



# Accurate Classification of COVID-19 Based on Incomplete Heterogeneous Data using a KNN Variant Algorithm

Ahmed Hamed<sup>1</sup> · Ahmed Sobhy<sup>1</sup> · Hamed Nassar<sup>1</sup>

Received: 8 June 2020 / Accepted: 7 December 2020 / Published online: 4 March 2021  
© King Fahd University of Petroleum & Minerals 2020

## Abstract

Great efforts are now underway to control the coronavirus 2019 disease (COVID-19). Millions of people are medically examined, and their data keep piling up awaiting classification. The data are typically both incomplete and heterogeneous which hampers classical classification algorithms. Some researchers have recently modified the popular *KNN* algorithm as a solution, where they handle incompleteness by imputation and heterogeneity by converting categorical data into numbers. In this article, we introduce a novel *KNN* variant (*KNNV*) algorithm that provides better results as demonstrated by thorough experimental work. We employ rough set theoretic techniques to handle both incompleteness and heterogeneity, as well as to find an ideal value for *K*. The *KNNV* algorithm takes an incomplete, heterogeneous dataset, containing medical records of people, and identifies those cases with COVID-19. We use in the process two popular distance metrics, Euclidean and Mahalanobis, in an effort to widen the operational scope. The *KNNV* algorithm is implemented and tested on a real dataset from the Italian Society of Medical and Interventional Radiology. The experimental results show that it can efficiently and accurately classify COVID-19 cases. It is also compared to three *KNN* derivatives. The comparison results show that it greatly outperforms all its competitors in terms of four metrics: precision, recall, accuracy, and *F*-Score. The algorithm given in this article can be easily applied to classify other diseases. Moreover, its methodology can be further extended to do general classification tasks outside the medical field.

**Keywords** COVID-19 diagnosis · *KNN* · Incomplete data · Heterogeneous data · Rough set theory · Euclidean · Mahalanobis

## 1 Introduction

The coronavirus disease 2019 (COVID-19) [1] is currently wreaking havoc around the world. It is causing a major threat to human life, with severe economic consequences. Its symptoms include cough, fever, and respiratory complications. The hazardous side of COVID-19 is its rapid spreading because it is transmitted by contact and by small droplets produced when people cough, sneeze, or talk. To make matters worse, COVID-19 can survive on surfaces up to 72 h [2],

causing people to catch it by touching apparently normal patients.

The best way to improve a COVID-19 patient survival rate is through early detection of the disease [3], and here is where AI techniques, such as what is employed in the present work, can help. It is widely believed that AI has the potential to solve many problems related to COVID-19 if there is information about the patients. However, this information can be heterogeneous, in the sense that the features of the patients are of two types, categorical and numerical [4]. Categorical features are qualitative, e.g., gender and coughing or not coughing, whereas numerical features are quantitative, e.g., age and body temperature. The presence of both types in the information complicates processing. What is more, this information can also be incomplete, in the sense that some features may have missing values [5]. The value of a feature can be missing due to several factors, such as negligence, cost, or difficulty to obtain.

These two issues, heterogeneity and incompleteness, present tremendous challenges for the classification of

✉ Ahmed Hamed  
ahmed\_hamed@ci.suez.edu.eg

Ahmed Sobhy  
ahmed\_sobhy@ci.suez.edu.eg

Hamed Nassar  
nassar@ci.suez.edu.eg

<sup>1</sup> Faculty of Computers and Informatics, Suez Canal University, Ismailia, Egypt



COVID-19 cases within a dataset collected about numerous persons. Typically, classical classification algorithms do not handle heterogeneity and incompleteness. Therefore, one needs to adapt a classical algorithm to handle incomplete heterogeneous COVID-19 (IHC) datasets, and this is the goal of the present work.

Formally, IHC data can be defined by a triple  $(\mathcal{U}, \mathcal{A}, \mathcal{V})$ . The set  $\mathcal{U} = \{u_1, u_2, \dots, u_m\}$  is a non-empty finite set of patients, called the universe. The set  $\mathcal{A} = \{a_1, a_2, \dots, a_n, d\}$  is a non-empty finite set of features describing the patient. It contains  $n + 1$  features,  $n$  of which, namely  $a_1, a_2, \dots, a_n$ , are conditional features and the  $n + 1^{\text{st}}$  feature is a decision label. The  $n$  conditional features are of two types: categorical and numerical. The union of the set  $\mathcal{C}$  of categorical features and the set  $\mathcal{N}$  of numerical features is accordingly the set  $\mathcal{A}^-$  of conditional features, where

$$\mathcal{A}^- = \mathcal{A} - \{d\}.$$

If the value of a feature, categorical or numerical, is incomplete, this value is denoted in the present work by an asterisk “\*.”

If  $\mathbb{V} = \{\zeta_1, \zeta_2, \dots, \zeta_r\}$  is the value set of  $d$ , then  $d$  partitions the universe  $\mathcal{U}$  into  $r$  decision subsets (classes),  $\mathcal{U}_{\zeta_1}, \mathcal{U}_{\zeta_2}, \dots, \mathcal{U}_{\zeta_r}$ , where  $\mathcal{U}_{\zeta_k}$  is the subset of all patients with decision label value  $\zeta_k$ . Finally, the set  $\mathcal{V}$  is the union of the value sets of all features. That is,

$$\mathcal{V} = \{v_{u_i, a_j} \mid u_i \in \mathcal{U}, a_j \in \mathcal{A}^-\},$$

where  $v_{u_i, a_j}$  is the value of feature  $a_j$  of patient  $u_i$ . For example,  $v_{u_1, a_2} = 3$  means that feature  $a_2$  of patient  $u_1$  has the value 3. A specific instance of IHC data is called an IHC dataset. Table 1 shows a toy IHC dataset, used repeatedly in the sequel to illustrate the proposed KNNV algorithm.

In our context, classification is a two-step process [6]. In the first, called the training step, a classification algorithm learns from a set of patients  $u_i \in \mathcal{U}$  whose decision values are known. In the second, called the test step, the algorithm uses what it has learnt to classify an unknown patient  $u_0$  whose decision value is to be identified. A popular classification algorithm, used in the present work that can be modified to deal with IHC data, is the  $K$  nearest neighbor ( $K$  NN) algorithm [7]. Its main idea is to search for  $K$  neighbor patients nearest to the unknown patient  $u_0$  and then predict the decision value of the latter by a majority vote of those neighbors.

In the present work, we have chosen KNN rather than any other machine learning algorithm [8], such as support vector machines, decision trees, naïve Bayes, and backpropagation, for a good reason. These latter algorithms build a model from a training set of patients before receiving an unknown patient. This prior model buildup adds complexity to the operation.

Additionally, it is not easy to adapt these algorithms to handle both heterogeneity and incompleteness, prevalent in most real-world data nowadays. By contrast, KNN is a nonparametric classifier that does not build a prior model from the training set. Besides, it can be easily adapted, as done in the present article, to handle incomplete, heterogeneous data accurately and efficiently. Furthermore, unlike most other classification algorithms, KNN does not require the decision subsets  $\mathcal{U}_{\zeta_j}$  to be linearly separable.

The distances needed by KNN to identify the neighbors nearest to a given patient can be calculated via several distance metrics. In this article, we use two such metrics: Euclidean distance and Mahalanobis distance. Either one can be used to measure the distance between two patients, one of them being a member of a class of patients. However, while Euclidean distance does not pay attention to the other elements of the class in the measurement process, Mahalanobis does. Therefore, Mahalanobis distance is more suitable for assessing distances when the dataset is highly skewed or its features are correlated.

