Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 401–410
DOI: 10.15439/2023F378
ISSN 2300-5963 ACSIS, Vol. 35

Mutual Learning Algorithm for Kidney Cyst, Kidney Tumor, and Kidney Stone Diagnosis

Sabrina Tarin Chowdhury
ORCID ID: 0009-0005-5854-8161
Indiana University-Purdue University
Computer and Information Science
IN 46202, Indianapolis, USA
Email : sabchow@iupui.edu

Snehasis Mukhopadhyay
ORCID ID : 0009-0000-0836-2901
Indiana University-Purdue University
Computer and Information Science
IN 46202, Indianapolis, USA
Email : smukhopa@iupui.edu

Kumpati S. Narendra
Yale University
Center for Systems Science
CT 06520, New Haven, USA
Email : kumpati.narendra@yale.edu

***Abstract*— Mutual learning is a machine learning algorithm where multiple machine learning algorithms share knowledge among themselves to improve themselves. The utilization of mutual learning algorithms can effectively enhance the efficiency of machine learning and neural networks within a multi-agent system. This approach is particularly useful in scenarios where the system cannot be adequately trained with a large dataset. By exchanging data in a dynamic teacher-student network system, mutual learning can result in efficient learning outcomes. Typically, a large network serves as a static teacher and transfers data to smaller networks, referred to as student networks, to improve their efficiency. In this study, we aim to demonstrate that two small networks can dynamically alternate between the roles of teacher and student to share knowledge, resulting in improved efficiency for both networks. To exemplify this concept, we apply a mutual learning algorithm using convolutional neural networks (CNNs) and Support Vector Machine (SVM) to accurately identify the kidney diseases – cyst, tumor and stone using image classification algorithm.**

***Index Terms*—Mutual learning, teacher-student network, CNN, model distillation, Kidney Disease, Cyst, Tumor, Stone**

I. Introduction

MACHINE learning has a great potential to revolutionize the medical science. It can be a big aid to the current medical system specially in disease diagnosis in early stage. Moreover, in many third world countries, there is extreme shortage of doctors and hence, the doctors do not have the ample time and energy to invest behind a patient. In those cases, machine learning algorithm can provide work as a 'second brain' for the doctor to aid him in disease diagnosis. Even in the first world countries the machine learning algorithms can provide a third eye to the doctors. Couple of recent studies [1][2] shows that an estimated 5% of the outpatients get wrong diagnosis in US every year. Particularly when a patient is in serious medical condition, the misdiagnosis is common. A study shows that almost 20% of the serious patients are misdiagnosed at the level of primary care [3]. Misdiagnosis can result in serious harm of the patient and almost one-third of the misdiagnosed patient face harmful consequences [4].

Nevertheless, use of machine learning in medical diagnosis is still limited due to several facts. Experiments proved that for attaining considerable accuracy level, machine learning training dataset requires abundant amount of patient data [5][6][7][8][9][10]. However, the machine learning diagnostic algorithms could not reach the accuracy of the doctors in differential diagnosis [11][12][13] yet specially where there can be multiple possible causes of a patient disease symptoms. Lots of research has been done in disease diagnosis but few has shown considerable accuracies (accuracy>90%) because as stated earlier, wrong diagnosis can be potentially dangerous for the patient. For example, machine learning algorithms for heart disease detections show accuracies in the range between 80% to 90% [14][15][16][17] while only one result shows accuracy of 94% [16] using SVM algorithm. Machine learning algorithm for diabetes detection shows accuracies between the 70% to 80% [18][19][20] [21][22] while only one result using Naïve Bayes algorithm shows accuracy of 95%. Lever disease detection algorithms [23][24] [25] shows even poorer accuracies (around 70%) while only couple of results shows accuracies over 96% [25] using Naïve Bayes and functional tree algorithm. Research [26] [27][28] shows poor accuracies for Hepatitis detection also (ranges between 70% to 90%) while only one result [26] shows accuracy of 96% using Naïve Bayes algorithm.

Medical diagnosis AI must have very high accuracy (>97%) on unknown dataset [29][30]. For that, medical diagnosis AI must have training dataset greater than 10000 to build reliable system [29][30]. But building a big and comprehensive dataset in medical sector is not easy because patient data sharing has lots of confidentiality and legal bindings. Moreover, a key step in machine learning is to train the algorithm/network properly to achieve good accuracy. Most often, in order to achieve good enough accuracy, machine learning algorithms have to be trained using fairly large number of training datapoints. It may require large memory to execute and get power and computation resource hungry training algorithms which can be a tremendous problem in many systems. Hence, there is a big demand to find small and fast training mechanisms. Mutual learning [31][32][33] is one of the interesting concepts explored to execute faster and efficient training of machine learning algorithm and share knowledge among the algorithms.

Mutual learning algorithm is a machine learning algorithm where multiple machine learning algorithms learns from different sources and then share their knowledge among themselves (fig 1) so that all the agents can improve their classification and prediction accuracies simultaneously. Mutual learning algorithm can be an efficient mechanism for improving the machine learning and neural network efficiency in a multi-agent system. Most of the model distillation systems use a big network, known as teacher network, to pass its learning to a smaller network to train the later [34][35][36]. Static teacher-student network data passing is one-way which incurs several issues like mimicry loss [37]. Furthermore, the teacher network does not see any improvement in efficiency. On the other hand, in mutual learning, a variation of model distillation, there is no static teacher-student network. Rather, role of teacher and student network can change dynamically based on the training sample, and both networks can train each other (fig 1). Thus, efficiencies and accuracies of both

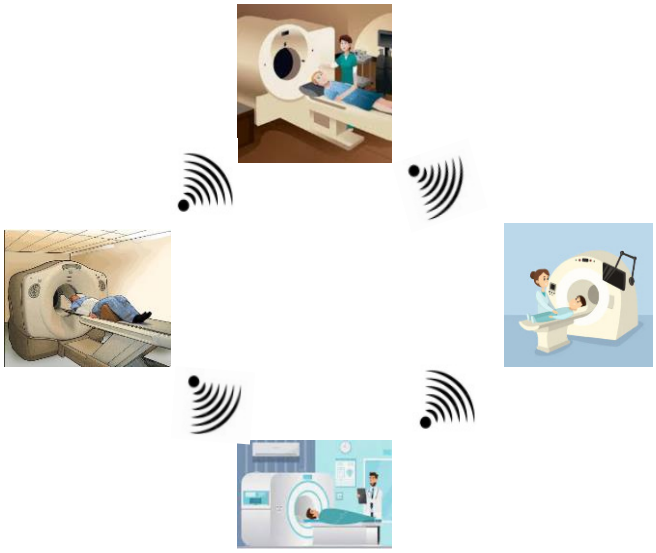


Figure 1: Medical diagnosis machines sharing knowledge.

networks improve. The concept can be particularly useful to increase the efficiency of multiple small networks simultaneously which can be used to do parallel processing. Furthermore, mutual learning can be particularly useful when a big single training dataset is not available. Rather, small, distributed sets of training data are available and some of the training data may need to be relabeled. In this way mutual learning can be very helpful in the field of machine learning for medical diagnosis. Since the machines are sharing data among themselves in non-human readable format, the issue of privacy breaching can also be avoided and gradually over the time the machine will improve their accuracy levels to an acceptable threshold for medical diagnosis.

