0, DEE2 T.24). The infection prediction results had demonstrated that the tree-based approach is highly effective in delivering high performance with a minimal number of features. However, other methods have also exhibited strong results with a small number of features in predicting symptomatic subjects; for example, in T.0 prediction in the DUKE sub-experiment, the ReliefF method achieved outstanding performance with only 3 features - highlighting that a small set of expressed genes may be sufficient for accurate symptom prediction. The analysis of genes with more than 4 occurrences at all time points aligns closely with the findings in the infection prediction problem, as illustrated in Figure 4.16. It is evident once again that genes associated with the immune system significantly contribute to

improved classification success when selected frequently. The GO term "antigen processing and presentation of peptide antigen" refers to the complex biological process where cells present peptide antigens a key part of the cellular immune response.

To summarize the experiment-based results for both problems collectively, it's evident that models have consistently and accurately predicted all individuals infected with the virus as well as those who developed symptoms. Furthermore, the GSEA-based representation type also achieved satisfactory performance for each sub-experimental dataset. When examining the impact of feature selection methods, it becomes clear that they are particularly beneficial for both prediction problems, especially in terms of achieving high predictive accuracy with a very small number of features. In addition, performing the ORA on the most frequently occurring genes has revealed a direct impact of genes related to the “Immune System” on the prediction performance.

Table 4.22 Average results for infection prediction of best models according to Feature Representation types.

Feature	Classifier	FS Method	NF	AUPRC	ACC	AUROC
G.+GSEA	GNB	Lasso	1731.4	0.856	0.727	0.658
P.+GSEA	GNB	Lasso	1457.2	0.849	0.723	0.639
GSEA	GNB	Lasso	1608.9	0.848	0.705	0.638
Probe	GNB	-	22277.0	0.847	0.712	0.626
Gene	SVM*	mRMR	37.4	0.844	0.655	0.718

Table 4.23 Average results for symptomatic prediction of best models according to Feature Representation types.

Feature	Classifier	FS Method	NF	AUPRC	ACC	AUROC
G.+GSEA	RF	-	49144.0	0.821	0.716	0.780
P.+GSEA	RF*	Lasso	1737.8	0.817	0.686	0.771
GSEA	LR*	-	36834.0	0.812	0.659	0.775
Probe	LR*	-	22277.0	0.808	0.640	0.761
Gene	KNN*	-	12310.0	0.801	0.682	0.743

Notwithstanding the usefulness of separate models, different combinations stand out for either symptom development or infection prediction depending on the time point/sub-experiment pair, as each is interpreted separately. Hence, averages of each method combination were computed to assess the generalizability of approaches. While calculating average metrics, the class probability distributions of each sample in different sub-experiments by each method combination were unified. This yielded average results for each sub-experiment; however, since our sub-datasets also had another dimension,

time point, results from 6-time points were averaged as well. Tables 4.22 and Table 4.23 respectively show the average metrics of method combinations for infection prediction and symptom prediction. In both prediction tasks, it is observed that GSEA-based approaches generally demonstrate better performance when averages are considered. Moreover, in infection prediction, the Gaussian Naïve Bayes algorithm stands out as the classifier, while the Lasso approach emerges as the preferred method for feature selection. The primary reason for the Bayes algorithm's superior performance is the small sample size used in the sub-experiments. Typically, machine learning algorithms tend to learn patterns in the data more effectively as the sample size increases. For analyses with limited data, probabilistic models are recommended. Given the class independence assumption, naive Bayes classifiers are able to efficiently utilize high dimensional features even with limited training data compared to more advanced methods. Hence, even though different combinations seem to be better when considered individually for time points and sub-experiments, the evaluation of the combinations by the average of the experiments as well as the time points shows the advantage of the Naive Bayes algorithm in infection prediction.

4.2.3.2 Results for Virus-Merge-Based Models

Table 4.24 The results of the best-performing models according to feature representation type for each virus-merged subset at time points T.0 and T.24 on the infection prediction task. An asterisk (*) indicates that the hyper-parameters were not optimized. NF column shows the number of used features after the feature selection methods.

Exp.	TimePoint 0						TimePoint 24					
	FS	NF	Feature	Clf.	AUPRC	ACC	FS	NF	Feature	Clf.	AUPRC	ACC
H1N1	Fisher S.	15	G.+GSEA	NuSVC	1.000	0.846	-	12310	Gene	XGB	0.952	0.846
	Fisher S.	14	GSEA	NuSVC	1.000	0.846	Lasso	57	Probe	XGB*	0.926	0.538
	Fisher S.	23	P.+GSEA	NuSVC	1.000	0.846	Tree B.	16	GSEA	NuSVC*	0.919	0.769
	Lasso	57	Probe	LR	0.951	0.769	-	49144	G.+GSEA	XGB	0.919	0.846
	ReliefF	67	Gene	XGB*	0.943	0.846	Fisher S.	937	P.+GSEA	NuSVC	0.905	0.769
H3N2	mRMR	7	Gene	XGB*	0.950	0.615	ReliefF	5	Gene	KNN	0.958	0.846
	Lasso	2329	P.+GSEA	KNN*	0.938	0.615	-	22277	Probe	KNN	0.950	0.538
	Tree B.	12	G.+GSEA	GNB	0.905	0.615	Lasso	1218	P.+GSEA	LR	0.917	0.846
	Tree B.	10	Probe	GNB	0.902	0.692	Lasso	1271	GSEA	LR	0.917	0.769
	Lasso	2627	GSEA	KNN	0.897	0.538	Lasso	1173	G.+GSEA	LR*	0.909	0.846
HRV	ReliefF	37	G.+GSEA	KNN*	0.886	0.692	Lasso	57	Gene	XGB*	0.958	0.692
	ReliefF	37	GSEA	KNN*	0.886	0.692	Tree B.	25	P.+GSEA	XGB*	0.919	0.615
	ReliefF	37	P.+GSEA	KNN*	0.886	0.692	Fisher S.	6	GSEA	XGB	0.918	0.615
	ReliefF	46	Gene	KNN*	0.872	0.615	Fisher S.	7	G.+GSEA	XGB*	0.915	0.692
	ReliefF	24	Probe	KNN	0.872	0.615	Fisher S.	16	Probe	SVM	0.906	0.615

Table 4.25 The results of the best-performing models according to feature representation type for each virus-merged subset at time points T.48 and T.72 on the infection prediction task. An asterisk (*) indicates that the hyper-parameters were not optimized. NF column shows the number of used features after the feature selection methods.

Exp.	TimePoint 48						TimePoint 72					
	FS	NF	Feature	Clf.	AUPRC	ACC	FS	NF	Feature	Clf.	AUPRC	ACC
H1N1	ReliefF	156	Gene	SVM	0.918	0.615	-	22277	Probe	XGB	0.919	0.615
	ReliefF	6	GSEA	XGB	0.900	0.692	-	59111	P.+GSEA	XGB*	0.880	0.615
	ReliefF	6	P.+GSEA	XGB	0.887	0.692	Lasso	4994	GSEA	GNB	0.875	0.769
	ReliefF	6	G.+GSEA	LR	0.874	0.615	Lasso	1979	G.+GSEA	GNB	0.850	0.769
	Lasso	58	Probe	LR	0.863	0.615	Fisher S.	151	Gene	GNB	0.850	0.692
H3N2	Tree B.	9	G.+GSEA	LR	1.000	0.846	Lasso	23	Gene	GNB	1.000	1.000
	Lasso	21	Gene	KNN*	0.988	0.923	Lasso	37	Probe	LR*	1.000	1.000
	ReliefF	49	GSEA	RF	0.985	0.846	ReliefF	271	G.+GSEA	KNN	0.988	0.923
	ReliefF	49	P.+GSEA	RF	0.985	0.846	ReliefF	271	GSEA	KNN	0.988	0.923
	-	22277	Probe	KNN	0.966	0.538	ReliefF	271	P.+GSEA	KNN	0.988	0.923
HRV	Tree B.	21	Gene	LR*	0.921	0.769	Lasso	47	Probe	RF*	0.912	0.769
	Lasso	45	Probe	SVM	0.899	0.692	Tree B.	18	P.+GSEA	XGB	0.899	0.769
	-	59111	P.+GSEA	RF	0.863	0.538	-	12310	Gene	KNN	0.885	0.692
	-	49144	G.+GSEA	XGB	0.854	0.692	Tree B.	19	GSEA	KNN*	0.865	0.615
	ReliefF	6	GSEA	KNN	0.848	0.692	mRMR	100	G.+GSEA	KNN*	0.860	0.692

Table 4.26 The results of the best-performing models according to feature representation type for each virus-merged subset at time points T.96 and T.120 on the infection prediction task. An asterisk (*) indicates that the hyper-parameters were not optimized. NF column shows the number of used features after the feature selection methods.

Exp.	TimePoint 96						TimePoint 120					
	FS	NF	Feature	Clf.	AUPRC	ACC	FS	NF	Feature	Clf.	AUPRC	ACC
H1N1	ReliefF	8	Probe	XGB	0.939	0.692	Tree B.	11	Gene	GNB	0.875	0.692
	-	12310	Gene	GNB	0.875	0.769	-	22277	Probe	GNB	0.875	0.769
	-	49144	G.+GSEA	LGB	0.846	0.692	Tree B.	13	G.+GSEA	GNB	0.848	0.692
	-	36834	GSEA	LGB	0.846	0.692	-	36834	GSEA	LGB	0.846	0.692
	-	59111	P.+GSEA	LGB	0.846	0.692	-	59111	P.+GSEA	LGB	0.846	0.692
H3N2	mRMR	7	Gene	LR	1.000	1.000	Lasso	29	Probe	LR	1.000	1.000
	Lasso	16	Probe	SVM	1.000	1.000	Lasso	17	Gene	XGB	1.000	0.923
	-	49144	G.+GSEA	LR	0.985	0.923	Fisher S.	2	G.+GSEA	LR	0.985	0.923
	-	36834	GSEA	LR	0.985	0.923	-	36834	GSEA	LR	0.985	0.923
	-	59111	P.+GSEA	LR	0.985	0.923	-	59111	P.+GSEA	LR	0.985	0.923
HRV	-	59111	P.+GSEA	XGB*	0.919	0.692	ReliefF	117	G.+GSEA	NuSVC	0.979	0.846
	ReliefF	16	Probe	GNB	0.915	0.615	ReliefF	118	P.+GSEA	NuSVC	0.971	0.769
	-	36834	GSEA	RF	0.907	0.692	ReliefF	116	GSEA	KNN	0.969	0.846
	Fisher S.	54	G.+GSEA	KNN*	0.858	0.692	ReliefF	19	Probe	GNB	0.934	0.615
	Tree B.	24	Gene	NuSVC	0.857	0.538	Lasso	62	Gene	SVM	0.904	0.692

The prediction results obtained from the combined training and test samples of sub-experiments according to same viruses are shown in Tables 4.24 to 4.26 for infection

prediction problem. In contrast to the experiment-based analysis, the results obtained from the models with and without feature selection are presented in the same tables, categorized by time point and virus type. When examining the results of infection prediction without feature selection, it is observed that GSEA-based approaches give the best results for all three virus types on the prediction of pre-infection (T.0). Conversely, no prominent feature representation types stand out at other time points. Furthermore, the superior performance of the GSEA-based approaches continued after feature selection was applied to the prediction of T.0 time points, except for the best performing model for the H3N2 virus. It is noteworthy that, the prediction performance for samples associated with H1N1 experiments tends to decrease up to T.72 time point in almost all models. However, the highest performance in predicting H1N1 infected samples is obtained by utilizing the ssGSEA-based representation at the pre-exposure time point (T0), with an AUPRC value of 1. At no other time point of the H1N1 predictions could an AUPRC value of 1 be achieved. On the H3N2 virus related results, the predictive performance is increasing steadily after the exposure of virus. The best performing models always achieved an AUPRC of 1, especially 48 hours after exposure. In the case of HRV virus, an AUPRC value of around 0.92 was obtained except for the T.0 time point.

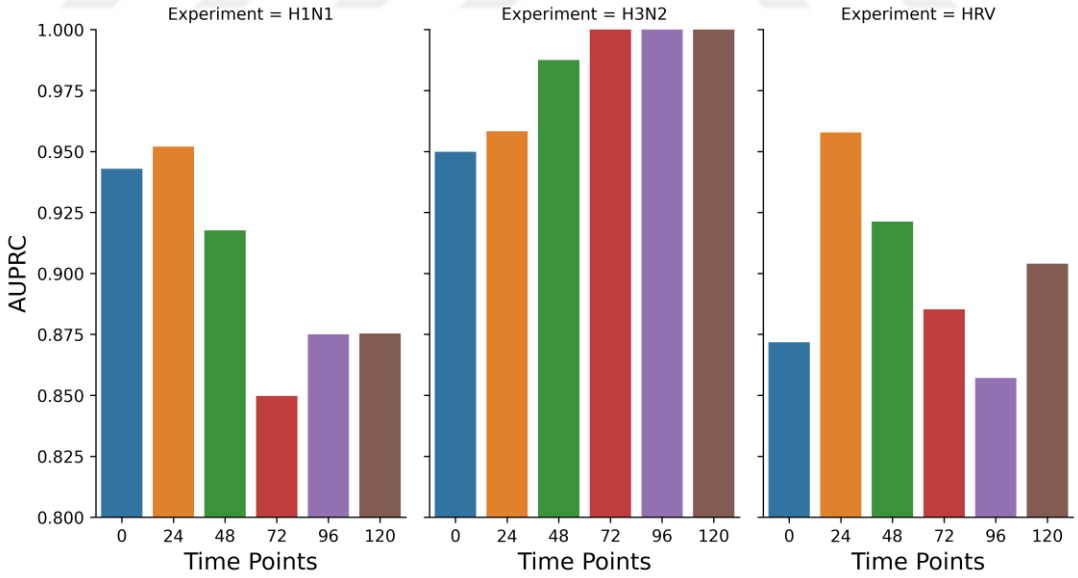

Figure 4.17 Prediction performance of the best performing gene-level representation used models according to the time points and virus types.

In the original article and the reported information on the dataset, there are also onset times at which volunteers became infected after exposure, depending on the experiment. According to reports, H1N1-injected volunteers were infected between 24-

48 hours, H3N2-injected volunteers were infected between 30-48 hours, and HRV-injected volunteers were infected 24-48 hours after exposure on average. Figure 4.17 shows the predictive performance of the best performing gene level expression-used models based on time points for each respiratory virus. When compared prediction models to the time points at which individuals become infected or feel symptoms, it is seen that there is a slight relation between them. For instance, HRV models demonstrated peak performance at the T.24 time point, which aligns with volunteers typically exhibiting symptoms 24-48 hours after exposure to HRV virus. This association suggests that machine learning models can predict changes in gene expression values just before symptoms appear. As a further investigation, selected features were extracted from each of the best performing models of expression values, i.e. genes and probes, according to virus types. For this purpose, the top-15 most frequently selected genes were identified using the best-performing models for each virus-experiment at the gene and probe-level. The frequency of occurrence of these genes in each virus-experiment was then extracted. Figure 4.18 represent frequency of each mostly selected genes according to virus types.