To improve the performance of KNN, three issues need to be addressed. First, a proper  $K$  value has to be used. A small  $K$  increases the influence of noise on prediction, while a large  $K$  increases computational complexity. Classical KNN uses the same  $K$  for all unknown patients to be classified, whereas  $K$  is to a great extent case dependent [9]. The “one  $K$  fits all” policy leads to a high percentage of misclassification, as shown in Experimental Work Section. Third,  $K$  NN computes distances inaccurately, as discussed in Sect. 3, which is another source of misclassification. The KNN variant (KNNV) algorithm proposed in this article addresses all three issues, thus improving identification of COVID-19 cases within IHC datasets fairly accurately.

The proposed KNNV classification algorithm chooses  $K$  adaptively for each unknown patient and computes accurately the distances between patients to identify the nearest neighbors. It mitigates the incompleteness problem and the heterogeneity problem through novel rigorous rules spelled out in the sequel. For validation, KNNV is tested on a publicly available IHC dataset from the Italian Society of Medical and Interventional Radiology (SIRM) [10]. The test results show excellent classification. In addition, KNNV is compared using the same dataset against algorithms of its category. The comparison results show impressive superiority of KNNV.

The rest of this article is organized as follows. Section 2 covers related work. Section 3 describes the proposed algorithm. In Sect. 4, experimental work to validate the algorithm and compare it with potential competitors is presented, and discussion about the findings is given. Finally, concluding remarks are provided in Sect. 5.

**Table 1** A toy IHC dataset made of 13 records

	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$	$a_8$	$d$
$u_1$	Male	*	Yes	Yes	Yes	0.97	*	0.78	COVID-19
$u_2$	*	Yes	*	No	Yes	*	0.39	0.64	COVID-19
$u_3$	Female	No	*	No	Yes	0.77	0.79	0.39	Flu
$u_4$	Male	Yes	No	Yes	No	0.08	0.8	0.37	Flu
$u_5$	Female	Yes	Yes	*	No	0.8	0.1	*	COVID-19
$u_6$	Female	*	*	No	Yes	0.42	0.55	0.39	Flu
$u_7$	Male	Yes	Yes	No	Yes	0.98	*	*	COVID-19
$u_8$	Male	Yes	No	Yes	Yes	0.43	0.42	0.36	Flu
$u_9$	*	Yes	No	No	Yes	0.96	*	0.11	Flu
$u_{10}$	Female	*	No	Yes	Yes	*	0.34	0.55	COVID-19
$u_{11}$	Male	Yes	Yes	Yes	*	0.38	0.39	0.81	Flu
$u_{12}$	Male	Yes	Yes	Yes	No	0.85	*	*	COVID-19
$u_{13}$	Female	No	Yes	*	*	0.31	0.59	0.37	Flu

Clearly,  $\mathcal{U} = \{u_1, u_2, \dots, u_{13}\}$  is the set of patients and  $\mathcal{A} = \{a_1, a_2, \dots, a_8, d\}$  is the set of features. Note that  $\mathcal{C} = \{a_1, a_2, \dots, a_5\}$  is the set of categorical features and  $\mathcal{N} = \{a_6, a_7, a_8\}$  the set of numerical features, with  $d$  being the decision label. Due to the missing values, denoted by “\*,” this dataset is roughly 20% incomplete

## 2 Related Work

Researchers in different fields are now actively participating to fight COVID-19, and information science researchers are no exception. Roosa et al. [11] propose a forecasting algorithm to predict the spread of COVID-19 in China. They use phenomenological models to develop and assess short-term forecasts of the cumulative number of confirmed COVID-19 cases in the Chinese Hubei province. A related attempt is made by Pirouz et al. [12], who propose a binary classification system using artificial neural networks (ANN) to predict the number of confirmed COVID-19 cases in Hubei province. McCall [13] reports that AI is doing a paradigm shift in health care and is, therefore, considered a promising tool to trap COVID-19. He recommends using AI algorithms to predict the location of the next outbreak. Hu et al. [14] propose an AI system for real-time forecasting of COVID-19 with the aim of estimating the life time of the virus. They propose a modified stacked auto-encoder system for modeling the transmission dynamics of the pandemic throughout China. They use a variety of features such as maximum, minimum, and average daily temperature, humidity, and wind speed as inputs to an ANN, with the aim of predicting the confirmed number of COVID-19 patients in the next 30 days. Du et al. [15] propose a hybrid AI model that combines the strengths of both a natural language processing module and a long short-term memory network. Their objective is to analyze the change in the infectious capacity of COVID-19 patients within a few days after they catch the virus. Santosh [16] proposes an AI-driven system able to predict the time of the next COVID-19 outbreak, while forecasting the COVID-19 possibility to spread across the globe. Boldog et

al. [17] propose a computational tool able to assess the risks of COVID-19 outbreaks outside China. They compute the probability of a major outbreak in a country through testing three features: cumulative number of patients in the non-locked down Chinese provinces, connectivity of that country with China, and efficacy of control measures in that country. All the above proposals focus on predicting outbreaks, overlooking classification of existing cases. For this latter objective, the endeavors next have been made.

Gozes et al. [18] propose an AI based computational tomography (CT) image classification algorithm for the detection, quantification and tracking of COVID-19. This algorithm has the ability to distinguish COVID-19 patients from other patients. They use a deep learning model to classify COVID-19 from CT images. Ai et al. [19] propose a CT image algorithm for COVID-19 identification that uses a reverse-transcription polymerase chain reaction test. Barstugan et al. [20] propose a feature extraction process for CT images and a discrete wavelet transform algorithm to improve COVID-19 classification. Specifically, they use a grey level co-occurrence matrix, local directional pattern, grey-level run length matrix, and grey-level size zone matrix. Afterward, they use support vector machines to classify the pandemic based on the extracted features. Xu et al. [21] propose a CT image classification algorithm for early classification of COVID-19 using deep learning techniques. The algorithm is able to distinguish COVID-19 patients from Flu patients. They first segment the CT images using a 3D deep learning model; then, the segmented images are binary classified. Wang et al. [22] use CT images to extract COVID-19 conditional graphical features that can then be used to distinguish COVID-19 patients from other patients. Li et al. [23]

propose a CT image classification algorithm for the early classification of COVID-19. They exploit both deep learning and ANN to extract visual features from chest CT images. The trouble with all these endeavors is that they depend on medical imaging, which may not be readily available or accessible. An alternative classification direction, where the present work belongs, depends on documented personal and medical examination data as described next.

Peng et al. [24] use AI techniques to improve classification accuracy of COVID-19. They use sparse rescaled linear square regression, evolutionary non-dominated radial slots, attribute reduction with multi-objective decomposition-ensemble optimization, gradient boosted feature selection, and recursive feature elimination. Rao and Vazquez [25] classify COVID-19 from data collected about travel history along with a phone-based online survey. These data are then used to divide patients into four decision subsets: no risk, minimal risk, moderate risk, and high risk. Maghdid et al. [26] propose a framework for early classification of COVID-19 using on-board smart-phone sensors. Specifically, they make use of temperature, inertial, proximity, color, and humidity sensors embedded in smart-phones. Their setup allows for low-cost classification. All the above attempts have one thing in common—they impute missing values in incomplete data. However, imputation is harmful because it changes data distribution and also breaks down potentially important relations between conditional features and the decision label [27]. Besides, these attempts do not handle heterogeneous data directly. They convert categorical values to 0's and 1's as a turn around, negatively impacting classification accuracy. The present work, which falls in the same category, avoids all these drawbacks.

As for distance metrics used to assess the nearness of neighbors in KNN and its derivatives, the typical metric is Euclidean distance. However, Mahalanobis distance becomes more suitable if the data are skewed or the features are correlated, as it takes into consideration data distribution. Jaafar et al. [28] report that Euclidean distance deteriorates KNN accuracy if the data is unbalanced. Therefore, they propose Mahalanobis distance for more accurate classification. Yi et al. [29] propose a classification system based on KNN suitable for robotic systems. Since robots work in real-world environments, where features are strongly correlated, they too use Mahalanobis distance. To mitigate the computational complexities involved in Mahalanobis distance, due to calculating the inverse covariance matrix of data, they employ principle component analysis (PCA) for data reduction. Fan et al. [30] also use Mahalanobis distance with KNN in the context of a framework to enhance the security of power systems, where features are typically highly correlated. In the present work, we use both Euclidean and Mahalanobis, for two reasons. First, we want to expose their differences and their impact on classification. Second, we want to demon-

strate that, regardless of which one is used, the proposed KNNV is still successful and still superior to its competitors.

