Medical Outcome Prediction for Intensive Care Unit Patients

Simone A. Ludwig, North Dakota State University, USA

Stefanie Roos, Darmstadt University, Germany

Monique Frize, Carleton University, Canada

Nicole Yu, Carleton University, Canada

ABSTRACT

The rate of people dying from medical errors in hospitals each year is very high. Errors that frequently occur during the course of providing health care are adverse drug events and improper transfusions, surgical injuries and wrong-site surgery, suicides, restraint-related injuries or death, falls, burns, pressure ulcers, and mistaken patient identities. Medical decision support systems play an increasingly important role in medical practice. By assisting physicians in making clinical decisions, medical decision support systems improve the quality of medical care. Two approaches have been investigated for the prediction of medical outcomes: "hours of ventilation" and the "mortality rate" in the adult intensive care unit. The first approach is based on neural networks with the weight-elimination algorithm, and the second is based on genetic programming. Both approaches are compared to commonly used machine learning algorithms. Results show that both algorithms developed score well for the outcomes selected.

Keywords: *Genetic Algorithms, Genetic Programming, Hours of Ventilation, Intensive Care, Machine Learning Algorithms, Mortality, Neural Networks*

INTRODUCTION

A study of the health care system in the United States reported that at least 44,000 people, and perhaps as many as 98,000 people, die in hospitals each year as a result of medical errors, many of which could have been prevented. Medical errors can be defined as the failure of a planned action to be performed as intended or the use of the wrong action to achieve an aim. Problems that frequently occur during the course of providing health care are adverse drug events and improper transfusions, surgical injuries and wrong-site surgery, suicides, restraint-related injuries or death, falls, burns, pressure ulcers, and mistaken patient identities. High error rates with serious consequences are most likely to occur in intensive care units, operating rooms, and emergency departments (Institute of Medicine, 1999).

Medical decision support systems play an increasingly important role in medical practice to address the above stated problems. By assisting physicians with making clinical decisions,

DOI: 10.4018/jcmam.2010040102

medical decision support systems are expected to improve the quality of medical care (Wennber & Cooper, 1999).

Sim et al. (2001) define clinical or medical decision support systems as software designed to be a direct aid to clinical decision-making, where the characteristics of an individual patient are matched to a computerized clinical knowledge base; patient-specific assessments or recommendations are then presented to the clinician and/or the patient for a decision. Numerous medical decision support systems have been developed to assist medical practice. In 2001, Kaplan reviewed 27 clinical decision support systems reported in the literature (Kaplan, 2001), while Metaxiotis et al. (2000) list 13 well known systems developed for diagnosis, test result interpretation and knowledge management. The range of clinical decision support systems spans the realms of home health care, to enterprise-wide systems, to medical research laboratories. When developing a new CDSS, several factors need to be considered to increase the likelihood that it will be integrated into the health care delivery in a variety of clinical environments. These factors need to be applied at all stages of the development life cycle of the CDSS. The criteria for a successful deployment of a CDSS can be divided into three main areas: (i) The data entry and the decision algorithms; (ii) the human-computer interaction, which includes the data acquisition and the manner in which information is requested from the system; and its usability; (iii) the output of the CDSS, including the format and type of information supplied (Frize et al., 2010).

The application of machine learning methods in medicine is the subject of considerable ongoing research, which mainly concentrates on modeling some of the human actions or thinking processes and recognizing diseases from a variety of input sources (e.g. cardiograms, CAT (Computed Axial Tomography) / MRI (Magnetic Resonance Imaging) / ultrasound scans, photomicrographs, etc.). Other application areas are knowledge discovery (Neves et al., 1999), and biomedical systems, which include genetics and DNA analysis. The use of machine learning has also been applied to biomedical science related systems. There is already a growing interest in the application of learning systems for the interpretation of gene expression data (Brown et al., 1999; Slonim et al., 2000).

In a medical diagnosis problem, what is needed is a set of examples that are representative of all the variations of the disease. The examples need to be selected very carefully if the system is to perform reliably and efficiently. However, development of machine learning systems for medical decision-making is not a trivial task. Difficulties include the acquisition, collection and organization of the data that will be used for training the system. This becomes a major problem, especially when the system requires large data sets over long periods of time, which in most cases is not available due to the lack of an efficient recording system, or because of privacy issues. Another difficulty arises when trying to automate some processes as not all of them can be automated due to ethical and safety issues. Deciding what could and needs to be automated directly influences the design and implementation of the learning system.

