Ranked MSD: A New Feature Ranking and Feature Selection Approach for Biomarker Identification

Ghanshyam Verma, Alokkumar Jha, Dietrich Rebholz-Schuhmann, Michael G. Madden

▶ **To cite this version:**

Ghanshyam Verma, Alokkumar Jha, Dietrich Rebholz-Schuhmann, Michael G. Madden. Ranked MSD: A New Feature Ranking and Feature Selection Approach for Biomarker Identification. 3rd International Cross-Domain Conference for Machine Learning and Knowledge Extraction (CD-MAKE), Aug 2019, Canterbury, United Kingdom. pp.147-167, 10.1007/978-3-030-29726-8_10 . hal-02520052

HAL Id: hal-02520052
https://inria.hal.science/hal-02520052

Submitted on 26 Mar 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Ranked MSD: A New Feature Ranking and Feature Selection Approach for Biomarker Identification

Ghanshyam Verma[1,2], Alokkumar Jha[1,2], Dietrich Rebholz-Schuhmann[3], and Michael G. Madden[1,2]

[1]Insight Centre for Data Analytics, National University of Ireland Galway, Ireland
[2]School of Computer Science, National University of Ireland Galway, Ireland
[3]ZB MED - Information Center for Life Sciences, University of Cologne, Germany
{ghanshyam.verma, alokkumar.jha}@insight-centre.org, rebholz@zbmed.de, michael.madden@nuigalway.ie

Abstract. In the era of big data when a huge amount of data is continuously being generated, it is common for situations to arise where the number of samples is much smaller than the number of features (variables) per sample. This phenomenon is often found in biomedical domains, where we may have relatively few patients, compared to the amount of data per patient. For example, gene expression data typically has between 10,000 and 60,000 features per sample. A separate issue arises from the "right to explanation" found in the European General Data Protection Regulation (GDPR), which may prevent the use of black-box models in applications where explainability is required. In such situations, there is a need for robust algorithms which can identify the relevant features from experimental data by discarding irrelevant ones, yielding a simpler subset that facilitates explanation. To address these needs, we have developed a new algorithm for feature ranking and feature selection, named *Ranked MSD*. We have tested our proposed approach on two real-world gene expression data sets, both of which relate to respiratory viral infections. This Ranked MSD feature selection algorithm is able to reduce the feature set size from 12,023 genes (features) to 65 genes on the first data set and from 20,737 genes to 31 genes on the second data set, in both cases without any significant loss in disease prediction accuracy. In an alternative configuration, our proposed algorithm is able to identify a small subset of features that gives better accuracy than that of the full feature set. Our proposed algorithm can also identify important biomarkers (genes) with their importance score for a particular disease and the identified top-ranked biomarkers can play a vital role in drug discovery and precision medicine.

Keywords: Machine learning · Respiratory viral infection · Feature Ranking · Feature Selection · Classification · Explainable AI.

1 Introduction

It has been observed that the use of ML (machine learning) algorithms has been increased in healthcare applications that deeply impact the life of patients [24]. The term "black-box model" is used for those ML models that fail to explain their predictions in a way that humans can understand and make some meaningful conclusions for decision making. According to Rudin [17] due to the lack of proper explanation and transparency of black box models, there might be severe consequences of using them for decision making specifically in health, finance and in the domains where people are directly involved. Therefore, rather than using black-box models without a proper explanation if we use models that are explainable or use secondary analyses to generate explanations from black-box models, then it can surely help in better decision making, particularly in the medical domain where medical professionals want to understand how and why a machine decision has been made. Moreover, algorithms that can facilitate meaningful explanations could enhance the trust of medical professionals in future AI or ML based systems [11].

In supervised machine learning, a classification algorithm is learned by applying it to a set of training samples or instances, where each instance contains a vector of attribute values (also called features) and a class [16]. For example, in genomics, the features are generally genes (of which there are typically thousands) and the class label might denote whether or not a patient is infected. Here the problem is that we have thousands of genes and all the genes are not relevant for a particular problem or disease. We are only interested in very few most important genes which can be targeted for further study or drug discovery. Therefore, it is very important to identify and select the very few most important genes. One way to solve this problem is to use an appropriate feature selection technique. Most of the machine learning algorithms are designed in such a way that they learn which are the most important features to use for decision making. In theory, they should never select irrelevant and unhelpful features. But there is a difference between theory and practice. In practice, irrelevant or distracting features often confuse machine learning systems and lead to deterioration of classification accuracy [31]. Having a large set of irrelevant features require excessive computational time and memory space. Moreover, a large number of irrelevant features make it very hard to interpret the representation of the target concept. Because of these bad effects of irrelevant features, it is common to perform feature selection before applying any learning algorithm.

When we want to perform feature selection, there are two different broad types of approaches. The first type are *filter* methods, which make an assessment of feature importance based on general characteristics of the data, using criteria that are independent of the subsequent machine learning algorithm. The second type are *wrapper* methods. Wrapper methods start with an empty set of features, and iteratively add/remove features until an optimal feature set is found [31]. An important property of filter methods is that they can assign a score to all the features, based on which features can be ranked in the desired order. This is useful when we need to select the top few features for further analysis. On the

other hand, wrapper methods do not, in general, have any mechanism to rank features.

In this work, our overall goal is to identify the important biomarkers or genes for a particular disease, and assign an importance score to each biomarker. Therefore, the focus of this paper is on filter methods, and wrapper methods are not applicable. In this paper, we propose a feature ranking algorithm named *Ranked MSD* which gives a ranked list of all the features with their importance score. We also propose two more feature selection algorithms $\mathcal{F}_{equal}$ and $\mathcal{F}_{best}$. $\mathcal{F}_{equal}$ gives most strongly relevant features and $\mathcal{F}_{best}$ gives all the relevant features by discarding irrelevant features. The overall approach can identify the most important biomarkers and can help in explaining the predictions.

The rest of the paper is structured as follows. In section 2, we describe related work. Section 3 describes the two real-world data sets used to perform experiments. In section 4, we explain the proposed algorithms. In section 5 and 6, we present the overall experimental design and methodology used. In section 7, we discuss how we can explain predictions using our approach. In section 8, we discuss results in detail with comparative analysis. In section 9, we present the identified biomarkers using proposed algorithms. In section 10, we discuss the significance of identified biomarkers, and finally we conclude in section 11.

2 Related Work

A basic and natural way to interpretability is to provide explanations of an ML model's predictions in the form of input features [14]. This is the reason most of the work that tried to explain the predictions of black-box models used in some sense the features that have some influence on the class of interest [14, 23, 26]. Riberio et al. proposed an approach called LIME [23] which can explain the prediction of a classifier by providing a small list of features that either contribute to the prediction or are evidence against the prediction. In our work, using Ranked MSD approach we are also suggesting a small list of features with their importance score that can explain the predictions of a classifier. This feature importance score also denotes class discriminative power of that feature. For computing contribution of features, LIME uses K-Lasso an approach based on Lasso [8] and we are using our proposed approach which is explained in Algorithm 1. Filter methods can be used to compute contribution or importance of features. Most of the filter methods use feature ranking as a principle mechanism for feature selection [9]. Feature ranking is a type of preprocessing that ranks features in ascending or descending order of their relevance to the class label based on a computed score for each variable or feature. A suitable threshold is then used to select the top ranked features [4]. Filter methods are not dependent on the choice of the classifier or predictor. However, under certain assumptions, it may produce optimal solution for a given predictor [28]. One of the most important properties of a good feature is that it contains useful information of the different possible classes in the data. This property is known as feature relevance [16], and relates to the usefulness of a feature in discrimination of classes.