### 3 Proposed Approach

In this section, we introduce the KNNV classification algorithm as a tool to identify COVID-19 cases in IHC datasets. The contribution of KNNV is twofold. First, for each unknown patient  $u_0$ , it computes a special  $K$  value that suits that patient most. Second, accurate distance calculations are employed. We start by providing some preliminaries that include the essentials of classical KNN, which will be used as a basis for our proposed KNNV.

#### 3.1 Preliminaries

In classical KNN [28], given a set  $\mathcal{U}$  of patients described by a set  $\mathcal{A}^-$  of features, a normalization function is first applied to the set  $\mathcal{N} \subset \mathcal{A}^-$  to scale the numerical features to the interval  $[0, 1]$  to prevent features with large values from outweighing those with smaller values. A normalized feature value  $\widehat{v}_{u_i, a_j}$  is obtained from its raw counterpart  $v_{u_i, a_j}$  by

$$\widehat{v}_{u_i, a_j} = \begin{cases} \frac{v_{u_i, a_j} - \min_{u_k} (v_{u_k, a_j})}{\max_{u_k} (v_{u_k, a_j}) - \min_{u_k} (v_{u_k, a_j})} & \text{if } v_{u_i, a_j} \neq * \\ * & \text{if } v_{u_i, a_j} = * \end{cases},$$

where  $\min_{u_k} (v_{u_k, a_j})$  and  $\max_{u_k} (v_{u_k, a_j})$  are the minimum and maximum values, respectively, of feature  $a_j$  across all patients  $u_k$ .

For any unknown patient  $u_0$ , KNN searches the IHC dataset for a subset  $\mathbb{U}_{K, u_0} \subset \mathcal{U}$  of patients, of size  $K > 1$ , whose elements  $u_i \in \mathbb{U}_{K, u_0}$  are closer to  $u_0$  than all other patients. Distances between the unknown patient  $u_0$  and another patient  $u_i \in \mathbb{U}_{K, u_0}$  are measured with respect to some set  $\mathcal{M} \subseteq \mathcal{A}^-$  of features using a suitable distance metric. In this article, we will use both Euclidean distance and Mahalanobis distance.

The Euclidean distance  $l(u_0, u_i)$  between patients  $u_0$  and patient  $u_i$  with respect to a set  $\mathcal{M}$  of features is given by

$$l(u_0, u_i) = \sqrt{\sum_{a_k \in \mathcal{M}} (v_{u_0, a_k} - v_{u_i, a_k})^2}. \quad (1)$$

This formula can be written in matrix form as

$$l(u_0, u_i) = \sqrt{(\mathcal{V}_{u_0} - \mathcal{V}_{u_i})^T (\mathcal{V}_{u_0} - \mathcal{V}_{u_i})}, \quad (2)$$

where  $\mathcal{V}_{u_i}$  is the feature vector of patient  $u_i$ , with  $\mathbf{X}^T$  being the transpose of matrix  $\mathbf{X}$ .



On the other hand, the Mahalanobis distance  $\tilde{l}(u_0, u_i)$  between patient  $u_0$  and patient  $u_i$  is measured with respect to a set  $\mathcal{M}$  of features as follows.

$$\tilde{l}(u_0, u_i) = \sqrt{(\mathcal{Y}_{u_0} - \mathcal{Y}_{u_i})^T \mathbf{C}^{-1} (\mathcal{Y}_{u_0} - \mathcal{Y}_{u_i})}, \tag{3}$$

where  $\mathbf{C}^{-1}$  is the inverse of the covariance matrix of the dataset that includes both  $u_0$  and  $u_i$  and the set  $\mathcal{M}$  of features. It can be seen that Mahalanobis distance reduces to Euclidean distance if the dataset covariance matrix is the identity matrix, which is the case if there is no correlation between the features. The question now is how to calculate the differences under the square roots above, especially when there are missing values.

In most applications of  $KNN$  [29], there are common rules typically applied to calculate the differences. For categorical features, the difference between two feature values is calculated as follows (basically a Hamming distance approach is used).

- If both values are existing and identical (e.g.,  $v_{u_i, a_k}$  is male and  $v_{u_0, a_k}$  is male) or both are missing, the difference is considered 0.
- Else, i.e., if both values are existing and different (e.g.,  $v_{u_i, a_k}$  is male and  $v_{u_0, a_k}$  is female) or if one is missing, then the difference is considered 1.

On the other hand, for numerical features, the difference between two feature values is calculated as follows.

- If both values are existing, the difference is calculated by normal subtraction.
- Else (i.e., if one value is missing), the difference is considered the existing value (which is tantamount to assuming 0 for the missing value).

We note from these rules a major flaw in  $KNN$ . Since numerical values are normalized to  $[0, 1]$ , the absolute difference between any two existing values is in  $[0, 1]$ . In the meantime, since categorical features are in  $\{0, 1\}$ , the absolute difference is in  $\{0, 1\}$ . Therefore, in calculating distances between patients, categorical and missing values have a greater impact on classification than existing numerical values, which leads to misclassification. This flaw is remedied in our proposed  $KNNV$ , described in the next Section.

The last step in classical  $KNN$  is to assign the unknown patient to a certain decision label value (i.e., class). Specifically, given the set  $\mathbb{U}_{K, u_0} \subseteq \mathcal{U}$  of the  $K$  nearest neighbors of an unknown patient  $u_0$ , the patient is assigned a decision

label  $v_{u_0, d} \in \{\zeta_1, \zeta_2, \dots, \zeta_r\}$  given by

$$v_{u_0, d} = \operatorname{argmax}_{\zeta_j} \sum_{u_i \in \mathbb{U}_{K, u_0}} \delta_{v_{u_i, d}, \zeta_j}, \tag{4}$$

where  $\delta_{x, y}$  is the Kronecker delta function defined as

$$\delta_{x, y} = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases}.$$

In (4), a majority vote is essentially carried out. Specifically, the sum is evaluated  $r$  times, for  $\zeta_1, \zeta_2, \dots, \zeta_r$ , and the  $\zeta_j$  that results in the highest value of the sum is the voted decision label. The sum each time is evaluated in  $K$  steps, for the  $K$  elements  $u_i$  of  $\mathbb{U}_{K, u_0}$ . Initially 0, the sum is incremented by 1 if the label  $v_{u_i, d}$  of  $u_i$  is identical to the label  $\zeta_j$  currently being considered by  $\operatorname{argmax}_{\zeta_j}$ , and is not incremented otherwise.

### 3.2 KNNV Algorithm

The problem with using a fixed  $K$  for all patients, as described above, is that it is hard to find its proper value. Furthermore, this fixed value typically depends on a threshold, whose change results in a different set of nearest patients, making classification highly volatile. It would be more advantageous, then, to determine  $K$  dynamically for each patient, based on its feature vector, and that is what  $KNNV$  does.

There are two aspects that make  $KNNV$  novel. First, it treats categorical features differently from numerical features. It uses rough set theory (RST) techniques to handle categorical features and classical distance metrics to handle numerical features. As such,  $KNNV$  does not convert categorical feature values into numbers, thereby making all the features numerical and then using distance metrics to identify nearest neighbors, as is done in classical  $KNN$  (see for example, [28–30]). By applying RST techniques to categorical features,  $KNNV$  solves two problems at once: incompleteness of those features and vagueness of the proper value of  $K$ . To this end, two RST-based definitions useful for handling categorical features are in order.

