Published in final edited form as:

Mach Learn Appl. 2022 December 15; 10: . doi:10.1016/j.mlwa.2022.100437.

An automated treatment plan alert system to safeguard cancer treatments in radiation therapy

Paul M. Kump[a,*], **Junyi Xia**[b], **Sridhar Yaddanapudi**[c], **Erwei Bai**[d]

[a]Department of Electrical Engineering, SUNY Maritime College, 6 Pennyfield Ave., Bronx, NY 10465, USA

[b]Department of Radiation Oncology, Mount Sinai Hospital, New York City, NY, USA

[c]Department of Radiation Oncology, University of Iowa, IA, USA

[d]Department of Electrical and Computer Engineering, University of Iowa, Iowa City, IA, USA

Abstract

In radiation oncology, the intricate process of delivering radiation to a patient is detailed by the patient's treatment plan, which is data describing the geometry, construction and strength of the radiation machine and the radiation beam it emits. The patient's life depends upon the accuracy of the treatment plan, which is left in the hands of the vendor-specific software automatically generating the plan after an initial patient consultation and planning with a medical professional. However, corrupted and erroneous treatment plan data have previously resulted in severe patient harm when errors go undetected and radiation proceeds. The aim of this paper is to develop an automatic error-checking system to prevent the accidental delivery of radiation treatment to an area of the human body (i.e., the treatment site) that differs from the plan's documented intended site. To this end, we develop a method for structuring treatment plan data in order to feed machine-learning (ML) classifiers and predict a plan's treatment site. In practice, a warning may be raised if the prediction disagrees with the documented intended site.

The contribution of this paper is in the strategic structuring of the complex, intricate, and nonuniform data of modern treatment planning and from multiple vendors in order to easily train ML algorithms. A three-step process utilizing up- and down-sampling and dimension reduction, the method we develop in this paper reduces the thousands of parameters comprising a single treatment plan to a single two-dimensional heat map that is independent of the specific vendor or construction of the machine used for treatment. Our heat-map structure lends itself well to feed well-established ML algorithms, and we train-test random forest, softmax, k-nearest

*Corresponding author. pkump@sunymaritime.edu (P.M. Kump).

CRediT authorship contribution statement

Paul M. Kump: Methodology, Investigation, Software, Formal analysis, Writing – original draft, Writing – review & editing. **Junyi Xia:** Conceptualization, Methodology, Funding acquisition, Data curation, Writing – review & editing. **Sridhar Yaddanapudi:** Conceptualization, Data curation, Writing – review & editing. **Erwei Bai:** Methodology, Investigation, Funding acquisition, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: The authors declare that project funding was awarded by National Cancer Institute of the National Institutes of Health.

neighbors, shallow neural network, and support vector machine using real clinical treatment plans from several hospitals in the United States. The paper demonstrates that the proposed method characterizes treatment sites so well that ML classifiers may predict head-neck, breast, and prostate treatment sites with an accuracy of about 94%. The proposed method is the first step towards a thorough, fully automated error-checking system in radiation therapy.

Keywords

Cancer classification; Radiation heat map; Nonuniform treatment data

1. Introduction

1.1. Background

The global use of medical radiation as a treatment for cancer is increasing, with half of all cancer patients receiving radiation therapy sometime during the course of their treatments (NCI, 2021). Moreover, the average lifetime radiation dose has increased by seven-hundred percent over the last four decades. Several factors are contributing to this global trend, including increasing human life expectancy and significant technological advancements in radiation oncology, for example, in both the equipment and the methodologies used to deliver radiation treatment (Ganesh, 2014).

From initial patient consultation to treatment, the radiation therapy process is a complex one with many steps. During consultation, the patient discusses treatment options with the oncologist. If the patient is to receive radiation therapy, the patient proceeds with a simulation process that includes a computerized tomography (CT) scan for designing the treatment plan. The oncologist outlines the tumor in the image as well as organs at risk that radiation should avoid. A computerized system then designs the treatment plan, which is reviewed and approved by the oncologist. After approval, the treatment plan undergoes quality and safety checks by the medical physicist. Finally, the radiation therapist delivers the treatment using a medical linear accelerator (LINAC), which emits high-energy radiation photon or electron beams.

Mounted to a gantry and shaping the beam emitted by the LINAC is an attached multi-leaf collimator (MLC), which, when properly configured, restricts the radiation beam to the treatment target in order to avoid radiation damage to the surrounding healthy tissues. Fig. 1 shows a Varian TrueBeam LINAC along with a close-up of its MLC, which features two opposing banks of 60 tungsten leaves each. Although new technologies such as modern LINACs provide physicians with advanced tools for cancer treatment, their complexity has created many sources of errors, such as software flaws, faulty programming, poor safety procedures, or inadequate staffing and training (Bogdanich, 2010).

The New York Times reported a fatal accident (Bogdanich, 2010) where a failure to detect a computer error caused a lethal dose of radiation to be delivered to the brainstem instead of the tumor. Although this accident could be prevented by pre-treatment dose measurements or careful treatment monitoring, similar accidents could occur in a clinic with a suboptimal

plan-check workflow. No comprehensive study is available for the number of radiation errors resulting from death (French, 2002). Providers may fail to report events for many reasons, including fear of publicity or punitive actions, lack of knowledge of the event, or misunderstandings about the reporting system (Rosenthal & Booth, 2005). In 2011, the American Society of Radiation Oncology partnered with the American Association of Physicists in Medicine to establish a nationwide voluntary indicent learning system, RO-ILS (Hoopes et al., 2015), to collect and analyze radiation oncology incidents. As of the second quarter in 2022, 800 participating institutions reported a total of 24,487 incidents, with at least 2% categorized as severe (ASTRO, 2022). Thus, it can be expected that 220 patients experience severe radiation therapy incidents annually in the United States given the approximately 3600 radiation therapy institutions that populate the nation.

Clearly, there is a need to develop a classification method to verify whether the treatment site associated with the treatment plan is consistent with the treatment prescription. Such a system could prevent accidents similar to the one mentioned above because it would alert users when the treatment plan is incorrect or erroneous. As a safety check, we propose an automated treatment plan alert system to be inserted into the radiation therapy process flow, as shown in Fig. 2. Once the treatment plan is reviewed and approved by the oncologist, checked by the medical physicist, and uploaded to the local radiation oncology information system, our system reviews the plan parameters and predicts the intended treatment site. If the predicted treatment site agrees with the documented intended site, treatment may proceed as planned. If not, medical professionals can manually review the plan and make changes as necessary, ultimately exporting the corrected plan.

1.2. Related and current work

Previous work in this area aims only to detect, rather than classify, anomalies in radiation treatment plans. For example, Kalet et al. (2015) applied a Bayesian model to detect abnormal treatment plans by learning parameter distributions from a set of historical treatment plans. Much of the literature analyzes simple treatment plans which are derived from the use of an older, now seldomly used technique in which the radiation beam is confined to a rectangular region (so-called “four-field box plans”). For example, Azmandian et al. (2007) used a K-means clustering method to detect planning errors of four-field box plans, achieving a detection rate of 77% to 100%. Kisling et al. (2020) used a deep learning model trained on beam apertures and digitally reconstructed radiographs to verify the clinical acceptability of four-field box plans.