The aim of this paper is to compare two classifiers with other machine learning techniques for the prediction of medical outcomes, such as the hours of ventilation necessary for patients in the Intensive Care Unit (ICU), and the prediction on whether the patient is likely to survive. Our two classifiers developed are based on Artificial Neural Networks and Genetic Programming, which are compared to well-known classifiers.

The paper is structured as follows. The first section describes related work, in particular different classification models. The datasets and our two classifiers used for this investigation are described, as well as the other classifiers used to perform the comparison. The results are then provided and compared, and the conclusions and future work are presented.

RELATED WORK

Supervised classification is one of the tasks most frequently carried out in the area of medical informatics. The most prominent classification algorithms can be categorized into logic-based algorithms, neural network algorithms, statistical learning algorithms, instance-based learning algorithms and support-vector machine algorithms. A summary of all algorithms is given below.

There are two groups of learning-based models: decision trees and rule-based classifiers. Decision trees are trees that classify instances by sorting them based on feature values. Each node in a decision tree represents a feature in an instance to be classified, and each branch represents a value that the node can receive. Instances are classified starting at the root node and sorted based on their feature values. The most well-known algorithm is C4.5 (Quinlan, 1993). C4.5 is an extension of Quinlan's earlier ID3 algorithm (Quinlan, 1979).

Decision trees can be translated into a set of rules by creating a separate rule for each path from the root to a leaf in the tree (Quinlan, 1993). However, rules can also be directly induced from training data using a variety of rule-based algorithms. Classification rules represent each class by a disjunctive normal form relation. The goal is to construct the smallest rule-set that is consistent with the training data. A well-known rule-based algorithm is RIPPER (Repeated Incremental Pruning to Produce Error Reduction) (Cohen, 1995).

Neural network approaches are based on the perceptron (Rosenblatt, 1962). A single-layer perceptron network consists of one or more artificial neurons in parallel. Each neuron in the layer provides one network output, and is usually connected to all of the external (or environmental) inputs. The perceptron learning algorithm works as follows. First, the weight and threshold values of the neuron are set to random values. Then, the input is presented. Afterwards, the output of the neuron is calculated, and the weights of the neurons are adjusted. These steps are repeated until a defined error criterion is satisfied. As perceptrons can only classify linearly separable sets of instances, multi-layered perceptrons were invented. A multi-layer neural network consists of large number of units jointed together in a pattern of connections. The pattern of connections is ordered into three layers: input, hidden and output layer. There are several algorithms with which the network can be trained; however, the most well-known and widely used learning algorithm to estimate the values of the weights is the Backpropagation algorithm (Bryson & Ho, 1969).

Statistical approaches are characterized by having an explicit underlying probability model, which provides a probability that an instance belongs in each class, rather than simply a classification. Bayesian networks are the most well known representative of statistical learning algorithms. A Bayesian network is a graphical model for probability relationships among a set of variables. The Bayesian network structure S is a directed acyclic graph (DAG) and the nodes in S are in one-to-one correspondence with the features X. The arcs represent causal influences among the features, which the lack of possible arcs in S encodes conditional interdependencies. Moreover, a feature is conditionally independent from its non-descendants given its parents. Typically, the task of learning a Bayesian network can be divided into two subtasks: first, the learning of the DAG structure of the network, and then the determination of its parameters (Jensen, 1996). Naive Bayesian network are very simple Bayesian networks, which are composed of DAG with only one parent (representing the unobserved node) and several children (corresponding to observed nodes) with a strong assumption of independence among child nodes in the context of their parent (Good, 1950).

Instance-based learning algorithms are lazy-learning algorithms (Mitchell, 1997), as they delay the induction or generalization process until classification is performed. Lazy-learning algorithms require less computation time during the training phase than eager-learning algorithms such as decision trees, neural and

Bayes networks, but more computation time during the classification process. One of the most straightforward instance-based learning algorithms is the nearest neighbour (kNN) algorithm, which is based on the principle that the instances within a dataset generally exist in close proximity to other instances that have similar properties (Cover & Hart, 1967). If the instances are tagged with a classification label, then the value of the label of an unclassified instance can be determined by observing the class of its nearest neighbours. The kNN locates the k nearest instances to the query instance and determines its class by identifying the single most frequent class label.