**Definition 1** (*Feature similarity relation,  $\mapsto$* ) Let  $u_0$  and  $u_i$ ,  $i \neq 0$ , be two patients, and let  $v_{u_0, a_n}$  and  $v_{u_i, a_n}$  be their values, respectively, for some categorical feature  $a_n$ . We say that the feature value  $v_{u_0, a_n}$  is similar to the feature value  $v_{u_i, a_n}$ , and denote that by  $v_{u_0, a_n} \mapsto v_{u_i, a_n}$ , if and only if  $v_{u_0, a_n} = *$  and  $v_{u_i, a_n} \neq *$ .

Now, we employ Definition 1 to define the neighborhood relation  $N$ , which relates an unknown patient  $u_0$  to another patient  $u_i$  with respect to the set  $\mathcal{C}$  of categorical features.

**Definition 2** (*Neighborhood relation,  $N$* ) Let  $u_0$  and  $u_i$ ,  $i \neq 0$ , be two patients, and let  $v_{u_0, a_n}$  and  $v_{u_i, a_n}$  be their values,

respectively, for some categorical feature  $a_n$ . We define the neighborhood relation  $N$  as the set of pairs  $(u_0, u_i)$ , such that for every  $a_n \in \mathcal{C}$ , the value  $v_{u_i, a_n}$  is either similar or equal to the value  $v_{u_0, a_n}$ , or the value  $v_{u_0, a_n}$  is either similar or equal to the value  $v_{u_i, a_n}$ . Using quantifiers, this relation is expressed as

$$N = \{(u_0, u_i) \mid \forall u_i \in \mathcal{U}, \forall a_n \in \mathcal{C} : v_{u_i, a_n} \mapsto v_{u_0, a_n} \vee v_{u_0, a_n} \mapsto v_{u_i, a_n} \vee v_{u_i, a_n} = v_{u_0, a_n}\}.$$

With the above in mind, the proposed KNNV regards the neighbors of a given unknown patient  $u_0$  as three types, categorical, numerical, and true, defined as follows.

**Definition 3 (Categorical neighbor)** A patient  $u_i$  is a *categorical* neighbor of another patient  $u_0$  if, based on their categorical feature values, the two patients are neighbors according to Definition 2.

Based on this definition, there are no “near” and “far” categorical neighbors. That is, if we search for categorical neighbors, we either get *all* of them or we get none. Also, there is no such thing as the  $K$  nearest categorical neighbors of a given patient  $u_0$ .

**Definition 4 (Numerical neighbor)** A patient  $u_i$  is a *numerical* neighbor of another patient  $u_0$  if their numerical feature values are close to each other. Closeness here is measured, as is done in classical KNN, by distance metrics such as Euclidean and Mahalanobis.

In calculating these distances, a note concerning how to calculate the difference between two feature values, of two different patients, when one is missing is in order. Unlike the common practice of assuming 0 for the missing value, KNNV assumes the existing value instead. This is tantamount to ignoring this feature altogether in the calculation. The same thing applies if the two values are missing.

Based on this definition, and since a numerical neighbor is categorized as such based on distance metrics, there can be near and far neighbors relative to a given patient  $u_0$ . One patient can be nearer than another to a given patient  $u_0$  if the distance from  $u_0$  to the former is smaller than that to the latter. As such, we may search for an arbitrary number  $K$  of numerical neighbors nearest a given patient  $u_0$ .

**Definition 5 (True neighbor)** A patient  $u_i$  is a *true* neighbor of another patient  $u_0$ , if it is both a categorical neighbor and a numerical neighbor of  $u_0$ .

Having said that, it is now easy to informally explain how KNNV works. To classify an unknown patient  $u_0$ , KNNV performs the following steps in order:

1. It eliminates from the dataset any feature whose existing values are all identical, for two reasons. First, if all the values of this feature are existing and identical, then this

feature is irrelevant to the classification process. Second, if the existing values are identical, but there are missing values, there is no clear road to substitute for the missing values without jeopardizing data semantics.

2. It searches the dataset for the categorical neighbors of  $u_0$ , according to Definition 3.
3. Assuming the search returns  $J$  categorical neighbors, it searches the dataset for the  $J$  nearest numerical neighbors of  $u_0$ , according to Definition 4.
4. With  $J$  categorical neighbors and  $J$  numerical neighbors at hand, it identifies the true neighbors among them, according to Definition 5.
5. It identifies the class (decision label value) of each true neighbor and assigns  $u_0$  to the class to which the largest number of true neighbors belong (majority vote).

Having introduced KNNV informally, we will next provide its formal specification, given that its pseudocode is listed in Algorithm 1. To this end, we associate with each unknown patient  $u_0$  three neighborhood sets, which are obtained by KNNV in the shown order:

1. The set  $\psi_{\mathcal{C}, u_0}$  of categorical neighbors, which contains the patients that are neighbors of  $u_0$  with respect to the categorical features  $\mathcal{C}$ . Specifically,

$$\psi_{\mathcal{C}, u_0} = \{u_i \mid u_i \in \mathcal{U} \wedge (u_0, u_i) \in N\}. \quad (5)$$

Let  $J$  denote the size of this set. That is,  $J = |\psi_{\mathcal{C}, u_0}|$ .

2. The set  $\psi_{\mathcal{N}, u_0}$  of nearest numerical neighbors, which contains the  $J$  patients nearest  $u_0$  with respect to numerical features  $\mathcal{N}$ . To find this set, let  $L = (l_1, l_2, \dots, l_{|\mathcal{U}|})$  be the list of distances, with respect to  $\mathcal{N}$ , between the unknown patient  $u_0$  and the patients  $u_1, u_2, \dots, u_{|\mathcal{U}|}$ , respectively. Further, let  $\mathcal{Z} = \{z_1, z_2, \dots, z_K\}$  be the set of indices of the  $J$  smallest distances in  $L$ . That is,  $\mathcal{Z}$  identifies the nearest  $J$  patients to  $u_0$  with respect to  $\mathcal{N}$ . The set  $\mathcal{Z}$  is now used to define the second neighborhood set  $\psi_{\mathcal{N}, u_0}$  as follows.

$$\psi_{\mathcal{N}, u_0} = \{u_i \mid u_i \in \mathcal{U} \wedge i \in \mathcal{Z}\}. \quad (6)$$

3. The set  $\psi_{\mathcal{A}^-, u_0}$  of true neighbors, which contains the patients nearest  $u_0$  with respect to all features  $\mathcal{A}^-$ . This set is basically the intersection of the above two sets. Specifically,

$$\psi_{\mathcal{A}^-, u_0} = \psi_{\mathcal{C}, u_0} \cap \psi_{\mathcal{N}, u_0}.$$

This set is crucial because its elements are what decide the decision label of the unknown patient  $u_0$ . We just note that if the intersection is the null set, then  $\psi_{\mathcal{A}^-, u_0}$  is made from the union of  $\psi_{\mathcal{C}, u_0}$  and  $\psi_{\mathcal{N}, u_0}$ . That is,

$$\psi_{\mathcal{A}^-,u_0} = \begin{cases} \psi_{\mathcal{C},u_0} \cap \psi_{\mathcal{N},u_0} & \text{if } \psi_{\mathcal{C},u_0} \cap \psi_{\mathcal{N},u_0} \neq \emptyset \\ \psi_{\mathcal{C},u_0} \cup \psi_{\mathcal{N},u_0} & \text{otherwise} \end{cases} . \quad (7)$$

The set  $\psi_{\mathcal{A}^-,u_0}$  of true neighbors is sufficient to discover the decision label  $v_{u_0,d}$  of that patient, which is the ultimate goal of KNNV. Simply, the algorithm just carries out the majority vote given by (4) among its elements, using  $\cup_{K,u_0} = \psi_{\mathcal{A}^-,u_0}$  and  $\mathcal{M} = \mathcal{A}^-$ , to find  $v_{u_0,d}$ .