Rather than four-field box plans, the treatment plans analyzed in the current paper are more relevant for modern technologies and much more intricate and comprehensive. Specifically, our study includes Volumetric Modulated Arc Therapy (VMAT) treatment plans, which typically involve more than 100 gantry angles in a single plan; whereas only four gantry angles comprise the aforementioned four-field box plans. Our plans include MLC aperture information, MLC leaf widths, gantry angles, MLC angles, and beam energies—all of which, except for leaf widths, potentially variable as treatment is being delivered. Furthermore, the modern treatment plan data are represented with vendor-specific formats

which vary across clinics and LINACs (Boyer et al., 2001, pp. 1–5). This last fact can make it impossible to train ML classifiers without a data preprocessing stage.

In fact, before developing the methods in the current paper, the authors experimented with many different ways of representing treatment plan data for ML classification, with varying results, none rivaling those of the current paper. ith so much raw data contained within oncology databases, identifying a convenient and useful set of features to capture salient treatment plan information is a difficult task. With only a modest classification accuracy, the authors' previous best results (Bai & Xia, 2021) were obtained by representing each treatment plan by the average variation in its gantry angle and in its leaf positions. Clearly the data representation (Bai & Xia, 2021) was inadequate, most likely since dose, an important factor, was not considered.

1.3. Paper overview

This paper describes an automatic error-checking system for preventing the accidental delivery of radiation treatment to an unintended area of the patient's body. The main contribution of this work is the development of the sophisticated, nuanced process of extracting only the useful, but low-dimensional information from the expansive, detailed and nonuniform plans of modern LINACs and its important role in classifying cancer treatment sites. Only using leaf positions and dose information from the many parameters comprising a treatment plan, the three-step method we develop is to (1) convert each treatment plan to a corresponding heat map, (2) up- or down-sample heat maps to achieve a uniform size across all heat maps, and (3) apply dimension reduction to convert each heat map to a low-dimensional vector.

Using real treatment plans retrieved from multiple clinics, we test our three-step method by training many ML classifiers and observing their ability to classify each plan's treatment site provided the low-dimensional data extracted by our method. The data are so well clustered among similar intended uses, the paper shows, that ML classification algorithms can predict the intended use with a classification accuracy of about 94%—a result which is difficult to place into proper context, since the literature contains no previous results relating to the classification of modern treatment plans and from multiple clinics. However, by structuring raw radiation oncology data in an ML-friendly format, the paper lays the groundwork for a fully autonomous error-checking system in modern radiation therapy.

The rest of the paper is organized as follows: Section 2 describes the data set, the data structuring method, and the classifiers employed to test the structuring method. Section 3 presents the classification results of training/testing the classifiers, and we report the performance for many different tunings of each classifier. Section 4 summarizes the paper and provides insights towards the practical deployment of the method.

2. Data and methods

2.1. Data curation and description

The $n = 697$ radiation treatment plans analyzed in this study were head-neck, breast, and prostate treatment sites from two separate clinics in the United States. Treatment plans from

one clinic were generated from the Pinnacle (Philips Healthcare) treatment planning system, those from the other clinic from the Varian Eclipse system. Regardless of the system, data were exported for analysis from the MOSAIQ (Elekta AB, Stockholm, Sweden) oncology information system using customized SQL (Structured Query Language) queries. The exact count of each treatment site in the study is shown in Table 1.

Thousands of parameters characterize the raw unstructured treatment plan data stored in oncology databases. For example, the shape of the beam radiated by the LINAC is described by the positions of the gantry, MLC and leaves which demand many parameters to describe precisely. Further complicating the matter is that these parameters may change throughout the delivery of radiation in a single plan. With no universal set of parameters or data representation in existence, representing the data in a structured manner is a challenge. In our study, two different LINACs each employ its own parameters and conventions. For example, Table 2, which lists the subset of raw treatment plan data used in this study, shows machines can employ either $N_L = 60$ or $N_L = 80$ leaves in each of two leaf banks. (See Fig. 1, right.)

Referencing Table 2, the data vectors A_Leaf_Set and B_Leaf_Set describe the exact position of each leaf in leaf banks A and B, respectively, relative to an x-y grid centered on the MLC. This is illustrated in Fig. 1 (right), where the faint coordinate axes overlay on the MLC, with one leaf bank on the bottom, and the other, the top. In units of centi-Gray, the Dose data refer to the strength of the beam radiating through the aperture formed by the leaves. The Site_Name is the documented intended treatment site, either head-neck, breast, or prostate in our data set. Some of the many more parameters comprising an individual treatment plan that are not used in this study include the width of each leaf, the MLC angle, and the gantry angle. A single treatment plan is essentially a sequential listing of all these parameters to describe the dynamic process as treatment is delivered over time.

During treatment, the LINAC's gantry rotates around the patient, temporarily pausing its rotation at points in space to deliver a dose of radiation. We define a *control point* as the state of the LINAC's parameters (A_Leaf_Set, B_Leaf_Set, and Dose in Table 2) each time it pauses to deliver radiation. The number of control points N comprising a treatment plan varies by plan, from several to hundreds. The subsequent method described next utilizes the dose and leaf positions at each control point to construct a radiation heat map, building upon intuition that suggests each of the three cancer treatment sites considered in this study has a unique shape.

2.2. Data structure

Our method is to first compute a heat map for each plan by utilizing individual plan parameters, namely MLC leaf positions and planning doses. Then, heat maps are adjusted to unify their sizes across all plans, a process that involves up- or down-sampling some heat maps, depending on characteristics of the LINAC used in the plan. Finally, the algorithm flattens each heat map to a high-dimensional vector and then, using a popular dimension-reduction technique, reduces the dimension to one that is much lower in order to train an ML classifier. Each of these three steps is now discussed in detail.

2.2.1. Heat maps—At each control point, the collective leaf positions describe an aperture through which the radiation beam passes. One such aperture is shown in Fig. 3 as white, unshaded. The dose can be considered as only passing through the aperture, and the N apertures in a single plan can be laid over one another in order to accumulate the dose passing through the apertures over the entire plan. Doing so creates a kind of heat map for the plan that is N_L by N_P in size, where N_P is the number of positions each leaf can take.