In the following sub-sections, we will discuss two state-of-the-art filter methods, against which we will compare our proposed feature ranking algorithm.

2.1 Correlation Criteria

As described by Guyon et al. [9], the Pearson correlation coefficient can be defined as:

$$R(i) = \frac{Cov\left(X_i, Y\right)}{\sqrt{Var\left(X_i\right) Var\left(Y\right)}} \tag{1}$$

Here Cov denotes covariance and Var denotes the variance. X_i denotes the i^{th} feature vector and Y denotes the outcome. $R(i)$ represents the fraction of the total variance around the mean value $\overline{y}$, therefore, the $R(i)^2$ can be used as a variable ranking criterion, with which we can rank features in ascending or descending order. $R(i)^2$ can be used for two-class classification, for which each class label is mapped to a given value of Y, e.g., 0 & 1.

2.2 Information Gain

Another well-known feature selection approach is Information Gain, which can be classed as an information theoretic ranking criterion [4, 9, 16]. It is based on Shannon's definition of entropy which can be represented as:

$$H\left(Y\right) = -\sum_{y} p\left(y\right) log\left(p\left(y\right)\right) \tag{2}$$

This formula represents the information content or uncertainty in any variable Y. Now if we observe a new variable X, then the conditional entropy can be represented by the following formula:

$$H\left(Y|X\right) = -\sum_{x}\sum_{y} p\left(x, y\right) log\left(p\left(y|x\right)\right) \tag{3}$$

This formula says that by observing a variable X, the uncertainty of the output or variable Y is reduced. Now the formula for Information Gain IG can be represented as:

$$IG(Y, X) = H(Y) + H(X) - H(Y|X) \tag{4}$$

Here $H(Y)$ is the information content or uncertainty of class variable Y, the second term $H(X)$ is the information content of observed variable X and $H(Y|X)$ is the conditional entropy .

3 The Respiratory Viral Data Sets

We have conducted experiments on two real-world data sets, both of which are related to respiratory viral infections. The first data set is collected from 7 Respiratory Viral Challenge studies which is available for open access on Gene Expression Omnibus (GEO) using accession number GSE73072[4]. This first data

[4] https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE73072.

set consists of a total of 151 human volunteers. All the volunteers were healthy when they enrolled for the study. After enrolment in the study, all subjects were inoculated with one of the 4 viruses (H1N1, H3N2, HRV, RSV). Their blood samples were taken at different pre-defined time-points, thus delivering gene expression profiles from non-infected individuals as well as from infected ones [19]. The details related to labels and other additional details of this data set can be found on GEO (accession number GSE73072). From the start, total 151 subjects were enrolled in these 7 challenge studies, however, we have to exclude 47 subjects from the study because those subjects' gene expression data are either inconsistent (faulty) or missing, and faulty data can be misleading and harmful while model-building. Detailed information about the excluded subjects can be found in a paper that used this dataset before [30].

The second dataset contains gene expression profiles of 133 adults whose samples are taken in three different seasons - Fall, Winter and Spring. Baseline samples are taken at the time of enrolment of volunteers (Fall season). Day 0, Day 2, Day 4, Day 6 and Day 21 samples are taken during the winter season (Influenza season). Samples of all the volunteers are taken again in Spring season. For each volunteer, samples are taken at up to seven time points before, during, and after the occurrence of illness (Influenza and other acute respiratory viral infections). Among those seven time points, the samples taken before illness (baseline), at day 21 and during Spring season are healthy samples and rest of the samples are infected samples as they are taken during the illness (day 0, day 2, day 4 and day 6). A total of 890 microarray samples were collected. Any samples that failed Quality Control were excluded from the study (N = 10), leaving 880 high-quality arrays from which the subsequent analysis was conducted. Out of these 880 samples, 373 samples are healthy samples and rest of 507 samples are infected samples [33]. There were in total 47,254 probe IDs in each microarray sample from which 20,737 probe IDs have unique gene mapping; therefore, we left with a total of 20,737 genes for further analysis. This data set is also openly accessible on GEO via accession number GSE68310[5].

The first dataset contains in total 12,023 genes and the second data contains in total 20,737 genes; however, a large number of genes have little or no contribution in finding the progression of a particular disease, so it is crucial to find that small number of genes which actually provide diagnostic signals and contribute the most at the time of a particular disease progression. It is also important to understand their importance for that disease prediction and for finding treatment targeting those genes. In this work, using the proposed *Ranked MSD* feature selection algorithm, we are interested in finding the strongly relevant features (genes) which are potential biomarkers and contributing the most in respiratory viral disease prediction. Our software implementing our algorithm is open-source and freely available; R code for it can be accessed here: https://github.com/researher/Ranked_MSD.

[5] https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE68310.

4 Proposed Algorithms

We have developed a feature ranking algorithm and two additional schemes (Algorithm 2 and 3) for feature subset selection. One can observe these three algorithms as one but here we have designated them as 3 individual algorithms for simplicity of explanation; their purpose is to rank all the features, identify *strongly relevant* and *relevant* features respectively for a particular problem.

Algorithm 1 Ranked MSD Feature Ranking Algorithm

Input: The training data $Y_{tr} = [G_1, G_2, ..., G_m]_{n \times m}$, where n = number of samples, m = total number of features; a vector of class labels $C = [C_1, C_2, ..., C_p]$, where p = total number of classes and $p \geq 2$

Output: $\mathcal{RF}_{All}$ = Ranked list of all the m features

1: Create *Reference* vector $R_{1 \times m}$, where R is the feature wise mean of all the samples belong to class C_1 of Y_{tr}

2: Create *Target* matrix $T_{j \times m}$, where j = number of samples in T and T contains all the samples from Y_{tr} except class C_1

3: Compute Mean Squared Difference (MSD) for each feature

$$MSD_{1 \times m} = \frac{\sum_1^j (T_{j \times m} - R_{1 \times m})^2}{j}$$ ▷ MSD is a named vector that contains feature names and MSD values.

4: Compute $\mathcal{RF}_{All}$ by arranging the MSD in the descending order of their values.