In this paper, we demonstrate the concept of such mutual learning via the different kidney disease (cyst, tumor and stone in kidneys) detection using scanned kidney images in ref[38] dataset. We used CNN and SVM algorithm to implement the kidney image pattern recognition. We show how the mutual learning can improve the efficiency of both the networks simultaneously and how it can reduce the overall training time significantly. The accuracy keeps improving over the time as more and more training data is shared between machine learning algorithms. We also show that the increasing the number of networks in mutual learning can significantly improve the efficiency of all the networks involved without significantly increasing the training time. The mutual learning is implemented between both homogeneous and heterogeneous agents and comparison between relative accuracy improvements are discussed and analyzed in detail.

II. BACKGROUND ON MUTUAL LEARNING

For centuries, philosophers have delved into the study of learning theory, while psychologists, engineers, and computer scientists have joined this exploration over the past seventy years. As a vast, multidisciplinary field, learning theory has been the subject of investigation using a multitude of methods. Historically, the majority of research has focused on a single agent, typically a learner or student, operating in a deterministic or stochastic environment. However, this report marks a significant departure from traditional learning approaches as it investigates the dynamics of multiple agents learning from each other.

The fundamental inquiry, posed in various iterations, revolves around how two or more agents or entities, operating within the same or similar environment and attempting to solve the same or similar problem, can share information to increase operational efficiency.

Mutual learning problems are prevalent and encompass a wide spectrum, ranging from straightforward deterministic optimization to exceedingly complex ones that are challenging to articulate accurately. The problems addressed in this report span multiple areas, such as deterministic optimization in high-dimensional spaces, stochastic reinforcement learning in static/stationary environments (learning automata), employing both deterministic and stochastic schemes, learning in dynamic environments, such as those defined by Markov Decision Processes, and learning/adaptation by multiple agents in dynamic environments described by deterministic or stochastic difference and differential equations.

Mutual learning can occur between two humans, a human and a machine, or between two machines. Researchers in fields such as social psychology are particularly interested in the former. However, the importance of human-machine interactions has become increasingly evident, especially in the context of interactions between human-driven and fully autonomous vehicles. We anticipate that machine-machine learning will lead to complex yet intriguing problems that will keep investigators occupied for many years. The quantitative approach used in this and future reports will not only facilitate efficient collaboration between machines, but also shed light on the limitations of such collaboration. Specifically, the study aims to address the question of whether two agents, each utilizing schemes that result in optimal behavior in stationary environments, may arrive at an incorrect conclusion when learning from one another.

A. Related Research

In the study conducted by Ikemoto *et al* [39], human-robot mutual learning and co-adaptation were explored, inspired by human parenting behavior. In the context of artificial neural networks, Zhang *et al* [37] examined the problem of a group of deep neural networks learning from each other for a classification task. The researchers concluded that small neural networks with mutual learning could outperform a single powerful teacher network. Nie *et al* [40] investigated mutual learning to achieve superior performance in two related yet distinct computer vision tasks, namely human parsing and pose estimation. Another relevant research theme is multi-agent learning systems, where agents focus on different subtasks of a complex problem and work together to solve it, similar to mathematical game theory. Panait and Luke [41] provide an overview of this well-established field, emphasizing inter-agent communication, task decomposition, and scalability in multi-agent systems. In contrast to multi-agent systems, mutual learning involves agents that collaborate to solve the same or similar tasks and act as (partial) teachers to each other to enhance their learning.

III. KIDNEY DISEASE DIAGNOSIS WITH MACHINE LEARNING : LITERATURE REVIEW

In literature, many machine learning studies has been done on pattern recognition-based kidney diseases detection.

However, most of the work focused on chronic kidney disease detection [42][43][44] since it can be fatal for the patient. Few works [45][46][47] has been done on machine learning based kidney cyst, tumor and stone detection. Ref [45] used CNN to show an accuracy of 99.52% while ref [46] showed 99.30% accuracy using VGG16. Ref [47] showd impressive 99.98% accuracy using DenseNet201. All these works are conducted on a certain dataset and parameters are optimized for maximum accuracy. The principle concern is, how these trained networks would behave for a new unknown set of data. It is unlikely that they will show similar accuracy for unknown dataset. For achieving good universal accuracy, the algorithms are needed to be trained over times by datasets from various sources and types.

IV. THE PATTERN RECOGNITION PROBLEM

For every pattern recognition problem, there is a sample space S consisting of elements. These elements, also known as pattern samples or samples, are the focus of a specific problem. For example, in character recognition, a sample would refer to a specific character, while in medical diagnosis, it would be a set of symptoms. The goal of a pattern recognizer is to develop a rule that divides the sample space into partitions where all elements belonging to the same partition are equivalent. Essentially, the sample space S is divided into equivalence classes.

A. Design of a Pattern Recognizer

The basic structure of the pattern recognizer consists of the following three stages:

1) Physical Measurement: In the first stage, each sample (converted from physical measurements) corresponds to a set of ordered numbers.

2) Feature Extraction: In the second stage those features which are judged to be important for the recognition problem are derived from the elements in stage 1 (this is more of an art than a science).

3) Classification : This is the crucial part of the procedure in which the elements are classified on the basis of their features.

The above separation of the problem into the three stages of physical measurement, feature extraction and classification is mainly for convenience. The choice of the features is critical to the success of the classification process, but the former depends on the physical measurements made on the samples. If the original set S can be expressed as $S = \cup_i C(i)$, where C(i) is an equivalence class, the objective of pattern recognition is to find a mapping such that all elements of C(i) are mapped to the same class.

B. Methods for Pattern Recognition

Historically, the methods proposed for pattern recognition , belong to two distinct periods. During the 1960s,70s, and part of 80s most of the methods assumed that the two sets could be separated by a hyperplane in the feature space. Hence the problem was to determine the orientation of the hyperplane based on the test samples. A very large number of outstanding text books exist in which the convergence of the hyperplane to the desired orientation, based on the information contained in the training samples, is rigorously proved.

The rise of methods based on artificial neural networks followed the period referred to earlier. Significantly more complex decision surfaces than hyper-planes (manifolds in the feature space) could be used to perform pattern classification. The methods were significantly less analytic in nature, but the success of the methods in real problems eventually made them the preferred methods in practical applications. In much of the literature in the 1960s, 70s, and 80s, the discriminant surfaces were linear hyperplanes and the classification rule was based on whether a sample lies above or below the hyperplane (i.e., whether the projection of the sample on the normal to the hyperplane is positive or negative). In such situations, classification is the process by which the hyperplane is determined by the training samples, and, if a solution exists, using the hyperplane to classify test samples whose classifications are unknown.

Pattern recognition based on the above methods are well known. When 'Mutual Learning' is used for such problems, it is assumed that agents (or machines) with different training sets (datasets) are attempting to solve the same problem. Our interest lies in the questions that can arise when they communicate with each other and whether they can improve their performance in some sense by such communication.

V. THE MUTUAL LEARNING ALGORITHM

In this section, we describe our main contribution of the paper, i.e., the mutual learning algorithm. As stated earlier, we propose two different algorithms for mutual learning.

A. Algorithm I - Similarity Matching Based Mutual Learning

In this algorithm, the two agents take turns in serving the other agent's teacher, i.e., there are no predefined assigned roles as teacher and student between the two agents. When an agent encounters a novel data-point that is not present in either agent's dataset, the two agents engage in mutual communication where each one looks for points that are 'similar' in its training set, and the corresponding data labels. The agent that wins this competition, i.e., has (labeled) examples that are more 'similar' to the novel data point than the other agent, serves as the teacher and the other agent takes the role of the student for the novel data point. The two agents augment their training with their respective (labeled) training datasets intermittently with such mutual learning with exploratory 'novel' datapoints not included in the training set.