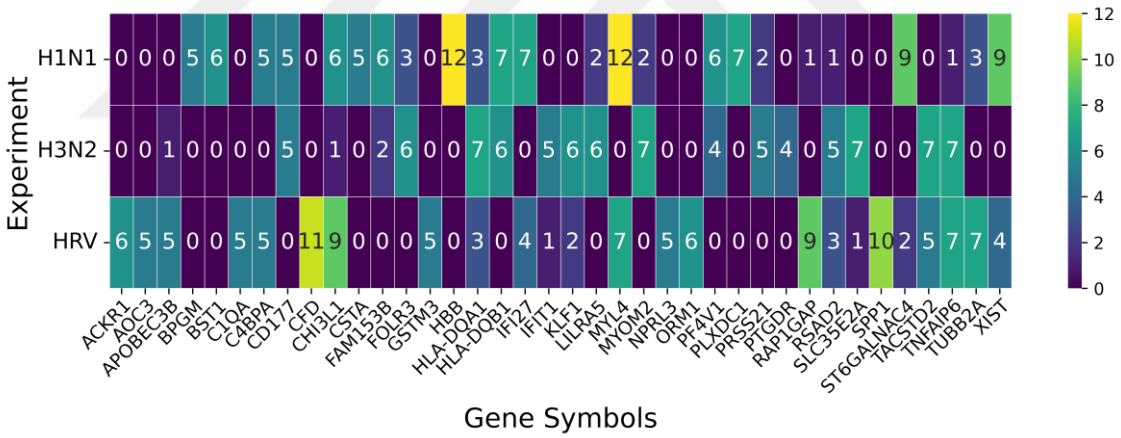

Figure 4.18 Frequency of the most frequently selected top 15 genes according to different virus-based experiments. In order to calculate frequencies of each gene, genes in which mostly selected in both probe- and gene-level representation models were taken into account.

As can be seen from the frequency of occurrence, no specific gene(s) was selected frequently in all 3 virus types. Nevertheless, some genes have an occurrence in 2 experiments. For instance, the gene “MYL4” was selected 12 and 7 times in H1N1 and HRV experiments, respectively. Genes of the HLA family (HLA-DQA1, HLA-DQB1) were selected for both the H3N2 and H1N1 viruses. In the context of influenza viruses (H1N1, H3N2), HLA class II molecules play a crucial role in presenting viral antigens to

CD4 T cells, which are essential for an effective immune response generation. Hence, these genes have the potential to influence the immune response to influenza during natural infection as well as vaccination [226]. The gene TNFAIP6 was selected 7 times in both HRV and H3N2 experiments. TNFAIP6 is thought to be one of the genes strongly induced by HRV, leading to high expression of TNFAIP6 in nasal secretions as part of the inflammatory and immune response to HRV infection [223]. Complement factor D (the gene CFD), frequently selected in HRV experiments, is one of the critical elements of the alternative complement pathway, which is such an important part of the innate immune system for host defense against pathogens [227]. The association between selected identified genes and the immune system had been confirmed previously by an overrepresentation analysis in sub-experiment-based analyses.

A similar analysis was performed on the mostly selected genes (see gene symbols in Figure 4.14) obtained from virus-based experiments. Consequently, “defense response (GO:0006952)”, “leukocyte-mediated immunity (GO:0002443)”, “immune effector process (GO:0002252)”, “immune response (GO:0006955)”, “inflammatory response (GO:0006954)”, “innate immune response (GO:0045087)” from the GO terms and the pathways “immune system (R-HSA-168256)”, “Neutrophil degranulation (R-HSA-6798695)” from the Reactome database are significantly enriched. Thus, the association between the immune system and predicting infection continues when different experiments are combined, as evidenced by the results.

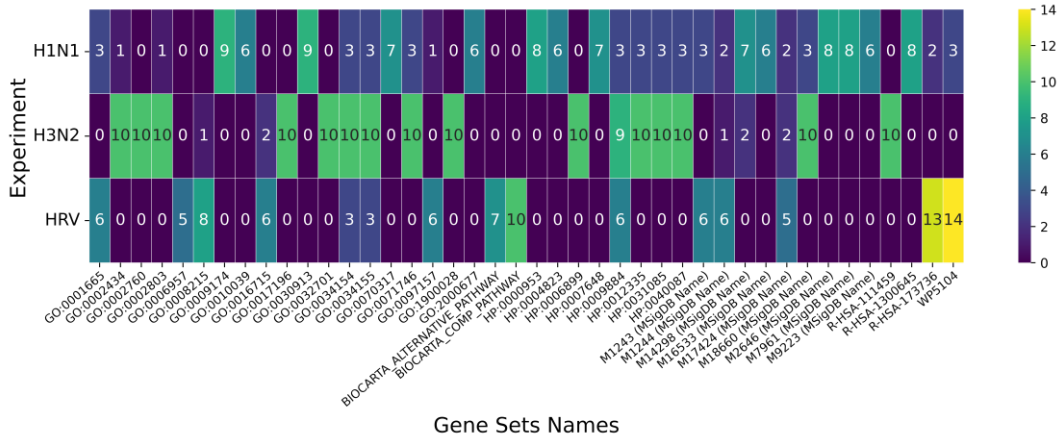

Figure 4.19 Frequency of the most frequently selected top 15 pathways and gene sets according to different virus-based experiments.

In our experiments, we also performed frequency analysis on the gene sets selected by best-performing GSEA-based representation types, as represented in Figure 4.19. The

MSigDB database is a comprehensive repository containing curated gene sets from well-known databases such as KEGG, Reactome, and Gene Ontology as well as gene sets from studies published in the literature. The gene sets in the figure labeled with “MSigDB Name” are from publications. Other prefixes used in the figure are Gene Ontology (GO), Human Phenotype Ontology (HP), Reactome Database (R-HSA), and WikiPathways (WP).

Another point to consider with the selected genes & gene sets is the relations between them. For example, the frequently selected gene set HP:0004823 of the H1N1 experiment contains the HBB gene, which was selected most frequently gene in the same experiment. In the H3N2 experiment, the gene set labelled “HP:0012335” (called as Abnormality of folate metabolism on the source repository) was selected 10 times. When this gene set was analyzed, it was found to contain HLA-DQA1 and HLA-DQB1, which are among the most frequently selected genes in H3N2 experiments. In the HRV-related experiment, CFD was the most frequently selected gene. This gene is included in the following pathways: “Acquired partial lipodystrophy Barraquer Simons syndrome pathway (WP5104)”, “Reactome alternative complement activation pathway (R-HSA-173736)”, and “Biocarta alternative complement pathway”. These pathways are also the most frequently selected gene sets in the same experiment. The fact that these gene sets contain the CFD gene might be related to the improvement in predictability of GSEA-based representations that select these gene sets. Therefore, differentiation in the expression value of the CFD gene could be one of the most discriminating factors in predicting whether individuals are exposed to the HRV virus.

Tables 4.27 to 4.29 indicate the models' results for predicting symptom development. Unlike the infection prediction problem, no virus type or time point achieved an AUPRC value of 1 before time point T.72 except the best model of the H3N2 experiment. Nevertheless, the majority of the models delivered strong predictive performance, exceeding an AUPRC of 0.94. Among the feature selection methods, the mRMR approach shows an improvement in this prediction problem. In addition, the Lasso and ReliefF methods stand out slightly more than the others.

Contrary to the infection prediction problem, probe-level representation achieved better performance at some time points than other types of representation. For example, the highest AUPRCs for H1N1 and H3N2 viruses at time point T.24 was obtained by

probe-level representation. On the other hand, GSEA-based approaches have generally achieved better performance than other methods, especially after the T.48 time point.

Table 4.27 The results of the best-performing models according to feature representation type for each virus-merged subset at time points T.0 and T.24 on the symptom prediction task. An asterisk (*) indicates that the hyper-parameters were not optimized. NF column shows the number of used features after the feature selection methods.

Exp.	TimePoint 0						TimePoint 24					
	FS	NF	Feature	Clf.	AUPRC	ACC	FS	NF	Feature	Clf.	AUPRC	ACC
H1N1	Fisher S.	65	G.+GSEA	XGB	0.941	0.769	-	22277	Probe	LR	0.955	0.769
	mRMR	22	Gene	RF*	0.920	0.923	-	49144	G.+GSEA	RF	0.944	0.769
	Lasso	973	P.+GSEA	KNN	0.892	0.769	Lasso	861	GSEA	LR	0.944	0.846
	Fisher S.	17	Probe	RF*	0.876	0.769	Lasso	308	P.+GSEA	LR	0.930	0.692
	Lasso	492	GSEA	RF*	0.856	0.769	-	12310	Gene	LR	0.925	0.769
H3N2	mRMR	17	Gene	KNN*	0.984	0.846	mRMR	31	Probe	SVM	0.981	0.846
	Tree B.	13	Probe	KNN	0.924	0.846	Lasso	4362	P.+GSEA	KNN	0.968	0.846
	Lasso	1643	GSEA	KNN	0.921	0.615	Fisher S.	15	GSEA	GNB	0.955	0.769
	Lasso	1551	P.+GSEA	XGB	0.886	0.769	Fisher S.	19	G.+GSEA	GNB	0.933	0.769
	Tree B.	10	G.+GSEA	XGB	0.883	0.846	-	12310	Gene	LR	0.928	0.769
HRV	Lasso	5086	P.+GSEA	RF*	0.940	0.692	Lasso	3466	GSEA	XGB*	0.966	0.769
	ReliefF	38	GSEA	LR	0.938	0.846	-	22277	Probe	NuSVC	0.944	0.538
	ReliefF	22	Gene	RF*	0.935	0.692	mRMR	30	Gene	LR	0.923	0.846
	ReliefF	38	G.+GSEA	LR	0.916	0.846	Lasso	3489	P.+GSEA	SVM	0.923	0.692
	-	22277	Probe	LGB	0.769	0.538	Lasso	3343	G.+GSEA	NuSVC*	0.916	0.615

Table 4.28 The results of the best-performing models according to feature representation type for each virus-merged subset at time points T.48 and T.72 on the symptom prediction task. An asterisk (*) indicates that the hyper-parameters were not optimized. NF column shows the number of used features after the feature selection methods.

Exp.	TimePoint 48						TimePoint 72					
	FS	NF	Feature	Clf.	AUPRC	ACC	FS	NF	Feature	Clf.	AUPRC	ACC
H1N1	Lasso	539	G.+GSEA	RF	0.988	0.923	Fisher S.	6	G.+GSEA	LR	1.000	1.000
	-	22277	Probe	LR	0.955	0.769	Fisher S.	6	GSEA	LR	1.000	1.000
	Lasso	731	GSEA	RF*	0.944	0.769	Fisher S.	7	P.+GSEA	LR	1.000	1.000
	-	12310	Gene	NuSVC*	0.941	0.846	Tree B.	17	Gene	NuSVC	0.941	0.923
	Lasso	666	P.+GSEA	NuSVC*	0.930	0.923	Fisher S.	63	Probe	KNN	0.921	0.846
H3N2	Lasso	4505	P.+GSEA	XGB*	1.000	0.769	ReliefF	40	Gene	XGB	1.000	0.923
	Lasso	1535	G.+GSEA	XGB*	0.991	0.846	Lasso	4636	P.+GSEA	XGB	1.000	0.923
	ReliefF	1399	Gene	KNN*	0.948	0.769	mRMR	23	GSEA	XGB	0.966	0.846
	Fisher S	13	GSEA	LR	0.945	0.692	-	49144	G.+GSEA	KNN	0.960	0.923
	-	22277	Probe	KNN*	0.936	0.846	-	22277	Probe	KNN	0.950	0.923
HRV	Lasso	3759	P.+GSEA	NuSVC*	0.916	0.692	-	49144	G.+GSEA	LR*	0.945	0.769
	mRMR	137	G.+GSEA	KNN	0.915	0.692	ReliefF	33	Gene	LR	0.945	0.769
	-	36834	GSEA	NuSVC*	0.889	0.538	Lasso	2624	P.+GSEA	NuSVC	0.945	0.692
	Lasso	41	Gene	XGB*	0.885	0.615	Lasso	2676	GSEA	SVM	0.938	0.846
	ReliefF	470	Probe	LR	0.877	0.538	-	22277	Probe	NuSVC*	0.938	0.538

Table 4.29 The results of the best-performing models according to feature representation type for each virus-merged subset at time points T.96 and T.120 on the symptom prediction task. An asterisk (*) indicates that the hyper-parameters were not optimized. NF column shows the number of used features after the feature selection methods.

Exp.	TimePoint 96						TimePoint 120					
	FS	**NF**	**Feature**	**Clf.**	**AUPRC**	**ACC**	**FS**	**NF**	**Feature**	**Clf.**	**AUPRC**	**ACC**
H1N1	mRMR	54	Gene	KNN*	1.000	1.000	Lasso	627	P.+GSEA	RF	0.965	0.846
	mRMR	28	G.+GSEA	LR	1.000	0.923	Fisher S.	35	Probe	KNN*	0.962	0.923
	Fisher S.	3	GSEA	LR*	1.000	0.846	-	36834	GSEA	LR*	0.955	0.846
	Fisher S.	3	P.+GSEA	LR*	1.000	0.846	ReliefF	2	Gene	XGB	0.951	0.846
	Lasso	21	Probe	XGB*	1.000	0.538	-	49144	G.+GSEA	KNN*	0.944	0.846
H3N2	mRMR	43	Gene	XGB	1.000	0.923	mRMR	13	Probe	XGB	1.000	0.923
	Lasso	4913	G.+GSEA	XGB*	1.000	0.846	ReliefF	29	G.+GSEA	XGB*	1.000	0.846
	ReliefF	146	P.+GSEA	XGB	1.000	0.769	mRMR	23	Gene	XGB	1.000	0.846
	-	36834	GSEA	XGB*	0.981	0.923	ReliefF	29	GSEA	XGB*	1.000	0.846
	-	22277	Probe	NuSVC	0.981	0.769	ReliefF	29	P.+GSEA	XGB*	1.000	0.846
HRV	ReliefF	5265	GSEA	NuSVC	0.966	0.692	Lasso	83	Gene	GNB	0.959	0.846
	ReliefF	9129	G.+GSEA	LR*	0.945	0.846	Fisher S.	22	G.+GSEA	SVM*	0.945	0.769
	ReliefF	4087	P.+GSEA	LR	0.945	0.846	Lasso	2754	P.+GSEA	SVM	0.945	0.692
	ReliefF	530	Gene	SVM	0.923	0.846	-	22277	Probe	KNN*	0.940	0.615
	-	22277	Probe	NuSVC*	0.916	0.615	Tree B.	23	GSEA	XGB	0.935	0.692

We also identified frequently selected genes by virus type and used them as input for ORA. The most frequently selected genes in the H1N1 experiment were C1orf115 and DDX3X, in the H3N2 experiment were UPF3A and RUBCNL. These genes were not found or observed in previous analyses like infection prediction. On the other hand, the following genes became prominent in the HRV experiment; MPZL1, HLA-DQA1, HLA-DQB1, and DEFA4. Moreover, no significant results were obtained for genes other than those selected for the HRV virus experiments. Significant genes from the HRV experiments were enriched in gene sets related to the immune system, which is consistent with our infection prediction results.

4.2.3.3 Results for All-Merge-Based Models

The final experiment group in the respiratory analysis is “ALL”, where all samples from the sub-experiments are merged into a single set. Our main dataset, GSE73072, consists of 7 datasets related to 4 respiratory viruses derived from different challenges. Therefore, performing machine learning and further analyses on this combination of datasets would provide comprehensive findings and associations regarding the generalization of respiratory virus prediction with machine learning.