Formally, let (i, j) represent a position on a coordinate grid, where $i = 1,\ldots, N_L$ is a leaf index, and $j = 1,\ldots, N_P$ is a positional index; the extended edge of any leaf in either bank can be described by such a position. For example, the position (4, 100) in bank A implies the 4th leaf in bank A extends to the 100th positional index, which corresponds to a unique length in tenths of millimeters. At a single control point k, the set of all leaves and their positions forms an outline, and the corresponding kth aperture A_k is defined as the set of all (i, j) falling inside of this outline, see Fig. 3. Now suppose d_k represents the dosage, in units of centigray, at the kth control point, and $I_k(i, j)$ is an indicator function such that

$$I_k(i,j) = \begin{cases} 1, & (i,j) \in A_k \\ 0, & \text{else.} \end{cases}$$

Then a plan's heat map is computed as

$$H(i,j) = \sum_{k=1}^{N} d_k \, I_k(i,j) \,. \tag{1}$$

The authors wish to note that such a simple calculation describes the dose distribution *as it leaves the machine*, not as it penetrates the patient.

Several typical heat maps generated by the application of Eq. (1) are shown in Fig. 4, two examples of each of the three treatment sites. As can be seen in the figure, heat maps for each treatment site share some interesting characteristics. First, heat maps for breast treatment have a larger active area than the other two treatment sites. Heat maps of breast plans have a very distinctive "blocky" (rectangular) shape, and they do not have the gradient shading that the other two treatment sites have. The distinction between maps of prostate plans and those of head-neck plans is more subtle—it seems they differ in shape, and the head-neck heat maps may be hotter. Consequently, it can be expected that any reasonable ML algorithm accepting heat maps as input should easily be able to classify breast plans, possibly confusing some prostate plans with head-neck plans, and vice-versa.

2.2.2. Up/down-sampling—Different LINACs are used in different hospitals, and their respective MLCs can differ in their number of leaves N_L. In each leaf bank, some machines have $N_L = 60$ leaves, others $N_L = 80$ leaves. Recall that the size of all heat maps is N_L by N_P. Therefore, heat maps are either 60×4000 or 80×4000 in size. In other words, some heat maps are comprised of 240,000 pixels, others 320,000. This nonuniformity in size presents a problem for any ML algorithm, because they all aim to find the relationship between each pixel (and pixel combinations) and the treatment site. An algorithm will not

learn a relationship for one number of pixels and another relationship for a different number of pixels.

In order to accommodate different number of leaves in different hospitals and various LINACs, heat maps may be up- or down-sampled so their sizes are uniform across all heat maps. An advantage to down-sampling is that the resultant heat map will have fewer pixels, which is preferred for numerical computation. The authors adopt the down-sampling approach in this paper.

Heat maps that are 80 × 4000 in size may be down-sampled by a factor of $M = \frac{60}{80} = \frac{3}{4}$ along the leaf index axis, so that its length is reduced from 80 to 60, and, consequently, the number of pixels from 320,000 to 240,000. As in Eq. (1), let $H_{80}(i, j)$ denote a heat map of size 80 × 4000, where the subscript 80 is meant to explicitly show the number of leaves. While there are many different down-sampling methods, we apply perhaps the simplest:

$$\widehat{H}(i, j) = H_{80}(\lfloor i \,/\, M \rfloor, j),\ i = 1, \ldots, 60, \tag{2}$$

where $\widehat{H}(i, j)$ denotes the down-sampled heat map of size 60 × 4000, and ⌊·⌋ denotes the floor function. This straightforward scheme is easily computed and samples from the entire range of the original leaf index axis.

In general, down-sampling as in Eq. (2) can produce aliasing and necessitate the use of an anti-aliasing filter. In the current study, the authors found the effect of aliasing to be so small that additional filters make only minimal, almost unnoticeable improvements. For comparison, Fig. 5 shows one original 80-leaf heat map of each treatment site and the corresponding down-sampled version. It is clear from the figure that the down-sampled version is nearly identical to its original. Preserved are the important characteristics such as the rectangular shape of the breast plan and the gradient of the prostate and head-neck plans, thus any aliasing from the down-sampling process is insignificant. Filters are not used in the algorithm, with the goal of keeping the algorithm as simple as possible without sacrificing performance.

2.2.3. Dimension reduction—After down-sampling, the size of all heat maps is m = 240,000 pixels. In ML applications, this number is referred to as the *dimension* of the data set and, for the current study, will still present a problem to ML algorithms, because it is too large as compared to the number of plans, or examples, in the data set (n = 697). Computationally, such a situation when $n < m$ leads to ill-conditioned optimizations or undefined matrix inverses. Even if the problem was well-conditioned, training an ML model with such a large dimension m could be computationally expensive and practically infeasible. A common issue in ML applications, especially in computer vision problems, the solution is almost always to preprocess the data with a dimension-reduction technique, for example, principal component analysis (PCA) (Hastie et al., 2008, p. 534). With PCA, the idea is to represent the data in an optimal subspace of dimension $p < m$, circumventing the issues of ill-conditioned problems and computational infeasibility.

In order to apply PCA, we first flatten all n heat maps – which are each essentially a matrix of 60 × 40,000 numbers – to vectors of size $1 \times m$, and concatenate them all in a matrix $X \in \mathbb{R}^{n \times m}$. One column of X corresponds to one pixel – the same pixel – in every heat map. In accordance with PCA, X is then linearly transformed to a new matrix $T \in \mathbb{R}^{n \times m}$, but the columns of T no longer correspond to pixels. Instead, its columns, called "principle components" (PCs), are linear combinations of the columns of X such that the first column of T captures the axis of maximum variance in the data, the second column the second-most variance, et seq. Since the columns are sorted according to variance, it is possible to truncate T, only retaining the first p columns, while still retaining much of the data variance. With no theoretically optimal way of selecting the new dimension p, one has to rely on heuristics or cross-validation to do so.

One such heuristical approach is to make a plot, as in Fig. 6, of cumulative fractional variance versus the number of retained PCs. More formally, if λ_l denotes the variance explained by the lth PC, then an expression for the fractional variance explained by this PC is $\lambda_l / \sum_{r=1}^{m} \lambda_r$. Thus the cumulative fractional variance of retaining the first p PCs is given by

$$\text{retained var}_p = \sum_{l=1}^{p} \left(\frac{\lambda_l}{\sum_{r=1}^{m} \lambda_r} \right). \tag{3}$$

With more than 0.99 fractional variance after just 185 PCs, the bar chart in Fig. 6 demonstrates the retained variance in Eq. (3) quickly approaches 1. This quality is highly desirable since it implies the remaining 240,000 – 185 = 239,185 PCs account in total for less than 1% of the data variance, and the dimension can be immensely reduced while still retaining much of the information. For selecting a specific p, the approach is most often to select the number of PCs just after the "elbow" of the curve in Fig. 6. See, for example, (Tabachinick & Fidell, 2001, p. 476). With this in mind, the authors chose to reduce the dimension to $p = 50$ for this study, still retaining about 95% of the data variance. Retaining more than 50 dimensions can increase the chance that the algorithm will overfit the data and drive down the accuracy of the classifier (Hastie et al., 2008, p. 536). With the size of the data reduced to 697 × 50 from 697 × 240,000, ML algorithms may be trained without numerical or computation issues. Each plan is effectively compressed to a 50-element vector, and these 50 numbers form the plan's features.