The expectation in such mutual learning is that, by leveraging the 'expertise' of the other agent on specific 'unseen' examples, an agent may be able to overcome the inadequacy of its training data, and will be able to learn faster. That is, each agent will be able to achieve superior classification performance than what is possible without such mutual learning. The following describes the proposed mutual learning algorithm in a more precise manner.

Let D be a (training) dataset of ground truth with N examples. At each iteration k, each agent picks an example from D with probability p(k). With probability (1-p(k)) they choose a random input with unknown class label. If they choose an example from D, learning proceeds as in the isolated learning case. If they choose a random input X, each agent A_1 and A_2 determines their output classification for M <= N examples in D that are closest to X. Whichever agent A_i has higher number of correct labels for these M ground truth examples, is considered the teacher, with the other agent A_j

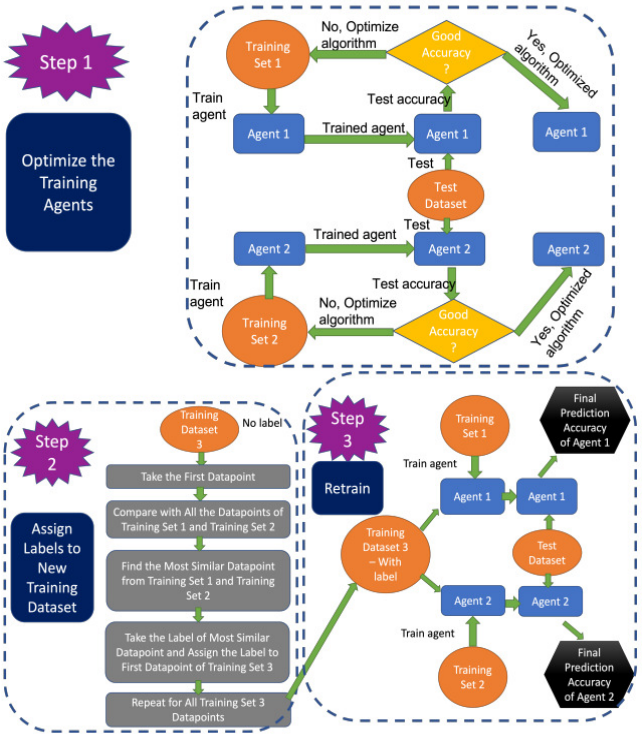


Figure 2: Pictorial representation of Algorithm 1

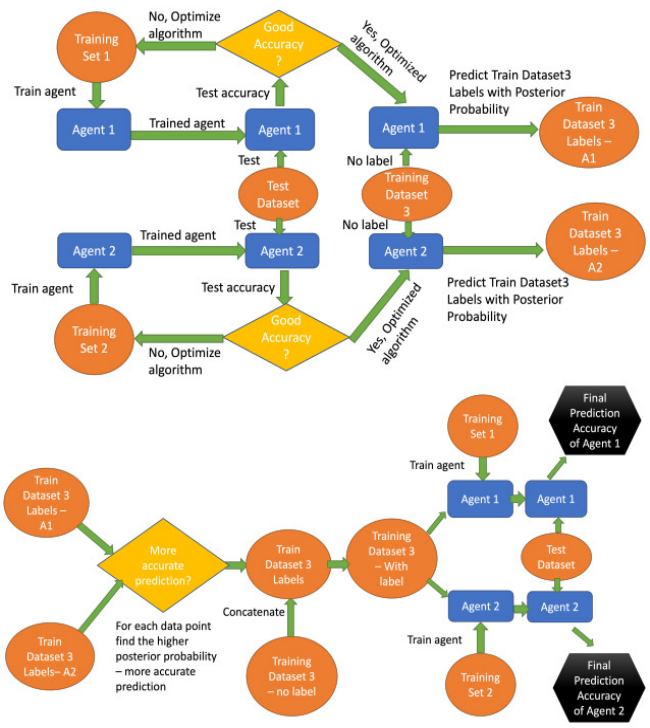


Figure 3: Pictorial representation of Algorithm II

being the student. The student A_j then updates itself for X treating the output label generated by the teacher A_i for X as the target. X can be viewed as an off-line experiment, while choosing any example from D is an on-line experiment. p(k) starts at k=1 with a value of 0.5 (say) but is increased towards 1 with increasing k as, say,

$$p(k) = 0.5*(2- 1/\sqrt{k}))$$

Therefore, eventually both agents only use ground truths. This provides the guarantee that eventually both agents will only use on-line experiments, i.e., isolated learning with ground truths, and therefore is guaranteed to perform no worse than isolated learning. What we hope to demonstrate that with an appropriate scheduling of p(k), the mutually learning team can achieve a given high level of accuracy with fewer total online experiments than that required by an isolated learning agent, by making used of the off-line experiments.

Fig.2 shows a pictorial representation of algorithm 1. Fist, two agents are trained with different training datasets first (fig 2 top). When an agent encounters a novel datapoint, it matches the novel data point with all the datapoints in both the training datasets of agent1 and agent2. In this way it tries to find the 'most similar' and previously seen datapoint. The new novel datapoint is labeled the same as the 'most similar' datapoint (fig 1 bottom). All the novel datapoints are labelled in the same way and both the agents are retrained with new novel dataset and their respective old training dataset (fig 2 bottom).

B. Algorithm II - Previous Knowledge Based Mutual Learning

In this algorithm the two agents are trained with different training datasets first (fig 3 top). When an agent encounters a novel data point, both the agents predict the label of the data point based on their own knowledge and previous training. The confidence level or the prediction accuracy probability for each agent are also calculated at the same time (fig 3 top). The agent with higher confidence wins the competition and the novel data point is labeled according to the winner agent's prediction (fig 3 bottom). The winner agent then works as the teacher and the other agent becomes the student for that particular new data point. In this way, for a particular unseen data point, we are trying to see which agent has the most capability to predict based on previous knowledge and training. In this way the 'less capable' agent for that particular agent learns something new and as a result, it will be able to predict for similar datapoint faster and with better accuracy in future. The two agents augment their training with their respective (labeled) training datasets intermittently with such mutual learning with exploratory 'novel' datapoints not included in the training set.