Table 4.30 The results of the best-performing models according to feature representation type for each virus-merged subset at time points

FS	NF	Feature	Clf.	AUPRC	ACC	FS	NF	Feature	Clf.	AUPRC	ACC
Infection Prediction Time Point 0						**Symptom Develop Prediction Time Point 0**					
ReliefF	311	Gene	SVM*	0.897	0.659	mRMR	48	G.+GSEA	KNN	0.820	0.705
ReliefF	36	G.+GSEA	RF*	0.884	0.727	mRMR	27	P.+GSEA	LGB	0.802	0.636
ReliefF	36	GSEA	RF*	0.884	0.727	mRMR	277	GSEA	LGB*	0.790	0.636
ReliefF	36	P.+GSEA	RF*	0.884	0.727	Lasso	215	Gene	LGB*	0.752	0.545
Tree B.	88	Probe	XGB	0.877	0.659	Lasso	164	Probe	LGB*	0.746	0.659
Infection Prediction Time Point 24						**Symptom Develop Prediction Time Point 24**					
Tree B.	85	Probe	RF*	0.897	0.727	mRMR	27	P.+GSEA	SVM	0.809	0.682
Lasso	7843	GSEA	XGB	0.892	0.705	Lasso	7370	GSEA	LR*	0.801	0.659
ReliefF	4602	G.+GSEA	LGB	0.882	0.682	Lasso	7006	G.+GSEA	SVM	0.799	0.682
Lasso	178	Gene	KNN*	0.879	0.682	-	12310	Gene	KNN*	0.798	0.591
-	59111	P.+GSEA	LR*	0.869	0.659	ReliefF	706	Probe	SVM*	0.764	0.568
Infection Prediction Time Point 48						**Symptom Develop Prediction Time Point 48**					
Lasso	7850	G.+GSEA	RF*	0.858	0.705	-	36834	GSEA	LR	0.825	0.705
Lasso	7826	P.+GSEA	RF*	0.852	0.705	-	59111	P.+GSEA	LR*	0.820	0.750
-	22277	Probe	RF	0.840	0.705	ReliefF	402	Gene	SVM*	0.816	0.659
Lasso	98	Gene	XGB*	0.840	0.614	ReliefF	2017	Probe	SVM	0.813	0.682
Lasso	6894	GSEA	LR*	0.838	0.659	Tree B.	81	G.+GSEA	SVM*	0.809	0.795
Infection Prediction Time Point 72						**Symptom Develop Prediction Time Point72**					
Tree B.	82	G.+GSEA	LR*	0.878	0.773	Tree B.	76	G.+GSEA	SVM	0.935	0.750
Fisher S.	121	Gene	NuSVC	0.872	0.659	ReliefF	120	Gene	LR*	0.908	0.727
mRMR	155	GSEA	XGB	0.868	0.682	Fisher S.	8	P.+GSEA	LR	0.903	0.750
-	59111	P.+GSEA	LR*	0.864	0.750	Fisher S.	7	GSEA	LR	0.902	0.750
Tree B.	91	Probe	RF	0.840	0.727	Fisher S.	26	Probe	SVM	0.871	0.659
Infection Prediction Time Point 96						**Symptom Develop Prediction Time Point 96**					
Lasso	166	Probe	XGB	0.900	0.727	ReliefF	1076	P.+GSEA	SVM	0.934	0.727
ReliefF	16	G.+GSEA	XGB	0.889	0.727	-	49144	G.+GSEA	LR	0.934	0.705
ReliefF	16	GSEA	XGB	0.889	0.727	ReliefF	535	GSEA	SVM*	0.932	0.727
ReliefF	16	P.+GSEA	XGB	0.889	0.727	-	12310	Gene	NuSVC*	0.930	0.682
mRMR	12	Gene	LR	0.889	0.659	ReliefF	51	Probe	GNB	0.924	0.841
Infection Prediction Time Point 120						**Symptom Develop Prediction Time Point 120**					
Tree B.	67	P.+GSEA	KNN	0.876	0.727	-	36834	GSEA	XGB	0.950	0.773
-	49144	G.+GSEA	LR*	0.870	0.636	Lasso	6791	G.+GSEA	LR*	0.942	0.727
ReliefF	2425	GSEA	RF	0.866	0.705	ReliefF	3922	Probe	NuSVC*	0.939	0.682
ReliefF	22	Probe	RF	0.863	0.659	Lasso	6880	P.+GSEA	LR*	0.938	0.727
Lasso	181	Gene	RF	0.852	0.705	-	12310	Gene	NuSVC	0.933	0.705

Table 4.30 shows the performance results obtained on the "ALL" experiment group. As expected, models couldn't achieve better performance than virus-based and sub-experiment-based analyses. The reason we presume is that the merging of different virus-related sub-datasets leads to more diverse and complex patterns and hence models have difficulty to capture patterns effectively. Because the sub-dataset could have different

distributions and combining them may cause statistical bias and challenges in generalization. Nevertheless, almost all models obtained an adequate prediction result, nearly AUPRC value of 0.85. Although there is no significant increase in the prediction of infection over time, in predicting the development of symptom problem, models predictivity is increases with time after the exposure. Despite the fact that no specific representation approach that always obtain best results, GSEA-based approaches achieved the best AUPRC values in both prediction problems at time points T.48, T.72 and T.120. In addition, the best models were usually obtained when feature selection was applied to the datasets.,

In particular, the ReliefF and Lasso methods exhibited better performance than others in infection prediction problems. The tree-based method, on the other hand, could not achieve good performance compared to the results of the virus-based and sub-experiment-based methods. Since the data set used in the "All" experimental group combines 7 sub-datasets and the bootstrapping of the tree-based algorithms leads to the selection of a random subset from the data set, it is likely that the selected features are not significant with respect to the virus or the sub-experiment. Consequently, the tree-based feature selection may not be able to select discriminative features that correctly classify samples.

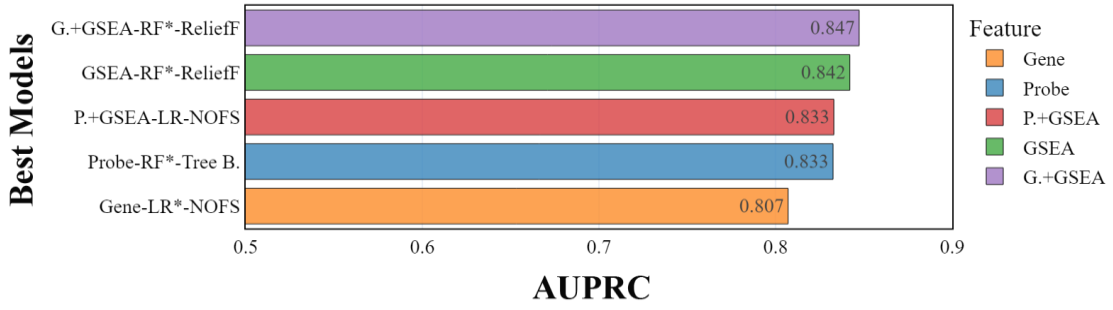

Figure 4.20 Average prediction performance of each model combination according to different feature representation types for infection prediction problem.

The best combination of classifiers, representation types, and feature selection according to the prediction problem was also determined by taking the average prediction of different time points. Figures 4.20 and 4.21 illustrate these results for infection prediction and symptomatic individual prediction, respectively. For example, the best of the gene representation-based combination, gene level expression values without feature selection + logistic regression classifier, could only achieve an AUPRC value of 0.807 as the worst result for infection prediction on average. On the contrary, the extending of

gene-level expression values with enrichment scores as features (G.+GSEA representation) achieved an AUPRC value of 0.849 when ReliefF and RF algorithms were utilized as feature selection and classifier, respectively.

The best model was followed by the GSEA representation, which solely relied on enrichment scores and showed a near-prediction performance with an AUPRC of 0.842. Integrating probe-level expression values with enrichment scores also achieved better performance than only gene-level and probe-level representation types. These findings suggest that the use of GSEA scores with/without the combination of expression values is promising for further improvement of prediction accuracy.

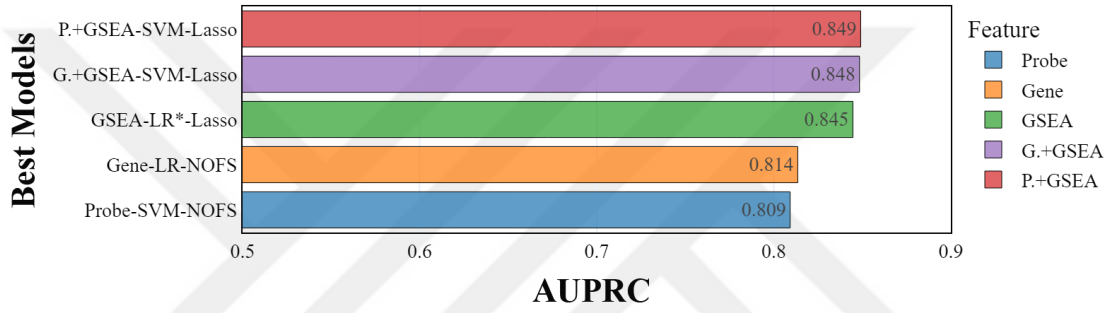

Figure 4.21 Average prediction performance of each model combination according to different feature representation types for symptom develop prediction problem.

Similar achievements of the GSEA-based representation types were also achieved in the prediction of symptomatic subjects. Combined feature representation of probe-level expression and enrichment scores (P.+GSEA) achieved an AUPRC value of 0.849 as the best-performing model. Closer predictivity performance was obtained by combining gene-level expression values and enrichment scores. These results for both prediction problems reflect that the use of enrichment scores, which allow an extended feature size, leads to improved generalization of prediction performance.

4.2.3.4 Comparison Results with Viral DREAM Challenge

A further analysis of the thesis is the comparison of the proposed methods with the results of the Viral DREAM Challenge. DREAM is a community-driven organization focused on advancing biomedical and systems biology research through crowdsourcing competitions. Competitions usually aim to address on a specific biomedical research question, narrowed down to a specific disease. As the competitions are open to

researchers around the world, a wide range of ideas and solutions can be presented. This allows for the most effective solution to the problem being sought [228].

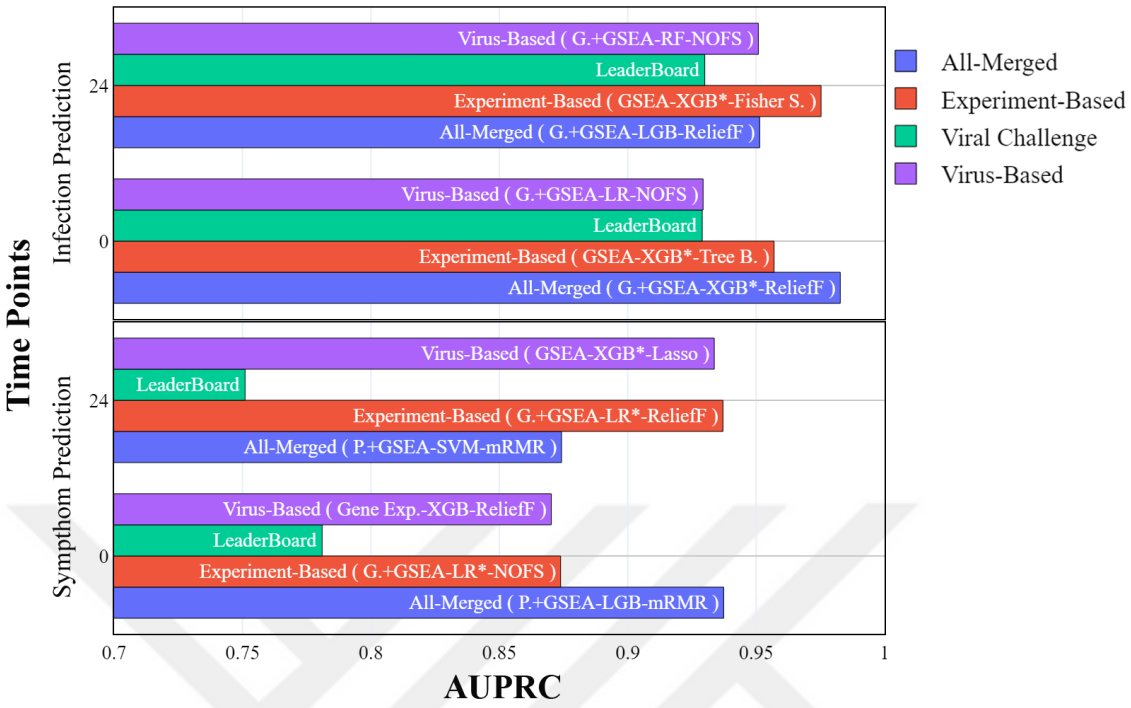

Figure 4.22 Comparison of the best performing models of different experimental groups (Experimental, Virus-based and All-Merged) with the winning results of the Viral DREAM Challenge according to different T.0 and T.24 time points.

The Respiratory Viral DREAM Challenge was one of these competitions which aimed to develop early predictors of respiratory susceptibility and infectivity based on pre-and post-exposure gene expression profiles [214]. Participants were expected to make predictions about viral shedding and the presence of symptoms before and 24 hours after exposure. According to the results of the leaderboard stage, the winner teams achieved an AUPRC of 0.9298 for predicting infection, while the prediction of symptomatic cases stuck around an AUPRC of 0.78. Furthermore, the heme metabolism pathway showed a strong relation with symptom development as a result of the enrichment analyses on the susceptible genes identified by participants. During the leaderboard stage of the challenge, probe-level expression of GSE73072 was used and testing samples were selected only from DEE4, DEE5, and HRV DUKE experiments. As explained detailly in the dataset section, we had also chosen the same testing subject from DEE4, DEE5 and HRV DUKE sub-experiment in our analysis.

Figure 4.22 illustrates the comparison of the best-performing models of each experiment group with the winning results on the Viral Challenge. In order to ensure a

fair comparison, we have recalculated the AUPRC scores keeping only the samples same as Viral Challenge testing samples, i.e. using only test samples derived from DEE4, DEE5, HRV DUKE.

In the viral challenge, only the pre-infection (T.0) and early post-infection (T.24) time points predictions were considered. Therefore, we filtered results by keeping only these two prediction points. In addition, each time point (i.e. T.0 and T.24) / prediction task (i.e. infection prediction and symptom presence prediction) pair was evaluated separately in the challenge, in other words, there was no expectation that a proposed model would be best for all time points and problem predictions. Therefore, we figured out only the best-performing models for each different experiment group (experiment-based, virus-based, all-sample merged).

The model combination G.+GSEA representation type with XGB classifier and ReliefF feature selection, which also trained with the merged of all-sub-dataset, achieved an outstanding AUPRC value of 0.983 on the infection prediction for the pre-infection period (Timepoint T.0). Moreover, the number of used features on this model was only 36. Considering that the winning model of the Viral Challenge used 22777 features (since it's probe-level expression values), it can be interpreted that our best model achieved a reasonably high score despite the small number of features. In the experiment-based analysis, where each sub-experiment dataset was trained and tested separately, the GSEA-based representation using only enrichment scores as features also outperformed the winning model of the leaderboard stage of the Viral Challenge, achieving an AUPRC value of 0.956.

Similar results are also observed for the early post-infection (T.24) prediction problem. While the best-performing model of the Viral challenge was stuck at an AUPRC of 0.93, our model consisting of GSEA features, XGB classifier, and Fisher score feature selection achieved an AUPRC of 0.975 when each sub-experiment was trained separately. This model was followed by the gene expression + GSEA-based representation, using a random forest classifier, with an AUPRC of 0.951.

On the second prediction task, the prediction of the presence of symptoms, the winning models of the challenge obtained 0.781 and 0.751 AUPRC values for the prediction of pre-infection and early post-infection time points, respectively. When these

results were compared with the infection prediction performance, it was clear that there was still room for improvement in the symptom prediction problem.

Our models, shown in Figure 4.21, significantly outperformed the results of the winning team in the leaderboard stage. Extended features of probe-level expression with enrichment scores (P.+GSEA) had achieved the highest score in symptom prediction with an AUPRC value of 0.9373 at T.0. In predicting the symptom development following the exposure, the gene expression + GSEA feature representation method achieved the highest predictivity of 0.9370 AUPRC value in the experiment-based group where each sub-experiment was independently trained. Other best-performing models for experiment groups such as virus-based (merging samples from identical viruses) and ALL (merging all samples), demonstrated significant improvement exceeding 10% with an AUPRC value of no less than 0.87.

Chapter 5

Conclusions and Future Prospects

5.1 Conclusions

This thesis has comprehensively investigated the predictive performance of machine learning methods on two disease types, genetic and infectious. These diseases are Behçet's disease and respiratory infections, respectively. For both disease types, data types containing individuals' genetic profiles (GWAS data and Gene Expression) were used as inputs, and disease presence predictions were compared using the most common machine learning methods. Furthermore, different approaches have been proposed by integrating the community or external information into disease prediction depending on the type of experiment. The general finding as a result of our experiments according to disease prediction are as follows:

- Although only two different experiments were conducted during the thesis, the high prediction performance was achieved despite the different types of input data. It can be concluded that machine learning methods are capable of handling individual genetic data and predicting the presence of disease.

