ELSEVIER

Journal of Power Sources 62 (1996) 121–134

JOURNAL OF POWER SOURCES

Prediction of individual cell performance in a long-string lead/acid peak-shaving battery: application of artificial neural networks

Richard E. Young, Xiang Li, S.P. Perone *

Department of Chemistry, San Jose State University, San Jose, CA 95192, USA

Received 23 January 1996; revised 2 April 1996

Abstract

This work represents the culmination of several years of study of an operating large energy storage battery with the purpose of determining if computerized pattern recognition of maintenance data (and/or available fabrication data) could be used for the early detection of poorly performing cells. Also investigated was the possible identification of cells with predicted high performance. Previous studies using k-nearest neighbor pattern recognition have been augmented with the investigation of artificial neural network analysis. Both methods have achieved practical levels of prediction, but the neural network prediction results are somewhat better. It was possible to select 70% of the high-performing cells, without any false selections from the low-performing cells; it was possible to identify nearly 96% of the poor-performance cells, with none of the high-performance cells mis-selected. These results suggest the feasibility of the routine application of neural networks for performance prediction as part of a maintenance strategy for long-string energy storage systems.

Keywords: Lead/acid batteries; Artificial neural networks

1. Introduction

The use of large lead/acid energy storage batteries consisting of hundreds to thousands of cells has been under study as one possible solution for electric power (utility) load management [1]. It is desirable that all cells of a battery have a similar (high) capacity rating to prevent low capacity cells from going into reversal [2]. (Cell reversal is the state in which the electrodes reverse polarity during deep discharge, leading to heating, gassing, and possible irreversible damage.) Hence, it would be beneficial to identify low and high performing cells in advance so that they could be segregated to improve the performance of the battery. Previous studies [3–13] have concerned themselves with identifying these groups of cells by applying the pattern recognition techniques of k-nearest neighbor and non-linear mapping to battery fabrication and maintenance data. Although the results of these studies were encouraging, they were still inadequate for realistic applications. Prediction for the performance of the cells had an overall classification accuracy at best of 73.8% using non-uniform conditions for training and test sets.

The approach used in this study was to determine if a neural network could produce classification accuracies superior to what were achieved in any of the previous studies [9–13], examining easily acquired battery maintenance events. The goal was to achieve classifications such that greater than 90% of the poor performing cells could be removed from the battery, or that greater than 90% of the high performing cells could be selected. The features and neural network parameters that help achieve this goal might provide the ability to select cells for creating a consistently high performance battery with a low possibility of cell reversals, and also might prove instructive as to the physical/chemical properties of low and high performing cells.

1.1. Description of the battery system

This study concerns itself with a lead/acid battery manufactured by GNB, Inc., Kankakee, IL, in June 1983. The 324-cell battery, fabricated according to the Electric Power Research Institute (EPRI) specifications, was capable of delivering 500 kW for a 1-h discharge (1040 Ah cell capacity limit) or 1.2 MWh for a 5-h discharge (2080 Ah cell capacity limit) [3]. The 340 cells produced for this battery were fabricated and tested in five batches: four batches of 80 cells each and a fifth batch of 20 cells. Each cell was numbered with the batches labeled 'circuits' 1 through 5. Detailed records of fabrication materials and measurements were made for each cell. After completion of the initial acceptance tests, the battery was installed (in December 1983) at the Battery

* Corresponding author.

0378-7753/96/$15.00
PII S0378-7753(96)02423-8

Energy Storage Test (BEST) facility in Newark, NJ, in 54 six-cell modules (comprising 324 of the 340 manufactured cells). The performance of over 200 charge/discharge cycles was observed at the BEST facility. Then, in July 1987, the battery was installed at Crescent Electric Membership Corporation (CEMC), Statesville, NC, an area electric power distributor. It has since been operated as a peak-shaving energy storage system at a maximum discharge of 500 kW for 1 h and a minimum discharge of 200 kW for 3 h.