5: Return $\mathcal{RF}_{All}$

The name of the proposed feature ranking algorithm is *Ranked MSD* which needs only training data in input and returns a ranked list of all the features ($\mathcal{RF}_{All}$) with their importance score. In the first step of Algorithm 1, we create a *Reference* vector $R_{1 \times m}$ which contains the feature wise mean of all the samples belong to a particular class say class one or base class. In our application, all these samples of class one belong to the negative class (all healthy samples) thus the *Reference* vector R represents the gene expression values of healthy samples on an average. In other words, the *Reference* vector R can be used as a representative of the base class. In the second step we create a *Target* matrix $T_{j \times m}$ which contains all the samples from training data except class one i.e. healthy samples. In the third step, we compute the Mean Squared Difference (MSD) for every feature (gene) from the formula shown in Algorithm 1. This MSD serves as a scoring function for the proposed algorithm. This scoring function preserves the difference between healthy gene expression values and infected gene expression values and gives the highest score to the highly differentially expressed gene and so on. Then in the next step, we compute the ranked list of all the features ($\mathcal{RF}_{All}$) by arranging the computed MSD score in descending order of their values. Here the MSD vector is a named vector which has feature names as heading and computed scores as values. Therefore when we arrange them in descending order of their values the most important genes are the top ones.

Once all the features are ranked, we may wish to select a subset of relevant features from ($\mathcal{RF}_{All}$). A simple solution is to take ranked features one by one

sequentially and evaluate them by training a classifier until we get the best accuracy. In the worst case, this method has high time complexity of order $O(m * T)$ where m is the total number of features and T is time complexity to train a classifier which depends on the classification algorithm used (and may have higher-order complexity). The proposed Algorithms 2 and 3 provide a less time-consuming solution for this problem in comparison to the sequential search. To understand the algorithm 2 and 3, first, we need to define the following terms.

Definition 1: (Feature Set with First Statistically Equal Accuracy) - $\mathcal{F}_{equal}$ is the feature set that has minimum number of top variables or genes (subset of full feature set) and gives accuracy which is statistically (according to a t-test) equivalent to the accuracy of using the full feature set.

Definition 2: (Feature Set with Best Accuracy) - $\mathcal{F}_{best}$ is a subset of the full feature set which gives the best possible accuracy.

Algorithm 2 Feature set with $\mathcal{F}_{equal}$ accuracy

Input: The training data $Y_{tr} = [G_1, G_2, ..., G_m]_{n \times m}$ and Ranked list of all the m features ($\mathcal{RF}_{All}$), Significance level (α).
Output: $\mathcal{F}_{equal}$ ▷ $\mathcal{F}_{equal}$ = Feature Size with First Statistically Equal Accuracy

function SEARCHFEQUAL(Y_{tr}, $\mathcal{RF}_{All}$, $L = 1$, $R = m$, α)
  $M = ceiling(\frac{L+R}{2})$
  Train the desired classifier using top $M-1$, M and $M+1$ features from $\mathcal{RF}_{All}$
  Perform ***t-test*** between $Acc(M-1, M, M+1)$ and $Acc(m)$ individually
  if ((*p-value(M-1)* $< \alpha$) & (*p-value(M)* $< \alpha$) & (*p-value(M+1)* $> \alpha$)) **then**
    $\mathcal{F}_{equal} = \mathcal{RF}_{All}[M+1]$
    Return $\mathcal{F}_{equal}$
  else
    if ((*p-value(M-1)* $< \alpha$) & (*p-value(M)* $> \alpha$) & (*p-value(M+1)* $> \alpha$)) **then**
      $\mathcal{F}_{equal} = \mathcal{RF}_{All}[M]$
      Return $\mathcal{F}_{equal}$
    else
      if ((*p-value(M-1)* $< \alpha$) & (*p-value(M)* $< \alpha$) & (*p-value(M+1)* $< \alpha$)) **then**
        SEARCHFEQUAL(Y_{tr}, $\mathcal{RF}_{All}$, $L = M$, $R = R$, α)
      else
        SEARCHFEQUAL(Y_{tr}, $\mathcal{RF}_{All}$, $L = L$, $R = M$, α)
      end if
    end if
  end if
end function

Algorithms 2 and 3 are designed in such a way that they can give us $\mathcal{F}_{equal}$ and $\mathcal{F}_{best}$ respectively, since depending on the problem that we wish to solve, we may need to identify strongly relevant features or all relevant features: $\mathcal{F}_{equal}$ contains *strongly relevant* features, such as the most important biomarkers if applied to the biomedical domain. $\mathcal{F}_{best}$ gives us the feature set with best ac-

curacy, which therefore contains all relevant features (including the strongly relevant features as a subset). To find irrelevant features we just need to remove all relevant features produced by $\mathcal{F}_{best}$ from the list of all the features $\mathcal{RF}_{All}$ ($\mathcal{RF}_{All}$ - $\mathcal{F}_{best}$).

In our application, we are interested in identifying potential biomarkers. Therefore, we use $\mathcal{F}_{equal}$ because it gives us the smallest optimal feature set without any significant loss in disease prediction accuracy. This observation is backed up by the results obtained (See section 8).

Algorithm 3 Feature set with $\mathcal{F}_{best}$ accuracy

Input: The training data $Y_{tr} = [G_1, G_2, ..., G_m]_{n \times m}$ and Ranked list of all the m features ($\mathcal{RF}_{All}$)
Output: $\mathcal{F}_{best}$ ▷ $\mathcal{F}_{best}$ = Feature set that leads to best accuracy

$i = size[\mathcal{F}_{equal}]$ ▷ Start from $i = 2$ if not computing $\mathcal{F}_{equal}$
$Overall\_Best\_Acc = 0$, $Best\_Acc = Acc(m)$
while $i \leq m$ **do** ▷ m = total number of features
  $L = i, R = i * 2$
  while $(L + 1) < R$ **do**
    $M = ceiling(\frac{L+R}{2})$
    Train the desired classifier using top M features from $\mathcal{RF}_{All}$
    if $(Acc(M) > Best\_Acc)$ **then**
      $Best\_Acc = Acc(M)$
      $Best\_Feature\_Size = M$
      $R = M$
    else
      $L = M$
    end if
  end while
  if $(Best\_Acc > Overall\_Best\_Acc)$ **then**
    $Overall\_Best\_Acc = Best\_Acc$
    $\mathcal{F}_{best} = \mathcal{RF}_{All}[Best\_Feature\_Size]$
  end if
  $i = i * 2$
end while
Return $\mathcal{F}_{best}$

Algorithm 2 identifies and returns $\mathcal{F}_{equal}$. Finding $\mathcal{F}_{equal}$ using Algorithm 2 is much less time-consuming than a full linear search through the list of features, as it recursively applies binary search to find candidate entries for $\mathcal{F}_{equal}$. Every time the *SearchFEqual* function is called, it calculates the accuracy using top $M - 1, M$ and $M + 1$ features and performs t-tests to find whether or not they are statistically equal to the accuracy of full feature set. A t-test might show statistical equivalence with multiple feature sets in a range, but we want the feature set that has the minimum number of features. There are 8 possibilities based on 3 feature subsets $M - 1, M$ and $M + 1$ and 2 options which are statistically

equal or not. These 8 possibilities are $000, 001, 010, 011, 100, 101, 110, 111$ where 1 denotes that the statistically equal accuracy found and 0 denotes not found using a particular feature subset. For example, if $M-1$ is 0, M is 0 and $M+1$ is 1 means we found the $\mathcal{F}_{equal}$. So there are 2 possibilities for $\mathcal{F}_{equal}$ that is 001 and 011 and if these are true it returns the $\mathcal{F}_{equal}$, otherwise, it checks for other possibilities. In the remaining 6 possibilities, if 000 is true then we move to the right side else in rest of the other cases we move to the left side as $\mathcal{F}_{equal}$ would be in the left side. The time complexity of finding $\mathcal{F}_{equal}$ using Algorithm 2 is $O(log_2 m * 3T)$ where m is the total number of features, T is the time complexity to train a classifier, and the constant 3 can be neglected.