- Although most of the genes selected as significant for disease prediction are compatible with in vivo or in vitro studies reported in the literature, some selected genes have never been reported to be associated with the related disease. This may be due to nature of machine learning and complex diseases. Complex diseases usually result from the contribution of multiple genomic variants and genes, due to fact that genes can interact with other genes, proteins, and pathways. Therefore, a minor value change in a gene which seemingly non-significant biologically may help to classify samples. This is because machine learning models learn by trying to classify input values optimally. In other words, they analyze relevant data (such as gene expression) statistically, rather than relying solely on domain knowledge

such as biological or genetic knowledge. Therefore, some genes that might not be considered significant based on in vivo analyses could play a crucial role in improving prediction accuracy.

- It can be noted that the use of external knowledge in the prediction phase has a positive effect in both experiments. In most machine learning based disease prediction studies, only clinical or omics data such as genomic, proteomic, and metabolomic data are used as input to represent the samples. Recently, an integrative approach combining multi-omics data of the samples has been proposed. However, there are many more domain-based biological studies related to infection and disease progression, including gene sets, pathways, and demographic and environmental factors. Our experiments have shown that this external information can be beneficial for predictive performance if somehow integrated into the model.

In the thesis, it would be more beneficial to interpret the findings separately for each disease since the experiments involve two different diseases with different types of data. If we examine the results of the first phase of Behçet's disease experiment (when all the features are used), the first notable result is the low predictive performance, which remains at around 60% when all the SNPs are used. Even when feature selection methods were applied to the entire SNP set, a slight improvement was obtained. On the other hand, the DKSS method, which selects SNPs using active subnetworks in the literature, achieved 96% accuracy using nearly 8076 SNPs, outperforming the closest feature method by 30% more accuracy. This result can be seen as evidence that a specific subset of features, rather than all features, is more useful in the prediction of genetic diseases with machine learning methods. Moreover, the higher results obtained after applying the P-value criteria (in the second phase) also confirm that all features should be filtered by a correct technique.

In the second phase, where 18479 features were used after filtering by P-value, almost perfect classification performance was achieved with more than 99+ accuracy rate. Furthermore, even if fewer number of these features were used after the feature selection methods performed, the prediction performance still remained at around 97% accuracy. These results indicate that the P-value criterion in GWAS data filters out the SNPs that are most relevant for disease prediction. However, relying only on the P-value criteria

may not be sufficient when identifying the most effective SNPs in terms of disease prediction. To address this issue, the common SNPs selected by the best-performing feature selection methods were identified and ranked. Even though the top-7 SNPs in the ranked list are SNPs with the lowest P-value, the list contains SNPs with high P-value values such as rs522686, rs703191, and rs1208571. This result can be presented as evidence that the combination of multiple genetic factors is significant in the disease prediction problem, especially in multifactorial genetic diseases. SNPs were then mapped to genes, and it was observed that the top selected gene is HLA-B (rs1058026). It was expected because this gene was marked as a highly associated gene with Behçet's Disease. It also showed that the usage of domain knowledge from literature could significantly contribute to disease prediction. Additionally, other highly selected genes include HCP5 (rs1131896, rs2848713), KIRREL3 (rs522686), LAMP5-AS1 (rs16995979), MICA (rs2256028), and SCD5 (rs6535384). However, some further in vivo and in vitro experiments should be carried out to state these genes are also significant for Behçet's disease.

In our second experiment, infectious disease prediction, we have conducted more comprehensive and detailed analyses compared to those in Behçet's disease. This is because the dataset we used provides considerably broader opportunities in terms of prediction. The GSE73072 dataset is a combination of seven distinct sub-experiments related to four respiratory viruses, as well as keeps information on infection and symptomatic status of individuals. In addition, gene expression values from samples were collected a day before inoculation (i.e., T.-24 or T.-30 hours), immediately before inoculation (T.0), and at predetermined intervals. Therefore, we could carry out multiple analyses such as symptom prediction, pre- and post-exposure prediction, merging the same virus samples, etc. To analyze this dataset comprehensively, we handled the dataset in 3 experimental groups, each of them analyzed individually:

1. Experiment-Based Analysis: Each sub-dataset analyzed separately.

2. Virus-Based Analysis: Samples related in same virus are merged to examine virus related findings

3. Combined Analysis ("ALL"): All samples are merged as a single dataset.

Within each of these experimental groups, several machine learning, feature selection and feature representation methods were compared, yielding more than 60,000 lines of results. These results can be summarized as follows:

- Firstly, the infection and symptomatic status of individuals was predicted with acceptable accuracy, achieving an average AUPRC of approximately 0.9. This result was achieved in almost all experimental groups as well as in the time point predictions. In particular, the prediction performance at T.0 showed that machine learning methods can predict whether individuals will become infected and develop symptoms after exposure to the virus.

- Although 7 sub-datasets were collected at different times and for different experiments, classification performance were closed to each other when they are merged or analyzed separately. For instance, the DEE2 H3N2 and DEE5 H3N2 sub-experiments yielded AUPRC values of 1 and 0.943, respectively, for infection predictions at time point T.0. On the other hand, the performance remained at an AUPRC of 0.984, which can be considered acceptable. Consequently, the merging of samples from different datasets to be predicted can be considered an applicable approach in terms of generalizability, if they are related to the same disease. Similar results were obtained for the symptom development prediction problem.

- In some experiments, remarkably high prediction performance has been achieved using only one or a few features. For instance, in the DEE2 sub-experiment, a Tree-Based approach with a selection of 5 ssGSEA features yielded an AUPRC value of 1 for infection prediction at time point T.0. This notable performance would potentially be seen associated with the scarcity of training and test data in the experiment-based group. Nevertheless, a combination of the fisher score and ssGSEA representation model achieved an AUPRC value of 0.902 at time point T.72 prediction on the "ALL" dataset, which has more samples as all samples are merged. Therefore, it can be concluded that a small number of genetic information can be adequate rather than use all features to make predictions about respiratory infection problems.

- During the analyses, the effect of feature representation types on respiratory infection prediction was also addressed. Microarray technology actually reflects

the intensity level of probes derived from blood-like samples. Probes are the short sequences of single-stranded DNA. If gene-level expression values are needed, these probes should be mapped into genes to determine which gene expressed more. Consequently, in our analysis, we have compared the usage of both probe-level and gene-level expression values as input for machine learning models. Additionally, we have proposed the ssGSEA-based representation approach to the respiratory prediction problem. The ssGSEA-based representation essentially uses enrichment scores as features that express how representative the predefined gene sets or pathways are of the expression values of the samples. From the results we have obtained, it has been observed that there is not a single method that consistently outperforms the others. Best-performing representation type varied depending on the experiment group, type of sub-experiments as well as predicted time points. For instance, in the symptom prediction problem of the "ALL" experiment group, the ssGSEA-based representation approaches achieved the highest performance in all 6 time point predictions. However, the same representation types could only obtain the best results in 1 out of 6 time point predictions in H3N2-virus-related experiments. Nevertheless, it can be concluded that representing the samples with enrichment scores derived from gene expression values is acceptable approach.

- When evaluating the feature selection methods, Tree-Based, Lasso, ReliefF, and mRMR were found to be the most effective methods. A common ground of these four methods is that they all consider the interdependencies of features during the selection process. Hence, these methods evaluate features as a group rather than individually. This finding is supported by biological evidence, which suggests that diseases are often influenced by multiple genetic factors, as previously stated. Another notable finding related to feature selection is that best-performing results are often obtained when the feature selection method is applied. Therefore, it can be concluded that feature selection techniques can significantly improve the accuracy of predicting infectious diseases, particularly respiratory diseases.

- The CFD was selected as most significant gene almost all different time point prediction in the H3N2-related experiment. HBB, on the other hand, was mostly selected gene for the H1N1 experiment. Despite the lack of study indicating the association of these genes to respiratory viruses, the results of machine learning

demonstrate a clear impact of their expression on the accurate prediction of an individual's infection. To conclusively prove the impact of these genes on disease progression or infection, further genetic or biological studies and experiments should be carried out.

- Another significant finding resulting from the experiments is the strong association between the prediction of respiratory infection and the Immune System. Upon further analysis of the common genes selected in the best-performing models across all three experimental groups using Over Representation Analysis (ORA), it has become evident that these genes are closely associated with the “Immune System” or “Immune Response”.

- The primary objective of the respiratory infection prediction experiments was to outperform the best results of the Viral DREAM Challenge, considered to be one of the most comprehensive challenges in the field of respiratory infection prediction, by using different approaches. Our proposed methods were compared with the results of the winner of the challenge (see Section 4.2.2.2), showing that GSEA-based representation approaches outperformed them in all prediction problems and time points. In particular, the models we introduced for predicting symptom development achieved nearly 20% better results. This suggests that the GSEA-based feature representation approach can be valuable in prediction problems that utilize gene expression as input data.

5.2 Societal Impact and Contribution to Global Sustainability

Today, accurate and early diagnosis is one of the most important factors in preventing the spread of diseases and the progression of health problems. Rapid and accurate diagnosis can reduce healthcare costs and potential health problems for individuals, especially children. However, diagnosis of any disease is challenging with traditional approaches due to the complex mechanisms of the disease, the similarity of symptoms, and the difficulty in identifying the underlying factors leading to the disease. Therefore, healthcare systems need to evolve such an approach to provide robust, faster,

and more accurate solutions. Moreover, the challenges of the coronavirus pandemic (COVID-19) also brought to light the urgent need for a transformation of our healthcare services to make health systems a more resilient and sustainable [229].

This topic, sustainability in healthcare, is also partially included in the 17 Sustainable Development Goals (SDGs) introduced by the United Nations Development Programme (UNDP) in 2015 [230]. These 17 goals serve as a shared blueprint for achieving peace, prosperity, and sustainability for both people and the planet. One of these goals, Goal 3 in the list, is related to improving health outcomes and access to quality health care. This goal also includes a sub-target directly related to communicable, i.e. infectious diseases.

This thesis contains a comprehensive and comparative analysis of the prediction of two different diseases, one genetic and the other infectious. Besides machine learning methods in prediction, feature selection and feature representation approaches included in the experiments in detail. Therefore, all experiments constituting the thesis output are related to personalized medicine. Hence, by showing how machine learning approaches behave in two different types of disease and how accurate in prediction they are, this study contributes to the sustainability of personalized medicine as well as healthcare systems. In addition, some of the genes that are expected to be associated with the disease according to in silico analysis are revealed for the diseases handled. Hence, this study is a contribution to the sustainability of healthcare systems through machine learning prediction for both Behçet's disease and respiratory infection. Furthermore, in the respiratory infection problem, we also find the significant genes that might be important for post-exposure infection. From this point of view, our study highlights the prospects of how models based on artificial intelligence, which has gained an important place in the development of technology, will play a role in current and future medical technology, healthcare systems, and the diagnosis of diseases.

5.3 Future Prospects

During the study of this thesis, 2 papers on disease prediction were published in high quartile journals. In addition, a paper that included an enrichment score-based representation for the problem of infection prediction was submitted to another journal.

Nevertheless, there are still many opportunities in the disease prediction problem of both infectious and genetic disease. Because the paradigm of the health-science has been shifting technological-solutions with the power of the artificial intelligence.

One of them is to develop a feature selection & extraction tool that incorporates both GSEA and ssGSEA approaches. As explained in previous sections, the standard GSEA approach can be performed over the whole dataset, not for the sample level. Therefore, the ssGSEA method was used to extract information to be used in sample-level prediction for our experiments. However, determining which gene sets or pathways to use as input for the ssGSEA process can be done more reliably and statistically, rather than randomly. For example, enriched gene sets can be identified using standard GSEA and only training samples, and then ssGSEA can be performed using only these sets. Prediction accuracy could also be improved by optimizing the parameters of GSEA or ssGSEA if a such tool is developed. Our next study after the thesis could be the development of this tool.

Another possible topic for prospects in disease prediction can be integrative feature representation using various types of data, including biological, community knowledge, patient symptom description, etc. with advanced techniques. As is well known, the revolution of Large Language Models (LLMs) has had a profound impact on science and technology. Although LLMs are originally a subset of Natural Language Processing (NLP), which provides the ability to generate human-like text and answer complex questions, they have demonstrated remarkable benefits in various fields such as industry, education, and the arts. These models can contribute to disease prediction problems by analyzing medical records, disease explanations, and symptom descriptions declared by patients.

BIBLIOGRAPHY

[1] J. A. Reuter, D. V. Spacek, and M. P. Snyder, ‘High-Throughput Sequencing Technologies’, *Molecular Cell*, vol. 58, no. 4, pp. 586–597, May 2015, doi: 10.1016/j.molcel.2015.05.004.

[2] H. Satam *et al.*, ‘Next-Generation Sequencing Technology: Current Trends and Advancements’, *Biology*, vol. 12, no. 7, Art. no. 7, Jul. 2023, doi: 10.3390/biology12070997.

[3] M. V. Schneider and S. Orchard, ‘Omics Technologies, Data and Bioinformatics Principles’, in *Bioinformatics for Omics Data: Methods and Protocols*, B. Mayer, Ed., in Methods in Molecular Biology. , Totowa, NJ: Humana Press, 2011, pp. 3–30. doi: 10.1007/978-1-61779-027-0_1.

[4] H. Bhaskar, D. C. Hoyle, and S. Singh, ‘Machine learning in bioinformatics: A brief survey and recommendations for practitioners’, *Computers in Biology and Medicine*, vol. 36, no. 10, pp. 1104–1125, Oct. 2006, doi: 10.1016/j.compbiomed.2005.09.002.

[5] N. Auslander, A. B. Gussow, and E. V. Koonin, ‘Incorporating Machine Learning into Established Bioinformatics Frameworks’, *Int J Mol Sci*, vol. 22, no. 6, p. 2903, Mar. 2021, doi: 10.3390/ijms22062903.

[6] R. S. Olson, W. L. Cava, Z. Mustahsan, A. Varik, and J. H. Moore, ‘Data-driven advice for applying machine learning to bioinformatics problems’, in *Biocomputing 2018*, WORLD SCIENTIFIC, 2017, pp. 192–203. doi: 10.1142/9789813235533_0018.

[7] K. K. Jain, *Textbook of Personalized Medicine*. New York, NY: Springer, 2009. doi: 10.1007/978-1-4419-0769-1.

[8] B. Larijani, H. R. Aghaei Meybodi, N. Sarhangi, and M. Hasanzad, ‘Principles of Precision Medicine’, in *Precision Medicine in Clinical Practice*, M. Hasanzad, Ed., Singapore: Springer Nature, 2022, pp. 1–11. doi: 10.1007/978-981-19-5082-7_1.

[9] G. Rani and P. K. Tiwari, *Handbook of Research on Disease Prediction Through Data Analytics and Machine Learning*. IGI Global, 1AD. Accessed: Feb. 28, 2024. [Online]. Available: https://www.igi-global.com/book/handbook-research-disease-prediction-through/www.igi-global.com/book/handbook-research-disease-prediction-through/237838

[10] I. Y. Iourov, S. G. Vorsanova, and Y. B. Yurov, ‘Pathway-based classification of genetic diseases’, *Molecular Cytogenetics*, vol. 12, no. 1, p. 4, Feb. 2019, doi: 10.1186/s13039-019-0418-4.

[11] G. J. Lonergan, D. B. Cline, and S. L. Abbondanzo, ‘Sickle Cell Anemia’, *RadioGraphics*, vol. 21, no. 4, pp. 971–994, Jul. 2001, doi: 10.1148/radiographics.21.4.g01jl23971.

[12] A. Raza *et al.*, ‘Predicting Genetic Disorder and Types of Disorder Using Chain Classifier Approach’, *Genes*, vol. 14, no. 1, Art. no. 1, Jan. 2023, doi: 10.3390/genes14010071.