Quarterly maintenance data have been recorded for the GNB battery since being installed at CEMC in 1987. Statistically representative sets of cells were chosen for capacity tests conducted in March 1989 (109 cells) and April 1990 (121 cells). Then, in September 1991, capacity tests were performed on all 324 cells. The GNB battery was still operating at CEMC at the time of completion of the study reported here (May 1995), without any significant number of cell failures. No further capacity data have been obtained since the 1991 study.

1.2. Previous studies predicting and modeling cell performance

The idea of using manufacturer's pre-test results for predicting cell lifetime was first investigated for nickel–cadmium cells using statistical analysis, cluster analysis, and multi-variate pattern recognition techniques [14]. Later, this type of inquiry was extended to lead/acid cells [4–6]. Encouraging results from these studies laid the foundation for extensive investigations into the GNB battery, using these same techniques as applied to fabrication and routine maintenance data, with the objective of cell performance prediction [6–12].

Using fabrication and routine maintenance data for predicting cell performance was considered attractive by the first researchers for several reasons [12,14]. Using the fabrication data might allow one to group cells for specific functions at the outset [14]. If the initial manufacturer's test data were not available, then routine maintenance data might be used for the same predictive functions [12]. Routine maintenance data might also be used for performance prediction rather than periodically conducting expensive capacity tests [11].

1.3. Prior investigations of cell performance prediction for the GNB lead/acid battery

The Perone and Spindler [4,6] study of initial fabrication and test data for lifetime classification analysis of lead/acid golf cart batteries laid the foundation for the fabrication and test plan specified by EPRI for the GNB battery [6]. Data have been collected and analyzed at every stage in the GNB battery's life [10–13]. Three separate studies endeavored to determine whether cell performance could be predicted using different components of these data. A first study [4–9] explored the use of the initial fabrication/test data. A second study investigated the use of routine maintenance data [10–12]. And a third study examined the use of fabrication/test and maintenance data together [13].

1.3.1. Studies of battery fabrication/test data

The fabrication/test data of the GNB battery were examined for features that could predict cell performance [9], where cell capacity was the performance measure. The study consisted of two parts. The first part was to determine whether accurate classifications could be made with two classes (low and high performance cells). Accuracy was determined using leave-one-out k-nearest neighbor (KNN) analysis. An overall accuracy of 92% was achieved on the training set (data for cells of known performance) using several different feature sets. Non-linear mapping (NLM) was used to determine which feature sets, though providing accurate classifications, also produced the best separations in hyper-space. Those features giving good results with both KNN analysis and NLM were considered to contain classification information. A prediction set (data for cells of known performance, but not included in the training set) was not analyzed.

In the second part of the study the cells were classified into three classes (high, low, and medium performance cells). The assumption was made that class divisions could be made based on data having a Gaussian distribution and classes were established with normalized mean and standard deviation units (the mean was set at zero and standard deviation was set at one). Cells with capacity greater than one standard deviation above the average were defined as a high performance class; cells with capacity greater than one standard deviation below the average were defined as a low performance class; and cells with capacity within one standard deviation of the average were defined as a medium performance class. Because of this over-simplified class definition guideline some erroneous classifications were encountered. As a result, NLM was used to re-classify some cells. For the re-classified training set the overall training accuracy achieved with various combinations of features ranged from 78 to 86%. For the individual classes the highest accuracy was 92% for the low, 81% for the medium, and 85% for the high performance cells. As in the previous battery classification studies, the maximum classifications were achieved with different feature sets for each class. A suitable prediction set was not available at the time of Petesch's study [9], but this was evaluated later [13] (see below).

Petesch [9] concluded that, through multi-variate analysis, one could extract from the initial fabrication data information related to cell performance even seven years after manufacture.