Support vector machines (SVMs) is the newest commonly used supervised machine learning technique (Vapnik, 1995). SVMs revolve around the notion of a "margin" – either side of a hyperplane that separates two data classes. Maximizing the margin and thereby creating the largest possible distance between the separating hyperplane and the instances on either side of it has been shown to reduce an upper bound on the expected generalization error.

APPROACHES

For the classification of the medical ICU outcomes "death" and "hours of ventilation", our two approaches are investigated further by comparing them to the most common machine learning algorithms of WEKA. The datasets, evaluation measures, and classifiers are explained in detail below.

Dataset

In this study, there were two data sets from an adult intensive care unit, each having the following features:

- Age of the patient
- Chronic: 1 if illness is chronic, 0 otherwise
- Emergency case: 1 if it is an emergency, 0 otherwise
- Post operation: 1 if patient had an operation before, 0 otherwise
- Gender: 1 male, -1 female
- Body temperature
- MAP: mean arterial pressure
- Heart rate
- Respiratory rate
- FiO_2 (inspired oxygen)-concentration in blood
- PO_2 (partial pressure of oxygen) in blood
- pH-value (arterial)
- Na: serum sodium (mMol/L of blood)
- K: serum potassium (mMol/L of blood)
- Serum creatinine (mMol/L of blood)
- Hematocrit: (volume of red blood cells) / (volume of blood total)
- WBC: white blood cell count (total/mm^3 in 1000's)
- GCS: Glasgow Coma Score (level of consciousness)

One of the datasets contains data of the outcome "survival of patient", and the other contains the "hours of ventilation" needed by the patient. In the second case, the learning task is a classification into the classes "at most 8 hours of ventilation" and "more than 8 hours of ventilation". The complete data set contains 1491 entries with 18 features, 14 numerical, 4 boolean. The sets are divided into two smaller sets, according to one Boolean attribute, detailing whether the patient had an operation before coming to the ICU. In the following, the sets will be referred to as Mortality post-OP, Mortality non-OP, Ventilation post-OP and Ventilation non-OP. There are 884 entries for the post-OP case, 608 cases for non-OP patients; 12.50% of the non-OP patients' die, and 3.17% of the post-OPs. The prevalence of patients needing more than 8 hours of ventilation is 35.53% in the non-OP set, and 28.28% in the post-OP set. Table 1 shows the distribution of the data sets.

Evaluation Measures

The following measures are recorded: (1) sensitivity: the percentage of positives correctly identified (death or needing more than 8 hours of ventilation is considered 'positive'); (2) specificity: The percentage of negatives cor-

Table 1. Distribution of data set split into 4 categories: ventilation non-OP, ventilation post-OP, mortality non-OP, and mortality post-OP

Data set	number of cases	mortality rate [%]	more than 8 hours of ventilation [%]
non-OP	608	12.50	35.53
post-OP	884	3.17	28.28

rectly identified; (3) CCR (Correct Classification Rate): percentage of correct predictions in total; (4) area under ROC (Receiver Operating Curve). The fitness function used for this study is the log-sensitivity:

$$logsens = -sensitivity^{n} * \log(1 - sensitivity * specificity) \quad (1)$$

WEKA Algorithms

The classifier of the open source machine-learning package called WEKA (Witten and Frank, 2005) will be compared to our two approaches. WEKA is a collection of machine learning algorithms for solving real-world data mining problems. Within WEKA there are several machine learning algorithms available for classification, which are neural networks, support vector machines, decision trees, Bayesian classifiers, and lazy learning methods. A set of the most common machine learning algorithms were chosen for this investigation and a brief description is provided below:

- BN: BN is a Bayes Network learning algorithm using various search algorithms and quality measures.
- IBk: K-nearest neighbours classifier, which can select an appropriate value of K based on cross-validation, and also performs distance weighting.
- J48: This algorithm, as explained earlier, contains the class for generating a pruned or unpruned C4.5 decision tree.
- JRip: This class implements a propositional rule learner RIPPER.
- K*: K* is an instance-based classifier, i.e. the class of a test instance is based upon the class of those training instances similar to it, as determined by some similarity function.
- MP: MP is a multilayer perceptron that uses backpropagation to classify instances. The nodes in this network are all sigmoid (except for when the class is numeric, in which case the output nodes become unthresholded linear units).
- NB: Class for a Naive Bayes classifier using estimator classes. Numeric estimator precision values are chosen based on the analysis of the training data.
- SMO: Support vector machines are a set of related supervised learning methods that analyze data and recognize patterns, used for classification and regression analysis. SMO (Sequential minimal optimization) in particular implements the sequential minimal optimization algorithm for training a support vector classifier. This implementation globally replaces all missing values and transforms nominal attributes into binary ones. It also normalizes all attributes by default.