2.3. Classifiers

The preceding method of extracting salient treatment plan information is evaluated by observing the accuracy of well-established ML algorithms in classifying the correct treatment site. We consider random forest, softmax, k-nearest neighbors, artificial neural network (ANN), and support vector machine (SVM) as the predictive model. For brevity, we only broadly describe the theory of the classifiers and their associated tuning parameters —just enough to understand how and why specific choices of parameters yield the observed results. Rather than rehash the mathematical details, references are provided for the reader

to investigate these details, if desired. e particularly enjoy (Kuhn & Johnson, 2018) which details each of the algorithms considered in this study.

2.3.1. Random forest—One of the most powerful ML algorithms, the random forest classifier is an ensemble of many decision trees (Geron, 2018, p. 181). A decision tree, on its own, is a classifier that is trained by repeatedly splitting the training data in an optimal fashion, maximizing information entropy each time the data is split on a feature. Each feature is used exactly once to split the data, and each path of the tree terminates with a leaf node, which contains the class. The class predicted by each tree in a random forest is a vote in a majority-wins prediction of the forest. During training, each tree is only allowed a random subset of features to consider, which encourages a diverse collection of trees, in turn reducing overfitting (a problem for the simple decision tree classifier) and improving accuracy (Geron, 2018, p. 181). The number of trees comprising the forest is a hyperparameter which needs to be chosen. Usually, and at the cost of increasing computational effort, the classifier's accuracy improves as the number of trees increases, but with marginal gains that level off asymptotically.

2.3.2. Softmax—Softmax regression estimates the probability of an example belonging to each class and selects the class with the highest probability (Geron, 2018, p. 139). The probability of each class is learned in an optimal way by training a linear model, the output of which is then logit-transformed to a range of numbers between 0 and 1, such that the three class probabilities sum to 1. Often it is the case that the softmax classifier overfits the training data, so an $l-2$ regularization term may be added to the linear function in order to reduce error and improve performance (Hastie et al., 2008, p. 119). Too much regularization, however, will underfit the training data and hinder the performance at test time. Thus, choosing an optimal amount of regularization is key for good performance.

2.3.3. K-nearest neighbors—The k-nearest neighbors algorithm (Hastie et al., 2008, p. 14) is the third classifier considered in this study and perhaps the simplest. In this algorithm, hyperparameter k is first chosen, and a test data point is classified by assigning the class which is most frequent among the k training data points nearest to that test point, using Euclidean distance as the distance metric. The choice of k will affect the classifier's accuracy, where a value too small or too large will either overfit or underfit the training data, respectively (Hastie et al., 2008, p.15).

2.3.4. Artificial neural network—ANNs, because of their versatility and scalability, are currently at the very core of large and complex ML tasks (Geron, 2018, p. 239). However, as more of a "black-box" algorithm, ANN models can be difficult to interpret, and their architecture and hyperparameters difficult to tune (Geron, 2018, p. 239). Further, with many trainable parameters comprising their architecture, they may not perform well in smaller applications with small data sets. In these situations, traditional ML algorithms may be preferred over ANNs, especially when a traditional algorithm is already known to perform well. With only $n = 697$ treatment plans, deep (i.e., large, complex) ANN architectures are not an option for this study, though we still investigate the performance of some shallow ANNs with varying architectures.

2.3.5. Support vector machine—The last classifier considered, the SVM, is a well-established classifier that finds a separating boundary in feature space in order to maximize the width of the gap between two classes (El-Naqa et al., 2002). Though a binary classifier, the SVM can be applied to our three-class data set by twice applying a one-versus-all strategy (Geron, 2018, p. 475). When classes are overlapping in feature space and not linearly separable, the SVM formulation is extended by introducing slack variables into the margin optimization and controlling the amount of slack using a hyperparameter called the *box constraint*. When the box constraint is small, a large margin is chosen at the expense of a greater number of misclassifications in the training set. A large box constraint, on the other hand, encourages accurate classification in the training set, but with a smaller margin.

3. Classification results

Employing the sklearn library on Python 3.8, we test each classifier with several different tunings/achitectures, and in its own five-fold cross-validation loop. Accordingly, the data set was split into five equally sized partitions, four to train the algorithm (80% of the data), and one to test its performance (20% of the data). As opposed to training-only accuracy which can be overly optimistic, testing the algorithm on data that was withheld during training time provides a more accurate estimate of the algorithm's performance in real-world situations (Hastie et al., 2008, p. 129). With cross-validation, this training-testing process is repeated five times so that each of the five partitions serves as the test set exactly once, and, correspondingly, five accuracies are observed.

The algorithm was tested on a Microsoft Surface Book running Windows 10 with Intel Core i7-6600 CPU @ 2.60 GHz and 16 GB of RAM. With this modest computer, the 697 heat maps were generated in about 11.4 s of time, or 16.4 ms per map. Training and testing were also very quick, and similar for each algorithm tested; on average, training required about 7.51 s of time, and testing, 321 ms.

Grid search was employed to determine the optimal tuning of each tested algorithm. Through trial and error, we considered a set (grid) of reasonable hyperparameters from which to tune each model. The cross-validation process was carried out for each tuning, and the mean accuracy and variance across the five folds were recorded. In this way, a single, unusual test partition with several outliers will not sabotage the observed accuracy. Reporting the performance versus each tuning enables us to identify an optimal tuning and any trends that may emerge.

There are several implementation details to describe at this point:

- The test data are withheld when training the ML classifier. In order to truly withhold the test data from the whole algorithm, the test data are also withheld from PCA when finding the optimal subspace.
- Some pixels have the same value (0, black) across all heat maps in a training set. These pixels have zero variance and present a computational problem for PCA. Therefore, these pixels are identified and removed from the training set prior to the PCA algorithm. They are also removed from the test set. Depending on the

training set, this process slightly reduces the dimension of the heat maps to about 211,800 from 240,000.

- Computationally, PCA requires the normalization of all pixels so they have zero mean across all heat maps in the training set (and optionally unit variance). Both the mean and variance of the training set are calculated and utilized to normalize the training set *and also the test set*; the latter is to ensure the test set is transformed with exactly the same parameters as the training set.

These implementation details are best summarized as a block diagram, see Fig. 7.

3.1. Random forest

Table 3 shows the results of the random forest classifier on the treatment plan data set, considering 5, 10, 20, 50, and 100 number of trees. Achieving more than 94% accuracy, the random forest performs very well—actually, the best of all those tested. As expected, the classification accuracy, which is averaged across the five cross-validation folds, increases with the number of trees, and with marginal increase beginning around 50 trees. The variance of the accuracy across the five folds is small and does not seem to be affected by the number of trees.