1.3.2. Studies of routine maintenance data

The routine maintenance data were examined by Chen [10] and Perone and co-workers [11,12] for features that could predict cell performance. Like the fabrication/test study done by Petesch [9], Chen's study was also done with two classification schemes: classification into two classes (low and high) and three classes (low, medium, and high).

Table 1
Best performance prediction results with combined feature sets, KNN pattern recognition, nonlinear mapping-optimized class distributions [13]

Accuracy (%) [b]	False Positive	% correct of selected	Features [a]	
			Maintenance	Fabrication
a) *Class 1*				
91.7	6	64.7	Water level trend	Dry weight; average specific gravity after discharge; maximum capacity first five cycles
58.3	3	70.0	Cell voltage trend; specific gravity trend; average cell voltage over all events	Acid added during formation and before shipping
50.0	2	75.0	Cell voltage trend; average cell voltage over all events	Dry weight; acid added during formation and before shipping
b) *Class 2*				
76.9	2	83.3	Water level trend	Dry weight; Average specific gravity after discharge; maximum capacity from first five cycles
84.6	3	78.6	Average water level over all events	Acid added during and before shipping (relative to total acid); average specific gravities before and after five test cycles; maximum capacity from first five cycles

[a] Maintenance events, 1989; capacity test, 1990.
[b] 42 total cells: 12 class 1 (high capacity); 13 class 2 (low capacity); 17 class 3 (intermediate).

Class divisions were defined, as in Petesch's study, by normalized standard deviations. Chen's study demonstrated the predictive power of the maintenance data using non-traditional data indexing. Several indexes were tried, but a 'battery activity' index proved the most useful. This index was based on the total volume of water required to be added to the battery each quarter for one year prior to a capacity test. The amount of water required was determined to reflect the overall activity of the battery during a given quarter. A training set consisted of maintenance data collected for one-year prior to a given capacity test event. For the two-class training set, several features gave 100% overall accuracy. For the two-class prediction set, several sets of three to five features gave around 80% overall accuracy. For the three-class training sets, the four best features gave 76%–88% overall accuracy with the high performance cells providing nearly 100% accuracy. However, no meaningful results were realized with the prediction set.

1.3.3. *Studies of combined fabrication/test and maintenance data*

Though it was apparent that fabrication/test data and routine maintenance data contained information useful in predicting cell performance, individually, they were inadequate to make practical predictions when using the KNN and the NLM techniques. A study was, therefore, performed to explore the simultaneous use of the fabrication/test data set and the routine maintenance data. The results were presented by Li [13].

Three classes were looked at in Li's study (low, medium, and high performance cells). Class boundaries were defined as before [9–12] by normalized standard deviations. KNN analysis was used for developing classification clusters, and NLM was used for fine tuning the classifications. The leave-one-out KNN technique was used for determining classification accuracy. Two sets of training and prediction sets were created from the maintenance data as follows:

Training set	Prediction set
Prior to March 1989	Prior to April 1990
Capacity test	Capacity test
Prior to April 1990	Prior to September 1990
Capacity test	Capacity test

The combined fabrication/maintenance data yielded an overall prediction accuracy of 60%. However, the major breakthrough came with the reworking of the three-class problem. That is, the three-class problem was redefined as a series of three two-class problems: low versus medium/high; medium versus low/high, and high versus low/medium. This adaptation allowed higher individual classification accuracies to be achieved than was possible before on prediction sets. Maximum prediction accuracies were 85% for the low performance cells, 65% for the medium performance cells, and 92% for the high performance cells. Different feature sets were optimum for each of the three prediction objectives.

A more definitive evaluation of prediction ability must take into account the occurrences of false positives. When this factor is considered, the most effective prediction procedures using the combined feature set established by Li's work are summarized in Table 1. Note that several different results are provided. The 'best' choice depends upon specific objectives. For example, if the objective is to obtain a set of high performing cells with the fewest possible mistakes, the example in Table 1 where only 50% of class 1 cells are identified would be the best choice, because the highest accuracy (75%) for the cells selected would be achieved. For class 2 (low performing) cells, a large percentage (84.6%) could be identified, but nearly 12% of those selected would not be class 2 cells.