Neural Network with Weight-Elimination (NNWEA) Approach

The neural network implementation uses a weight-elimination algorithm, processing the data through the ANN and then applying the weight-elimination algorithm. Weight-elimination attempts to reduce small weights to zero (Frize et al., 1995, Ennett & Frize, 2003), effectively removing them from the neural network. This is essentially a form of pruning the network, which has been used to improve ANN

classification results. The back-propagation portion updates the weights in order to maximize the log sensitivity value, an early stopping criterion used to balance sensitivity and specificity and slightly favor sensitivity (Ennett et al., 2002), given in Equation (1). Each ANN run is stopped when the best performance has been reached and maintained for the last 500 epochs. The ANN was set to create classifiers using between 3 and 11 hidden nodes. The best performing classifier was chosen from the nine classifiers. The process has been entirely automated (Rybchynski, 2005; Ennett et al., 2004). The values are scaled between -1 and 1 by using a modified Z-score transformation equation. The fourteen input variables (as given in the dataset subsection) were normalized using Equation (2) where α' is the normalized value, α is the value, μ is the mean and σ is the standard deviation:

$$\alpha' = \frac{\alpha - \mu}{3 * \sigma} \quad (2)$$

Categorical variables (chronic, emergency surgery, sex) were given a value of -1 or 1. Once the variables were normalized, they were split into a training (46.0%), test (28%) and verification set (25%). Since the mortality dataset had a low prevalence of cases in both the post-OP (3.17%) and non-OP (12.50%) databases, we randomly re-sampled from this population in order to improve the performance of the ANN in terms of sensitivity (Ennett & Frize, 2000). For the post-OP dataset, the training set was raised to 5.06% of cases from the original 3.37%. For the verification sets for the outcome ventilation (that is, positive cases defined as "more than 8 hours of ventilation"), was randomly sampled to create ten different sets. The verification sets were used to measure the performance of the ANN and to determine the mean and standard deviation of the ANN performance. A satisfactory performance on the verification sets was indicative that the classifier generalizes well on new, unseen cases. Poor performance suggests overtraining of the network.

Automatic Genetic Programming (AGP) Approach

The origins of evolutionary computation reach back to the 50's of the last century. Genetic programming, in itself, was not considered until the middle of the 80's. The term first appeared in (Cramer, 1985), and the main development took place in the early and middle 90's, particularly through work by Koza (1992). Genetic programming uses the concepts of genetics and Darwinian natural selection to generate and evolve entire computer programs. Genetic programming largely resembles genetic algorithms in terms of its basic algorithm. The notions of mutation, reproduction (crossover) and fitness are essentially the same; however, genetic programming requires special attention when using those operations. While genetic algorithms are concerned with modifying fixed-length strings, usually associated with parameters to a function, genetic programming is concerned with actually creating and manipulating the (non-fixed length) structure of the program (or function). Therefore, genetic programming is more complex than genetic algorithms (Banzhaf et al., 1998) and works as follows. The solution is developed by first creating a number of initial programs, which are then recombined and changed in each evolution step. The set of programs is referred to as the population; any single program is an individual. Run of evolution is the term used to describe the whole process of finding a solution. Before starting an evolution, one has to define (at least) the following:

- Fitness measure/function: a function that evaluates how close a program is to the optimal solution. For the prognosis, the fitness value, as given by Equation (1), is to be minimized.
- Population size: the number of programs that are supposed to be used for evolving a solution.
- Function and terminal set: functions, constants, and variables the programs are allowed to use.

- Genetic operators: selection, crossover and mutation operators and the probability for using the later two. There are a variety of different selection, crossover and mutation operators available to choose from.
- Termination criterion: the evolution usually either ends if a sufficiently good solution is found, or if the maximum number of iterations is reached.