[13] T. Lee and H. Lee, ‘Prediction of Alzheimer’s disease using blood gene expression data’, *Sci Rep*, vol. 10, no. 1, Art. no. 1, Feb. 2020, doi: 10.1038/s41598-020-60595-1.

[14] S. Khanal, J. Chen, N. Jacobs, and A.-L. Lin, ‘Alzheimer’s Disease Classification Using Genetic Data’, in *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, Dec. 2021, pp. 2245–2252. doi: 10.1109/BIBM52615.2021.9669730.

[15] A. S. Alatrany, A. Hussain, M. Jamila, and D. Al-Jumeiy, ‘Stacked Machine Learning Model for Predicting Alzheimer’s Disease Based on Genetic Data’, in

2021 14th International Conference on Developments in eSystems Engineering (DeSE), Dec. 2021, pp. 594–598. doi: 10.1109/DeSE54285.2021.9719449.

[16] J. Kälsch *et al.*, 'Normal liver enzymes are correlated with severity of metabolic syndrome in a large population based cohort', *Sci Rep*, vol. 5, no. 1, Art. no. 1, Aug. 2015, doi: 10.1038/srep13058.

[17] D. Shigemizu *et al.*, 'The Construction of Risk Prediction Models Using GWAS Data and Its Application to a Type 2 Diabetes Prospective Cohort', *PLOS ONE*, vol. 9, no. 3, p. e92549, Mar. 2014, doi: 10.1371/journal.pone.0092549.

[18] J. Li, J. Ding, D. U. Zhi, K. Gu, and H. Wang, 'Identification of Type 2 Diabetes Based on a Ten-Gene Biomarker Prediction Model Constructed Using a Support Vector Machine Algorithm', *BioMed Research International*, vol. 2022, p. e1230761, Mar. 2022, doi: 10.1155/2022/1230761.

[19] Z.-Z. Chen and R. E. Gerszten, 'Metabolomics and Proteomics in Type 2 Diabetes', *Circulation Research*, vol. 126, no. 11, pp. 1613–1627, May 2020, doi: 10.1161/CIRCRESAHA.120.315898.

[20] M. M. Ahsan and Z. Siddique, 'Machine learning-based heart disease diagnosis: A systematic literature review', *Artificial Intelligence in Medicine*, vol. 128, p. 102289, Jun. 2022, doi: 10.1016/j.artmed.2022.102289.

[21] M. Sudharsan and G. Thailambal, 'Alzheimer's disease prediction using machine learning techniques and principal component analysis (PCA)', *Materials Today: Proceedings*, vol. 81, pp. 182–190, Jan. 2023, doi: 10.1016/j.matpr.2021.03.061.

[22] N. H. Nguyen *et al.*, 'Machine Learning-based Prediction Models for Diagnosis and Prognosis in Inflammatory Bowel Diseases: A Systematic Review', *Journal of Crohn's and Colitis*, vol. 16, no. 3, pp. 398–413, Mar. 2022, doi: 10.1093/ecco-jcc/jjab155.

[23] J. R. Nair and R. J. Moots, 'Behcet's disease', *Clinical Medicine*, vol. 17, no. 1, pp. 71–77, Feb. 2017, doi: 10.7861/clinmedicine.17-1-71.

[24] D. Saadoun and B. Wechsler, 'Behçet's disease', *Orphanet Journal of Rare Diseases*, vol. 7, no. 1, p. 20, Apr. 2012, doi: 10.1186/1750-1172-7-20.

[25] C. Maldini, K. Druce, N. Basu, M. P. LaValley, and A. Mahr, 'Exploring the variability in Behçet's disease prevalence: a meta-analytical approach', *Rheumatology*, vol. 57, no. 1, pp. 185–195, Jan. 2018, doi: 10.1093/rheumatology/kew486.

[26] N. M. Leonardo and J. McNeil, 'Behcet's Disease: Is There Geographical Variation? A Review Far from the Silk Road', *International Journal of Rheumatology*, vol. 2015, p. e945262, Dec. 2015, doi: 10.1155/2015/945262.

[27] A. Akman and E. Alpsoy, 'Behçet Hastalığı: Etyopatogenezde Güncel Bilgiler', vol. 43, no. Supp:2, pp. 32–38.

[28] S. Ohno, K. Aoki, S. Sugiura, E. Nakayama, K. Itakura, and M. Aizawa, 'HL-A5 AND BEHCET'S DISEASE', *The Lancet*, vol. 302, no. 7842, pp. 1383–1384, Dec. 1973, doi: 10.1016/S0140-6736(73)93343-6.

[29] A. Gul and S. Ohno, 'HLA-B*51 and Behçet Disease', *Ocular Immunology and Inflammation*, vol. 20, no. 1, pp. 37–43, Feb. 2012, doi: 10.3109/09273948.2011.634978.

[30] M. Giza, D. Koftori, L. Chen, and P. Bowness, ‘Is Behçet’s disease a “class 1-opathy”? The role of HLA-B*51 in the pathogenesis of Behçet’s disease’, *Clinical and Experimental Immunology*, vol. 191, no. 1, pp. 11–18, Jan. 2018, doi: 10.1111/cei.13049.

[31] L. Ortiz-Fernández and A. H. Sawalha, ‘Genetics of Behçet’s Disease: Functional Genetic Analysis and Estimating Disease Heritability’, *Frontiers in Medicine*, vol. 8, 2021, Accessed: Feb. 20, 2024. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fmed.2021.625710

[32] E. F. Remmers *et al.*, ‘Genome-wide association study identifies variants in the MHC class I, IL10, and IL23R-IL12RB2 regions associated with Behçet’s disease’, *Nat Genet*, vol. 42, no. 8, Art. no. 8, Aug. 2010, doi: 10.1038/ng.625.

[33] I. Sousa *et al.*, ‘Association of CCR1, KLRC4, IL12A–AS1, STAT4, and ERAP1 With Behçet’s Disease in Iranians’, *Arthritis & Rheumatology*, vol. 67, no. 10, pp. 2742–2748, 2015, doi: 10.1002/art.39240.

[34] N. Hammam *et al.*, ‘Development of machine learning models for detection of vision threatening Behçet’s disease (BD) using Egyptian College of Rheumatology (ECR)–BD cohort’, *BMC Medical Informatics and Decision Making*, vol. 23, no. 1, p. 37, Feb. 2023, doi: 10.1186/s12911-023-02130-6.

[35] J. M. Kim, J. G. Kang, S. Kim, and J. H. Cheon, ‘Deep-learning system for real-time differentiation between Crohn’s disease, intestinal Behçet’s disease, and intestinal tuberculosis’, *Journal of Gastroenterology and Hepatology*, vol. 36, no. 8, pp. 2141–2148, 2021, doi: 10.1111/jgh.15433.

[36] İ. Güler and E. D. Übeyli, ‘Detection of ophthalmic arterial doppler signals with Behcet disease using multilayer perceptron neural network’, *Computers in Biology and Medicine*, vol. 35, no. 2, pp. 121–132, Feb. 2005, doi: 10.1016/j.compbiomed.2003.12.007.

[37] Z. Ye *et al.*, ‘A metagenomic study of the gut microbiome in Behcet’s disease’, *Microbiome*, vol. 6, no. 1, p. 135, Aug. 2018, doi: 10.1186/s40168-018-0520-6.

[38] H. Tang *et al.*, ‘TMT and PRM-Based Quantitative Proteomics Identify Potential Biomarkers for Behçet Syndrome’. Rochester, NY, Aug. 12, 2021. doi: 10.2139/ssrn.3903947.

[39] L. H. Bajrai *et al.*, ‘Gene Expression Profiling of Early Acute Febrile Stage of Dengue Infection and Its Comparative Analysis With Streptococcus pneumoniae Infection’, *Frontiers in Cellular and Infection Microbiology*, vol. 11, 2021, Accessed: Feb. 20, 2024. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fcimb.2021.707905

[40] M. C. Inhorn and P. J. Brown, ‘The Anthropology of Infectious Disease’, *Annual Review of Anthropology*, vol. 19, pp. 89–117, 1990.

[41] S. Agrebi and A. Larbi, ‘Use of artificial intelligence in infectious diseases’, in *Artificial Intelligence in Precision Health*, D. Barh, Ed., Academic Press, 2020, pp. 415–438. doi: 10.1016/B978-0-12-817133-2.00018-5.

[42] I. Abubakar, ‘Infectious Disease Epidemiology’, in *Infectious Disease Epidemiology*, I. Abubakar, H. R. Stagg, T. Cohen, and L. C. Rodrigues, Eds., Oxford University Press, 2016, p. 1. doi: 10.1093/med/9780198719830.003.0001.

[43] A. Delivorias and N. Scholz, ‘Economic impact of epidemics and pandemics’, 2020.

[44] G. Sun, T. Matsui, Y. Hakozaki, and S. Abe, ‘An infectious disease/fever screening radar system which stratifies higher-risk patients within ten seconds using a neural network and the fuzzy grouping method’, *Journal of Infection*, vol. 70, no. 3, pp. 230–236, Mar. 2015, doi: 10.1016/j.jinf.2014.12.007.

[45] M. R. Saybani *et al.*, ‘RAIRS2 a new expert system for diagnosing tuberculosis with real-world tournament selection mechanism inside artificial immune recognition system’, *Med Biol Eng Comput*, vol. 54, no. 2, pp. 385–399, Mar. 2016, doi: 10.1007/s11517-015-1323-6.

[46] K. Moulaei, M. Shanbehzadeh, Z. Mohammadi-Taghiabad, and H. Kazemi-Arpanahi, ‘Comparing machine learning algorithms for predicting COVID-19 mortality’, *BMC Medical Informatics and Decision Making*, vol. 22, no. 1, p. 2, Jan. 2022, doi: 10.1186/s12911-021-01742-0.

[47] S. Akbulut, Z. Küçükakçali, and C. Çolak, ‘MACHINE LEARNING-BASED CLASSIFICATION OF HBV AND HCV-RELATED HEPATOCELLULAR CARCINOMA USING GENOMIC BIOMARKERS’, *İst Tıp Fak Derg*, vol. 85, no. 4, Art. no. 4, Oct. 2022, doi: 10.26650/IUITFD.1130442.

[48] K. Y. Tai, J. Dhaliwal, and K. Wong, ‘Risk score prediction model based on single nucleotide polymorphism for predicting malaria: a machine learning approach’, *BMC Bioinformatics*, vol. 23, no. 1, p. 325, Aug. 2022, doi: 10.1186/s12859-022-04870-0.

[49] W. I. A. W. M. Nawi *et al.*, ‘Developing forecasting model for future pandemic applications based on COVID-19 data 2020–2022’, *PLOS ONE*, vol. 18, no. 5, p. e0285407, May 2023, doi: 10.1371/journal.pone.0285407.

[50] S. Dixon, R. Keshavamurthy, D. H. Farber, A. Stevens, K. T. Pazdernik, and L. E. Charles, ‘A Comparison of Infectious Disease Forecasting Methods across Locations, Diseases, and Time’, *Pathogens*, vol. 11, no. 2, Art. no. 2, Feb. 2022, doi: 10.3390/pathogens11020185.

[51] A. Ajith, K. Manoj, H. Kiran, P. J. Pillai, and J. J. Nair, ‘A Study on Prediction and Spreading of Epidemic Diseases’, in *2020 International Conference on Communication and Signal Processing (ICCSP)*, Jul. 2020, pp. 1265–1268. doi: 10.1109/ICCSP48568.2020.9182147.

[52] X. Kuang, F. Wang, K. M. Hernandez, Z. Zhang, and R. L. Grossman, ‘Accurate and rapid prediction of tuberculosis drug resistance from genome sequence data using traditional machine learning algorithms and CNN’, *Sci Rep*, vol. 12, no. 1, Art. no. 1, Feb. 2022, doi: 10.1038/s41598-022-06449-4.

[53] A. Rajput and M. Kumar, ‘Anti-Ebola: an initiative to predict Ebola virus inhibitors through machine learning’, *Mol Divers*, vol. 26, no. 3, pp. 1635–1644, Jun. 2022, doi: 10.1007/s11030-021-10291-7.

[54] W. H. Organisation, *World Health Statistics 2020, Monitoring Health for the SDGs*. World Health Organisation Geneva, Switzerland, 2020.

[55] G. Yuan *et al.*, ‘Early identification and severity prediction of acute respiratory infection (ESAR): a study protocol for a randomized controlled trial’, *BMC*

Infectious Diseases, vol. 22, no. 1, p. 632, Jul. 2022, doi: 10.1186/s12879-022-07552-7.

[56] R. E. Dixon, 'Economic costs of respiratory tract infections in the United States', *The American Journal of Medicine*, vol. 78, no. 6, Supplement 2, pp. 45–51, Jun. 1985, doi: 10.1016/0002-9343(85)90363-8.

[57] H. F. Boncristiani, M. F. Criado, and E. Arruda, 'Respiratory Viruses', in *Encyclopedia of Microbiology (Third Edition)*, M. Schaechter, Ed., Oxford: Academic Press, 2009, pp. 500–518. doi: 10.1016/B978-012373944-5.00314-X.

[58] J.-M. Harerimana, L. Nyirazinyoye, D. R. Thomson, and J. Ntaganira, 'Social, economic and environmental risk factors for acute lower respiratory infections among children under five years of age in Rwanda', *Archives of Public Health*, vol. 74, no. 1, p. 19, May 2016, doi: 10.1186/s13690-016-0132-1.

[59] R. Lambkin-Williams, N. Noulin, A. Mann, A. Catchpole, and A. S. Gilbert, 'The human viral challenge model: accelerating the evaluation of respiratory antivirals, vaccines and novel diagnostics', *Respiratory Research*, vol. 19, no. 1, p. 123, Jun. 2018, doi: 10.1186/s12931-018-0784-1.

[60] E. Kuchar, K. Miśkiewicz, A. Nitsch-Osuch, and L. Szenborn, 'Pathophysiology of Clinical Symptoms in Acute Viral Respiratory Tract Infections', in *Pulmonary Infection*, M. Pokorski, Ed., in Advances in Experimental Medicine and Biology. , Cham: Springer International Publishing, 2015, pp. 25–38. doi: 10.1007/5584_2015_110.

[61] R. R. Jansen *et al.*, 'Frequent Detection of Respiratory Viruses without Symptoms: Toward Defining Clinically Relevant Cutoff Values', *Journal of Clinical Microbiology*, vol. 49, no. 7, pp. 2631–2636, Dec. 2020, doi: 10.1128/jcm.02094-10.

[62] C. L. Byington *et al.*, 'Community Surveillance of Respiratory Viruses Among Families in the Utah Better Identification of Germs-Longitudinal Viral Epidemiology (BIG-LoVE) Study', *Clinical Infectious Diseases*, vol. 61, no. 8, pp. 1217–1224, Oct. 2015, doi: 10.1093/cid/civ486.

[63] S.-Y. Zhang, Q. Zhang, J.-L. Casanova, and H. C. Su, 'Severe COVID-19 in the young and healthy: monogenic inborn errors of immunity?', *Nat Rev Immunol*, vol. 20, no. 8, Art. no. 8, Aug. 2020, doi: 10.1038/s41577-020-0373-7.

[64] I. Esteban *et al.*, 'Asymptomatic COVID-19 in the elderly: dementia and viral clearance as risk factors for disease progression.', *Gates Open Res*, vol. 5, p. 143, Apr. 2022, doi: 10.12688/gatesopenres.13357.2.

[65] M. Pichon, B. Lina, and L. Josset, 'Impact of the Respiratory Microbiome on Host Responses to Respiratory Viral Infection', *Vaccines*, vol. 5, no. 4, Art. no. 4, Dec. 2017, doi: 10.3390/vaccines5040040.

[66] G. J. Walker *et al.*, 'Viruses associated with acute respiratory infection in a community-based cohort of healthy New Zealand children', *Journal of Medical Virology*, vol. 94, no. 2, pp. 454–460, 2022, doi: 10.1002/jmv.25493.