From Li's work [13], it was clear that the use of a feature set combining initial fabrication and recent maintenance data could predict cell performance at a high enough level of accuracy to merit consideration for practical management of long-string battery energy storage systems. However, the original objective of this work was to evaluate the effectiveness of maintenance data alone for performance prediction, because cell fabrication/test data are rarely available with the quality and detail provided for the GNB battery studied here. Thus, the work reported below considered only maintenance data.

1.4. Neural networks for cell performance prediction from maintenance data

Routine battery maintenance data were evaluated for performance prediction, using artificial neural networks (ANN) as a pattern recognition tool. Because of the fundamental differences in ANN principles, and the ability to handle nonlinear relationships, it was anticipated that significant improvement might be obtained compared with conventional pattern recognition methods.

1.4.1. Introduction to neural networks

A neural network is a mathematical modeling procedure originally thought of as mimicking the operation of neurons in the human brain [15]. Implemented on a computer, a neural network maps between two sets for purposes of classifying, predicting, pattern recognition, or other specialized processing (such as signal analysis). A neural network gains its aptitude by encoding patterns into the activation levels of a system of parallel distributed information processors [16,17]. Most importantly, they 'learn' by exposure to examples. This is one of the major advantages of using a neural network. That is, rather than devising complicated models, one presents the neural network with plentiful and representative examples, and it will extract its own model [16,18].

Neural networks fall in the same category as other multivariate techniques such as linear discriminant, KNN, machine learning, and statistical least-squares techniques [19]. However, neural networks have more capabilities than any of the other techniques. They are nonlinear [20], provide more functional forms [20], and are nonparametric [15]. Because they are nonparametric, assumptions about the data fitting a particular density function are not made [19]. Thus, in circumstances where theoretical, analytical, or numeric solutions are inadequate (e.g. the relationships between features are unknown), a neural network may be able to associate many obscurely interrelated variables into a usable multidimensional mapping [18,21].

1.4.2. Neural network architecture

A neural network consists of a number of distinct layers each with various numbers of mathematical neurons (also known as processing elements, nodes, or units). The overall structure of a neural network is illustrated in Fig. 1 and an individual processing element is portrayed in Fig. 2.

First, in neural network architecture, there is an input layer. The number of processing elements in this layer corresponds to the number of inputs. Each processing element in this layer receives only one input from outside of the network. Each node in the input layer fans out its input, without modification, to each processing element in the next [22]. Each transfer of output from one neuron to the input of another neuron is

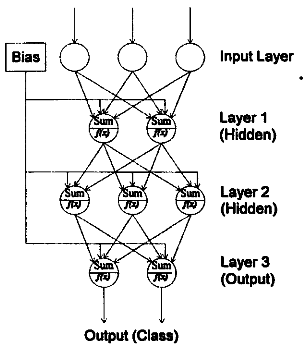

Fig. 1. Structure of a feed forward back propagation neural network consisting of three layers (the input layer is not counted).

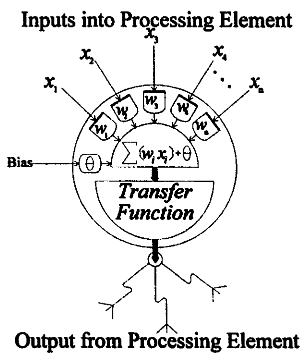

Fig. 2. Structure of an individual processing element. The inputs (x) are multiplied by weight factors (w), summed, transformed with a transfer function, and then distributed to other processing elements.

called a connection [23]. The next layer is called a hidden layer. There may be one or several hidden layers. Each processing element in the first hidden layer receives input from each input layer node. Each input in each processing element is multiplied by a separate weight factor. These products are then summed, scaled, and combined with a bias factor (Fig. 2). The product of a scaling factor and a bias factor are then added. The internal activation is then fed through a transfer function (called a squashing function by some) that effects a nonlinear transformation. Two such functions commonly used in neural networks are the sigmoid function and the hyperbolic tangent function.