After setting these parameters, the initial population can be created. Unless one already has some idea about how the solution might look like, the programs are built randomly. Each evolution step works as follows: Until a certain percentage of the population size (crossover rate) is reached, new programs are constructed as follows: 2 programs are selected according to the chosen selection method. The programs are "crossed over", that means certain parts of them are swapped. In tree-based genetic programming, a subtree is selected in each program and the two subtrees are swapped. The remaining part of the new population consists of copied programs from the old population (reproduction is the term used for copying old programs) or newly created programs. With a certain probability, the mutation rate, an individual is changed. Mutation can have various forms, most commonly it only changes one function/terminal in a program to a different one. This process is repeated until the termination criterion is reached. The result of the run is usually the program with the best fitness value found during the whole evolution (Poli et al., 2010).

The Java Genetic Algorithms Package (JGAP) (JGAP, 2010) was chosen as the programming platform. JGAP is a Genetic Algorithms and Genetic Programming package written in Java. It is designed to require minimum effort to use, but is also designed to be highly modular. It provides basic genetic mechanisms that can be used to apply evolutionary principles to solve problems.

In order to achieve high accuracy measures such as sensitivity, specificity, CCR and ROC, the following approach was used to automatically fine-tune the GP approach. First, feature selection is performed, then different function sets are analyzed and afterwards the number of generations and population size are investigated.

In all experiments a crossover rate of 0.9 and a mutation rate of 0.1 is used, as well as a population of size 500 is evolved for 300 generations is used for the function set and feature selection. Per generation 10% of the new population is made up by randomly created new individuals, the maximal crossover depth is 12, maximal initial depth is 5, and the maximal number of nodes is 60.

Feature selection resulted in the following features chosen as GP operators as shown in Table 2. As can be seen, for each dataset different features were selected. The common features used for the ventilation datasets are FiO_2, respiratory rate and emergency case. The common features for the mortality datasets are MAP, serum creatinine and chronic.

Different function sets were tested. The function set with the highest CCR and ROC values consisted of the following: *addition*, *subtraction*, *multiplication*, *division*, *larger than*, *less than*, *if*, *logical and*, *logical or*, *power function*, *square root*. The reason why the function set worked best is because of the *if*, the *logical and*, and the *logical or*, as with-

Table 2. Feature selection for GP approach

	Features
Ventilation non-OP	FiO_2, respiratory rate, GCS, PO_2, emergency case, body temperature, pH value, Na value
Ventilation post-OP	Respiratory rate, FiO_2, Na value, emergency case
Mortality non-OP	GCS, FiO_2, MAP, serum creatinine, age, chronic
Mortality post-OP	Emergency case, chronic, heart rate, gender, MAP, PO_2, serum creatinine

out these operators the evolved program only consists of one condition.

Population sizes of 500, 700 and 1000 are tested for 100, 200, 300 and 500 generations each. For all data sets the best fitness is indeed achieved for 1000 individuals and 500 generations: 1.493, which corresponds to an improvement of slightly more than 2.24%. Overall, a larger population size seems to have a higher impact than an increase in the number of generations.

EXPERIMENTS AND RESULTS

Tables 3 to 6 show the results of this investigation. The values of sensitivity, specificity, CCR and ROC value are shown for all algorithms tested. The first eight algorithms are run with WEKA, and the other two classifier implementations (NNWEA and AGP) are run independently. In order to assess these results, it is useful to follow a guideline that physician partners with the Carleton research group favor. For example, if a patient is predicted to live, but dies, this is considered worse than if a patient is predicted to die, but lives. In considering this guideline, we therefore need to ensure that the specificity is close to or over 85%, and that the sensitivity is as high as possible, which means better than 65%.

The results below are judged according to this principle. The CCR is not that important to our clinician partners. For ROC, a value greater or equal to 0.80 shows fairly good discrimination ability. For ventilation, it is important to predict long-term use, as this will impact both the patient status and the planning of resources such as the availability of artificial ventilation machines.

As can be seen from Table 3, the AGP and the NNWEA approaches score best in considering a combination of sensitivity and specificity, and the ROC value is over 0.80. The AGP approach does a little better than the NNWEA on sensitivity and a little worse than the NNWEA for specificity. Neither approach reaches the optimal value of 0.85; however, as explained above, this is not very critical. These results can help to plan the use of ventilators in the ICU.