While calling the $SearchFEqual$ function, we have to pass value of α as one of the parameters. Here α, the decision-making significance threshold, denotes the probability of type-I error that we are willing to accept in a particular experiment during Null Hypothesis Significance Test (NHST) and it determines the probability of a type-II error (β) for a study [21]. A type-I error occurs when we reject the null hypothesis incorrectly and a type-II error occurs when we fail to reject the null hypothesis when the alternative hypothesis is true. It is not possible to remove both the errors at a given time because if we decrease the probability of type-I error, it increases the probability of type-II error due to the nonlinear but negative and monotonic nature of the relationship between α and β. In general, a low value of α should be chosen if it is important to avoid a type-I error and a low value of β if the research question makes it particularly important to avoid a type-II error [1]. In our case while finding $\mathcal{F}_{equal}$ our objective makes it particularly important to avoid a type-II error, therefore, we have to choose low value of β and as we know both α and β are connected: we can't lower one without raising the level of other. To find $\mathcal{F}_{equal}$, we are performing repeated t-tests and to avoid the chances of type-II error we have used high value of alpha ($\alpha = 0.05$). A value of alpha is considered low if it is around 0.01.

Algorithm 3 takes training data (Y_{tr}) and ranked list of all the features ($\mathcal{RF}_{All}$) in input and returns $\mathcal{F}_{best}$. This algorithm makes the assumption that the feature-ranking algorithm is able to rank the features successfully to gives best solutions, otherwise, it may give a sub-optimal solution. If we are interested in $\mathcal{F}_{equal}$ and have already calculated $\mathcal{F}_{equal}$ then it starts from $i = size[\mathcal{F}_{equal}]$, otherwise it starts from $i = 2$, increment by $i = i * 2$ and searches the full feature space. To find $\mathcal{F}_{best}$, it searches for the feature subset that gives the best accuracy by applying binary search within each i and $i * 2$ number of features and the feature size which gives overall best accuracy will be stored into $\mathcal{F}_{best}$. The $\mathcal{F}_{best}$ gives all the features which are relevant for a particular problem.

5 Experimental Design

The overall experimental design is illustrated in Fig. 1. We explain it by taking the example of Dataset 1. In Dataset 1, we have a total of 12023 features and 2042 samples. The data is divided into separate training and test sets. In all experiments, 80% of the data is used to train the classifiers and the remaining 20% is kept as a hold-out test set, using stratified sampling. To build the

ML model for each algorithm, we estimate model parameters over the training data using 10-fold cross-validation, repeated 3 times. Only the training data is used for feature selection. The proposed *Ranked MSD* feature selection algorithm is applied to rank the features and two feature subsets $\mathcal{F}_{equal}$ and $\mathcal{F}_{best}$ are obtained. Two existing feature selection methods, correlation criteria and information gain, were also applied to compare with our proposed method. Four well-known ML classifiers are trained using selected features after applying feature selection techniques and without applying any feature selection techniques, and performance evaluation is carried out as shown in Fig. 1

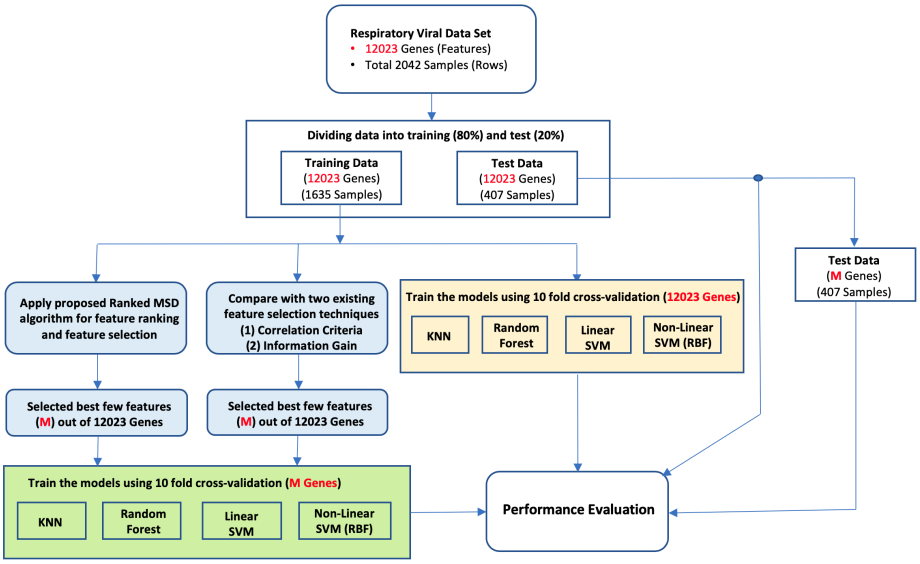

Fig. 1. Overall experimental design for evaluation of proposed Ranked MSD Algorithm.

6 Methodology

In this section, we briefly explain the methodology used for the evaluation of proposed Ranked MSD feature selection algorithm and potential biomarker identification. It is well known that no single ML algorithm is best for all kind of datasets, so we tested a selection of different ML approaches. The best performing classifier is then used for biomarker identification.

First, we used a very simple algorithm, k-NN, which is an instance-based learning algorithm; see for example [6]. k-NN is an important algorithm in the sense that it can give us good explanations if we have few features or a way to reduce our feature set to the most important features [20]. Moreover, it can be used to set a base to compare the results and to see the improvements yielded by more complex algorithms. We also used the Random Forest algorithm which is

an ensemble technique [7]. We then employed both linear SVM [3] and SVM with RBF kernel which has inbuilt capability to learn pattern from high dimensional data [25]. We have used R programming language version 3.4.1 for coding [22].

6.1 *k*-Nearest Neighbour (*k*-NN)

The k-NN utilizes the nearest neighbours of a data sample for prediction. The k-NN has two stages, the first stage is the determination of the nearest neighbours i.e. the value of k and the second is the prediction of the class label using those neighbours. The "k" nearest neighbours are selected using a distance metric. We have used Euclidean distance for our experiments. There are various ways to use this distance metric to determine the class of the test sample. The most straightforward way is to assign the class that the majority of k-nearest neighbours have. In the present work, the optimum value of k is searched over the range of $k = 1$ to 30.

6.2 Random Forest

The Random Forest algorithm constructs an ensemble of many classification trees [18, 27]. Each classification tree is created by selecting a bootstrap sample from the whole training data and a random subset of variables with size denoted as $mtry$ is selected at each split. We have used the recommended value of $mtry$: $(mtry = \sqrt{(number\ of\ genes)})$ [7]. The number of trees in the ensemble is denoted as $ntree$. We have used $(ntree) = 10,001$ so that each variable can reach a sufficiently large likelihood to participate in forest building.