Assuming there are total 'N' number of classes in a multiclass classification problem. For a particular unseen new data point the probability of predictions from output neurons can be written as follows according to Gibb's measure

$$\sum_{k=1}^{N} Pr(Y_i = k) = \sum_{k=1}^{N} \frac{1}{Z} e^{\beta_k \cdot X_i} = \frac{1}{Z} \sum_{k=1}^{N} e^{\beta_k \cdot X_i} = 1$$

Here we assumed softmax activation function. Solving for Z (normalization constant) gives

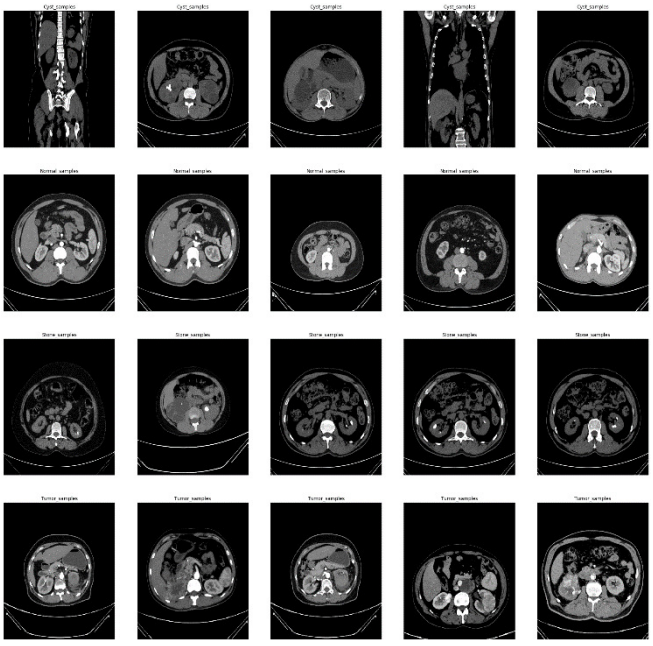
$$\sum_{k=1}^{N} e^{\beta_k \cdot X_i} = Z$$

Therefore, the prediction confidence for different classes are given as

$$Pr(Y_i = 1) = \frac{e^{\beta_1 \cdot X_i}}{\sum_{k=1}^{N} e^{\beta_k \cdot X_i}}, \quad Pr(Y_i = 2) = \frac{e^{\beta_2 \cdot X_i}}{\sum_{k=1}^{N} e^{\beta_k \cdot X_i}}$$

.....

$$Pr(Y_i = N) = \frac{e^{\beta_N \cdot X_i}}{\sum_{k=1}^{N} e^{\beta_k \cdot X_i}}$$



.

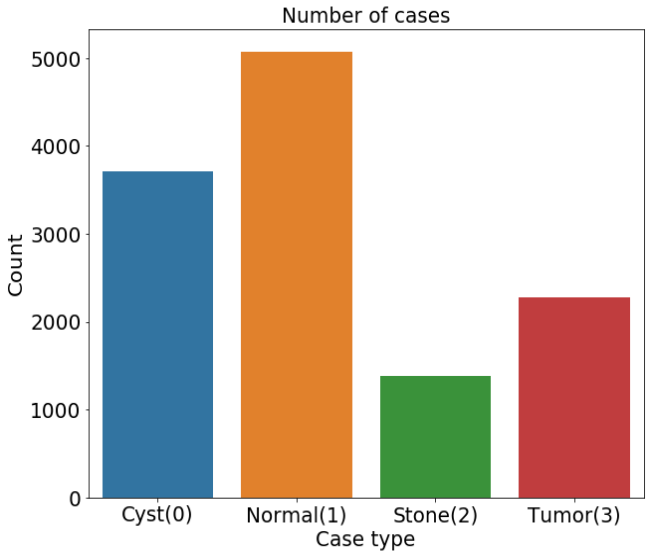


Figure 4 : (top) Samples of Kidney images. The first row shwos kidney images with Cyst. Second row are the scans of nomal kidneys. The third row contains the scans of kidneys with stone. The last row shows the scans of kidneys with tumors. (bottom) Histogram plot showing the number of samples in different data types in full training dataset.

The maximum probability is the predicted class for the unseen new data point. In this algorithm we determine the probabilities of predicted class from different agents and consider the prediction of the most 'confident' agent.

VI. SIMULATION FRAMEWORK FOR MUTUAL LEARNING DEMONSTRATION IN CNN

A. *Training Data*

The dataset is a collection of 12,446 unique data within it in which the cyst contains 3,709, normal 5,077, stone 1,377, and tumor 2,283 [38]. The size of each image is 865x700 pixels (figure 4-top). The training dataset contains 10,000 images and the testing dataset contains 2,446 images. The images roes through one hot encoding and the pixel values are normalized. The dataset is not well distributed especially if you consider the individual data points for different kidney diseases. Nevertheless, in our real world, we rarely have well distributed dataset rather it is expected that the we will see skewed data distribution. We intentionally kept the raw dataset in the same way so that we can demonstrate that our proposed mutual learning algorithm is not strongly affected by the skewness of the data.

The training dataset is equally divided into 4 smaller datasets – each containing 2,500 training points. The CNN is trained separately with the full training dataset and small training datasets and accuracy is tested against the testing dataset.

B. *CNN Model*

The CNN model consists of two parts - the data preprocessing part and the artificial neural network. The CNN preprocessing steps contains several layers. First, there is a 2D convolution layers both with 28 output filter with 3x3 kernel and ReLU activation function. The output of the convolution layer goes through a 2x2 maxpool layer. The input images again goes though a convolution layers both with 64 output filter with 3x3 kernel and ReLU activation function. The output from the convolution layer goes through a 2x2 maxpool layer, another convolution layers both with 64 output filter with 3x3 kernel and ReLU activation function. Then finally there is a flatten layer to convert the 2D data to 1D array.

The second layer in CNN is a fully connected artificial neural network (ANN). The ANN basically consist of four layers -three hidden dense layers and the output layer. The first hidden dense layer contains 640 neurons. Output from the first hidden layer goes through a dropout layer to avoid overfitting. Second hidden dense layer consists of 264 neurons, third hidden dense layer consists of 64 neurons and the output layer consists of 4 neurons. Each output neuron indicates a probability of kidney with cyst, normal kidney, kidney with stone and kidney with tumor respectively. The output neuron indicating highest probability is the final result.

C. *SVM Model*

In this work we used two types of SVM network – SVM with linear kernel and non-linear (sigmoid) kernel. The hyperparameters are optimized for both linear and non-linear SVM to achieve good accuracy with full training dataset.

VII. RESULTS AND DISCUSSION

First, the CNN network is trained with the complete training dataset and tested for efficiency using the testing dataset. Next, the CNN network is trained with each of the smaller training datasets, and efficiency is tested against the testing dataset each time. Two CNN networks are then trained simultaneously using different small and randomly chosen training datasets, followed by interaction between the networks for data exchange and mutual training. The efficiency of both CNN networks is then tested using the testing dataset. The experiment is further repeated with four teacher-student networks.

A. *CNN Network Testing Efficiency with Full Training Dataset and One-fourth Training Dataset*

The CNN is first trained with the full training dataset first and the accuracy is tested against the testing dataset. Training with the full dataset is a time and memory hungry procedure. The whole process took almost 35 minutes in a machine with 1.6 GHz dual-core processor and 8GB 2,133 MHz RAM. The

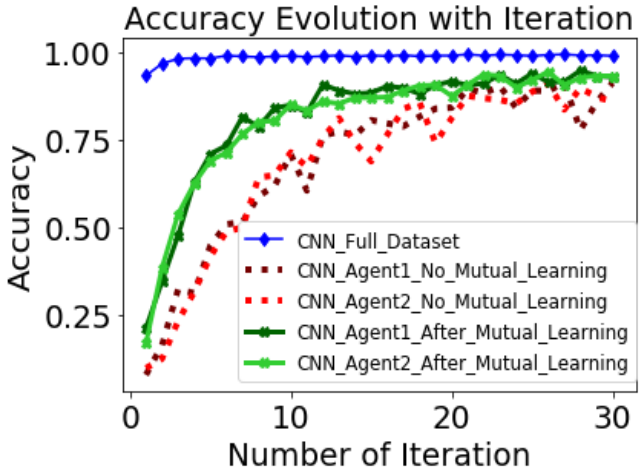


Fig. 5. CNN network accuracy plot with iteration for mutual learning. The top blue plot indicates the accuracy when the CNN is trained with the full training dataset. The light green and dark green plots indicates the CNN network accuracy with iteration after mutual learning. The light red and dark red lines are the accuracy plots before mutual learning.

maximum accuracy is 99.54%. The accuracy plot with iteration for full training dataset is shown in fig. 5 (top blue curve). The CNN is then trained with the each of the small datasets and accuracy is tested every time. In fig. 5 accuracy plots are shown for two of the small training sets. The maximum accuracy for all the training sets is found to be 84.1%.