For the next dataset (see Table 4), the scores are as follows: K* ranks best in terms of sensitivity; AGP ranks highest in terms of specificity; the highest value of CCR is achieved by J48; and the ROC value is largest using BN. However, for the overall score, our two approaches meet the guideline and perform better than all other approaches.

Table 5 shows the results of the mortality dataset looking at non-OP patients. A sensitivity of 1 can be observed for SMO; however,

Table 3. Ventilation dataset – non-OP

	Sensitivity	Specificity	CCR	ROC
BN	0.793	0.759	0.781	0.853
IBk	0.823	0.560	0.729	0.693
J48	0.803	0.606	0.734	0.679
JRip	0.844	0.653	0.776	0.776
K*	0.911	0.338	0.707	0.772
MP	0.811	0.630	0.747	0.746
NB	0.809	0.690	0.766	0.834
SMO	0.844	0.588	0.753	0.716
NNWEA	0.840	0.750	0.760	0.820
AGP	0.913	0.693	0.772	0.803

Table 4. Ventilation dataset – post-OP

	Sensitivity	Specificity	CCR	ROC
BN	0.910	0.820	0.884	0.934
IBk	0.869	0.548	0.778	0.693
J48	0.930	0.816	0.898	0.869
JRip	0.935	0.784	0.892	0.860
K*	0.952	0.408	0.798	0.841
MP	0.929	0.732	0.873	0.902
NB	0.897	0.680	0.836	0.898
SMO	0.938	0.726	0.878	0.835
NNWEA	0.920	0.860	0.880	0.930
AGP	0.906	0.882	0.890	0.894

the specificity is 0, which implies that the model specifically classified all unseen instances to one of the two classes (death). Therefore, SMO is not a good choice for this particular dataset. K* achieves the highest sensitivity, NNWEA achieves the highest specificity, CCR is highest using SMO, and the ROC value is largest using BN. For the overall performance, the NNWEA performs best, although more tuning would be needed to increase the specificity, although this would be slightly at the expense of the sensitivity.

The last table (Table 6) shows the same behavior of SMO. Besides that, again K* achieves a very high sensitivity value of 0.998. The highest values for specificity and ROC are achieved by AGP and the highest CCR value is gained using BN. Here the best overall performing approach is the AGP, which meets the guideline.

In summary, looking at our two approaches, the values of all measures are very comparable, in particular for the two ventilation datasets. For the mortality datasets, much higher values for specificity are achieved by the WEKA algorithms, but sensitivity is fairly low for these approaches. The AGP approach scored best in quite a few categories, as did the NNWEA approach.

Table 5. Mortality dataset – non-OP

	Sensitivity	Specificity	CCR	ROC
BN	0.925	0.368	0.855	0.852
IBk	0.912	0.329	0.839	0.610
J48	0.919	0.303	0.842	0.589
JRip	0.957	0.276	0.872	0.629
K*	0.959	0.237	0.868	0.762
MP	0.930	0.395	0.863	0.805
NB	0.838	0.579	0.806	0.826
SMO	1.000	0.000	0.875	0.500
NNWEA	0.520	0.820	0.690	0.800
AGP	0.921	0.737	0.760	0.829

Table 6. Mortality dataset – post-OP

	Sensitivity	Specificity	CCR	ROC
BN	0.995	0.179	0.969	0.849
IBk	0.988	0.071	0.959	0.537
J48	0.984	0.107	0.956	0.585
JRip	0.987	0.143	0.960	0.567
K*	0.998	0.000	0.967	0.736
MP	0.987	0.071	0.958	0.704
NB	0.950	0.536	0.937	0.854
SMO	1.000	0.000	0.968	0.500
NNWEA	0.670	0.790	0.840	0.770
AGP	0.937	0.893	0.895	0.915

CONCLUSION

Intensive care medicine most often involves rapid decision-making on the basis of a huge amount of information. ICU physicians often rely on conventional wisdom and personal experience to arrive at assessments and judgments. This requires an intuitive weighting of various factors to achieve an optimal balance between clinical situations that are often competing. There is increasing interest in computer-based decision support tools to automate aspects of the medical decision-making that takes place in complex clinical areas such as the ICU.

For the classification of the medical ICU outcomes "death" and "hours of ventilation", our two approaches (one based on Artificial Neural Networks and the other based on Genetic Programming) were investigated and found to perform better overall than most common machine learning algorithms of WEKA, given certain clinical expectations. As seen from the results of the experiments using the ICU datasets, our two approaches (NNWEA and AGP) are performing very well and are comparable to the algorithms provided by WEKA. In particular, the AGP and NNWEA approaches scored the highest values in some categories.