The output from each node's transfer function is then sent to nodes in a successive hidden layer, or to the output layer if there are no more hidden layers.

1.4.3. Neural network development

There are three phases in neural network development. In the first phase the neural network is trained in a process called 'supervised learning'. A training set of data is presented to the neural network causing the weights in each processing element, initially set to small random numbers, to be modified to minimize the difference between the actual outputs and the desired outputs. Desired outputs are defined to reflect the known true class of each item represented by each pattern. When an individual pattern is presented to the neural network this is called a 'training cycle'. When the network adjusts itself to minimize error this is termed an 'epoch'. An epoch may occur after only one or after many training cycles.

In the second phase, the training set is taken through the trained network, but without the weights being adjusted. This 'recall' phase compares the output values to the correct values of the training data allowing one to determine how well the neural network learned the training set. Poor results mean either that the network has not been properly trained, or that there is a problem with the data. On the other hand, obtaining good results with the training set still does not necessarily mean that the network was properly trained. The network must be evaluated with a test set (prediction set).

In the 'test' phase the network is presented with patterns (test or prediction set) of the same origin as the training set which were not encountered during training. This procedure determines how well the network can interpolate for patterns it has not seen before. These three phases are repeated numerous times with adjustments made to the neural network between each set of phases by the user to try to improve the performance for the next set of training, recall, and testing. Finally, after the network's performance has met the desired criteria of success defined by the user, it is deployed with real world data where the outcomes are unknown.

2. Experimental

2.1. Hardware and software

An IBM/PC compatible 486-DX2/66MHz computer with 8 Mbyte RAM, 512 Kbyte SRAM cache, and 550 Mbyte hard disk drive were used for development of the neural networks and database management.

Excel® (version 5.0; Microsoft Corporation, Redmond, WA) was used to manage the databases. Management of the databases included importing files previously created using SYMPHONY™ (Lotus Corporation, Cambridge, MA), reorganizing and editing the data, pre-processing of the data using mathematical and statistical functions, and exporting the data in a format compatible with the neural network software. The neural networks were developed using NeuralWorks Professional II/PLUS (version 5.0; NeuralWare, Inc., Pittsburgh, PA). All software was executed under Windows™ (version 3.11; Microsoft Corporation) with MS-DOS® (version 6.2; Microsoft Corporation).

2.2. Raw database

The raw database contained all of the unprocessed maintenance data. This included quarterly maintenance task data (float voltages, specific gravities, water additions, and electrolyte levels) for each of the 324 cells from August 1987 through September 1991 and the results of all of the capacity tests done in March 1989, April 1990, and September 1991. Capacity test data consisted of the results of capacity tests for 109 cells from March 1989, 121 cells from April 1990, and 323 cells from September 1991.

2.3. Definition of class boundaries

Cell capacity, expressed as a percentage of the nominal value (2080 Ah), was the figure-of-merit used to distinguish how well a cell performed. The assumption was made that class divisions could be made based on their having a Gaussian distribution. Classes were established based on their normalized mean and standard deviation units (the normalized mean was zero and standard deviation was one). Cells with capacity greater than one standard deviation above the average were defined as a high performance class (class 1); cells with capacity greater than one standard deviation below the average were defined as a low performance class (class 2), and cells with capacity within one standard deviation of the average were defined as a medium performance class (class 3). Because there were many more members of class 3 than either of the other classes, class 3 cells represented in the data base were selected randomly, with the total number approximately equal to those of classes 1 or 2. A class assignment must be associated with each pattern in the feature database for supervised learning.