B. Mutual Learning with Two Teacher-Student Networks Using Algorithm I

Two CNN networks are trained with two randomly selected small training sets (dataset 1 and dataset 2) first. The two networks then share their knowledge with each other and get trained further. For that, the labels are removed from another small training dataset (dataset 3).

Each datapoint (kidney scnas) are compared with the data points of dataset 1 and dataset 2 that are used to train the two CNNs. The comparison is done by comparing each corresponding pixels of the image and calculating the root mean square value of the difference. The closest data point from the two training datasets is assumed to contain the right label, The rest of the data points in dataset 3 are relabeled in this way and the two CNNs are trained accordingly.

Fig. 5 shows the accuracy plots before and after the mutual learning. The accuracy clearly got better after applying the mutual learning algorithm. The maximum accuracy before mutual learning was 84.1% while the maximum accuracy increased to 90.45% after mutual learning. In this way, mutual learning can help when system cannot be trained with big single dataset or there is scarcity of single big training dataset. Here both the network accuracy increases simultaneously which adds to the benefit of dynamic teacher-student network instead of static teacher-student network.

C. Mutual Learning with Two Teacher-Student Networks Using Algorithm II

Two CNN networks are trained with two different, randomly selected small one-fourth training sets again (dataset 1 and dataset 2). The two networks then share their knowledge with each other using the second mutual learning algorithm and get trained further. For that, the labels are removed from the another small training dataset (dataset 3).

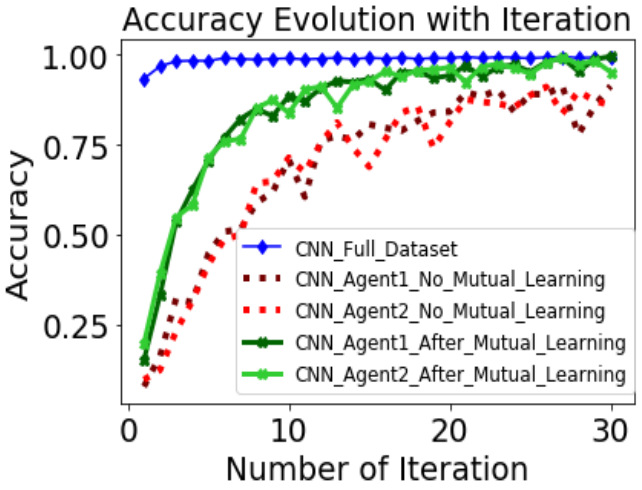


Fig. 6. CNN network accuracy plot with iteration for mutual learning. The top blue plot indicates the accuracy when the CNN is trained with the full training dataset. The light green and dark green plots indicates the CNN network accuracy with iteration after mutual learning. The light red and dark red lines are the accuracy plots before mutual learning.

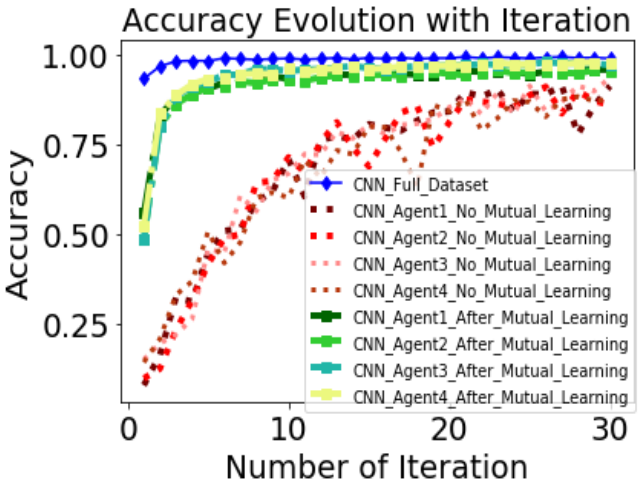


Fig 7. Accuracy plot with iteration for four CNN agents with mutual learning and without mutual learning

The new training data point labels are predicted using both the CNN agents along with their prediction confidence. Each data point in the dataset 3 set is relabeled according to more confident agent and both the agents are retrained after the labeling is finished. Fig. 6 shows the accuracy plots before and after the mutual learning. The accuracy clearly got better after applying the mutual learning algorithm. The maximum accuracy before mutual learning was 84.1% while the maximum accuracy increased to 93.6% after mutual learning. The accuracy is improvement is better than first algorithm. This is because the proximity calculation between figures can be more prone to error. Since, the accuracy of agents are already quite high, the confidence based relabeling is more accurate. But if the pre-mutual learning accuracy is low for the agents then algorithm I should work better than algorithm II. The most appropriate algorithm therefore depends on the pre-mutual learning accuracies of the agents and input types.

D. Mutual Learning with Four Teacher-Student Networks Using Algorithm I

Model distillation can be more efficient and the accuracy can be further improved if more agents share information

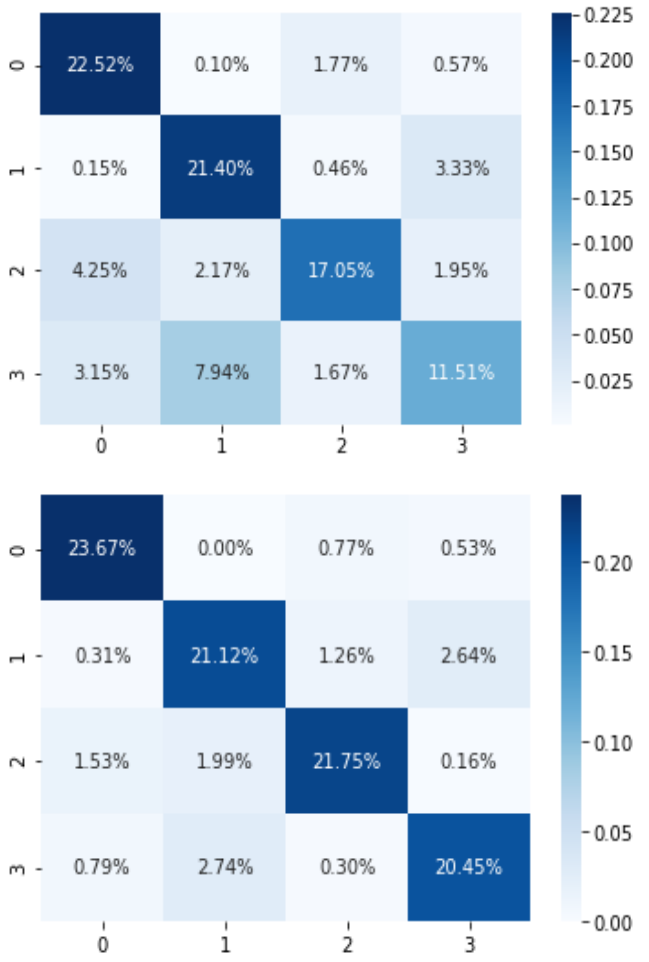


Fig. 8. Heatmap of confusion matrix for linear and non-linear SVN trained with full training dataset.

among themselves. To demonstrate the fact, we repeated the mutual learning algorithm with four networks. All the networks dynamically play the role of teacher and student. When one network plays the role of teacher, the three other networks play the role of student. Since four networks can share a lot more information with each other compared with two networks, all the networks become more well-trained and hence, the efficiency of all the networks increases simultaneously.