3.2. Softmax

The results of the softmax classifier on the treatment plan data are shown in Table 4 for $l-2$ regularization weights of 0.01, 0.1, 1, 10, and 100. From the table, it is observed that the optimal regularization is 0.1—more or less regularization results in decreased accuracy. At just over 89% averaged over five cross-validation folds, the classifier's optimal performance falls short of the random forest. The variance, like with the random forest, is small and the performance seems unaffected by the choice in hyperparameter.

3.3. K-nearest neighbors

For a finite number of data points as in the current study, no theoretical results exist to choose the value of k, so several values of k are considered, each one tested in a cross-validation loop. For example, Table 5 shows the performance of the k-nearest neighbors algorithm with $k = 1, 3, 5, 7, 11$. The best accuracy of 86.17% occurs with $k = 3$, though underwhelming compared to the accuracy of the random forest.

3.4. Artificial neural network

ANN architectures may vary in the number of hidden layers, and the number of non-linear activation units (neurons) in each layer. To keep the architecture simple, we choose to stack only a few hidden layers, each with its own rectified linear unit (ReLU) activation. The stack of hidden layers precedes the output layer consisting of a softmax activation function with three neurons in order to yield three probabilities—one for each class. Fig. 8 illustrates the architecture with three hidden layers. Each input example, after it is reduced with PCA, feeds the network and is assigned the class of highest probability. Limiting the analysis to small architectures, we choose combinations of 1, 2, 3 hidden layers and 50,100,500 neurons in each layer, and the results are shown in Table 6. According to the table, ANN does not

perform well, with a maximum accuracy of 74.29% occurring with an architecture of three hidden layers, 500 neurons in each layer.

3.5. Support vector machine

Though the SVM algorithm can be applied to find complex, non-linear decision boundaries in feature space using the so-called *kernel trick* (Hastie et al., 2008, p. 423), only linear decision boundaries are considered, but with several choices of the hyperparameter. Table 7 shows the SVM does not perform well, with a maximum accuracy of 62.40% occurring with a box constraint of 100.

3.6. Confusion matrix

Recall it is expected that breast cancer plans are easier to classify since breast heat maps appear distinctively different from head-neck and prostate heat maps (Fig. 4). And with the data set being imbalanced and favoring breast plans (Table 1), observing only the classification accuracy may be misleadingly optimistic. More informative of the algorithm's performance is a tally of predictions by class, like in a *confusion matrix*. More exactly, a confusion matrix is a table in which each row represents the true instances in each class, and each column the predicted instances. The entry located at row i, column j of the confusion matrix is the number of times the ith class was predicted as the jth class. Obviously, a diagonal matrix is ideal.

Of all the classifiers, the random forest is the most accurate, so only its confusion is investigated. Placed in a five-fold cross-validation loop, the classifier generates five confusion matrices, one for each test set partition, and each matrix is aggregated into a single matrix in order to concisely tally the confusion. The resulting confusion matrix is shown in Table 8. Correctly classified 301 times out of 305 (98.7%), breast plans were indeed the easiest plans for the algorithm to identify, as expected. Head-neck plans were correctly classified 170 times out of 186, or 91.4%. Prostate plans were correctly classified 188 times out of 206, or 91.3%. When a head-neck plan was misclassified (16 times), it was more often than not confused as a prostate plan (15 times) rather than a breast plan (1 time). Similarly, when a prostate plan was misclassified (18 times), it was more often than not confused as a head-neck plan (16 times) rather than a breast plan (2 times).

Like the confusion matrix, F-1 scores are preferable when classes are imbalanced. An F-1 score is the harmonic mean of precision and recall and is easily calculated from the confusion matrix. The scores are shown in Table 9 for each class, along with precision and recall. The average F-1 score, weighted by class population, is 0.945.

4. Conclusion

This paper presents a novel algorithm to automatically determine if a given cancer treatment plan is consistent with its intended use. Based on dose and collimator aperture information, the algorithm first calculates a radiation heat map for the plan, which contains enough information, the paper shows, to preserve the unique characteristics of head-neck, breast, and prostate treatment sites. Manual visual inspection of the heat map further supports this point, as the shape and shading of the heat map are typically excellent indicators

of the treatment site. Once the plan's heat map is calculated, the algorithm may up- or down-sample the map to standardize its size, as machines delivering treatment can vary in the number of leaves comprising the machine's mechanical leaf banks. The heat map is then reduced, using PCA, to a 50-dimensional vector, which is fed to a random forest model to classify the treatment site. A warning can be raised if the classified treatment site and the documented intended treatment site do not agree, indicating an additional manual review of the plan by a medical professional may be warranted.

Testing the algorithm in a cross-validation loop demonstrates its accuracy to be about 94.6%, provided the random forest contains a sufficient number of trees. The small variance in accuracy among the five cross-validation folds indicates the robustness of the algorithm. The accuracy, however, which is shown to be sensitive to the number of trees comprising the forest, steadily declines as the number of trees decreases from 50. Still, the random forest, with a sufficient number of trees, outperforms the many other classifiers considered in this study which do not perform nearly as well. The algorithm's confusion matrix demonstrates that breast plans are the easiest plans for the algorithm to correctly identify; whereas head-neck plans, when they are misclassified, are often confused with prostate plans, and vice-versa.

In practice, the parameters describing the optimal PCA subspace can be calculated offline – prior to receiving the query treatment plan – from the analysis of hundreds of historical treatment plans from various hospitals and clinics. Similarly, the random forest model can also be trained offline with the same historical data. With all training done offline, classifying the query treatment plan is computationally very quick once the model parameters are loaded: Simply calculate the plan's heat map, perform up- or down-sampling if necessary, project the map into the subspace, and ask the random forest for its predicted class. It is reasonable to expect the algorithm to be correct 94.6% of the time, only misclassifying a plan 5.4% of the time, in accordance with the cross-validation results.

This study uses a radiation heat map as the features to feed a random forest classifier, but potentially thousands of other features could be engineered from the raw treatment plan data. By manually inspecting the plans that were misclassified during testing, it may be possible to learn which characteristics are the most salient, and how features could be engineered to improve classification accuracy. It would be interesting to continue to explore other features as inputs to the classifier – either instead of or in addition to the heat map – and evaluate the performance, using 94.6% accuracy as the benchmark for comparison. Of additional interest is the algorithm's performance in classifying cancer treatment sites not considered in the study, for example, in the lungs. Presuming a sufficient amount of data, the algorithm may naturally be extended to classify additional treatment sites simply by incorporating the data from these new treatment sites into the training of the algorithm.

Acknowledgments

Research reported in this publication was supported by National Cancer Institute of the National Institutes of Health, USA under award number R42CA195819.

Data availability

The authors do not have permission to share data.

References

ASTRO (2022). Aggregate data report. https://www.astro.org/ASTRO/media/ASTRO/Patient%20Care%20and%20Research/PDFs/ROILS_2022_Q2.pdf.