[67] J. T. Lim, K. B. Tan, J. Abisheganaden, and B. L. Dickens, 'Forecasting upper respiratory tract infection burden using high-dimensional time series data and forecast combinations', *PLOS Computational Biology*, vol. 19, no. 2, p. e1010892, ub 2023, doi: 10.1371/journal.pcbi.1010892.

[68] G. Barlacchi, C. Perentis, A. Mehrotra, M. Musolesi, and B. Lepri, ‘Are you getting sick? Predicting influenza-like symptoms using human mobility behaviors’, *EPJ Data Sci.*, vol. 6, no. 1, Art. no. 1, Dec. 2017, doi: 10.1140/epjds/s13688-017-0124-6.

[69] E. Elbasi, A. Zreikat, S. Mathew, and A. E. Topcu, ‘Classification of influenza H1N1 and COVID-19 patient data using machine learning’, in *2021 44th International Conference on Telecommunications and Signal Processing (TSP)*, Jul. 2021, pp. 278–282. doi: 10.1109/TSP52935.2021.9522591.

[70] E. Bongen, F. Vallania, P. J. Utz, and P. Khatri, ‘KLRD1-expressing natural killer cells predict influenza susceptibility’, *Genome Medicine*, vol. 10, no. 1, p. 45, Jun. 2018, doi: 10.1186/s13073-018-0554-1.

[71] R. Barral-Arca, A. Gómez-Carballa, M. Cebey-López, X. Bello, F. Martinón-Torres, and A. Salas, ‘A Meta-Analysis of Multiple Whole Blood Gene Expression Data Unveils a Diagnostic Host-Response Transcript Signature for Respiratory Syncytial Virus’, *International Journal of Molecular Sciences*, vol. 21, no. 5, Art. no. 5, Jan. 2020, doi: 10.3390/ijms21051831.

[72] Y. Xu, Y.-H. Zhang, J. Li, X. Y. Pan, T. Huang, and Y.-D. Cai, ‘New Computational Tool Based on Machine-learning Algorithms for the Identification of Rhinovirus Infection-Related Genes’, *Combinatorial Chemistry & High Throughput Screening*, vol. 22, no. 10, pp. 665–674, Dec. 2019, doi: 10.2174/1386207322666191129114741.

[73] M. Radovic, M. Ghalwash, N. Filipovic, and Z. Obradovic, ‘Minimum redundancy maximum relevance feature selection approach for temporal gene expression data’, *BMC Bioinformatics*, vol. 18, no. 1, p. 9, Jan. 2017, doi: 10.1186/s12859-016-1423-9.

[74] S.-K. Hung *et al.*, ‘Developing and validating clinical features-based machine learning algorithms to predict influenza infection in influenza-like illness patients’, *Biomedical Journal*, vol. 46, no. 5, p. 100561, Oct. 2023, doi: 10.1016/j.bj.2022.09.002.

[75] G. Verma, A. Jha, D. Rebholz-Schuhmann, and M. G. Madden, ‘Using Machine Learning to Distinguish Infected from Non-infected Subjects at an Early Stage Based on Viral Inoculation’, in *Data Integration in the Life Sciences*, S. Auer and M.-E. Vidal, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, pp. 105–121. doi: 10.1007/978-3-030-06016-9_11.

[76] A. Zan *et al.*, ‘DeepFlu: a deep learning approach for forecasting symptomatic influenza A infection based on pre-exposure gene expression’, *Computer Methods and Programs in Biomedicine*, vol. 213, p. 106495, Jan. 2022, doi: 10.1016/j.cmpb.2021.106495.

[77] M. Teufel and P. Sobetzko, ‘Reducing costs for DNA and RNA sequencing by sample pooling using a metagenomic approach’, *BMC Genomics*, vol. 23, no. 1, p. 613, Aug. 2022, doi: 10.1186/s12864-022-08831-y.

[78] A. Mohammed Yakubu and Y.-P. P. Chen, ‘Ensuring privacy and security of genomic data and functionalities’, *Briefings in Bioinformatics*, vol. 21, no. 2, pp. 511–526, Mar. 2020, doi: 10.1093/bib/bbz013.

[79] E. Clough and T. Barrett, ‘The Gene Expression Omnibus Database’, in *Statistical Genomics: Methods and Protocols*, E. Mathé and S. Davis, Eds., in Methods in Molecular Biology. , New York, NY: Springer, 2016, pp. 93–110. doi: 10.1007/978-1-4939-3578-9_5.

[80] E. Clough *et al.*, ‘NCBI GEO: archive for gene expression and epigenomics data sets: 23-year update’, *Nucleic Acids Research*, vol. 52, no. D1, pp. D138–D144, Jan. 2024, doi: 10.1093/nar/gkad965.

[81] S. T. Sherry *et al.*, ‘dbSNP: the NCBI database of genetic variation’, *Nucleic Acids Research*, vol. 29, no. 1, pp. 308–311, Jan. 2001, doi: 10.1093/nar/29.1.308.

[82] M. A. Freeberg *et al.*, ‘The European Genome-phenome Archive in 2021’, *Nucleic Acids Research*, vol. 50, no. D1, pp. D980–D987, Jan. 2022, doi: 10.1093/nar/gkab1059.

[83] Z. Wang, M. A. Jensen, and J. C. Zenklusen, ‘A Practical Guide to The Cancer Genome Atlas (TCGA)’, in *Statistical Genomics: Methods and Protocols*, E. Mathé and S. Davis, Eds., in Methods in Molecular Biology. , New York, NY: Springer, 2016, pp. 111–141. doi: 10.1007/978-1-4939-3578-9_6.

[84] U. Sarkans *et al.*, ‘From ArrayExpress to BioStudies’, *Nucleic Acids Research*, vol. 49, no. D1, pp. D1502–D1506, Jan. 2021, doi: 10.1093/nar/gkaa1062.

[85] A. Kozomara, M. Birgaoanu, and S. Griffiths-Jones, ‘miRBase: from microRNA sequences to function’, *Nucleic Acids Research*, vol. 47, no. D1, pp. D155–D162, Jan. 2019, doi: 10.1093/nar/gky1141.

[86] A. L. Mitchell *et al.*, ‘MGnify: the microbiome analysis resource in 2020’, *Nucleic Acids Research*, vol. 48, no. D1, pp. D570–D578, Jan. 2020, doi: 10.1093/nar/gkz1035.

[87] S. S. Kshatri, D. Singh, T. Goswami, and G. R. Sinha, ‘Introduction to statistical modeling in machine learning: a case study’, in *Statistical Modeling in Machine Learning*, T. Goswami and G. R. Sinha, Eds., Academic Press, 2023, pp. 1–21. doi: 10.1016/B978-0-323-91776-6.00007-5.

[88] H. Belyadi and A. Haghighat, *Machine Learning Guide for Oil and Gas Using Python: A Step-by-Step Breakdown with Data, Algorithms, Codes, and Applications*. Gulf Professional Publishing, 2021.

[89] R. O. Sinnott, H. Duan, and Y. Sun, ‘A Case Study in Big Data Analytics: Exploring Twitter Sentiment Analysis and the Weather’, in *Big Data*, R. Buyya, R. N. Calheiros, and A. V. Dastjerdi, Eds., Morgan Kaufmann, 2016, pp. 357–388. doi: 10.1016/B978-0-12-805394-2.00015-5.

[90] C. Cortes and V. Vapnik, ‘Support-vector networks’, *Mach Learn*, vol. 20, no. 3, pp. 273–297, Sep. 1995, doi: 10.1007/BF00994018.

[91] A. Rani, N. Kumar, J. Kumar, J. Kumar, and N. K. Sinha, ‘Machine learning for soil moisture assessment’, in *Deep Learning for Sustainable Agriculture*, R. C. Poonia, V. Singh, and S. R. Nayak, Eds., in Cognitive Data Science in Sustainable Computing. , Academic Press, 2022, pp. 143–168. doi: 10.1016/B978-0-323-85214-2.00001-X.

[92] K.-B. Duan and S. S. Keerthi, ‘Which Is the Best Multiclass SVM Method? An Empirical Study’, in *Multiple Classifier Systems*, N. C. Oza, R. Polikar, J. Kittler,

and F. Roli, Eds., in Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2005, pp. 278–285. doi: 10.1007/11494683_28.

[93] S. Ray, 'A Quick Review of Machine Learning Algorithms', in *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, Feb. 2019, pp. 35–39. doi: 10.1109/COMITCon.2019.8862451.

[94] S. Zhang, X. Li, M. Zong, X. Zhu, and D. Cheng, 'Learning k for kNN Classification', *ACM Trans. Intell. Syst. Technol.*, vol. 8, no. 3, p. 43:1-43:19, Ocak 2017, doi: 10.1145/2990508.

[95] S. Rashka and V. Mirdzhalili, 'Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2', *Birmingham, Mumbai. Packt*, 2020.

[96] J. R. Quinlan, 'Learning decision tree classifiers', *ACM Comput. Surv.*, vol. 28, no. 1, pp. 71–72, Mar. 1996, doi: 10.1145/234313.234346.

[97] L. Breiman, 'Random Forests', *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.

[98] R. Srivastava, S. Kumar, and B. Kumar, 'Classification model of machine learning for medical data analysis', in *Statistical Modeling in Machine Learning*, T. Goswami and G. R. Sinha, Eds., Academic Press, 2023, pp. 111–132. doi: 10.1016/B978-0-323-91776-6.00017-8.

[99] R. E. Schapire, 'The strength of weak learnability', *Mach Learn*, vol. 5, no. 2, pp. 197–227, Jun. 1990, doi: 10.1007/BF00116037.

[100] A. Mayr, H. Binder, O. Gefeller, and M. Schmid, 'The Evolution of Boosting Algorithms', *Methods Inf Med*, vol. 53, no. 06, pp. 419–427, 2014, doi: 10.3414/ME13-01-0122.

[101] A. J. Ferreira and M. A. T. Figueiredo, 'Boosting Algorithms: A Review of Methods, Theory, and Applications', in *Ensemble Machine Learning: Methods and Applications*, C. Zhang and Y. Ma, Eds., New York, NY: Springer, 2012, pp. 35–85. doi: 10.1007/978-1-4419-9326-7_2.

[102] C. Bentéjac, A. Csörgő, and G. Martínez-Muñoz, 'A comparative analysis of gradient boosting algorithms', *Artif Intell Rev*, vol. 54, no. 3, pp. 1937–1967, Mar. 2021, doi: 10.1007/s10462-020-09896-5.

[103] G. Ke *et al.*, 'LightGBM: A Highly Efficient Gradient Boosting Decision Tree', in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2017. Accessed: Feb. 20, 2024. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html

[104] T. Chen and C. Guestrin, 'XGBoost: A Scalable Tree Boosting System', in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, in KDD '16. New York, NY, USA: Association for Computing Machinery, Ağustos 2016, pp. 785–794. doi: 10.1145/2939672.2939785.

[105] A. Meyer-Baese and V. Schmid, 'Statistical and Syntactic Pattern Recognition', in *Pattern Recognition and Signal Analysis in Medical Imaging (Second Edition)*, A. Meyer-Baese and V. Schmid, Eds., Oxford: Academic Press, 2014, pp. 151–196. doi: 10.1016/B978-0-12-409545-8.00006-6.

[106] N. Salmi and Z. Rustam, ‘Naïve Bayes Classifier Models for Predicting the Colon Cancer’, *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 546, no. 5, p. 052068, Jun. 2019, doi: 10.1088/1757-899X/546/5/052068.

[107] I. Rish, ‘An empirical study of the naive Bayes classifier’, in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, 2001, pp. 41–46.

[108] D. Chagné, L. Bianco, C. Lawley, D. Micheletti, and J. M. E. Jacobs, ‘Methods for the Design, Implementation, and Analysis of Illumina Infinium™ SNP Assays in Plants’, in *Plant Genotyping: Methods and Protocols*, J. Batley, Ed., in Methods in Molecular Biology. , New York, NY: Springer, 2015, pp. 281–298. doi: 10.1007/978-1-4939-1966-6_21.

[109] A. T. Weeraratna, J. E. Nagel, V. de Mello-Coelho, and D. D. Taub, ‘Gene Expression Profiling: From Microarrays to Medicine’, *J Clin Immunol*, vol. 24, no. 3, pp. 213–224, May 2004, doi: 10.1023/B:JOCI.0000025443.44833.1d.

[110] Y. Ilan, ‘Making use of noise in biological systems’, *Progress in Biophysics and Molecular Biology*, vol. 178, pp. 83–90, Mar. 2023, doi: 10.1016/j.pbiomolbio.2023.01.001.

[111] Y. Peng, Z. Wu, and J. Jiang, ‘A novel feature selection approach for biomedical data classification’, *Journal of Biomedical Informatics*, vol. 43, no. 1, pp. 15–23, Feb. 2010, doi: 10.1016/j.jbi.2009.07.008.

[112] B. Panay, N. Baloian, J. A. Pino, S. Peñafiel, H. Sanson, and N. Bersano, ‘Feature Selection for Health Care Costs Prediction Using Weighted Evidential Regression’, *Sensors*, vol. 20, no. 16, Art. no. 16, Jan. 2020, doi: 10.3390/s20164392.

[113] K. Kira and L. A. Rendell, ‘A Practical Approach to Feature Selection’, in *Machine Learning Proceedings 1992*, D. Sleeman and P. Edwards, Eds., San Francisco (CA): Morgan Kaufmann, 1992, pp. 249–256. doi: 10.1016/B978-1-55860-247-2.50037-1.

[114] K. Koras, D. Juraeva, J. Kreis, J. Mazur, E. Staub, and E. Szczurek, ‘Feature selection strategies for drug sensitivity prediction’, *Sci Rep*, vol. 10, no. 1, Art. no. 1, Jun. 2020, doi: 10.1038/s41598-020-65927-9.

[115] S. Yuan, Y.-C. Chen, C.-H. Tsai, H.-W. Chen, and G. S. Shieh, ‘Feature selection translates drug response predictors from cell lines to patients’, *Frontiers in Genetics*, vol. 14, 2023, Accessed: Feb. 20, 2024. [Online]. Available: https://www.frontiersin.org/journals/genetics/articles/10.3389/fgene.2023.1217414

[116] N. Pudjihartono, T. Fadason, A. W. Kempa-Liehr, and J. M. O’Sullivan, ‘A Review of Feature Selection Methods for Machine Learning-Based Disease Risk Prediction’, *Frontiers in Bioinformatics*, vol. 2, 2022, Accessed: Feb. 20, 2024. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fbinf.2022.927312

[117] B. Hu *et al.*, ‘Feature Selection for Optimized High-Dimensional Biomedical Data Using an Improved Shuffled Frog Leaping Algorithm’, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 15, no. 6, pp. 1765–1773, Nov. 2018, doi: 10.1109/TCBB.2016.2602263.

[118] F. Morstatter and H. Liu, ‘Advancing Feature Selection Research − ASU Feature Selection Repository’, 2010. Accessed: Feb. 20, 2024. [Online]. Available: https://www.semanticscholar.org/paper/Advancing-Feature-Selection-Research-%E2%88%92-ASU-Feature-Morstatter-Liu/7cb7ca94930461c3ae54542d7f2358c39b3c69b8

[119] C. Kuzudisli, B. Bakir-Gungor, N. Bulut, B. Qaqish, and M. Yousef, ‘Review of feature selection approaches based on grouping of features’, *PeerJ*, vol. 11, p. e15666, Jul. 2023, doi: 10.7717/peerj.15666.

[120] B. López, F. Torrent-Fontbona, R. Viñas, and J. M. Fernández-Real, ‘Single Nucleotide Polymorphism relevance learning with Random Forests for Type 2 diabetes risk prediction’, *Artificial Intelligence in Medicine*, vol. 85, pp. 43–49, Apr. 2018, doi: 10.1016/j.artmed.2017.09.005.