2.4. Feature database

The feature database was derived from the raw database and contained features which were hoped would prove useful in classifying the cells. These features included transformations and combinations of the data values found in the raw database. For example, prior to a capacity test several quarters

Table 2
Time indexing of raw database using 'months until capacity test'

Capacity test event	Maintenance event (t_1)	Maintenance event (t_2)	Maintenance event (t_3)	Maintenance event (t_4)
May 1989	February 1989	November 1988	August 1988	February 1988
April 1990	May 1990	November 1989	August 1989	May 1989
September 1991	September 1991	March 1991	November 1990	August 1990

Table 3
Neural network input codes for capacity test event features

Feature description	Float voltage	Specific gravity	Water addition	Electrolyte level
t_4 (see Table 1)	2	16	30	44
t_3 (see Table 1)	3	17	31	45
t_2 (see Table 1)	4	18	32	46
t_1 (see Table 1)	5	19	33	47
mean of (t_1, t_2, t_3, t_4)	6	20	34	48
$t_1 \times t_2 \times t_3 \times t_4$	7	21	35	49
slope of (t_1, t_2, t_3, t_4)	8	22	36	50
correlation coefficient of (t_1, t_2, t_3, t_4)	9	23	37	51
$t_1 - t_2$	10	24	38	52
$t_1 - t_3$	11	25	39	53
$t_1 - t_4$	12	26	40	54
$t_2 - t_3$	13	27	41	55
$t_2 - t_4$	14	28	42	56
$t_3 - t_4$	15	29	43	57

of float voltage readings were taken for each cell. The combination of several float voltages, transformed through linear regression, form a slope and a correlation coefficient which may demonstrate a trend. The slope and correlation coefficient would hence be elements in the feature database.

The raw database was reorganized based on a time index relative to the capacity test dates. Four comparably time-spaced maintenance events during the year prior to a capacity test were identified for each capacity test date. Time-until-capacity-test was the index for each maintenance event and each index unit was designated 't_1, t_2, t_3, or t_4', where t_1 refers to the maintenance event closest in time to the capacity test date. Table 2 shows the time index used for maintenance events associated with each capacity test event. Note that in one case the maintenance event assigned to index t_1 occurred after the capacity test (April–May 1990). This is acceptable for purposes of retrospective training and prediction, as there should be no change in capacity distribution by a maintenance event. Each maintenance event had associated with it the maintenance tasks of float voltages, specific gravities, water additions, and electrolyte levels. Additional features were also generated. These included combinations and relational transformations of the time indexed features associated with each capacity test event from Table 2. A total of 56 features were established for each cell for each capacity test event. Table 3 defines these features and indicates their neural network input code number. Each of the cells associated with a capacity test event in the feature database was assigned a class in the feature database.

The feature database also contained class assignments associated with each pattern. During the training phase the class assignment data fields are used for supervised training. During the recall and test phases the class assignments are used to gauge the performance of a neural network by comparing the patterns' actual classes to the neural network's class assignment of them. Hence only data for those cells associated with a capacity test event were retained for the feature database.

2.5. Data pre-processing

Inherent to the raw data were major variations in magnitude and scale. This included variations from feature to feature, as well as from event to event. The incidental variations in scale were eliminated by transforming the features via normalization according to Eq. (1)

$$(XN_j)_i = (X_j)_i / (X_{\mathrm{mean}})_j \tag{1}$$

where for the jth feature of the ith cell, $(XN_j)_i$ is the normalized value of the raw data $(X_j)_i$. $(X_{\mathrm{mean}})_j$ is the mean value of the jth feature for all cells of a particular maintenance event. This form of normalization retains the relative differences in ranges, which can be useful for transformed features which define trends over several events.