6.3 Support Vector Machine (SVM)

Assume that we have given a training set of instance-label pairs $(\boldsymbol{x}_i, y_i); \forall i \in \{1, 2, ..., l\}$ where $\boldsymbol{x}_i \in \mathbb{R}^n$ and $\boldsymbol{y} \in \{1, -1\}^l$, then the SVM [3, 10, 12] can be formulated and solved by the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{w}, b, \boldsymbol{\xi_i}} \quad & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{l}\xi_i, \\ \text{subject to} \quad & y_i\left(\boldsymbol{w}^T\phi\left(\boldsymbol{x}_i\right) + b\right) \geq 1 - \xi_i, \\ & \xi_i \geq 0. \end{aligned} \tag{5}$$

Here the parameter $C > 0$ is the penalty parameter of the error term [12] and $\xi_i \forall i \in \{1, 2, ..., l\}$ are positive slack variables [3]. For linear SVM, we did a search for best value of parameter C for a range of values $\left(C = 2^{-7}, 2^{-3}, ..., 2^{15}\right)$ and the one with the best 10-fold cross validation accuracy has finally been chosen.

We also used SVM with RBF kernel which is a non-linear kernel. There are four basic kernels that are frequently used: linear, polynomial, sigmoid, and RBF. We picked the RBF kernel, as recommended by Hsu et al. [12]. It has the following form:

$$K\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = \exp\left(\frac{-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2}\right); \frac{1}{2\sigma^2} > 0.$$

We performed a grid-search over the values of C and σ using 10-fold cross validation. The different pairs of (C, σ) values are tried in the range of $(C = 2^{-7}, 2^{-3}, ..., 2^{15}; \sigma = 2^{-25}, 2^{-13}, ..., 2^{3})$ and the values with the best 10-fold cross validation accuracy are picked for the final model building.

7 Explaining Predictions

Similar to the LIME [23], we also believe that it is possible to explain predictions of any classifier by explaining the contribution of important features that led to those predictions. Providing the explanation for an individual prediction is relatively easy and can be achieved by explaining the contribution of important features for that particular prediction. It is relatively hard to provide global interpretation, however, it can be achieved by either explaining a set of representative predictions of each class or explaining all the predictions as a whole [23]. Here, we are providing global interpretation by explaining all the predictions using a 3D-plot. Fig. 2 (B) is showing the contribution of the 3 most important features (genes) suggested by the proposed Ranked MSD approach. Fig. 2 contains the held-out test set predictions of GSE68310 gene expression data set using least important 3 genes (left) and using most important 3 genes (right). More clear and persuasive visual explanations can be provided if contributions of all the important feature can be plotted all together. We leave this exploration for future work.

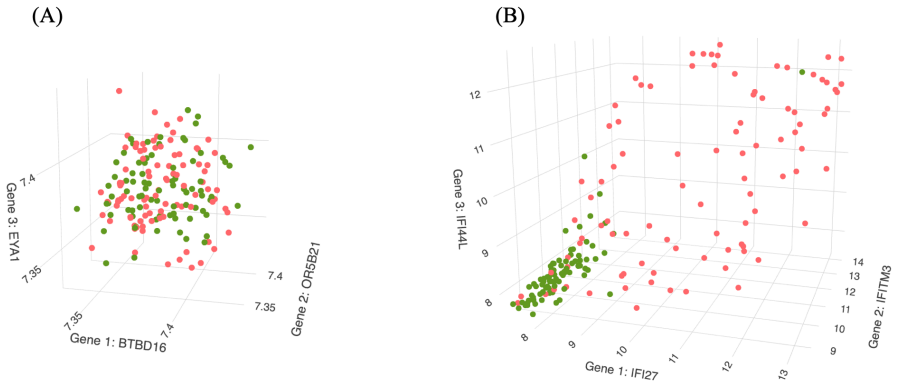

Fig. 2. Providing global interpretation of SVM with RBF Kernel model by plotting test data predictions. (A) 3D-plot of least important 3 genes which fail to achieve class separability and do not give any explanation (B) 3D-plot of most important 3 genes ranked by proposed algorithm. Using these 3 most important genes it is possible to achieve a greater class separability thus helping in explaining predictions. Green dots denote the healthy test samples and Red dots denote infected test samples. The axis denotes the gene expression values of the corresponding gene.

8 Results

We experimentally obtained the 10-fold cross-validation accuracy at full feature set, $\mathcal{F}_{equal}$ and $\mathcal{F}_{best}$ by applying proposed Ranked MSD algorithm on two datasets. We also compared the performance of proposed algorithm with two existing algorithms using four classifiers: k-NN, Random Forest, linear SVM, and SVM with RBF Kernel. The results from both datasets can be seen in Table 1 and Table 2. To show the performance of the proposed algorithm in comparison to existing algorithms, we have plotted graphs of feature size versus 10-fold cross-validation accuracy using the four classifiers for both the data sets (see Fig. 3 and Fig. 4). The shaded region in the figures showing the standard deviation calculated over 10 fold-cross validation accuracies (repeated 3 times so 30 accuracies in total).

Table 1. Comparison between the performance of proposed Ranked MSD and other feature selection algorithms using four well-known ML algorithms trained on full feature set, $\mathcal{F}_{equal}$ and $\mathcal{F}_{best}$ feature set of the first dataset. Here $\mathcal{F}_{equal}$ is feature size which gives statistically equal accuracy to that of full feature set and $\mathcal{F}_{best}$ is feature size which gives best accuracy.

Feature Ranking Algorithm	**ML Model**	**Total Features**	**Accuracy (All Features)**	$\mathcal{F}_{equal}$	**Accuracy** $\mathcal{F}_{equal}$	$\mathcal{F}_{best}$	**Accuracy** $\mathcal{F}_{best}$
Ranked MSD	KNN	12023	89.17%	22	88.60%	110	91.68%
Correlation Criteria	KNN	12023	89.17%	70	88.73%	1472	90.78%
Information Gain	KNN	12023	89.17%	8950	87.95%	11976	89.23%
Ranked MSD	Linear SVM	12023	91.52%	367	91.05%	2560	93.17%
Correlation Criteria	Linear SVM	12023	91.52%	561	90.68%	3040	92.62%
Information Gain	Linear SVM	12023	91.52%	11961	91.39%	12013	91.52%
Ranked MSD	Random Forest	12023	88.97%	45	88.32%	544	91.17%
Correlation Criteria	Random Forest	12023	88.97%	352	88.44%	3040	89.42%
Information Gain	Random Forest	12023	88.97%	11271	88.85%	11930	88.99%
Ranked MSD	SVM with RBF Kernel	12023	93.3%	65	92.25%	128	93.3%
Correlation Criteria	SVM with RBF Kernel	12023	93.3%	327	92.46%	1916	93.39%
Information Gain	SVM with RBF Kernel	12023	93.3%	6387	92.54%	11895	93.32%