Azmandian F, Kaeli D, Dy G, Hutchinson E, Ancukiewicz M, Niemierko A, & Jiang SB (2007). Towards the development of an error checker for radiotherapy treatment plans: a preliminary study. Physics in Medicine and Biology, 52, 6511–6524. [PubMed: 17951859]

Bai E, & Xia J (2021). A knowledge based automatic radiation treatment plan alert system. International Journal of Artificial Intelligence & Applications, 12.

Bogdanich W (2010). The radiation boom—radiation offers new cures, and ways to do harm. The New York Times, https://www.nytimes.com/2010/01/24/health/24radiation.html.

Boyer A, Biggs P, Galvin J, Klein E, LoSasso T, Low D, Mah K, & Yu C (2001). Basic applications of multileaf collimators. Medical Physics Publishing., https://www.aapm.org/pubs/reports/detail.asp?docid=71.

El-Naqa I, Yang Y, Wernick MN, Galatsanos NP, & Nishikawa RM (2002). A support vector machine approach for detection of microcalcifications. IEEE Trans Medical Imaging, 1552–1563. [PubMed: 12588039]

French J (2002). Treatment errors in radiation therapy. Radiation Therapy, 11(2), 149–159.

Ganesh T (2014). Incident reporting and learning in radiation oncology: Need of the hour. Journal of Medical Physics, 39, 203–205. [PubMed: 25525306]

Geron A (2018). Hands-on machine learning with scikit-learn & tensorflow (2nd ed.). O'Reilly.

Hastie T, Tibshirani R, & Friedman J (2008). The elements of statistical learning (2nd ed.). Springer.

Hoopes DJ, Dicker AP, Eads NL, Ezzell GA, Fraass BA, Kwiatkowski TM, Lash K, Patton GA, Piotrowski T, Tomlinson C, & Ford EC (2015). RO-ILS: Radiation oncology incident learning system: A report from the first year of experience. Practical Radiation Oncology, 5(5), 312–318. 10.1016/j.prro.2015.06.009. [PubMed: 26362705]

Kalet AM, Gennari JH, Ford EC, & Phillips MH (2015). Bayesian network models for error detection in radiotherapy plans. Physics in Medicine and Biology, 60, 2745–2749.

Kisling K, Cardenas C, Anderson BM, Zhang L, Jhingran A, Simonds H, Balter P, Howell M, Schmeler K, Beadle BM, & Court L (2020). Automatic verification of beam apertures for cervical cancer radiation therapy. Practical Radiation Oncology, 10, e415–e424. [PubMed: 32450365]

Kuhn M, & Johnson K (2018). Applied predictive modeling (2nd ed.). Springer.

NCI (2021). Radiation fact sheet. https://www.cancer.gov/aboutcancer/468treatment/types/radiation-therapy/radiation-fact-sheet.

Rosenthal J, & Booth M (2005). Maximizing the use of state adverse event data to improve patient safety. https://eadn-wc03-6094147.nxedge.io/cdn/wp-content/uploads/2009/04/use_of_adverse_data.pdf.

Tabachinick BG, & Fidell LS (2001). Using multivariate statistics (7th ed.). Pearson.

Fig. 1.
(Left) Varian TrueBeam LINAC and (right) its MLC comprised of tungsten leaves to shape the radiation beam delivered by the LINAC.

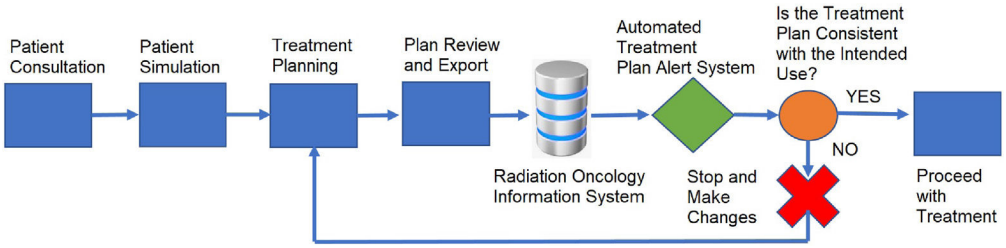

Fig. 2.

Flowchart of the radiation therapy process with an added automated treatment plan alert system.

Fig. 3.
An example of an aperture (white, unshaded) formed by collimator leaves.

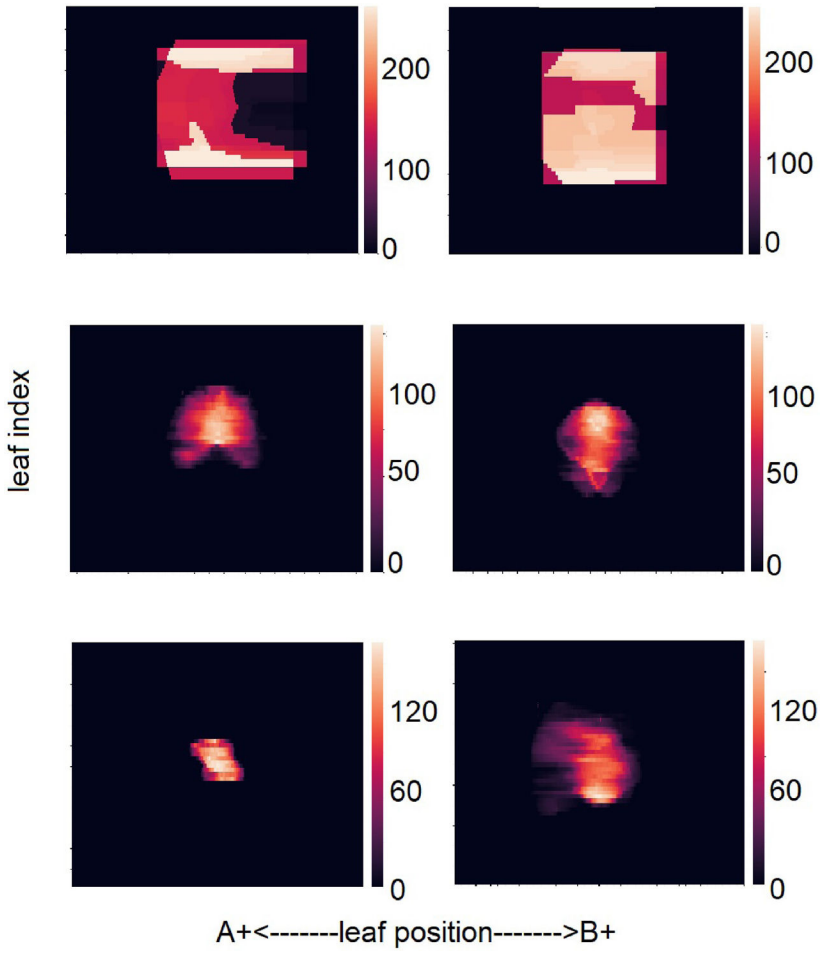

Fig. 4.

Examples of typical heat maps from six radiation treatment plans in the data set. (Top row:) breast, (middle row:) prostate, (bottom row:) head-neck. Radiation is expressed in units of centigray.