In case the majority vote results in a tie between  $q \leq r$  classes, KNNV assigns the unknown patient  $u_0$  to the closest class. To determine the closest class, let  $\mathcal{T}$  be the set of all the classes  $\zeta_k$  participating in the tie. Now, we calculate the average distance between  $u_0$  and each class  $\mathcal{U}_{\zeta_k} \in \mathcal{T}$  in the tie and then associate  $u_0$  to the class with the smallest average, i.e.,

$$v_{u_0,d} = \operatorname{argmin}_{\mathcal{U}_{\zeta_k} \in \mathcal{T}} \frac{1}{|\mathcal{U}_{\zeta_k}|} \sum_{u_i \in \mathcal{U}_{\zeta_k}} l(u_0, u_i), \quad (8)$$

where  $l(u_0, u_i)$  is the distance between the unknown patient  $u_0$  and patient  $u_i$  with respect to set  $\mathcal{N}$  of numerical features, calculated by a suitable metric such as Euclidean or Mahalanobis. If the averages come out equal for all the classes in the tie, a very unlikely event, then the unknown patient  $u_0$  is assigned randomly to one of these classes.

### 3.3 Illustrative Example

Below, we provide a detailed example to show how KNNV decides the decision label of the unknown patient  $u_0 = \langle \text{Female}, *, \text{Yes}, \text{No}, *, 0.43, 0.79, 0.34 \rangle$ , based on the toy IHC dataset shown in Table 1.

First, set  $\psi_{\mathcal{C},u_0}$  of categorical neighbors, with respect to  $\mathcal{C} = \{a_1, a_2, \dots, a_5\}$ :

In view of Definition 3, we search the toy dataset of Table 1 for the categorical neighbors of patient  $u_0$ . It can be seen that  $u_3$ , with categorical feature vector  $\langle \text{Female}, \text{No}, *, \text{No Yes} \rangle$ , is such a neighbor. This is because the categorical feature values of  $u_3$  and  $u_0$  are either equal (features  $a_1, a_4$ ) or similar (features  $a_2, a_3, a_5$ ). Likewise  $u_5, u_6$  and  $u_{13}$  are categorical neighbors of  $u_0$ . Therefore,  $\psi_{\mathcal{C},u_0} = \{u_3, u_5, u_6, u_{13}\}$  and  $J = |\psi_{\mathcal{C},u_0}| = 4$ . Consequently, the next step is to find the 4 nearest numerical neighbors of  $u_0$ .

Second, set  $\psi_{\mathcal{N},u_0}$  of the 4 nearest numerical neighbors of  $u_0$ , with respect to  $\mathcal{N} = \{a_6, a_7, a_8\}$ :

To this end, we compute the distance between the unknown patient  $u_0$  and every patient  $u_i \in \mathcal{U}$  with respect to the numerical features  $a_j \in \mathcal{N}$ . We calculate the distance twice, once Euclidean and once Mahalanobis.

### Algorithm 1: $K$ Nearest Neighbor Variant (KNNV) algorithm

```

Input : -IHC dataset: Records (feature vectors) of a set  $\mathcal{U}$  of patients, with set  $\mathcal{A}$  of features.
         -Feature vector of patient to be classified  $u_0$ .
Output:  $v_{u_0,d}$  //Decision label of  $u_0$ 
//Eliminate irrelevant features:
1 foreach  $a_j \in \mathcal{A}^-$  do
2   if the existing feature values  $v_{u_i,a_j}$  for all  $u_i \in \mathcal{U}$  are identical then
3     Delete feature  $a_j$  from the IHC //Irrelevant feature having single value
4   end
5 end
//Compose set  $\psi_{\mathcal{C},u_0}$  of categorical neighbors, according to Definition 3:
6 Find set  $\psi_{\mathcal{C},u_0}$  of categorical neighbors, as per (5)
7  $J := |\psi_{\mathcal{C},u_0}|$ 
//Compose set  $\psi_{\mathcal{N},u_0}$  of numerical neighbors, according to Definition 4:
8  $L := \emptyset$  //Initialize set of distances to unknown patient
9  $m := |\mathcal{U}|$  //Number of patients in the IHC dataset
10 for  $i = 1$  to  $m$  do
11    $L := L \cup \{l(u_0, u_i)\}$  //As per (1) if Euclidean and as per (3) if Mahalanobis
12 end
13 Construct set  $\mathcal{Z}$  with the indices of the  $J$  smallest distances in  $L$ 
14 Compose set  $\psi_{\mathcal{N},u_0}$  of the  $J$  nearest numerical neighbors of  $u_0$  as per (6)
//Compose set  $\psi_{\mathcal{A}^-,u_0}$  of true neighbors, according to Definition 5:
15 Construct set  $\psi_{\mathcal{A}^-,u_0}$  as per (7)
//Predict the decision label  $v_{u_0,d}$  of unknown object  $u_0$  by majority vote:
16 foreach  $\zeta_j \in \{\zeta_1, \zeta_2, \dots, \zeta_r\}$  do
17   Find the number  $\xi_j$  of objects with decision label  $\zeta_j$ 
18 end
19 if  $\xi_k$  is the largest in  $\{\xi_1, \xi_2, \dots, \xi_r\}$  and is unique then
//There is no tie
20   Assign  $u_0$  the decision label  $\zeta_k$ , i.e.  $v_{u_0,d} = \zeta_k$ 
21 else
//If  $\xi_k$  is not unique, there is a tie
22   Find decision label  $v_{u_0,d}$  as per (8)
23 end
24 end

```

- Euclidean distance, as per (1):  
 We start by finding the distance between  $u_0$  with the numerical feature vector  $\langle 0.43, 0.79, 0.34 \rangle$  and  $u_1$  with numerical feature vector  $\langle 0.97, *, 0.78 \rangle$ . For feature  $a_6$ , with both  $v_{u_1,a_6} \neq *$  and  $v_{u_0,a_6} \neq *$ , the difference is  $0.97 - 0.43 = 0.54$ . For  $a_7$ , the difference is considered 0 because  $v_{u_1,a_7} = *$ . For feature  $a_8$ , the difference is  $0.78 - 0.34 = 0.44$ . Therefore,  $l(u_0, u_1) = \sqrt{0.54^2 + 0^2 + 0.44^2} = 0.69$ .  
 Repeating the previous procedure with all  $u_i \in \mathcal{U}$  ends up with the list  
 $L = (0.69, 0.5, 0.34, 0.35, 0.78, 0.24, 0.55, 0.37, 0.57, 0.49, 0.61, 0.42, 0.23)$ . The next step is to find the set  $\mathcal{Z}$  of indices of the four (since  $J = 4$ ) nearest patients to

the unknown patient. By inspection,  $\mathcal{Z} = \{13, 6, 3, 4\}$ , corresponding to the distances  $\{0.23, 0.24, 0.34, 0.35\}$ . It follows from (6) that the set of the 4 nearest numerical neighbors is  $\psi_{\mathcal{N}, u_0} = \{u_3, u_4, u_6, u_{13}\}$ .

Third, set  $\psi_{\mathcal{A}^-, u_0}$  of true neighbors of  $u_0$ , with respect to  $\mathcal{A}^- = \{a_1, a_2, \dots, a_8\}$ :

As per (7), the third neighborhood set  $\psi_{\mathcal{A}^-, u_0}$  is the intersection of  $\psi_{\mathcal{E}, u_0}$  and  $\psi_{\mathcal{N}, u_0}$ . It follows that the third neighborhood set is  $\psi_{\mathcal{A}^-, u_0} = \{u_3, u_6, u_{13}\}$ . Noting that  $v_{u_3, d} = v_{u_6, d} = \text{Flu}$  and  $v_{u_{13}, d} = \text{COVID-19}$ , i.e., a vote of 2 to 1, the decision label of  $u_0$  is  $v_{u_0, d} = \text{Flu}$ .