Future work includes further improvements and refinements to the algorithms, as well as the implementation of the algorithms into a classifier suite. This classifier suite would contain all available algorithms and a selection policy, which would run all classifiers, measuring their sensitivity, specificity, CCR and ROC values. Given user specified input on the importance of these four measures, a weighted approach would return the best classifier for a particular dataset. It is likely that this combined approach will increase the performance of a decision support system for critical care in estimating these two outcomes. Other outcomes could be added such as length of stay in the unit, and onset of sepsis for example. Another aspect which could be investigated in the future is to test our approaches with a large medical data set, collected from a different clinical environment, in order to validate them more thoroughly.

REFERENCES

Banzhaf, W., Nordin, P., Keller, R. E., & Francone, F. D. (1998). *Genetic programming: an introduction on the automatic evolution of computer programs and its applications*. San Francisco, CA: Morgan Kaufmann Publishers.

Brown, M. P. S., Grundy, W. N., Lin, D., Cristianini, N., Sugnet, C., Ares, M., & Haussler, D. (1999). *Support Vector Machine Classification of Microarray Gene Expression Data* (Tech. Rep. UCSC-CRL-99-09). Santa Cruz, CA: Department of Computer Science, University of California, Santa Cruz.

Bryson, A. E., & Ho, Y.-C. (1969). *Applied optimal control: optimization, estimation, and control*. New York, NY: Blaisdell Publishing Company and Xerox College Publishing.

Cohen, W. (1995). Fast Effective Rule Induction. In . *Proceedings of*, *ICML-95*, 115–123.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, *13*(1), 21–27. doi:10.1109/TIT.1967.1053964

Cramer, N. L. (1985). A representation for the Adaptive Generation of Simple Sequential Programs. In J. Grefenstette (Ed.), *Proceedings of the International Conference on Genetic Algorithms and the Applications*, Pittsburgh, PA.

Ennett, C., & Frize, M. (2000). Selective Sampling to Overcome Skewed a Priori Probabilities with Neural Networks. In *Proceedings of the A.M.I.A. (American Medical Informatics Association) Annual Symposium*, Los Angeles, CA.

Ennett, C., & Frize, M. (2003). Weight-Elimination Neural Networks Applied to Coronary Surgery Mortality Prediction. *IEEE Transactions on Information Technology in Biomedicine*, *7*(2), 86–92. doi:10.1109/TITB.2003.811881

Ennett, C. M., Frize, M., & Charette, E. (2004). Improvement and automation of artificial neural networks to estimate medical outcomes. *Medical Engineering & Physics*, *26*(4), 321–328. doi:10.1016/j.medengphy.2003.09.005

Ennett, C. M., Frize, M., & Scales, N. (2002). Logarithmic-Sensitivity Index as a Stopping Criterion for Neural Networks. In *Proceedings of IEEE/EMBS* (Vol. 1, pp. 74-75).

Frize, M., Solven, F. G., Stevenson, M., Nickerson, B. G., Buskard, T., & Taylor, K. B. (1995, July). Computer-Assisted Decision Support Systems for Patient Management in an Intensive Care Unit. In *Proceedings of Medinfo95,* Vancouver, BC, Canada (pp. 1009-1012).

Frize, M., Weyand, S., & Bariciak, E. (2010, May). Suggested Criteria for Successful Deployment of a Clinical Decision Support System (CDSS). In *Proceedings of the MeMeA (Medical Measurements and Applications) Workshop,* Ottawa, ON, Canada (pp. 69-72).

Good, I. J. (1950). *Probability and the Weighing of Evidence*. London, UK: Charles Grin.

Institute of Medicine. (1999). *To Err Is Human: Building A Safer Health System*. Washington, DC: Author.

Jensen, F. (1996). *An Introduction to Bayesian Networks*. New York, NY: Springer.

JGAP. (2010). *Java Genetic Algorithms Package*. Retrieved from http://jgap.sourceforge.net

Kaplan, B. (2001). Evaluating informatics applications clinical decision support systems literature review. *International Journal of Medical Informatics*, *64*, 15–37. doi:10.1016/S1386-5056(01)00183-6

Koza, J. R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: MIT Press.