Based on the results obtained, it can be concluded that the scoring function used in the Ranked MSD algorithm is successfully able to rank the features in descending order of their importance, because we are able to see the increase in accuracy when we are adding the top-ranked features. For example, in case of SVM with RBF kernel (see Table 1), the $\mathcal{F}_{equal}$ using proposed Ranked MSD algorithm gives 92.25% accuracy using top 65 genes (strongly relevant) whereas total 12023 genes give 93.3% accuracy. Here the benefit of $\mathcal{F}_{equal}$ is that it can hold the potential biomarkers for the respiratory viral infection because it finds a small number of strongly relevant features which contribute to reaching accuracy of 92.25%. $\mathcal{F}_{best}$ yields 93.3% accuracy using the top 128 genes from $\mathcal{RF}_{All}$ and there is no improvement in accuracy if we add more genes, which shows that the

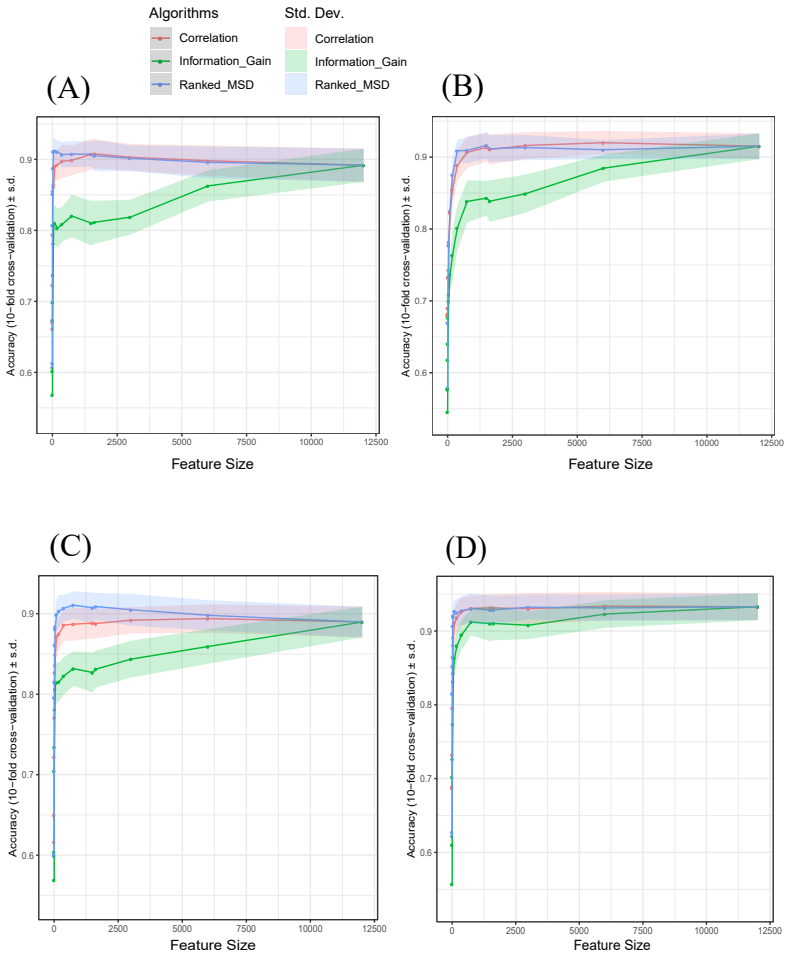

Fig. 3. Comparing performance of Ranked MSD algorithm with other existing feature selection techniques on first data set using (A) KNN (B) Linear SVM (C) Random Forest and (D) SVM with RBF Kernel.

rest of the genes after top 128 genes are irrelevant for this classifier. A similar behaviour can be observed for the second data set (see Table 2 and Figure 4).

In the area of drug discovery, we wish to target the smallest number of important genes. In such cases, the use of $\mathcal{F}_{equal}$ is valuable because we don't want to include all 12,023 or 20,737 genes as potential targets for drug discovery but those 65 genes (Dataset 1) or 31 genes (Dataset 2) which contribute to reaching the $\mathcal{F}_{equal}$ accuracy (See Table 1 and Table 2).

The other optimal feature subset, $\mathcal{F}_{best}$ can be used according when needed (See Table 1 and 2), in cases where one requires the best possible accuracy, while allowing a larger number of features to be selected; in such cases $\mathcal{F}_{best}$ provides all the features that are relevant. The standard deviation of repeated

Table 2. Comparison between the performance of proposed Ranked MSD and other feature selection algorithms using four well-known ML algorithms trained on full features set, $\mathcal{F}_{equal}$ and $\mathcal{F}_{best}$ feature set of the second dataset. Here $\mathcal{F}_{equal}$ is feature size which gives statistically equal accuracy to that of full feature set and $\mathcal{F}_{best}$ is feature size which gives best accuracy.

Feature Ranking Algorithm	**ML-Model**	**Total Features**	**Accuracy (All Features)**	$\mathcal{F}_{equal}$	**Accuracy** $\mathcal{F}_{equal}$	$\mathcal{F}_{best}$	**Accuracy** $\mathcal{F}_{best}$
Ranked MSD	KNN	20737	80.7%	2	79.82%	48	82.55%
Correlation Criteria	KNN	20737	80.7%	2	80.64%	224	83.05%
Information Gain	KNN	20737	80.7%	13	78.39%	632	83.53%
Ranked MSD	Linear SVM	20737	86.44%	45	85.82%	1725	90.16%
Correlation Criteria	Linear SVM	20737	86.44%	8	84.87%	6108	88.74%
Information Gain	Linear SVM	20737	86.44%	90	84.91%	1920	88.73%
Ranked MSD	Random Forest	20737	83.03%	10	81.42%	1664	85.44%
Correlation Criteria	Random Forest	20737	83.03%	7	81.83%	83	85.48%
Information Gain	Random Forest	20737	83.03%	93	81.75%	332	85.09%
Ranked MSD	SVM with RBF Kernel	20737	86.44%	31	84.35%	1239	90.07%
Correlation Criteria	SVM with RBF Kernel	20737	86.44%	8	84.83%	6136	88.97%
Information Gain	SVM with RBF Kernel	20737	86.44%	88	85.15%	1920	88.83%

10 fold cross-validation accuracies is not significantly high which suggests that the algorithm is able to produce stable results.

As the results show, our *Ranked MSD* algorithm is able to achieve significantly higher accuracy using very few genes compared to two well-known feature selection approaches, which indicates that our algorithm is selecting more highly informative genes than the other approaches. Based on the results of these experiments, we can conclude that the proposed Ranked MSD algorithm outperforms the existing correlation-based and entropy-based feature selection methods on investigated datasets. For further details, additional figures can be found in a supplementary file at this link: https://figshare.com/articles/Ranked_MSD/8312402.