## 2. Mahalanobis distance, as per (3):

We will repeat here the calculations done above using Euclidean distance, this time using Mahalanobis distance. First, we compute the covariance matrix  $\mathbf{C}$ , with size  $3 \times 3$  since we have three numerical features, as follows.

$$\mathbf{C} = \begin{bmatrix} 0.0995 & -0.0334 & 0.003 \\ -0.0334 & 0.0499 & -0.019 \\ 0.003 & -0.019 & 0.046 \end{bmatrix}.$$

The inverse matrix is

$$\mathbf{C}^{-1} = \begin{bmatrix} 13.39 & 10.24 & 3.36 \\ 10.24 & 31.61 & 12.39 \\ 3.36 & 12.39 & 26.64 \end{bmatrix}.$$

With Mahalanobis distance we represent the numerical feature vector as a column vector. Therefore, the difference between  $u_0$  and  $u_1$  is computed as follows.

$$\mathcal{V}_{u_0} - \mathcal{V}_{u_1} = \begin{bmatrix} -0.54 \\ 0 \\ -0.44 \end{bmatrix}.$$

Finally, the distance between  $u_0$  and  $u_1$  is computed as follows.

$$\begin{aligned} \tilde{l}^2(u_0, u_1) &= [-0.54 \quad 0 \quad -0.44] \begin{bmatrix} 13.39 & 10.24 & 3.36 \\ 10.24 & 31.61 & 12.39 \\ 3.36 & 12.39 & 26.64 \end{bmatrix} \\ &= \begin{bmatrix} -0.54 \\ 0 \\ -0.44 \end{bmatrix} = 10.65, \end{aligned}$$

which gives  $\tilde{l}(u_0, u_1) = 3.26$ .

Repeating the previous procedure with all  $u_i \in \mathcal{U}$  ends up with the list

$L = (3.26, 2.11, 1.31, 1.23, 3.41, 1.27, 2.01, 2.03, 2.1, 2.28, 2.56, 1.53, 1.34)$ .

The next step is to find the set  $\mathcal{Z}$  of indices of the 4, since  $J = 4$ , smallest distances. By inspec-

tion,  $\mathcal{Z} = \{3, 4, 6, 13\}$ , corresponding to the distances  $\{1.23, 1.27, 1.31, 1.34\}$ . It follows from (6) that the set of the 4 nearest numerical neighbors is  $\psi_{\mathcal{N}, u_0} = \{u_3, u_4, u_6, u_{13}\}$ .

Set  $\psi_{\mathcal{A}^-, u_0}$  of true neighbors of  $u_0$ , with respect to  $\mathcal{A}^- = \{a_1, a_2, \dots, a_8\}$ :

As per (7),  $\psi_{\mathcal{A}^-, u_0} = \{u_3, u_6, u_{13}\}$ . Noting that  $v_{u_3, d} = v_{u_6, d} = \text{Flu}$  and  $v_{u_{13}, d} = \text{COVID-19}$ , the decision label of  $u_0$  is  $v_{u_0, d} = \text{Flu}$ , which is what is obtained using Euclidean distance above. This is of course a coincidence and is not always the case.

## 4 Experimental Work

The KNNV algorithm is coded in MATLAB R16a and run on a PC with Centos 7, Intel(R) Core(TM) i7 CPU 2.4 GHz with 16 GB of main memory.

There are two objectives for the experiments carried out in this Section. The first is to validate the KNNV algorithm by showing that it operates successfully as intended and handles both heterogeneity and incompleteness in IHC data properly. The second objective is to show its superiority to some other KNN derivatives, namely modified KNN (MKNN) [31], KNN for imperfect data (KNN<sub>imp</sub>) [32], and cost-sensitive KNN (csKNN) [9]. For each algorithm, the precision, recall, accuracy, and  $F$ -score [33] metrics are evaluated. As an additional value, we show how to use KNNV as a tool to test the classification significance of each conditional feature with respect to the final classification decision.

The classification performance is analyzed using a 10-fold cross-validation method. That is, the whole IHC dataset is split into ten equal sub-datasets, nine serving for training and the tenth for testing, such that each patient appears in a test set once and in training sets nine times. After running the algorithm independently 10 times, the results are averaged and then presented. As the value of  $K$  affects the performance of the competitor algorithms, namely MKNN, KNN<sub>imp</sub> and csKNN, we first test their performance for  $K = 3, 5, \dots, 9$ , and choose for each algorithm the value giving the best results. Accordingly,  $K = 5$  was chosen for MKNN,  $K = 7$  for KNN<sub>imp</sub> and  $K = 5$  for csKNN.

Each experiment in this section is carried out twice: once using Euclidean and once using Mahalanobis. As a reminder, these metrics are used to determine the  $J$  nearest numerical neighbors of the patient under consideration, where  $J$  is the cardinality of the set of categorical neighbors. It is worth mentioning that we follow the same strategy for handling the missing values of numerical features with Mahalanobis distance as we do with Euclidean distance. Specifically, if either the value of feature  $a_n$  in one of the two patients ( $v_{u_0, a_n}, v_{u_i, a_n}$ ) is missing or both are missing, the difference is considered 0.



**Table 2** IHC dataset used in the experiments, with 68 COVID patients and 62 Flu patients described by 16 conditional features  $a_i$  and one decision feature  $d$

	Feature	Type	Value set
$a_1$	Age	Numerical	[4–90]
$a_2$	Gender	Categorical	{Male, female}
$a_3$	Fever	Categorical	{Yes, no}
$a_4$	Dyspnea	Categorical	{Yes, no}
$a_5$	Nasal	Categorical	{Yes, no}
$a_6$	Cough	Categorical	{Yes, no}
$a_7$	Partial pressure of oxygen (PO2)	Numerical	[32–292]
$a_8$	C-reactive protein (CRP)	Numerical	[0.75–23]
$a_9$	Asthenia	Categorical	{Yes, no}
$a_{10}$	Leukopenia	Categorical	{Yes, no}
$a_{11}$	Exposure to COVID-19 patients	Categorical	{Yes, no}
$a_{12}$	Coming from high risk zone	Categorical	{Yes, no}
$a_{13}$	Temperature	Numerical	[35.7–40]
$a_{14}$	Blood test	Categorical	{Yes, no}
$a_{15}$	Polymerase chain reaction (RT-PCR)	Categorical	{Positive, negative}
$a_{16}$	Medical history	Categorical	{Cancer, croonic, astham, COPD, chronic, DM}
$d$	Decision label	Categorical	{COVID-19, Flu}

### 4.1 Dataset

It is essential to have a dataset containing COVID-19 cases to test our  $K$  NNV classification algorithm on. However, we could not find such a dataset with a mix of COVID-19 and non-COVID-19 cases. The only solution to have the desired mixed dataset was then for us to construct one manually. First, we obtained a dataset of 68 COVID-19 cases from the SIRM database [10]. Second, for Non-COVID-19 cases, we obtained a dataset of 62 Flu cases from the Influenza Research Database (IRD) [34]. Then, we merged the two datasets and shuffled them randomly to obtain the IHC dataset we use in this Section, with two decision labels: COVID-19 and Flu.

We note that we faced a problem with the SIRM dataset. Specifically, the data was unstructured, in the sense that the feature values of each patient were described verbally as a paragraph. Therefore, we had to structure the data ourselves in a format consistent with that of the Flu dataset. The resulting IHC dataset, used in our experiments, contains features that are categorical, such as cough, and features that are numerical, such as partial pressure of oxygen (PO2). The features are described in Table 2. Additionally, since not all the patients have their records complete (for example, some patients have a blood test and others do not), there are missing values. In fact, about 44% of the feature values are missing.