Mextaxiotis, K., & Samouilidis, J.E. (2000). Expert systems in medicine: academic illusion or real power? *Information Management and Security*, 75-79.

Mitchell, T. (1997). *Machine Learning*. New York, NY: McGraw Hill.

Neves, J., Alves, V., Nelas, L., Romeu, A., & Basto, S. (1999). An Information System that Supports Knowledge Discovery and Data Mining in Medical Imaging. In *Proceedings of the ECCAI Advanced Course in Artificial Intelligence (ACAI) Workshop (W13) on Machine Learning in Medical Applications,* Chania, Greece.

Poli, R., Langdon, W. B., & McPhee, N. F. (2010). *A Field Guide to Genetic Programming.* Retrieved from http://www.gp-eld-guide.org.uk

Quinlan, J. R. (1979). Discovering rules by induction from large collections of examples . In Michie, D. (Ed.), *Expert Systems in the Microlectronic age* (pp. 168–201).

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. San Francisco, CA: Morgan Kaufmann Publishers.

Rosenblatt, F. (1962). *Priciples of Neurodynamics*. New York, NY: Spartan.

Rybchynski, D. (2005). *Design of an Artificial Neural Network Research Framework to Enhance the Development of Clinical Prediction Models*. Unpublished master's thesis, University of Ottawa, Ottawa, ON, Canada.

Sim, I., Gorman, P., Greenes, R., Hayes, R., Kaplan, B., & Lehmann, H. (2001). Clinical decision support systems for the practice of evidence-based medicine. *Journal of the American Medical Informatics Association*, *8*(6), 527–534.

Slonim, D., Tamayo, P., Mesirov, J., Golub, T., & Lander, E. (2000). Class prediction and discovery using gene expression data. In *Proceedings of the 4th Annual International Conference on Computational Molecular Biology (RECOMB),* Tokyo, Japan (pp. 263-272). Tokyo, Japan: Universal Academy Press.

Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Berlin, Germany: Springer Verlag.

Wennber, J., & Cooper, M. M. (1999). *The Dartmouth atlas of medical care in the United States: a report on the medicare program*. Chicago, IL: AHA Press.

Simone A. Ludwig joined the Department of Computer Science at North Dakota State University (NDSU) as Associate Professor in Fall 2010. Prior to joining NDSU, she worked at the University of Saskatchewan (Canada), Concordia University (Canada), Cardiff University (UK) and Brunel University (UK). She received her PhD degree and MSc degree with distinction from Brunel University (UK), in 2004 and 2000 respectively. Before starting her academic career Dr. Ludwig worked several years in the software industry. Her research interests include artificial intelligence, evolutionary computation, knowledge engineering, Semantic Grid/Web, and distributed/Grid computing.

Stefanie Roos is currently a graduate student of Darmstadt University, Germany. She started university in Fall 2006, receiving her Bachelor degree in Fall 2009. Stefanie was an exchange student at the University of Saskatchewan, Canada from Fall 2008 to Summer 2009. Her research interests include genetic programming and routing in decentralized networks.

Monique Frize has been a Professor for 20 years, first at University of New Brunswick (1989-1997), then at Carleton University and the University of Ottawa (1997-2010). She currently is Distinguished Professor at Carleton and Professor Emerita at University of Ottawa. Dr. Frize was a biomedical engineer for 18 years in hospitals (1971-1989). She published over 200 papers in journals and conference proceedings on artificial intelligence in medicine, infrared imaging and ethics. She is Senior Member of IEEE, Fellow of the Canadian Academy of Engineering (1992) and of Engineers Canada (2010), Officer of the Order of Canada (1993) and recipient of the 2010 Gold Medal from Professional Engineers Ontario and the Ontario Society of Professional Engineers. She received five honorary doctorates in Canadian universities since 1992. Her book: The Bold and the Brave: A history of women in science and engineering was released by University of Ottawa Press in November 2009.

Nicole Yu graduated with a Bachelor of Science Double Advanced Major in Chemistry and Biology given by Saint Francis Xavier University (Nova Scotia). With a deep interest for learning and a need for a bigger challenge, she proceeded to obtain her Masters of Applied Science in Biomedical Engineering from the Systems and Computer Engineering Department of Carleton University. When she isn't studying, she loves to play sports, snowboard, learn new hobbies and travel the world.