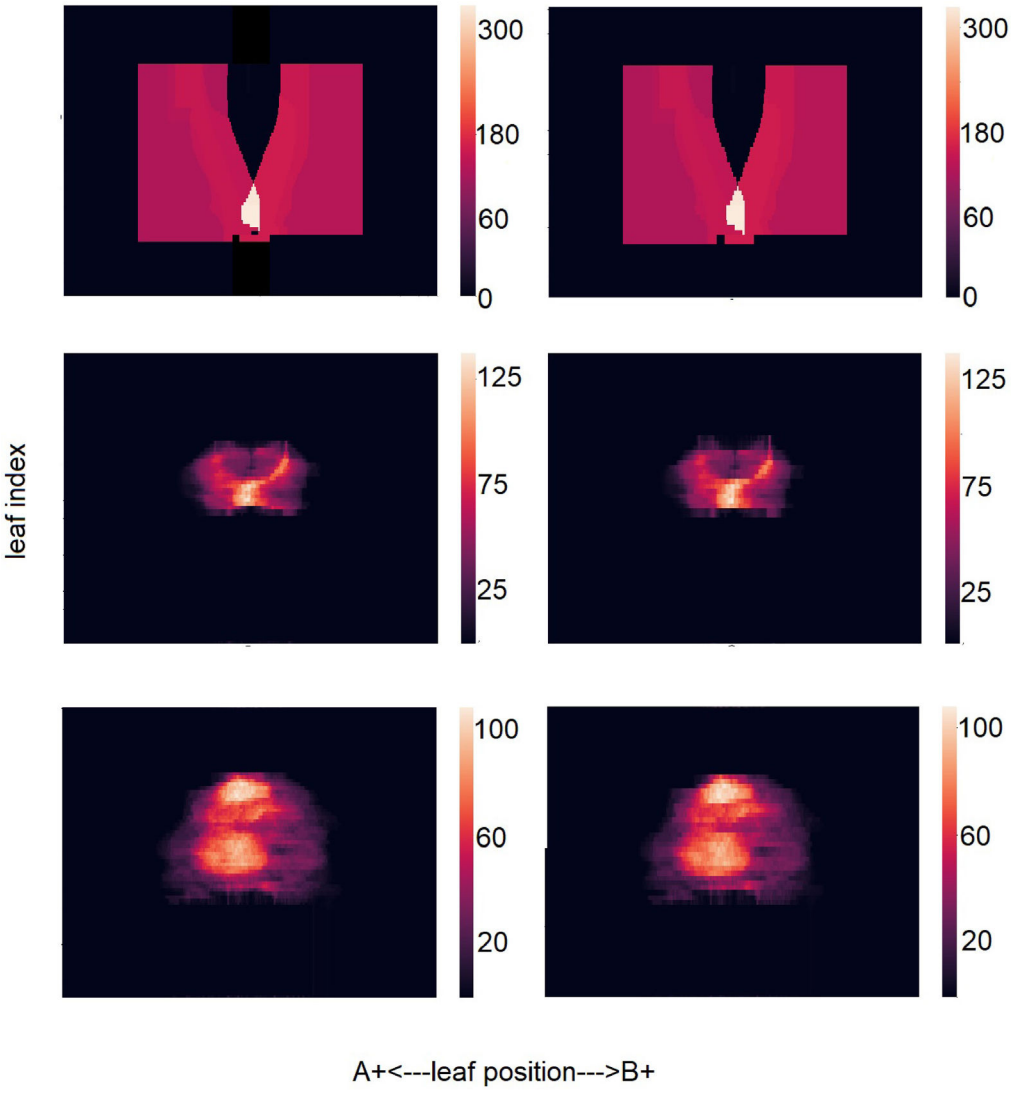

Fig. 5.

Down-sampling of heat maps. On the left is the original heat map of height 80, and on the right is the down-sampled version of height 60. (Top row:) breast, (middle row:) prostate, (bottom row:) head-neck.

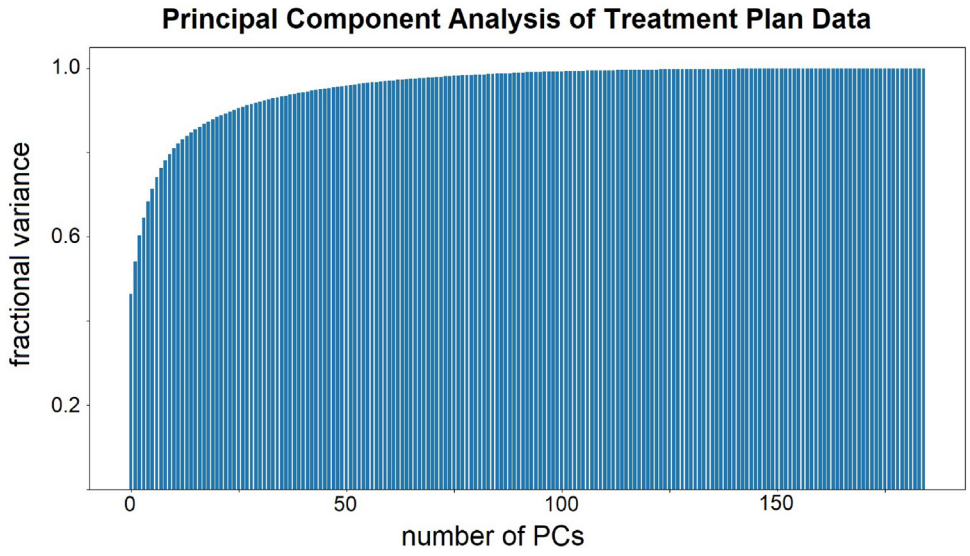

Fig. 6.

Principal component analysis of the treatment plan data set showing the cumulative fractional variance versus increasing number of retained principal components.