[121] K. Tadist, S. Najah, N. S. Nikolov, F. Mrabti, and A. Zahi, ‘Feature selection methods and genomic big data: a systematic review’, *Journal of Big Data*, vol. 6, no. 1, p. 79, Aug. 2019, doi: 10.1186/s40537-019-0241-0.

[122] W. Duch, ‘Filter Methods’, in *Feature Extraction: Foundations and Applications*, I. Guyon, M. Nikravesh, S. Gunn, and L. A. Zadeh, Eds., in Studies in Fuzziness and Soft Computing. , Berlin, Heidelberg: Springer, 2006, pp. 89–117. doi: 10.1007/978-3-540-35488-8_4.

[123] M. Rostami, K. Berahmand, E. Nasiri, and S. Forouzandeh, ‘Review of swarm intelligence-based feature selection methods’, *Engineering Applications of Artificial Intelligence*, vol. 100, p. 104210, Apr. 2021, doi: 10.1016/j.engappai.2021.104210.

[124] D. P. M. Abellana and D. M. Lao, ‘A new univariate feature selection algorithm based on the best–worst multi-attribute decision-making method’, *Decision Analytics Journal*, vol. 7, p. 100240, Jun. 2023, doi: 10.1016/j.dajour.2023.100240.

[125] C. Lai, M. J. T. Reinders, and L. Wessels, ‘Random subspace method for multivariate feature selection’, *Pattern Recognition Letters*, vol. 27, no. 10, pp. 1067–1076, Jul. 2006, doi: 10.1016/j.patrec.2005.12.018.

[126] Q. Gu, Z. Li, and J. Han, ‘Generalized Fisher Score for Feature Selection’. arXiv, Feb. 14, 2012. doi: 10.48550/arXiv.1202.3725.

[127] L. Sun, X.-Y. Zhang, Y.-H. Qian, J.-C. Xu, S.-G. Zhang, and Y. Tian, ‘Joint neighborhood entropy-based gene selection method with fisher score for tumor classification’, *Appl Intell*, vol. 49, no. 4, pp. 1245–1259, Apr. 2019, doi: 10.1007/s10489-018-1320-1.

[128] M. Gan and L. Zhang, ‘Iteratively local fisher score for feature selection’, *Appl Intell*, vol. 51, no. 8, pp. 6167–6181, Aug. 2021, doi: 10.1007/s10489-020-02141-0.

[129] K. Kira and L. A. Rendell, ‘The feature selection problem: traditional methods and a new algorithm’, in *Proceedings of the tenth national conference on Artificial intelligence*, in AAAI’92. San Jose, California: AAAI Press, Temmuz 1992, pp. 129–134.

[130] S. F. Rosario and K. Thangadurai, ‘RELIEF: Feature Selection Approach’, *ijird*, Oct. 2015, Accessed: Feb. 20, 2024. [Online]. Available:

https://www.internationaljournalcorner.com/index.php/ijird_ojs/article/view/136085

[131] M. R.-S. Ikonja, M. Robnik, and I. Kononenko, ‘Theoretical and Empirical Analysis of ReliefF and RReliefF’.

[132] R. J. Urbanowicz, M. Meeker, W. La Cava, R. S. Olson, and J. H. Moore, ‘Relief-based feature selection: Introduction and review’, *Journal of Biomedical Informatics*, vol. 85, pp. 189–203, Sep. 2018, doi: 10.1016/j.jbi.2018.07.014.

[133] J. R. Berrendero, A. Cuevas, and J. L. Torrecilla, ‘The mRMR variable selection method: a comparative study for functional data’, *Journal of Statistical Computation and Simulation*, vol. 86, no. 5, pp. 891–907, Mar. 2016, doi: 10.1080/00949655.2015.1042378.

[134] Z. Zhao, R. Anand, and M. Wang, ‘Maximum relevance and minimum redundancy feature selection methods for a marketing machine learning platform’, in *2019 IEEE international conference on data science and advanced analytics (DSAA)*, IEEE, 2019, pp. 442–452.

[135] S. Ramírez-Gallego *et al.*, ‘Fast-mRMR: Fast Minimum Redundancy Maximum Relevance Algorithm for High-Dimensional Big Data’, *International Journal of Intelligent Systems*, vol. 32, no. 2, pp. 134–152, 2017, doi: 10.1002/int.21833.

[136] H. Peng, F. Long, and C. Ding, ‘Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy’, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005, doi: 10.1109/TPAMI.2005.159.

[137] C. Ding and H. Peng, ‘Minimum redundancy feature selection from microarray gene expression data’, *J. Bioinform. Comput. Biol.*, vol. 03, no. 02, pp. 185–205, Apr. 2005, doi: 10.1142/S0219720005001004.

[138] K. Chrysostomou, ‘Wrapper Feature Selection’, in *Encyclopedia of Data Warehousing and Mining, Second Edition*, IGI Global, 2009, pp. 2103–2108. doi: 10.4018/978-1-60566-010-3.ch322.

[139] J. Bins and B. A. Draper, ‘Feature selection from huge feature sets’, in *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, Jul. 2001, pp. 159–165 vol.2. doi: 10.1109/ICCV.2001.937619.

[140] R. Kohavi and G. H. John, ‘The Wrapper Approach’, in *Feature Extraction, Construction and Selection: A Data Mining Perspective*, H. Liu and H. Motoda, Eds., in The Springer International Series in Engineering and Computer Science. , Boston, MA: Springer US, 1998, pp. 33–50. doi: 10.1007/978-1-4615-5725-8_3.

[141] J. Pirgazi, M. Alimoradi, T. Esmaeili Abharian, and M. H. Olyaee, ‘An Efficient hybrid filter-wrapper metaheuristic-based gene selection method for high dimensional datasets’, *Sci Rep*, vol. 9, no. 1, Art. no. 1, Dec. 2019, doi: 10.1038/s41598-019-54987-1.

[142] A. W. Whitney, ‘A Direct Method of Nonparametric Measurement Selection’, *IEEE Transactions on Computers*, vol. C–20, no. 9, pp. 1100–1103, Sep. 1971, doi: 10.1109/T-C.1971.223410.

[143] M. Tharmakulasingam, C. Topal, A. Fernando, and R. L. Ragione, ‘Backward Feature Elimination for Accurate Pathogen Recognition Using Portable

Electronic Nose', in *2020 IEEE International Conference on Consumer Electronics (ICCE)*, Jan. 2020, pp. 1–5. doi: 10.1109/ICCE46568.2020.9043043.

[144] T. Nisia and S. Rajesh, 'Enhanced Feature Selection Method Using Wrapper-based Random Search Strategy and Mutual Information for Remote Sensing Image Classification', *Proceedings of the Bulgarian Academy of Sciences*, vol. 75, no. 7, Art. no. 7, Jul. 2022, doi: 10.7546/CRABS.2022.07.12.

[145] Z. Wang, X. Xiao, and S. Rajasekaran, 'Novel and efficient randomized algorithms for feature selection', *Big Data Mining and Analytics*, vol. 3, no. 3, pp. 208–224, Sep. 2020, doi: 10.26599/BDMA.2020.9020005.

[146] W. Bouaguel, 'A New Approach for Wrapper Feature Selection Using Genetic Algorithm for Big Data', in *Intelligent and Evolutionary Systems*, K. Lavangnananda, S. Phon-Amnuaisuk, W. Engchuan, and J. H. Chan, Eds., in Proceedings in Adaptation, Learning and Optimization. Cham: Springer International Publishing, 2016, pp. 75–83. doi: 10.1007/978-3-319-27000-5_6.

[147] L. Zahedi, F. Ghareh Mohammadi, and M. H. Amini, 'A2BCF: An Automated ABC-Based Feature Selection Algorithm for Classification Models in an Education Application', *Applied Sciences*, vol. 12, no. 7, Art. no. 7, Jan. 2022, doi: 10.3390/app12073553.

[148] B. Venkatesh and J. Anuradha, 'A Review of Feature Selection and Its Methods', *Cybernetics and Information Technologies*, vol. 19, no. 1, pp. 3–26, Mar. 2019.

[149] M. A. Haque, *Feature Engineering & Selection for Explainable Models*. Leanpub, 2022. Accessed: Feb. 20, 2024. [Online]. Available: https://leanpub.next/feature-engineering-and-selection-for-explainable-models-a-second-course-data-scientists

[150] N. Mahendran and D. R. V. P m, 'A deep learning framework with an embedded-based feature selection approach for the early detection of the Alzheimer's disease', *Computers in Biology and Medicine*, vol. 141, p. 105056, Feb. 2022, doi: 10.1016/j.compbiomed.2021.105056.

[151] O. Demir-Kavuk, M. Kamada, T. Akutsu, and E.-W. Knapp, 'Prediction using step-wise L1, L2 regularization and feature selection for small data sets with large number of features', *BMC Bioinformatics*, vol. 12, no. 1, p. 412, Oct. 2011, doi: 10.1186/1471-2105-12-412.

[152] H. Zhou, J. Zhang, Y. Zhou, X. Guo, and Y. Ma, 'A feature selection algorithm of decision tree based on feature weight', *Expert Systems with Applications*, vol. 164, p. 113842, Feb. 2021, doi: 10.1016/j.eswa.2020.113842.

[153] J. R. Quinlan, 'Induction of decision trees', *Mach Learn*, vol. 1, no. 1, pp. 81–106, Mar. 1986, doi: 10.1007/BF00116251.

[154] J. R. Quinlan, *C4.5: Programs for Machine Learning*. Elsevier, 2014.

[155] R. A. Berk, 'Classification and Regression Trees (CART)', in *Statistical Learning from a Regression Perspective*, in Springer Series in Statistics. , New York, NY: Springer, 2008, pp. 1–65. doi: 10.1007/978-0-387-77501-2_3.

[156] A. Alsahaf, N. Petkov, V. Shenoy, and G. Azzopardi, 'A framework for feature selection through boosting', *Expert Systems with Applications*, vol. 187, p. 115895, Jan. 2022, doi: 10.1016/j.eswa.2021.115895.

[157] Y. Saeys, I. Inza, and P. Larrañaga, ‘A review of feature selection techniques in bioinformatics’, *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, Oct. 2007, doi: 10.1093/bioinformatics/btm344.

[158] T. Elemam and M. Elshrkawey, ‘A Highly Discriminative Hybrid Feature Selection Algorithm for Cancer Diagnosis’, *The Scientific World Journal*, vol. 2022, p. e1056490, Aug. 2022, doi: 10.1155/2022/1056490.

[159] T. Butler-Yeoman, B. Xue, and M. Zhang, ‘Particle swarm optimisation for feature selection: A hybrid filter-wrapper approach’, in *2015 IEEE congress on evolutionary computation (CEC)*, IEEE, 2015, pp. 2428–2435.

[160] C. Chu, A.-L. Hsu, K.-H. Chou, P. Bandettini, and C. Lin, ‘Does feature selection improve classification accuracy? Impact of sample size and feature selection on classification using anatomical magnetic resonance images’, *NeuroImage*, vol. 60, no. 1, pp. 59–70, Mar. 2012, doi: 10.1016/j.neuroimage.2011.11.066.

[161] M. Yousef, A. Kumar, and B. Bakir-Gungor, ‘Application of Biological Domain Knowledge Based Feature Selection on Gene Expression Data’, *Entropy*, vol. 23, no. 1, Art. no. 1, Jan. 2021, doi: 10.3390/e23010002.

[162] J. Qi and J. Tang, ‘Integrating gene ontology into discriminative powers of genes for feature selection in microarray data’, in *Proceedings of the 2007 ACM symposium on Applied computing*, in SAC ’07. New York, NY, USA: Association for Computing Machinery, Mar. 2007, pp. 430–434. doi: 10.1145/1244002.1244101.

[163] Z.-H. Zhou, *Machine Learning*. Singapore: Springer, 2021. doi: 10.1007/978-981-15-1967-3.

[164] G. Brown, ‘Ensemble Learning’, in *Encyclopedia of Machine Learning*, C. Sammut and G. I. Webb, Eds., Boston, MA: Springer US, 2010, pp. 312–320. doi: 10.1007/978-0-387-30164-8_252.

[165] S. Buyrukoğlu and A. Akbaş, ‘Efficiency of Ensemble Learning Algorithms in the Analysis of Effects of Covid-19 Pandemic on Electricity Consumption in Turkey’, *INOTECH*, vol. 1, no. 1, Art. no. 1, Jun. 2022.

[166] T. N. Rincy and R. Gupta, ‘Ensemble Learning Techniques and its Efficiency in Machine Learning: A Survey’, in *2nd International Conference on Data, Engineering and Applications (IDEA)*, Feb. 2020, pp. 1–6. doi: 10.1109/IDEA49133.2020.9170675.

[167] V. Bolón-Canedo and A. Alonso-Betanzos, ‘Ensembles for feature selection: A review and future trends’, *Information Fusion*, vol. 52, pp. 1–12, Dec. 2019, doi: 10.1016/j.inffus.2018.11.008.

[168] B. Seijo-Pardo, I. Porto-Díaz, V. Bolón-Canedo, and A. Alonso-Betanzos, ‘Ensemble feature selection: Homogeneous and heterogeneous approaches’, *Knowledge-Based Systems*, vol. 118, pp. 124–139, Feb. 2017, doi: 10.1016/j.knosys.2016.11.017.

[169] H. Nguyen, S. Shrestha, D. Tran, A. Shafi, S. Draghici, and T. Nguyen, ‘A Comprehensive Survey of Tools and Software for Active Subnetwork Identification’, *Frontiers in Genetics*, vol. 10, 2019, Accessed: Feb. 22, 2024. [Online]. Available: https://www.frontiersin.org/journals/genetics/articles/10.3389/fgene.2019.00155

[170] Y. E. Işık, Y. Görmez, Z. Aydın, and B. Bakir-Gungor, 'The Determination of Distinctive Single Nucleotide Polymorphism Sets for the Diagnosis of Behçet's Disease', *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 19, no. 3, pp. 1909–1918, May 2022, doi: 10.1109/TCBB.2021.3053429.

[171] C. Wieder *et al.*, 'Pathway analysis in metabolomics: Recommendations for the use of over-representation analysis', *PLOS Computational Biology*, vol. 17, no. 9, p. e1009105, Eyl 2021, doi: 10.1371/journal.pcbi.1009105.

[172] P. Khatri, M. Sirota, and A. J. Butte, 'Ten Years of Pathway Analysis: Current Approaches and Outstanding Challenges', *PLOS Computational Biology*, vol. 8, no. 2, p. e1002375, ub 2012, doi: 10.1371/journal.pcbi.1002375.

[173] S. Lee, Y. Park, and S. Kim, 'MIDAS: Mining differentially activated subpaths of KEGG pathways from multi-class RNA-seq data', *Methods*, vol. 124, pp. 13–24, Jul. 2017, doi: 10.1016/j.ymeth.2017.05.026.

[174] M. Fernandes and H. Husi, 'ORA, FCS, and PT Strategies in Functional Enrichment Analysis', in *Proteomics Data Analysis*, D. Cecconi, Ed., in Methods in Molecular Biology. , New York, NY: Springer US, 2021, pp. 163–178. doi: 10.1007/978-1-0716-1641-3_10.

[175] F. Maleki, K. Ovens, D. J. Hogan, and A. J. Kusalik, 'Gene Set Analysis: Challenges, Opportunities, and Future Research', *Frontiers in Genetics*, vol. 11, 2020, Accessed: Feb. 22, 2024. [Online]. Available: https://www.frontiersin.org/journals/genetics/articles/10.3389/fgene.2020.00654

[176] P. D. Karp, P. E. Midford, R. Caspi, and A. Khodursky, 'Pathway size matters: the influence of pathway granularity on over-representation (enrichment analysis) statistics', *BMC Genomics*, vol. 22, no. 1, p. 191, Mar. 2021, doi: 10.1186/s12864-021-07502-8.

[177] A. Subramanian *et al.*, 'Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles', *Proceedings of the National Academy of Sciences*, vol. 102, no. 43, pp. 15545–15550, Oct. 2005, doi: 10.1073/pnas.0506580102.