9 Biomarker Identification

In this section, we show the top 18 important biomarkers (see Table 3) which are obtained from taking the intersection of $\mathcal{F}_{equal}$ genes suggested from the best-performing classifier. We have selected SVM with RBF kernel as the best performing classifier, because it gives the consistently best accuracy with smallest optimal feature set of the classification algorithms evaluated. For Dataset 1, the $\mathcal{F}_{equal}$ size is 65 and for Dataset 2, the $\mathcal{F}_{equal}$ size is 31. The intersection of both optimal feature sets is 18, as illustrated in Figure 5(b). Table 3 lists the 18 biomarkers that are common to both datasets, with their importance scores as given by our *Ranked MSD* algorithm. The combined importance score of a biomarker is the average of its importance score from Dataset 1 and Dataset 2. These 18 biomarkers are found to be the most important ones for the progression of respiratory viral infection as they are common best biomarkers for both the

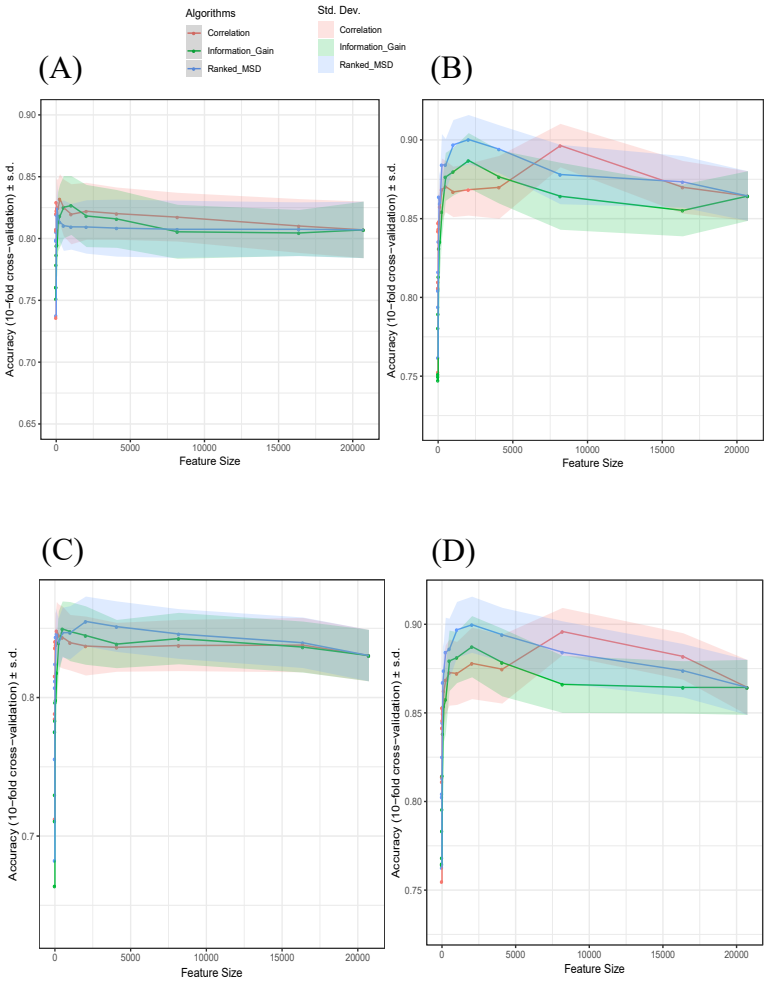

Fig. 4. Comparing performance of Ranked MSD algorithm with other existing feature selection techniques on second data set using (A) KNN (B) Linear SVM (C) Random Forest and (D) SVM with RBF Kernel.

respiratory viral data sets and play an important role in the discrimination of infected samples from non-infected ones.

10 Biological Significance of Biomarkers

To determine biological significance of identified 18 common biomarkers (see Table 3), we performed Molecular Enrichment Analysis (MEA) developed as an extension of Gene Set Enrichment Analysis (GESA) [29]. The biomarkers shown in Table 3 work as seed genes for the MEA analysis and the results of the enrichment can be seen in Figure 5 (a). We used KEGG pathway database [13]

Table 3. Top 18 biomarkers obtained from taking intersection of $\mathcal{F}_{equal}$ genes suggested by best performing classifier from both the data sets with their importance score given by the proposed Ranked MSD algorithm.

Sr. No.	Gene Symbol	Importance Score Data set 1	Importance Score Data set 2	Combined Importance Score
1	IFI27	7.440807172	8.744873626	8.092840399
2	RSAD2	7.781213672	3.788627429	5.78492055
3	IFI44L	6.574818892	4.978166941	5.776492916
4	RPS4Y1	9.091357246	2.008783926	5.550070586
5	ISG15	4.063142667	4.578217378	4.320680022
6	IFI44	5.014484011	2.860260613	3.937372312
7	IFITM3	2.495110022	5.359004043	3.927057032
8	HERC5	4.04471719	3.211724492	3.628220841
9	MX1	2.98390575	3.508595477	3.246250613
10	LY6E	2.249102841	3.834706217	3.041904529
11	IFIT3	3.441885872	2.50066336	2.971274616
12	OAS3	3.705077795	2.176990862	2.941034328
13	IFIT2	3.015140745	2.840130002	2.927635374
14	IFI6	3.192691447	2.236622559	2.714657003
15	OASL	3.303346964	1.954874216	2.62911059
16	HBG2	2.275345902	2.923786916	2.599566409
17	OAS2	2.345796307	2.446741233	2.39626877
18	XAF1	2.675522172	1.975049326	2.325285749

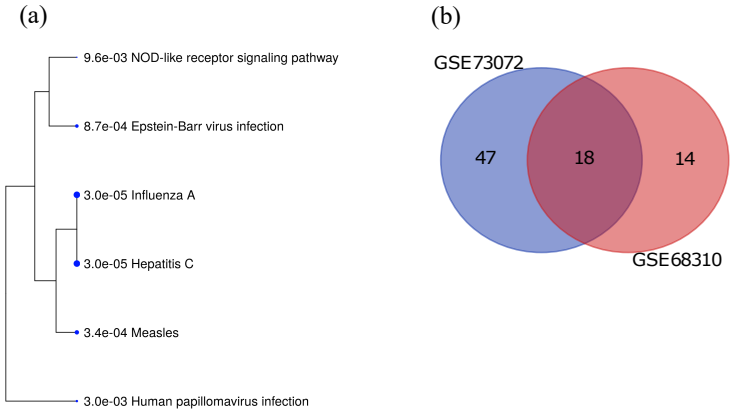

Fig. 5. (a) MEA analysis of seed genes and associated KEGG terms. The larger blue dots show higher enrichment with seed genes. (b) Venn-diagram of important genes obtained from data set 1 (GSE73072) and data set 2 (GSE68310). Total 18 genes are overlapping genes which are final biomarkers.

to perform the MEA analysis. Biomarkers retrieved only through gene expression can lead to non-relevant signatures and irrelevant phenotypes [5], therefore, a biological significance analysis is essential. The reproducible and clinically attainable results provided by us, proves the authenticity of biomarkers and can be

a great fit in precision medicine era. The 18 gene retrieved through our analysis yields Influenza A and Hepatitis C with enrichment score of 3.0e-05. It has been reported earlier that any gene signature profound in Influenza A also elevates the expression profile of Hepatitis C with similar intensity [2]. Further, Influenza A and measles are viruses that both cause respiratory symptoms thus enrichment of measles provide appropriate phenotype for our 18 gene signature [32]. Now, looking into the pathway from 5 (a) the enriched pathways NOD-like receptor signalling pathway is associated with higher immunity. Thus any targeted study through these 18 biomarkers will provide a clinical acceptable therapy for viral diseases specially in the case of Influenza. Thus it can be inferred that our biomarker panel covers immune response [15] with disease progression and provides a cohesive platform for precision medicine.