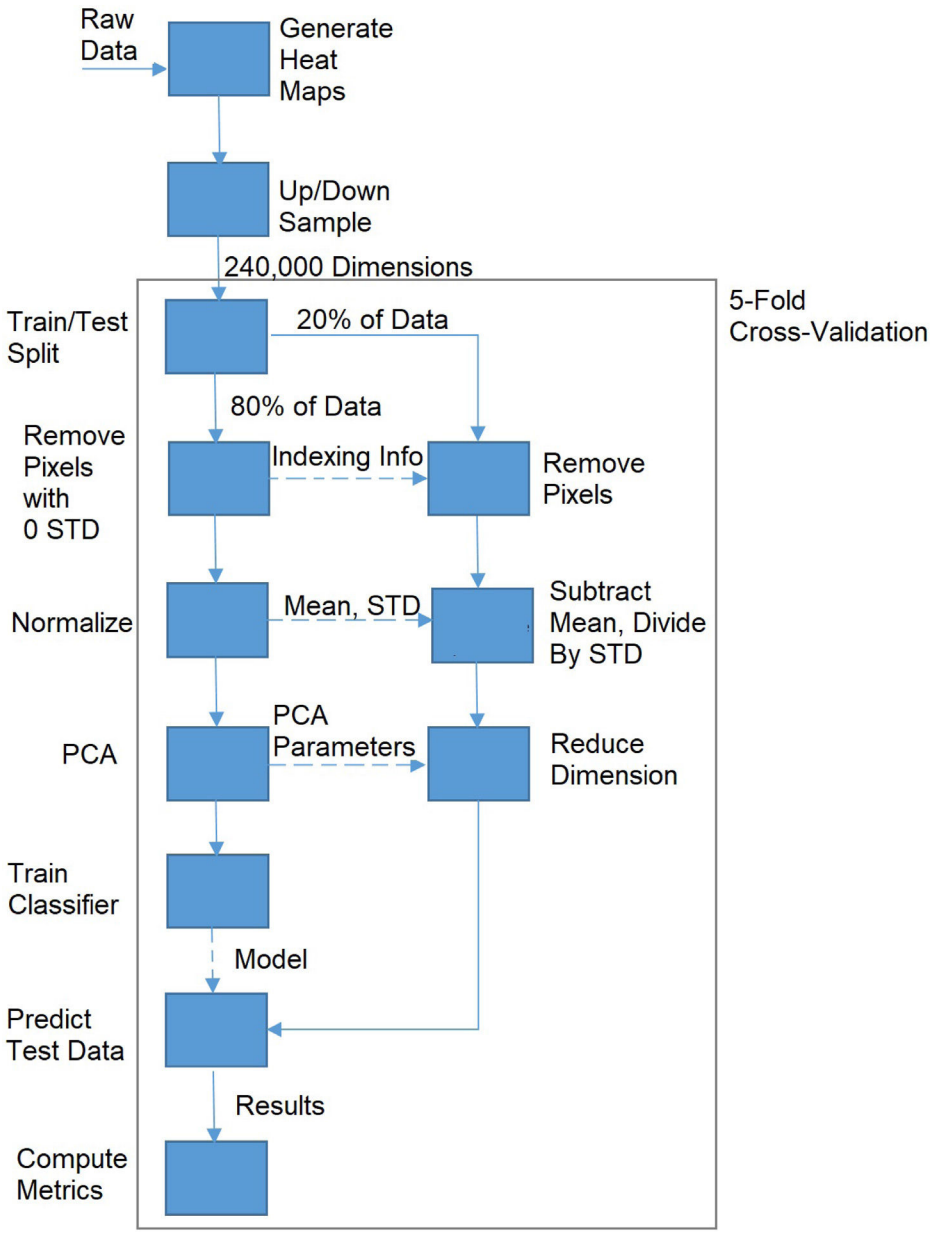

Fig. 7.

A block diagram of the entire algorithm, including cross-validation to measure accuracy. Solid arrows indicate the movement of data, while dashed arrows indicate the passing of parameters.

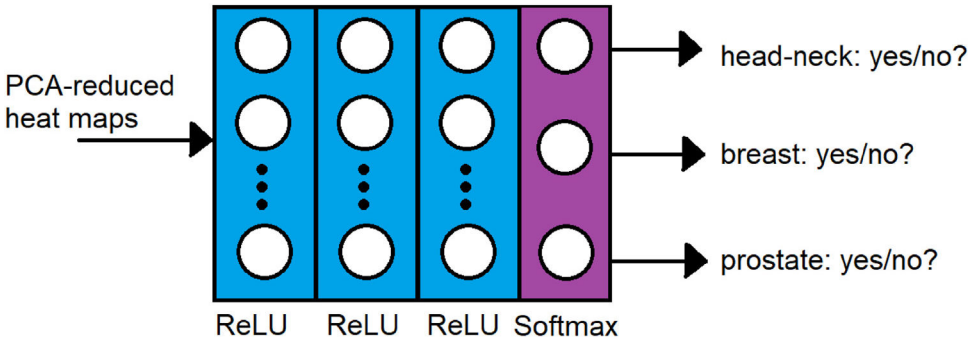

Fig. 8.

Architecture of the shallow networks considered in the study, shown with three hidden layers each with rectified linear unit (ReLU) activation. The number of neurons in each layer and the number of layers were hyperparameters that were varied and tested.

Table 1

Number of plans of each treatment site in the data set.

Head-neck	Breast	Prostate	Total
186	305	206	697

Table 2

Treatment plan data retrieved in raw form from oncology databases. Only shown are data used in this study.

Parameter identifier	Description	Units	Type
A_Leaf_Set	A 60– or 80–length vector of leaf positions in Bank A	Tenths of mm	Vector of ints
B_Leaf_Set	A 60– or 80–length vector of leaf positions in Bank B	Tenths of mm	Vector of ints
Dose	Amount of radiation delivered from the LINAC	centi-Gray	Float
Site_Name	Treatment site	N/A	String

Table 3

Classification accuracy of random forest classifier with varying number of trees (n_trees).

n_trees	**5**	**10**	**20**	**50**	**100**
Mean	87.02%	92.94%	93.68%	94.01%	94.55%
Variance ($\times 10^{-3}$)	3.17	1.60	2.27	1.33	2.73

Table 4

Classification accuracy of softmax classifier with varying amount of regularization.

l-2 weight	**0.01**	**0.1**	**1**	**10**	**100**
Mean	87.02%	89.56%	80.65%	74.71%	76.84%
Variance ($\times 10^{-3}$)	1.83	2.57	1.83	2.40	2.57

Table 5

Classification accuracy of *k*-nearest-neighbors classifier with different values of *k*.

k	1	3	5	7	11
Mean	78.53%	86.17%	78.96%	70.47%	74.71%
Variance ($\times 10^{-3}$)	3.17	1.27	3.57	2.23	1.40

Table 6

Classification accuracy of artificial neural network classifier with different architectures.

Hidden layers, neurons in each layer	(1,100)	(2,50)	(2,100)	(3,100)	(3,500)
Mean	55.61%	66.64%	73.86%	73.43%	74.29%
Variance ($\times 10^{-3}$)	6.90	2.04	3.07	3.27	2.83

Table 7

Classification accuracy of support vector machine classifier with varying box constraint.

Box constraint	**0.01**	**0.1**	**1**	**10**	**100**
Mean	57.73%	60.70%	59.43%	61.55%	62.40%
Variance ($\times 10^{-3}$)	3.73	8.60	3.17	3.17	0.93

Table 8

Random forest confusion matrix.

	Head-neck	Breast	Prostate	Total
Head-neck	170	1	15	186
Breast	2	301	2	305
Prostate	16	2	188	206

Table 9

F-1 scores for the random forest classifier.

	Precision	Recall	F-1
Head-neck	170/188 = 0.904	170/186 = 0.914	0.909
Breast	301/304 = 0.990	301/305 = 0.987	0.988
Prostate	188/205 = 0.917	188/206 = 0.913	0.915