Fig. 7 shows the accuracy plots before and after machine learning for ten agents. The maximum accuracy achieved in this case is 94.78% compared to 90.45% for two network mutual learning. The green curve set represents the mutual learning with ten teacher-student networks in fig. 7.

E. SVM Network Testing Efficiency with Full Training Dataset and One-fourth Training Dataset

The SVM with linear and non-linear kernel is first trained with the full training dataset first and the accuracy is tested against the testing dataset. Training with the full dataset is a time and memory hungry procedure. The whole process took couple of hours in a machine with 1.6 GHz dual-core processor and 8GB 2,133 MHz RAM. The maximum accuracy for SVM with linear kernel is 76.59% while maximum accuracy for SVM with non-linear kernel is 85.9% after 30 iteration. The heatmap of the confusion matrix for linear and non-linear SVM is shown in fig 8.

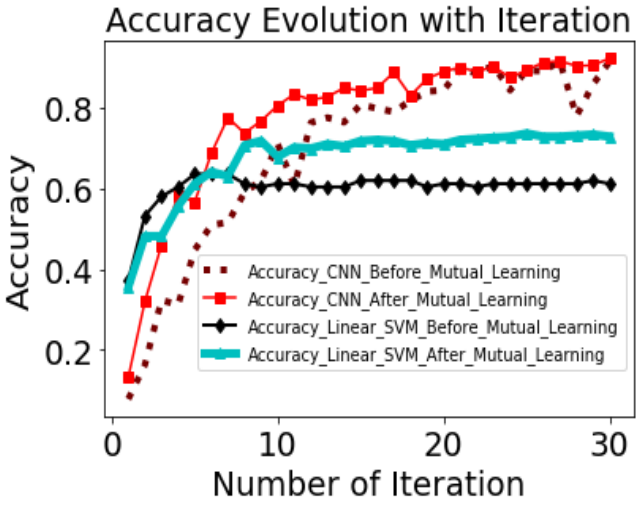


Fig. 9. Linear SVM anc CNN network accuracy plot with iteration for mutual learning. The brown and black plots indicates the CNN and linear SVM network accuracy cosecutively with iteration before mututal learning. The red and pastle lines are the accuracy plots after mutual learning for CNN and linear SVM network cosecutively.

Both SVM agents are then trained with small dataset and accuracy is tested every time. The maximum accuracy for linear SVM with small training set is 59.63%. while the accuracy is 44.1% for non-linear SVM.

F. Mutual Learning with Two SVM Networks Using Algorithm I

Homogeneous mutual learning is applied between two linear SVMs in the same ways as it was applied between two CNN agents. The linear SVM accuracy went up to 61.97% after mutual learning. Similarly, the accuracy went up to 55.74% after mutual learning between two non-linear SVMs.

G. Heterogeneous Mutual Learning with CNN and SVM Networks Using Algoritm II

Model distillation between homogeneous agents has been shown in literature. We have shown here that mutual learning is possible between heterogenous agents and the result is exciting. In figure 9, we have shown the accuracy plots before and after the mutual learning for linear SVM and CNN. The accuracy clearly got better after applying the mutual learning algorithm II. The maximum accuracy for linear SVM before mutual learning was 59.63% while the maximum accuracy increased to 74.04% after mutual learning. The maximum accuracy for CNN before mutual learning was 84.1% while the maximum accuracy increased to 85.76% after mutual learning. In this way, mutual learning can help both the agents get better accuracy.

One worth mentioning point is that we chose simple CNN and SVM algorithm because the basic purpose of the paper is not to show impressive accuracy of kidney disease diagnostic with machine learning. Rather our intention is to show that our proposed mutual learning algorithm works with different machine learning and neural network algorithm. Also most often we see that a particular algorithm is showing excellent accuracy for a particular dataset but might show poor accuracy for other dataset. A comprehensive way to avoid this fundamental issue is to keep training the algorithm with new datasets over the time. The mutual learning enables the machine learning algorithm to keep learning over time.

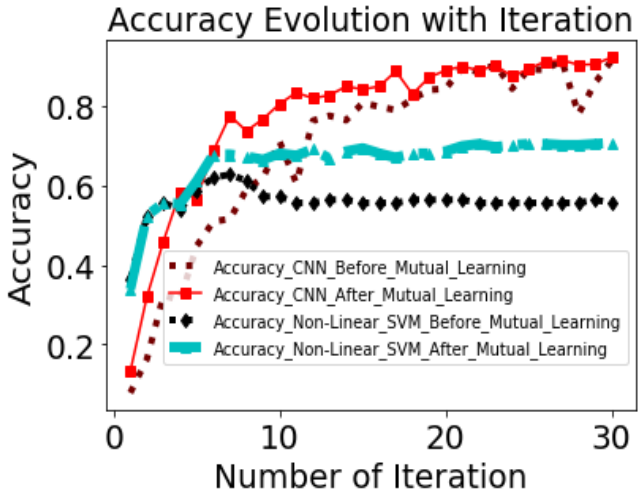


Fig. 10. Non-linear SVM anc CNN network accuracy plot with iteration for mutual learning. The brown and black plots indicates the CNN and non-linear SVM network accuracy cosecutively with iteration before mututal learning. The red and pastle lines are the accuracy plots after mutual learning for CNN and non-linear SVM network cosecutively.

In figure 10, we have shown the accuracy plots before and after the mutual learning for non-linear SVM and CNN. The accuracy clearly got better after applying the mutual learning algorithm. The maximum accuracy for non-linear SVM before mutual learning was 44.1% while the maximum accuracy increased to 72.21% after mutual learning. The maximum accuracy for CNN before mutual learning was 84.1% while the maximum accuracy increased to 84.79% after mutual learning.

An important observation is that the accuracy of SVM linear agent increased by 14.41% while accuracy of CNN only increased by 1.66%. Similarly, the accuracy of SVM linear agent increased by 28.11% while accuracy of CNN only increased by 0.69%. This is because CNN has already a much higher accuracy that SVM. Therefore, when they are engaged in mutual learning and trying to teach each other, the SVM learns a lot from CNN. But since the SVM accuracy was not high before mutual learning, it is not able to teach the CNN much and hence the CNN is less benefited from the mutual learning. Furthermore, the accuracy of CNN increases less in figure 9 vs in figure 10 because the non-linear SVM agent used in figure 10 has lower accuracy than the linear SVM agent used in figure 9. The machines here replicate our real life experience quite nicely.

H. Accuracy and Timing Comparison

The timing and accuracy comparison is shown below in table I. Clearly mutual learning gives a great advantage rather than training with big dataset because it significantly reduces the time and computational resource. It can give the flexibility to train multiple networks in parallel.