11 Conclusions and Future Work

In this work, we have aimed to tackle the issues that arise when the number of samples is much smaller than the number of features (commonly referred to as $n << p$). It becomes very hard to interpret the target concept in these situations. In addition, irrelevant features often confuse machine leaning systems and lead to deterioration of classification accuracy. Also, recent GDPR issues may make it difficult to use black-box models particularly in business and medicine. To address these problems, we have proposed a feature ranking algorithm named *Ranked MSD* with two additional algorithms to identify *strongly relevant* features and *relevant* features by discarding irrelevant features. Our experimental results show that the proposed *Ranked MSD* algorithm outperforms two well-known feature ranking methods, Correlation Criteria and Information Gain, and thereby can help with better disease prediction. Moreover, we have identified 18 biomarkers which are common biomarker across the two datasets that we have analysed and which have been identified as strongly relevant features by our approach. The importance of these 18 genes are confirmed with the four classifiers as they all yield improvements in accuracy using these top genes. To determine the biological significance of these 18 genes, we performed Molecular Enrichment Analysis (MEA); the results show that these biomarkers are strongly related to the target disease, and can therefore be considered as potential targets for drug discovery, and could play an important role in precision medicine.

In this work, we have demonstrated our proposed approach by applying it on datasets related to respiratory viral infections only. In future work, we aim to perform analyses on more datasets of different diseases and domains. We also aim to incorporate useful information that is openly available in the form of biomedical knowledge graphs.

Acknowledgements
This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289, co-funded by the European Regional Development Fund.

References

1. Banerjee, A., Chitnis, U., Jadhav, S., Bhawalkar, J., Chaudhury, S.: Hypothesis testing, type i and type ii errors. Industrial psychiatry journal **18**(2), 127 (2009)
2. Bolen, C.R., Robek, M.D., Brodsky, L., Schulz, V., Lim, J.K., Taylor, M.W., Kleinstein, S.H.: The blood transcriptional signature of chronic hepatitis c virus is consistent with an ongoing interferon-mediated antiviral response. Journal of Interferon & Cytokine Research **33**(1), 15–23 (2013)
3. Burges, C.J.: A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery **2**(2), 121–167 (Jun 1998)
4. Chandrashekar, G., Sahin, F.: A survey on feature selection methods. Computers & Electrical Engineering **40**(1), 16–28 (2014)
5. Cun, Y., Fröhlich, H.: Biomarker gene signature discovery integrating network knowledge. Biology **1**(1), 5–17 (2012)
6. Cunningham, P., Delany, S.J.: k-nearest neighbour classifiers. Multiple Classifier Systems **34**, 1–17 (2007)
7. Díaz-Uriarte, R., De Andres, S.A.: Gene selection and classification of microarray data using random forest. BMC bioinformatics **7**(1), 3 (2006)
8. Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., et al.: Least angle regression. The Annals of statistics **32**(2), 407–499 (2004)
9. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of machine learning research **3**(Mar), 1157–1182 (2003)
10. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Machine Learning **46**(1), 389–422 (Jan 2002)
11. Holzinger, A., Biemann, C., Pattichis, C.S., Kell, D.B.: What do we need to build explainable ai systems for the medical domain? arXiv preprint arXiv:1712.09923 (2017)
12. Hsu, C.W., Chang, C.C., Lin, C.J.: A practical guide to support vector classification (2010)
13. Kanehisa, M., Goto, S.: Kegg: kyoto encyclopedia of genes and genomes. Nucleic acids research **28**(1), 27–30 (2000)
14. Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., sayres, R.: Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In: Dy, J., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 80, pp. 2668–2677. PMLR, StockholmsmÃ€ssan, Stockholm Sweden (10–15 Jul 2018), http://proceedings.mlr.press/v80/kim18d.html
15. Kim, Y.K., Shin, J.S., Nahm, M.H.: Nod-like receptors in infection, immunity, and diseases. Yonsei medical journal **57**(1), 5–14 (2016)
16. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artificial intelligence **97**(1-2), 273–324 (1997)
17. Li, O., Liu, H., Chen, C., Rudin, C.: Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
18. Liaw, A., Wiener, M.: Classification and Regression by randomForest. R News **2**(3), 18–22 (2002), http://CRAN.R-project.org/doc/Rnews/
19. Liu, T.Y., Burke, T., Park, L.P., Woods, C.W., Zaas, A.K., Ginsburg, G.S., Hero, A.O.: An individualized predictor of health and disease using paired reference and target samples. BMC Bioinformatics **17**(1), 47 (Jan 2016)

20. Molnar, C., et al.: Interpretable machine learning: A guide for making black box models explainable. Christoph Molnar, Leanpub (2018)
21. Mudge, J.F., Baker, L.F., Edge, C.B., Houlahan, J.E.: Setting an optimal α that minimizes errors in null hypothesis significance tests. PloS one **7**(2), e32734 (2012)
22. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2013)
23. Ribeiro, M.T., Singh, S., Guestrin, C.: Why should i trust you?: Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. pp. 1135–1144. ACM (2016)
24. Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence **1**, 206–215 (May 2019)
25. Scholkopf, B., Sung, K.K., Burges, C.J.C., Girosi, F., Niyogi, P., Poggio, T., Vapnik, V.: Comparing support vector machines with Gaussian kernels to radial basis function classifiers. IEEE Trans. Signal Process. **45**(11), 2758–2765 (Nov 1997)
26. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 618–626 (2017)
27. Statistics, L.B., Breiman, L.: Random forests. In: Machine Learning. pp. 5–32 (2001)
28. Stork, E., Duda, R., Hart, P., Stork, D.: Pattern classification. New York [ua]: Academic Internet Publishers (2006)
29. Subramanian, A., Tamayo, P., Mootha, V.K., Mukherjee, S., Ebert, B.L., Gillette, M.A., Paulovich, A., Pomeroy, S.L., Golub, T.R., Lander, E.S., et al.: Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proceedings of the National Academy of Sciences **102**(43), 15545–15550 (2005)
30. Verma, G., Jha, A., Rebholz-Schuhmann, D., Madden, M.G.: Using machine learning to distinguish infected from non-infected subjects at an early stage based on viral inoculation. In: International Conference on Data Integration in the Life Sciences. pp. 105–121. Springer (2018)
31. Witten, I.H., Frank, E., Hall, M.A., Pal, C.J.: Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann (2016)
32. Zachariah, P., Posner, A., Stockwell, M.S., Dayan, P.S., Sonnett, F.M., Graham, P.L., Saiman, L.: Vaccination rates for measles, mumps, rubella, and influenza among children presenting to a pediatric emergency department in new york city. Journal of the Pediatric Infectious Diseases Society **3**(4), 350–353 (2014)
33. Zhai, Y., Franco, L.M., Atmar, R.L., Quarles, J.M., Arden, N., Bucasas, K.L., Wells, J.M., Nino, D., Wang, X., Zapata, G.E., et al.: Host transcriptional response to influenza and other acute respiratory viral infections–a prospective cohort study. PLoS pathogens **11**(6), e1004869 (2015)