When pattern recognition methods are applied, however, it is desired to eliminate completely the confounding effects of varying magnitudes of standard deviations between features. Thus, each feature was autoscaled according to Eq. (2)

$$(XS_j)_i = [(XN_j)_i - (XN_{\mathrm{mean}})_j] / (XNSD)_j \tag{2}$$

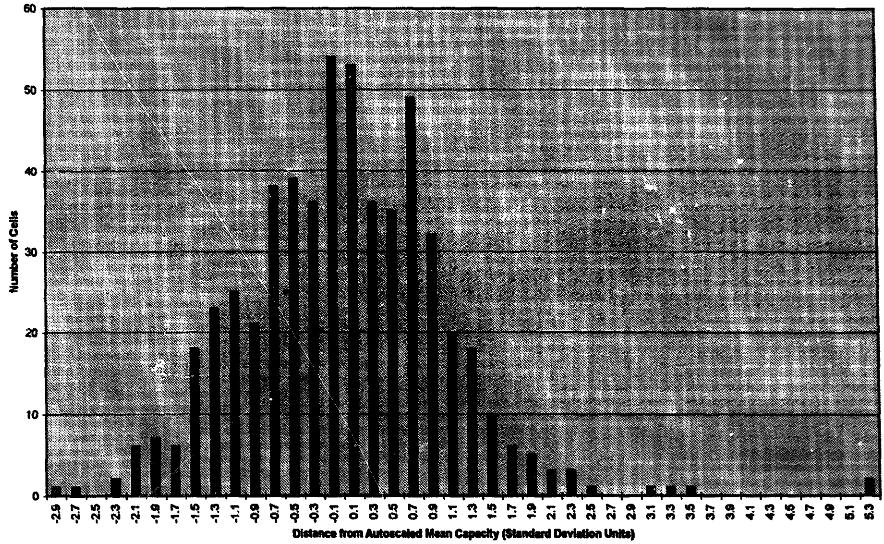

Fig. 3. Distribution of capacity among the cells at CEMC from the combined 1989, 1990, and 1991 capacity tests.

where for the jth feature of the ith cell, $(XS_j)_i$ is the autoscaled value of the normalized data $(XN_j)_i$ generated in Eq. (1). The normalized mean value, $(XN_{mean})_j$, is from Eq. (1), and $(XNSD)_j$ is the sample standard deviation of the normalized jth feature for all cells of a particular maintenance event.

2.6. Training and test sets

The choice of a training set is probably the most important aspect for success of a neural network project. Unlike classical statistical methods in which the number of samples necessary to obtain statistically significant results can be determined, with neural networks this number can only be estimated [24]. Typically the training set needs hundreds to thousands of examples so that an adequate number of patterns can be represented and learned [21,25,26]. In many instances the number of examples required is dependent upon the type of data. For example, a small data set can be justified as long as the training data set is relatively free of noisy data or idiosyncratic examples. The fewer the number of cases in the training set the less eccentricity is acceptable [27]. In this way the data are more uniform and less susceptible to outlier patterns disrupting the learning process.

Generally, all of the data are combined and split with 75–80% of the combined data forming the training set and the remaining 20–25% forming the test set [28]. In our study, all feature sets for each cell monitored in the three capacity test events were combined and their order randomized. (This was a significant procedural departure from our previous studies of the GNB battery data [9–16].) For the training set the first 75% of the combined and randomized data was chosen. The test set comprised the remaining 25% of the data. This produced a large enough training set to justify the use of a neural network. The randomization of the events between 1989, 1990, and 1991 provided a foundation for a generalized extrapolation of the results over time. Fig. 3 illustrates the Gaussian-like distribution of the combined capacities normalized and autoscaled.

2.7. Feature selection

For neural networks, highly correlated input features are not troublesome [29]. However, as with statistical methods, deletion of variables with insignificant consequence on the output improves the effectiveness of the modeling [30]. Hence it is advantageous to reduce the number of inputs to optimize neural network performance.