TABLE I. ACCURACY AND TIMING COMPARISON FOR CNN

	Single big training dataset	***Two small one-fourth Dataset***	***Mutual learning two agents Algorithm I***	***Mutual learning two agents Algorithm II***	***Mutual learning with four agents***
Maximum Accuracy	99.54%	84.1%	90.45%	93.6%	94.78%
Execution Time	~35 minutes	~6 minutes	~ 17 minutes	~ 15 minutes	~ 30 minutes

TABLE II. ACCURACY AND TIMING COMPARISON FOR SVM

	Single big training dataset	***Two randomly selected small one-fourth Dataset***	***Mutual learning with two SVM agents***	***Mutual learning with CNN and SVM agents***
Linear SVM Maximum Accuracy	76.59%	59.63%	61.97%	74.04%
Non-linear SVM Maximum Accuracy	85.9%	44.1%	55.74%	72.21%
Execution Time	~8 minutes	~2.5 minutes	~ 6 minutes	~ 6.5 minutes

VIII. CONCLUSION

This paper explores mutual learning in pattern classification, where two agents, P and Q, have separate sets of learning samples (patterns A and B) that require classification. The primary objective is to ensure that both agents classify all learning samples correctly, and the exchange of all samples is one possible solution. However, the paper seeks more efficient ways to determine misclassified samples between the two agents.

The paper concludes that detailed discussion between the two agents is necessary for successful classification, particularly regarding samples near the discriminant surfaces of the classifiers. When one agent misclassifies a learning sample of the other, the latter must continue its learning process until it correctly classifies the sample. Both agents then store different learning samples to accelerate the mutual learning process.

Furthermore, the paper presents a detailed description of a classification problem with simulation results that demonstrate the proposed mutual learning algorithm significantly enhances the participating agents' performance compared to isolated learning without mutual learning.

To summarize, the mutual learning algorithm has several benefits for machine learning systems. It not only improves the accuracy of all the networks involved, but it also enhances the speed of learning. This feature makes it a suitable option for practical systems with memory and computation resource constraints. Moreover, it enables many small networks to operate simultaneously, making it compatible with GPU-based systems. By increasing the number of networks, the accuracy of the system can be enhanced, providing the flexibility to adjust the system size as per the need. In short, the mutual learning algorithm can make machine learning faster, more flexible, and require fewer memory and computing resources.

ACKNOWLEDGMENT

The research reported here was supported by the National Science Foundation under grant numbers 1930601 (to Yale) and 1930606 (to IUPUI).

REFERENCES

[1] Singh, H., Meyer, A. N. & Thomas, E. J. "The frequency of diagnostic errors in outpatient care: estimations from three large observational studies involving us adult populations". *BMJ Qual. Saf.* **23**, 727–731 (2014).

[2] Singh, H., Schiff, G. D., Graber, M. L., Onakpoya, I. & Thompson, M. J. "The global burden of diagnostic errors in primary care", *BMJ Qual. Saf.* **26**, 484–494 (2017).

[3] Graber, M. L. "The incidence of diagnostic error in medicine", *BMJ Qual. Saf.* **22**, ii21–ii27 (2013).

[4] Singh, H., Giardina TD, Meyer AN, Forjuoh SN, Reis MD, Thomas EJ., "Types and origins of diagnostic errors in primary care settings". *JAMA Intern. Med.* **173**, 418–425 (2013).

[5] Liang, H., Tsui BY, Ni H, Valentim CCS, Baxter SL, Liu G, Cai W, Kermany DS, et al. "Evaluation and accurate diagnoses of pediatric diseases using artificial intelligence". *Nat. Med.* **1**, 433–438 (2019).

[6] Topol, E. J. "High-performance medicine: the convergence of human and artificial intelligence". *Nat. Med.* **25**, 44 (2019).

[7] De Fauw J., Ledsam J.R., Romera-Paredes B., Nikolov S., Tomasev N., Blackwell S., Askham H., Glorot X., O'Donoghue B., Visentin D., van den Driessche G., Lakshminarayanan B., Meyer C., Mackinder F., Bouton S., Ayoub K., Chopra R., King D., Karthikesalingam A., Hughes C.O., Raine R., Hughes J., Sim D.A., Egan C., Tufail A., Montgomery H., Hassabis D., Rees G., Back T., Khaw P.T., Suleyman M., Cornebise J., Keane P.A., Ronneberger O.. "Clinically applicable deep learning for diagnosis and referral in retinal disease". Nat Med. 2018 Sep;24(9):1342-1350. *doi: 10.1038/s41591-018-0107-6.* Epub 2018 Aug 13. PMID: 30104768.

[8] Yu, K.-H., Beam, A. L. & Kohane, I. S. "Artificial intelligence in healthcare". *Nat. Biomed. Eng.* **2**, 719 (2018).

[9] Jiang F, Jiang Y, Zhi H, Dong Y, Li H, Ma S, Wang Y, Dong Q, Shen H, Wang Y. "Artificial intelligence in healthcare: past, present and future." Stroke Vasc Neurol. 2017 Jun 21;2(4):230-243. *doi: 10.1136/svn-2017-000101*. PMID: 29507784; PMCID: PMC5829945.

[10] Esteva A, Robicquet A, Ramsundar B, Kuleshov V, DePristo M, Chou K, Cui C, Corrado G, Thrun S, Dean J. "A guide to deep learning in healthcare". Nat Med. 2019 Jan;25(1):24-29. *doi: 10.1038/s41591-018-0316-z.* Epub 2019 Jan 7. PMID: 30617335.

[11] Semigran, H. L., Levine, D. M., Nundy, S. & Mehrotra, A. "Comparison of physician and computer diagnostic accuracy". JAMA Intern. Med. 176, 1860–1861 (2016).

[12] Miller, R. "A history of the internist-1 and quick medical reference (qmr) computer-assisted diagnosis projects, with lessons learned". Yearb. Med. Inform. 19, 121–136 (2010).

[13] Razzaki, S., Baker, A., Perov, Y., Middleton, K., Baxter, J., Mullarkey, D., Sangar, D., Taliercio, M., Butt, M., Azeem Majeed, DoRosario, A., Mahoney, M., Saurabh, J., "A comparative study of artificial intelligence and human doctors for the purpose of triage and diagnosis". Preprint at *https://arxiv.org/abs/1806.10698* (2018).

[14] Vembandasamy, K., Sasipriya, R. and Deepa, E. (2015), "Heart Diseases Detection Using Naive Bayes Algorithm", IJISET-International Journal of Innovative Science, Engineering & Technology, 2, 441-444.

[15] Chaurasia, V. and Pal, S. (2013) "Data Mining Approach to Detect Heart Disease", International Journal of Advanced Computer Science and Information Technology (IJACSIT), 2, 56-66.

[16] Parthiban, G. and Srivatsa, S.K. (2012) "Applying Machine Learning Methods in Diagnosing Heart Disease for Diabetic Patients". International Journal of Applied Information Systems (IJAIS), 3, 25-30.

[17] Tan, K.C., Teoh, E.J., Yu, Q. and Goh, K.C. (2009) "A Hybrid Evolutionary Algorithm for Attribute Selection in Data Mining. Journal of Expert System with Applications", 36, 8616-8630. *https://doi.org/10.1016/j.eswa.2008.10.013*

[18] Iyer, A., Jeyalatha, S. and Sumbaly, R. (2015) "Diagnosis of Diabetes Using Classification Mining Techniques". International Journal of Data Mining & Knowledge Management Process (IJDKP), 5, 1-14. *https://doi.org/10.5121/ijdkp.2015.5101*

[19] Sen, S.K. and Dash, S. (2014) "Application of Meta Learning Algorithms for the Prediction of Diabetes Disease". International Journal of Advance Research in Computer Science and Management Studies, 2, 396-401.