[178] W. He *et al.*, 'Key genes and pathways in thyroid cancer based on gene set enrichment analysis', *Oncology Reports*, vol. 30, no. 3, pp. 1391–1397, Sep. 2013, doi: 10.3892/or.2013.2557.

[179] J. Shi and M. G. Walker, 'Gene Set Enrichment Analysis (GSEA) for Interpreting Gene Expression Profiles', *Current Bioinformatics*, vol. 2, no. 2, pp. 133–137, May 2007, doi: 10.2174/157489307780618231.

[180] D. H. Johnson, 'The Insignificance of Statistical Significance Testing', *The Journal of Wildlife Management*, vol. 63, no. 3, pp. 763–772, 1999, doi: 10.2307/3802789.

[181] H. Alibrahim and S. A. Ludwig, 'Hyperparameter optimization: Comparing genetic algorithm against grid search and bayesian optimization', in *2021 IEEE Congress on Evolutionary Computation (CEC)*, IEEE, 2021, pp. 1551–1559.

[182] M. Kuhn and K. Johnson, 'Over-Fitting and Model Tuning', in *Applied Predictive Modeling*, M. Kuhn and K. Johnson, Eds., New York, NY: Springer, 2013, pp. 61–92. doi: 10.1007/978-1-4614-6849-3_4.

[183] A. Stuke, P. Rinke, and M. Todorović, ‘Efficient hyperparameter tuning for kernel ridge regression with Bayesian optimization’, *Mach. Learn.: Sci. Technol.*, vol. 2, no. 3, p. 035022, Jun. 2021, doi: 10.1088/2632-2153/abee59.

[184] S. Shekhar, A. Bansode, and A. Salim, ‘A comparative study of hyper-parameter optimization tools’, in *2021 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*, IEEE, 2021, pp. 1–6.

[185] L. Yang and A. Shami, ‘On hyperparameter optimization of machine learning algorithms: Theory and practice’, *Neurocomputing*, vol. 415, pp. 295–316, Nov. 2020, doi: 10.1016/j.neucom.2020.07.061.

[186] A. Dhillon, A. Singh, and V. K. Bhalla, ‘Biomarker identification and cancer survival prediction using random spatial local best cat swarm and Bayesian optimized DNN’, *Applied Soft Computing*, vol. 146, p. 110649, Oct. 2023, doi: 10.1016/j.asoc.2023.110649.

[187] G. De Ath, R. M. Everson, A. A. M. Rahat, and J. E. Fieldsend, ‘Greed Is Good: Exploration and Exploitation Trade-offs in Bayesian Optimisation’, *ACM Trans. Evol. Learn. Optim.*, vol. 1, no. 1, p. 1:1-1:22, Nisan 2021, doi: 10.1145/3425501.

[188] Y. Jiao and P. Du, ‘Performance measures in evaluating machine learning based bioinformatics predictors for classifications’, *Quant Biol*, vol. 4, no. 4, pp. 320–330, Dec. 2016, doi: 10.1007/s40484-016-0081-2.

[189] T. Fawcett, ‘An introduction to ROC analysis’, *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, Jun. 2006, doi: 10.1016/j.patrec.2005.10.010.

[190] C. S. Beam, ‘Resolving power: A general approach to compare the discriminating capacity of threshold-free evaluation metrics’. arXiv, Mar. 31, 2023. doi: 10.48550/arXiv.2304.00059.

[191] B. Ozenne, F. Subtil, and D. Maucort-Boulch, ‘The precision–recall curve overcame the optimism of the receiver operating characteristic curve in rare diseases’, *Journal of Clinical Epidemiology*, vol. 68, no. 8, pp. 855–859, Aug. 2015, doi: 10.1016/j.jclinepi.2015.02.010.

[192] I. Strandén and O. F. Christensen, ‘Allele coding in genomic evaluation’, *Genetics Selection Evolution*, vol. 43, no. 1, p. 25, Jun. 2011, doi: 10.1186/1297-9686-43-25.

[193] L. C. Lazzeroni, Y. Lu, and I. Belitskaya-Lévy, ‘P-values in genomics: Apparent precision masks high uncertainty’, *Mol Psychiatry*, vol. 19, no. 12, Art. no. 12, Dec. 2014, doi: 10.1038/mp.2013.184.

[194] Y. Zhang, ‘On The Use of P-Values in Genome Wide Disease Association Mapping’, *J Biom Biostat*, vol. 7, no. 3, 2016, doi: 10.4172/2155-6180.1000297.

[195] F. Pedregosa *et al.*, ‘Scikit-learn: Machine learning in Python’, *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[196] E. Frank *et al.*, ‘Weka-A Machine Learning Workbench for Data Mining’, in *Data Mining and Knowledge Discovery Handbook*, O. Maimon and L. Rokach, Eds., Boston, MA: Springer US, 2010, pp. 1269–1277. doi: 10.1007/978-0-387-09823-4_66.

[197] J. Li *et al.*, 'Feature Selection: A Data Perspective', *ACM Comput. Surv.*, vol. 50, no. 6, p. 94:1-94:45, Aralık 2017, doi: 10.1145/3136625.

[198] B. Bakir-Gungor *et al.*, 'Identification of possible pathogenic pathways in Behçet's disease using genome-wide association study data from two different populations', *Eur J Hum Genet*, vol. 23, no. 5, Art. no. 5, May 2015, doi: 10.1038/ejhg.2014.158.

[199] S. F. Saccone *et al.*, 'SPOT: a web-based tool for using biological databases to prioritize SNPs after a genome-wide association study', *Nucleic acids research*, vol. 38, no. suppl_2, pp. W201–W209, 2010.

[200] H. Zhan, H. Li, L. Cheng, S. Yan, W. Zheng, and Y. Li, 'Novel Insights Into Gene Signatures and Their Correlation With Immune Infiltration of Peripheral Blood Mononuclear Cells in Behcet's Disease', *Frontiers in Immunology*, vol. 12, 2021, Accessed: Feb. 22, 2024. [Online]. Available: https://www.frontiersin.org/journals/immunology/articles/10.3389/fimmu.2021.794800

[201] T.-Y. Liu *et al.*, 'An individualized predictor of health and disease using paired reference and target samples', *BMC Bioinformatics*, vol. 17, no. 1, p. 47, Jan. 2016, doi: 10.1186/s12859-016-0889-9.

[202] G. G. JACKSON, H. F. DOWLING, T. O. ANDERSON, L. Riff, J. SAPORTA, and M. TURCK, 'Susceptibility and immunity to common upper respiratory viral infections—the common cold', *Annals of Internal Medicine*, vol. 53, no. 4, pp. 719–738, 1960.

[203] L. Gautier, L. Cope, B. M. Bolstad, and R. A. Irizarry, 'affy—analysis of Affymetrix GeneChip data at the probe level', *Bioinformatics*, vol. 20, no. 3, pp. 307–315, Feb. 2004, doi: 10.1093/bioinformatics/btg405.

[204] B. M. Bolstad, R. A. Irizarry, M. Åstrand, and T. P. Speed, 'A comparison of normalization methods for high density oligonucleotide array data based on variance and bias', *Bioinformatics*, vol. 19, no. 2, pp. 185–193, Jan. 2003, doi: 10.1093/bioinformatics/19.2.185.

[205] R. A. Irizarry, B. M. Bolstad, F. Collin, L. M. Cope, B. Hobbs, and T. P. Speed, 'Summaries of Affymetrix GeneChip probe level data', *Nucleic Acids Research*, vol. 31, no. 4, p. e15, Feb. 2003, doi: 10.1093/nar/gng015.

[206] M. Griffith, J. R. Walker, N. C. Spies, B. J. Ainscough, and O. L. Griffith, 'Informatics for RNA Sequencing: A Web Resource for Analysis on the Cloud', *PLOS Computational Biology*, vol. 11, no. 8, p. e1004393, Ağu 2015, doi: 10.1371/journal.pcbi.1004393.

[207] L. Yu, H. Qu, Q. Jia, X. Wang, and Z. Jia, 'Transformation, Normalization, and Batch Effect Removal', *Bio-101*, vol. 12, no. 14, p. e4462, Jul. 2022, doi: 10.21769/BioProtoc.4462.

[208] W. E. Johnson, C. Li, and A. Rabinovic, 'Adjusting batch effects in microarray expression data using empirical Bayes methods', *Biostatistics*, vol. 8, no. 1, pp. 118–127, Jan. 2007, doi: 10.1093/biostatistics/kxj037.

[209] A. Behdenna *et al.*, 'pyComBat, a Python tool for batch effects correction in high-throughput molecular data using empirical Bayes methods', *BMC*

Bioinformatics, vol. 24, no. 1, p. 459, Dec. 2023, doi: 10.1186/s12859-023-05578-5.

[210] M. A. Stalteri and A. P. Harrison, 'Interpretation of multiple probe sets mapping to the same gene in Affymetrix GeneChips', *BMC Bioinformatics*, vol. 8, no. 1, p. 13, Jan. 2007, doi: 10.1186/1471-2105-8-13.

[211] A. Liberzon, A. Subramanian, R. Pinchback, H. Thorvaldsdóttir, P. Tamayo, and J. P. Mesirov, 'Molecular signatures database (MSigDB) 3.0', *Bioinformatics*, vol. 27, no. 12, pp. 1739–1740, Jun. 2011, doi: 10.1093/bioinformatics/btr260.

[212] M. Reich, T. Liefeld, J. Gould, J. Lerner, P. Tamayo, and J. P. Mesirov, 'GenePattern 2.0', *Nat Genet*, vol. 38, no. 5, Art. no. 5, May 2006, doi: 10.1038/ng0506-500.

[213] Y. E. Işık and Z. Aydın, 'Comparative analysis of machine learning approaches for predicting respiratory virus infection and symptom severity', *PeerJ*, vol. 11, p. e15552, Jun. 2023, doi: 10.7717/peerj.15552.

[214] S. Fourati *et al.*, 'A crowdsourced analysis to identify ab initio molecular signatures predictive of susceptibility to viral infection', *Nat Commun*, vol. 9, no. 1, Art. no. 1, Oct. 2018, doi: 10.1038/s41467-018-06735-8.

[215] C. An, Y. W. Park, S. S. Ahn, K. Han, H. Kim, and S.-K. Lee, 'Radiomics machine learning study with a small sample size: Single random training-test set split may lead to unreliable results', *PLOS ONE*, vol. 16, no. 8, p. e0256152, Ağu 2021, doi: 10.1371/journal.pone.0256152.

[216] T.-T. Wong, 'Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation', *Pattern recognition*, vol. 48, no. 9, pp. 2839–2846, 2015.

[217] T. Head *et al.*, 'scikit-optimize/scikit-optimize: v0.5.2'. Zenodo, Mar. 25, 2018. doi: 10.5281/zenodo.1207017.

[218] S. L. Pett *et al.*, 'Increased Indoleamine-2,3-Dioxygenase Activity Is Associated With Poor Clinical Outcome in Adults Hospitalized With Influenza in the INSIGHT FLU003Plus Study', *Open Forum Infectious Diseases*, vol. 5, no. 1, p. ofx228, Jan. 2018, doi: 10.1093/ofid/ofx228.

[219] P. M. Varghese *et al.*, 'C4b Binding Protein Acts as an Innate Immune Effector Against Influenza A Virus', *Frontiers in Immunology*, vol. 11, 2021, Accessed: Feb. 22, 2024. [Online]. Available: https://www.frontiersin.org/journals/immunology/articles/10.3389/fimmu.2020.585361

[220] K. Zheng, Z. Ren, and Y. Wang, 'Serine-arginine protein kinases and their targets in viral infection and their inhibition', *Cell. Mol. Life Sci.*, vol. 80, no. 6, p. 153, May 2023, doi: 10.1007/s00018-023-04808-6.

[221] J. C. Rupp *et al.*, 'Host Cell Copper Transporters CTR1 and ATP7A are important for Influenza A virus replication', *Virol J*, vol. 14, no. 1, p. 11, Jan. 2017, doi: 10.1186/s12985-016-0671-7.

[222] C. Rajput, M. P. Walsh, B. N. Eder, E. E. Metitiri, A. P. Popova, and M. B. Hershenson, 'Rhinovirus infection induces distinct transcriptome profiles in polarized human macrophages', *Physiological Genomics*, vol. 50, no. 5, pp. 299–312, May 2018, doi: 10.1152/physiolgenomics.00122.2017.

[223] S. Abbasi *et al.*, ‘Impact of human rhinoviruses on gene expression in pediatric patients with severe acute respiratory infection’, *Virus Research*, vol. 300, p. 198408, Jul. 2021, doi: 10.1016/j.virusres.2021.198408.

[224] S. Heinonen *et al.*, ‘Rhinovirus Detection in Symptomatic and Asymptomatic Children: Value of Host Transcriptome Analysis’, *Am J Respir Crit Care Med*, vol. 193, no. 7, pp. 772–782, Apr. 2016, doi: 10.1164/rccm.201504-0749OC.

[225] M. Tsuda *et al.*, ‘Activation of granulocytes by direct interaction with dendritic cells’, *Clinical and Experimental Immunology*, vol. 150, no. 2, pp. 322–331, Nov. 2007, doi: 10.1111/j.1365-2249.2007.03490.x.

[226] C. M. Gelder *et al.*, ‘Associations between Human Leukocyte Antigens and Nonresponsiveness to Influenza Vaccine’, *The Journal of Infectious Diseases*, vol. 185, no. 1, pp. 114–117, Jan. 2002, doi: 10.1086/338014.

[227] J. Barratt and I. Weitz, ‘Complement Factor D as a Strategic Target for Regulating the Alternative Complement Pathway’, *Frontiers in Immunology*, vol. 12, 2021, Accessed: Feb. 22, 2024. [Online]. Available: https://www.frontiersin.org/journals/immunology/articles/10.3389/fimmu.2021.712572

[228] P. Meyer and J. Saez-Rodriguez, ‘Advances in systems biology modeling: 10 years of crowdsourcing DREAM challenges’, *Cell Systems*, vol. 12, no. 6, pp. 636–653, Jun. 2021, doi: 10.1016/j.cels.2021.05.015.

[229] R. Pastorino, C. Loreti, S. Giovannini, W. Ricciardi, L. Padua, and S. Boccia, ‘Challenges of prevention for a sustainable personalized medicine’, *Journal of Personalized Medicine*, vol. 11, no. 4, p. 311, 2021.

[230] W. M. Sweileh, ‘Bibliometric analysis of scientific publications on “sustainable development goals” with emphasis on “good health and well-being” goal (2015–2019)’, *Globalization and health*, vol. 16, pp. 1–13, 2020.

CURRICULUM VITAE

2009 – 2013	Baclehor, Management Information Systems, Mehmet Akif Ersoy University, Burdur, TURKEY
2015 – 2018	Master, Management Information Systems, Sivas Cumhuriyet University, Sivas, TURKEY
2018 – 2024	Doctoral Candidate, Electrical and Computer Engineering, Abdullah Gul University, Kayseri, TÜRKİYE
2015-	Research Assistant, Management Information Systems, Sivas Cumhuriyet University, Sivas, TURKEY

SELECTED PUBLICATIONS AND PRESENTATIONS

J1) Işık, Y. E., Görmez, Y., Aydın, Z., & Bakir-Gungor, B. "The Determination of Distinctive Single Nucleotide Polymorphism Sets for the Diagnosis of Behçet's Disease", IEEE/ACM Transactions on Computational Biology and Bioinformatics (2021)

J2) Işık, Y. E., & Aydın, Z. "Comparative analysis of machine learning approaches for predicting respiratory virus infection and symptom severity", PeerJ (2023).

C1) Görmez, Y., **Işık, Y. E.,** & Bakır-Güngör, B. "The Identification of Discriminative Single Nucleotide Polymorphism Sets for the Classification of Behçet's Disease". 3rd International Conference on Computer Science and Engineering, IEEE, pp. 443-447, (2018)