### 4.2 Experiment 1: Performance Evaluation of KNNV

We have assessed the performance of  $K$ NNV using the following four metrics:

$$\text{Precision} = \frac{TP}{TP + FN}, \tag{9}$$

$$\text{Recall} = \frac{TP}{TP + FP}, \tag{10}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{11}$$

$$F\text{-Score} = 2 \left( \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \right), \tag{12}$$

where

1. TP: Number of True Positive cases, COVID-19 patients that are properly classified as COVID-19,
2. FP: Number of False Positive cases, Flu patients that are wrongly classified as COVID-19,
3. TN: Number of True Negative cases, Flu patients that are properly classified as Flu, and
4. FN: Number False Negative cases, COVID-19 patients that are wrongly classified as Flu.

The values of the above metrics for  $K$ NNV and some related algorithm are reported in Table 3, for both Euclidean and Mahalanobis distances. The table shows vividly that, under either distance metric,  $K$ NNV achieves better results than the related algorithms. For example, the competitor

**Table 3** Average values of precision, recall, accuracy, and *F*-score achieved by *KNNV* and three related algorithms using both Euclidean and Mahalanobis distances

	Euclidean				Mahalanobis			
	<i>MKNN</i>	<i>csKNN</i>	<i>KNN<sub>imp</sub></i>	<i>KNNV</i>	<i>MKNN</i>	<i>csKNN</i>	<i>KNN<sub>imp</sub></i>	<i>KNNV</i>
Precision	0.59	0.57	0.81	<b>0.91</b>	0.42	0.39	0.71	<b>0.95</b>
Recall	0.61	0.66	0.76	<b>0.87</b>	0.49	0.70	0.64	<b>0.76</b>
Accuracy	0.66	0.51	0.67	<b>0.88</b>	0.49	0.49	0.71	<b>0.84</b>
<i>F</i> -Score	0.61	0.59	0.65	<b>0.88</b>	0.46	0.48	0.68	<b>0.84</b>

**Table 4** Maximum values of precision, recall, accuracy, and *F*-score achieved by *KNNV* and three related algorithms using both Euclidean and Mahalanobis distances

	Euclidean				Mahalanobis			
	<i>MKNN</i>	<i>csKNN</i>	<i>KNN<sub>imp</sub></i>	<i>KNNV</i>	<i>MKNN</i>	<i>csKNN</i>	<i>KNN<sub>imp</sub></i>	<i>KNNV</i>
Precision	0.77	0.65	0.89	<b>1</b>	0.53	0.41	0.77	<b>1</b>
Recall	0.81	1	0.93	<b>1</b>	0.60	1	0.67	<b>1</b>
Accuracy	1	0.69	1	<b>1</b>	0.60	0.55	0.77	<b>1</b>
<i>F</i> -Score	0.84	0.79	1	<b>1</b>	0.54	0.52	0.71	<b>1</b>

**Table 5** Minimum values of precision, recall, accuracy, and *F*-score achieved by *KNNV* and three related algorithms using both Euclidean and Mahalanobis distances

	Euclidean				Mahalanobis			
	<i>MKNN</i>	<i>csKNN</i>	<i>KNN<sub>imp</sub></i>	<i>KNNV</i>	<i>MKNN</i>	<i>csKNN</i>	<i>KNN<sub>imp</sub></i>	<i>KNNV</i>
Precision	0.39	0.44	0.60	<b>0.70</b>	0.34	0.36	0.58	<b>0.71</b>
Recall	0.24	0.32	0.44	<b>0.75</b>	0.33	0.29	0.51	<b>0.43</b>
Accuracy	0.33	0.43	0.58	<b>0.72</b>	0.41	0.31	0.51	<b>0.72</b>
<i>F</i> -Score	0.31	0.41	0.55	<b>0.77</b>	0.38	0.29	0.50	<b>0.60</b>

algorithms, *KNN<sub>imp</sub>*, achieve 0.81 for precision, whereas *KNNV* achieves 0.91. This means that *KNNV* outperforms *KNN<sub>imp</sub>* by about 10%. In view of the equations of the four evaluation metrics, this indicates that it has high values for both TP and TN, and low values for both FP and FN. This is attributed mainly to the RST techniques used in *KNNV*, to handle incompleteness and heterogeneity and to find the proper *K* value for the patient under classification.

The same results are reaffirmed in Tables 4 and 5 which outline the best and worst values of the four evaluation metrics, for *KNNV* and the related algorithms. Over the 10 runs made, *KNNV* achieves the best results, reason enough to conclude that *KNNV* can accurately identify COVID-19 cases even when the data are both heterogeneous and incomplete.

By comparing the Euclidean and Mahalanobis results, we observe two things. First, there is a difference between the two sets of results, but this difference is not significant (within 4%). This insignificance could be an indication that the dataset is either not highly skewed or not having great correlation among its features. The second observation is that under both distance metrics, *KNNV* keeps being superior to the three algorithms of the comparison.

### 4.3 Experiment 2: Feature Significance

To better understand COVID-19, this experiment is dedicated to investigate the impact of each individual feature on classifying the disease. In particular, the classification accuracy is obtained for each feature separately and independently in the aim to rank the impact of the feature on the classification decision. While testing numerical features,  $\psi_{\mathcal{C}, u_0} = \emptyset$ , and therefore, we set  $K = 1$ .

Table 6 shows the average classification accuracy of *KNNV* for each feature over 10 runs. A look at the table shows that nasal has the highest classification impact. This is reasonable as COVID-19 spreads mainly by droplets produced when people cough, sneeze, or talk. Fever, asthenia, and leukopenia come in second, third, and fourth, respectively. This agrees with what the world health organization (WHO) asserts in its report [35]: “at the beginning, the symptoms of COVID-19 are similar to those of a Flu.” Also, as per this report, if we are not sure about the nasal, fever, asthenia, and leukopenia features, we should look at the patient age. According to the report, if the patient is old, having

**Table 6** Mean classification accuracy of *K*NNV for each feature separately

	Feature	Accuracy	Rank
$a_1$	Age	0.50	9
$a_2$	Gender	0.52	5
$a_3$	Fever	0.59	2
$a_4$	Dyspnea	0.46	13
$a_5$	Nasal	0.65	1
$a_6$	Cough	0.47	12
$a_7$	Partial pressure of oxygen (PO <sub>2</sub> )	0.48	11
$a_8$	C-reactive protein (CRP)	0.52	6
$a_9$	Asthenia	0.54	3
$a_{10}$	Leukopenia	0.53	4
$a_{11}$	Exposure to COVID-19 patients	0.52	7
$a_{12}$	Coming from high risk zone	0.52	8
$a_{13}$	Temperature	0.44	14
$a_{14}$	Blood test	0.56	15
$a_{15}$	Polymerase chain reaction (RT-PCR)	0.49	10
$a_{16}$	Medical history	0.53	16

a weak immunity system, he/she is likely to be COVID-19 positive and should undertake other tests like RT-PCR and CRP. Indeed, Table 6 shows that the age, RT-PCR, and CRP features come after nasal, fever, asthenia, and leukopenia features. It also shows that blood analysis and medical history features have the least impact on COVID-19 classification, agreeing with the mentioned WHO report which does not even mention blood analysis and medical history as relevant in diagnosing COVID-19.

## 5 Conclusions

The *K*NNV algorithm proposed in this article is designed principally to classify COVID-19 in IHC datasets. Its design starts with the classical *K*NN algorithm as a basis; then, enhancements are added in a novel way. The novelty is basically in the use of RST techniques to handle both incompleteness and heterogeneity and to obtain an ideal *K* value for each patient to be classified. These additions have greatly improved the performance as demonstrated by the experimental work that is carried out.

The *K*NNV has been implemented and tested on a COVID-19 dataset from the Italian Society of Medical and Interventional Radiology (SIRM). It was also compared to three algorithms of its category. The test results show that *K*NNV can efficiently and accurately classify COVID-19 cases. The comparison results show that the algorithm greatly outperforms all its competitors in terms of four metrics: precision, recall, accuracy, and *F*-score, under both Euclidean

and Mahalanobis distance metrics. The approach given in this article can be applied for the identification of other diseases. Moreover, its ideas can be further developed to suite general classification problems outside the medical field.