[20] Kumari, V.A. and Chitra, R. (2013) "Classification of Diabetes Disease Using Support Vector Machine". International Journal of Engineering Research and Applications (IJERA), 3, 1797-1801.

[21] Sarwar, A. and Sharma, V. (2012) "Intelligent Naïve Bayes Approach to Diagnose Diabetes Type-2". Special Issue of International Journal of Computer Applications (0975-8887) on Issues and Challenges in Networking, Intelligence and Computing Technologies-ICNICT 2012, 3, 14-16.

[22] Ephzibah, E.P. (2011) "Cost Effective Approach on Feature Selection using Genetic Algorithms and Fuzzy Logic for Diabetes Diagnosis". International Journal on Soft Computing (IJSC), 2, 1-10. *https://doi.org/10.5121/ijsc.2011.2101*

[23] Vijayarani, S. and Dhayanand, S. (2015) "Liver Disease Prediction using SVM and Naïve Bayes Algorithms". International Journal of Science, Engineering and Technology Research (IJSETR), 4, 816-820.

[24] Gulia, A., Vohra, R. and Rani, P. (2014) "Liver Patient Classification Using Intelligent Techniques". (IJCSIT) International Journal of Computer Science and Information Technologies, 5, 5110-5115.

[25] Rajeswari, P. and Reena,G.S. (2010) "Analysis of Liver Disorder Using Data Mining Algorithm". Global Journal of Computer Science and Technology, 10, 48-52

[26] Ba-Alwi, F.M. and Hintaya, H.M. (2013) "Comparative Study for Analysis the Prognostic in Hepatitis Data: Data Mining Approach". International Journal of Scientific & Engineering Research, 4, 680-685.

[27] Karlik, B. (2011) "Hepatitis Disease Diagnosis Using Back Propagation and the Naive Bayes Classifiers". Journal of Science and Technology, 1, 49-62.

[28] Sathyadevi, G. (2011) "Application of CART Algorithm in Hepatitis Disease Diagnosis". IEEE International Conference on Recent Trends in Information Technology (ICRTIT), MIT, Anna University, Chennai, 3-5 June 2011, 1283-1287.

[29] Park C, Awadalla A, Kohno T, Patel S. "Reliable and trustworthy machine learning for health using dataset shift detection". In: Proceedings of the conference on NeurIPS, 2021, pp.1

[30] Hasani N, Morris MA, Rhamim A, Summers RM, Jones E, Siegel E, Saboury B. Trustworthy "Artificial Intelligence in Medical Imaging". PET Clin. 2022 Jan;17(1):1-12. *doi: 10.1016/j.cpet.2021.09.007*. PMID: 34809860; PMCID: PMC8785402.

[31] Hinton, Geoffrey E., Vinyals, O., Dean, J.,. "Distilling the Knowledge in a Neural Network." *ArXiv abs/1503.02531* (2015): n. pag.

[32] K. S. Narendra and S. Mukhopadhyay, "Mutual Learning: Part I - Learning Automata," 2019 American Control Conference (ACC), 2019, pp. 916-921, *doi: 10.23919/ACC.2019.8814751.*

[33] K. S. Narendra and S. Mukhopadhyay, "Mutual Learning: Part II -- Reinforcement Learning," 2020 American Control Conference (ACC), 2020, pp. 1105-1110*, doi: 10.23919/ACC45564.2020.9147838*

[34] Jimmy Ba and Rich Caruana. "Do deep nets really need to be deep?," In Advances in Neural Information Processing Systems. 2014.

[35] Adriana, R., Nicolas, B., Ebrahimi, K. S., Antoine, C., Carlo, G., & Yoshua, B. (2015). "Fitnets: Hints for thin deep nets". *Proc. ICLR*, *2*, 3.

[36] David Lopez-Paz, Ankit Singh Rawat Sashank J. Reddi Seungyeon Kim Sanjiv Kumar, "Unifying distillation and privileged information," International Conference on Learning Representations, 2016.

[37] Ying Zhang, Xiatian Zhu, Mao Ye,. 2018. "Deep mutual learning". In Conference on Computer Vision and Pattern Recognition, (CVPR), pages 4320–4328.

[38] Islam MN, Hasan M, Hossain M, Alam M, Rabiul G, Uddin MZ, Soylu A. "Vision transformer and explainable transfer learning models for auto detection of kidney cyst, stone and tumor from CT-radiography". Scientific Reports. 2022 Jul 6;12(1):1-4.

[39] S. Ikemoto, H. B. Amor, T. Minato, B. Jung and H. Ishiguro, 2012. "Physical human-robot interaction: Mutual learning and adaptation". IEEE robotics & automation magazine, 19(4), pp.24-35.

[40] Nie, X., Feng, J. and Yan, S., 2018. "Mutual Learning to Adapt for Joint Human Parsing and Pose Estimation". In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 502-517).

[41] Panait, L. and Luke, S., 2005. "Cooperative multi-agent learning: The state of the art". *Autonomous agents and multi-agent systems*, *11*(3), pp.387-434.

[42] Bai, Q., Su, C., Tang, W, Li Y.. "Machine learning to predict end stage kidney disease in chronic kidney disease". *Sci Rep* **12**, 8377 (2022). *https://doi.org/10.1038/s41598-022-12316-z*

[43] Dashtban A, Mizani MA, Pasea L, Denaxas S, Corbett R, Mamza JB, Gao H, Morris T, Hemingway H, Banerjee A. “Identifying subtypes of chronic kidney disease with machine learning: development, internal validation and prognostic validation using linked electronic health records in 350,067 individuals”. EBioMedicine. 2023 Mar;89:104489. *doi: 10.1016/j.ebiom.2023.104489*. Epub 2023 Feb 27. PMID: 36857859; PMCID: PMC9989643..

[44] U. Ekanayake and D. Herath, "Chronic Kidney Disease Prediction Using Machine Learning Methods," *2020 Moratuwa Engineering Research Conference (MERCon)*, Moratuwa, Sri Lanka, 2020, pp. 260-265, *doi: 10.1109/MERCon50084.2020.9185249*.

[45] Bhandari M, Yogarajah P, Kavitha MS, Condell J. “Exploring the Capabilities of a Lightweight CNN Model in Accurately Identifying Renal Abnormalities: Cysts, Stones, and Tumors, Using LIME and SHAP”. *Applied Sciences*. 2023; 13(5):3125. *https://doi.org/10.3390/app13053125*

[46] Islam, M.N., Hasan, M., Hossain, M.K. *et al.* “Vision transformer and explainable transfer learning models for auto detection of kidney cyst, stone and tumor from CT-radiography”. *Sci Rep* **12**, 11440 (2022). *https://doi.org/10.1038/s41598-022-15634-4*

[47] Badawy M, Abdulqader M. Almars, Hossam Magdy Balaha, Mohamed Shehata, Mohammed Qaraad, Mostafa Elhosseini, “A two-stage renal disease classification based on transfer learning with hyperparameters optimization”. Front Med (Lausanne). 2023 Apr 5;10:1106717. *doi: 10.3389/fmed.2023.1106717*. PMID: 37089598; PMCID: PMC10113505