**Funding** Not applicable.

## References

- World Health Organization: Coronavirus disease 2019 (COVID-19): situation report, 72 (2020)
- Cao, J.; et al.: Clinical features and short-term outcomes of 102 patients with corona virus disease 2019 in Wuhan, China. *Clin. Infect. Dis.* **71**(15), 748–755 (2020). <https://doi.org/10.1093/cid/ciaa243>
- Li, K.; et al.: CT image visual quantitative evaluation and clinical classification of coronavirus disease (COVID-19). *Eur. Radiol.* **30**, 4407–4416 (2020). <https://doi.org/10.1007/s00330-020-06817-6>
- Wang, Q.; et al.: Local neighborhood rough set. *Knowl.-Based Syst.* **153**, 53–64 (2018). <https://doi.org/10.1016/j.knsys.2018.04.023>
- Hamed, A.; Sobhy, A.; Nassar, H.: Distributed approach for computing rough set approximations of big incomplete information systems. *Inf. Sci.* **547**, 427–449 (2021). <https://doi.org/10.1016/j.ins.2020.08.049>
- Zhang, Y.; et al.: Large-scale multi-label classification using unknown streaming images. *Pattern Recognit.* (2020). <https://doi.org/10.1016/j.patcog.2019.107100>
- Deng, Z.; et al.: Efficient *k*NN classification algorithm for big data. *Neurocomputing.* **195**, 143–148 (2016). <https://doi.org/10.1016/j.neucom.2015.08.112>
- Shmueli, G.; et al.: *Data Mining for Business Analytics: Concepts, Techniques, and Applications in R*. Wiley, Hoboken (2017)
- Zhang, S.: Cost-sensitive *K*NN classification. *Neurocomputing* **391**, 234–242 (2020). <https://doi.org/10.1016/j.neucom.2018.11.101>
- Italian Society of Medical and Interventional Radiology (SIRM). <https://www.sirm.org/en/category/articles/covid-19-database/>
- Roosa, K.; et al.: Real-time forecasts of the COVID-19 epidemic in China from February 5th to February 24th, 2020. *Infect. Dis. Model.* **5**, 256–263 (2020). <https://doi.org/10.1016/j.idm.2020.02.002>
- Pirouz, B.; et al.: Investigating a serious challenge in the sustainable development process: analysis of confirmed cases of COVID-19 (new type of coronavirus) through a binary classification using artificial intelligence and regression analysis. *Sustainability* (2020). <https://doi.org/10.3390/su12062427>
- McCall, B.: COVID-19 and artificial intelligence: protecting health-care workers and curbing the spread. *Lancet Digit. Health.* (2020). [https://doi.org/10.1016/S2589-7500\(20\)30054-6](https://doi.org/10.1016/S2589-7500(20)30054-6)
- Hu, Z., et al.: Artificial intelligence forecasting of covid-19 in China (2020). arXiv preprint [arXiv:2002.07112](https://arxiv.org/abs/2002.07112).
- Zheng, N.; et al.: Predicting COVID-19 in China using hybrid AI model. *IEEE Trans. Cybern.* **50**(7), 2891–2904 (2020). <https://doi.org/10.1109/TCYB.2020.2990162>
- Santosh, K.C.: AI-driven tools for coronavirus outbreak: need of active learning and cross-population train/test models on multitudinal/multimodal data. *J. Med. Syst.* (2020). <https://doi.org/10.1007/s10916-020-01562-1>
- Boldog, P.; et al.: Risk assessment of novel coronavirus COVID-19 outbreaks outside China. *J. Clin. Med.* (2020). <https://doi.org/10.3390/jcm9020571>
- Gozes, O., et al.: Rapid AI development cycle for the coronavirus (covid-19) pandemic: initial results for automated detection &



- patient monitoring using deep learning ct image analysis (2020). arXiv preprint [arXiv:2003.05037](https://arxiv.org/abs/2003.05037)
19. Ai, T.; et al.: Correlation of chest CT and RT-PCR testing in coronavirus disease 2019 (COVID-19) in China: a report of 1014 cases. *Radiology* (2020). <https://doi.org/10.1148/radiol.202000642>
  20. Barstugan, M.; Ozkaya, U.; Ozturk, S.: Coronavirus (COVID-19) classification using CT images by machine learning methods (2020). arXiv preprint [arXiv:2003.09424](https://arxiv.org/abs/2003.09424)
  21. Butt, C.; et al.: Deep learning system to screen coronavirus disease 2019 pneumonia. *Appl. Intell.* (2020). <https://doi.org/10.1007/s10489-020-01714-3>
  22. Wang, S.; et al.: A deep learning algorithm using CT images to screen for corona virus disease (COVID-19). *medRxiv* (2020). <https://doi.org/10.1101/2020.02.14.20023028>
  23. Li, L.; et al.: Using artificial intelligence to detect COVID-19 and community-acquired pneumonia based on pulmonary CT: evaluation of the diagnostic accuracy. *Radiology* (2020). <https://doi.org/10.1148/radiol.202000905>
  24. Peng, M.; et al.: Artificial intelligence application in COVID-19 diagnosis and prediction. *SSRN Electron. J.* (2020). <https://doi.org/10.2139/ssrn.3541119>
  25. Rao, A.S.S.; Vazquez, J.A.: Identification of COVID-19 can be quicker through artificial intelligence framework using a mobile phone-based survey in the populations when cities/towns are under quarantine. *Infect. Control Hosp. Epidemiol.* (2020). <https://doi.org/10.1017/ice.2020.61>
  26. Maghdid, H.S.; et al.: A novel AI-enabled framework to diagnose coronavirus covid 19 using smartphone embedded sensors: design study. In: 2020 IEEE 21st International Conference on Information Reuse and Integration for Data Science (IRI), Las Vegas, NV, USA, 2020. pp. 180–187 (2020). <https://doi.org/10.1109/IRI49571.2020.00033>
  27. Cao, T.; et al.: Rough set model in incomplete decision systems. *J. Adv. Comput. Intell. Intell. Inform.* **21**, 1221–1231 (2017). <https://doi.org/10.20965/jaciii.2017.p1221>
  28. Jaafar, H.; Ramli, N.H.; Abdul Nasir, A.S.: An improvement to the k-nearest neighbor classifier for ECG database. In: IOP Conference on Series: Materials Science and Engineering, Penang, Malaysia. pp. 1–10 (2018)
  29. Yi, C.; et al.: A novel method to improve transfer learning based on Mahalanobis distance. In: 2017 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 2279–2283. IEEE (2018)
  30. Fan, H.; et al.: Post-fault transient stability assessment based on k-nearest neighbor algorithm with Mahalanobis distance. In: 2018 International Conference on Power System Technology (POWERCON), pp. 4417–4423. IEEE (2018)
  31. Ayyad, S.M.; Saleh, A.I.; Labib, L.M.: Gene expression cancer classification using modified K-Nearest Neighbors technique. *BioSystems.* **176**, 41–51 (2019). <https://doi.org/10.1016/j.biosystems.2018.12.009>
  32. Cadenas, J.M.; et al.: A fuzzy K-nearest neighbor classifier to deal with imperfect data. *Soft. Comput.* **22**, 3313–3330 (2018). <https://doi.org/10.1007/s00500-017-2567-x>
  33. Goutte, C.; Gaussier, E.: A probabilistic interpretation of precision, recall and F-score, with implication for evaluation. In: European Conference on Information Retrieval. pp. 345–359. Springer, Berlin (2005). [https://doi.org/10.1007/978-3-540-31865-1\\_25](https://doi.org/10.1007/978-3-540-31865-1_25)
  34. Influenza Research Database. <https://www.fludb.org/brc/home.spg?decorator=influenza>
  35. World Health Organization: Laboratory testing for coronavirus disease 2019 (COVID-19) in suspected human cases: interim guidance, 2 March 2020 (No. WHO/COVID-19/laboratory/2020.4). World Health Organization (2020)