If the weights associated with a particular input node are all small then that input, relative to the other inputs, has little impact on the solution obtained by the network [31]. A Hinton diagram [32] allows one to determine which inputs have relatively little impact, by portraying an x–y matrix of rectangular boxes, representing all intersections of nodes in the network. The magnitudes of weight-connecting nodes are indicated by the size and color of each rectangular box. Feature selection can be done by training a neural network to a satisfactory level with the input of many features, examining the Hinton diagram, and then eliminating those features associated with inputs that add little to the solution. The network is then re-trained with the remaining features. The elimination process is repeated until the performance of the neural network begins to deteriorate. In this way the strongest features are retained and the performance of the network can be optimized.

2.8. Measuring neural network performance

In order to judge the performance of a neural network one must be able to quantify its performance such that it can be compared with other neural networks and other classification techniques. Calculating classification accuracies, constructing confusion matrices, and determining risk factors are tools that provide different types of information about the classification performance of a classifier. Taken together they form a good picture of a classifier's capabilities.

2.8.1. Classification accuracy

A classification accuracy quantifies how well a classifier correctly identifies the actual class. It can be calculated either as an overall accuracy or as a class specific accuracy

$$\text{Overall classification accuracy} = (C_t/J)\,100\% \quad (3)$$

$$\text{Class specific classification accuracy} = (C_M/J_M)\,100\% \quad (4)$$

where C_t is the total number of correct classifications in the whole set of J patterns, and C_M is the number of correctly classified cases of class M containing J_M patterns.

2.8.2. Confusion matrix

A more powerful tool for evaluating the performance of a classifier is a confusion matrix. A confusion matrix provides much more information than the classification accuracies of Eqs. (3) and (4) because it not only conveys the percentages of correct classifications but also the types and percentages of mis-classifications. A confusion matrix is a table consisting of column headers indicating the true class assignments and row labels indicating the neural network class assignments. At the intersection of a row and column is the percentage of cases of a particular class that the neural network assigned to the class corresponding to that row. In the diagonal running from top left to bottom right are the class specific classification accuracies of Eq. (4). The other positions on the table indicate false positives. For example, Table 4 depicts a confusion matrix in which 86% of true class 1 cases were correctly classified, but 7% of the class 2 cases and 22% of the class 3 cases were mis-classified as class 1. Ideally then, if a neural network classified all of the cases correctly, there would be 100's running in a diagonal from top left to bottom right of the table with zeros everywhere else

Table 4
Example of a three-class confusion matrix [a]

Neural network identified class	True class		
	Class 1	Class 2	Class 3
Class 1	86	7	22
Class 2	4	90	3
Class 3	10	3	75

[a] Values signify the percentages of cases of the class indicated by the column that were classified by the neural network as belonging to the class indicated by the row.

3. Results and discussion

3.1. Preliminary investigation

Preliminary experiments established some fundamental neural network parameters which appeared to work best for the battery data sets. These parameters were: (i) 'extended delta-bar-delta' learning rule [33]; (ii) hyperbolic tangent transfer function [34] for hidden nodes; (iii) the NeuralWare® 'softmax output' transfer function [35] for output layer nodes, and (iv) input scaling between [−1, +1].

In the first round of the feature selection procedure all 56 of the features were input into a set of ten neural networks, each with a single hidden layer containing from one to ten hidden nodes. To gauge each neural network's performance, the test set's overall classification accuracy as well as class-specific accuracy were monitored. The training set's classification accuracies were useful only as confirmation that the test set's results were not fortuitous. The primary objective was to achieve the highest possible accuracies for classes 1 and 2, with the lowest confusion among these two classes. The Hinton diagram was used to eliminate the weaker features. At various points in the feature selection process, the stronger of the features which had previously been eliminated were reintroduced to see their effect. This cycle of feature reduction was repeated several times until a point was reached where deterioration of overall classification accuracy for the test set began to occur when additional features were eliminated.

As a result of the feature selection investigation, the single-hidden-layer neural networks with one hidden node proved to be adequate, with more hidden nodes not improving the overall classification accuracy. The neural network input codes for 13 features which appeared most significant were 2, 5, 6, 7, 10, 14, 17, 20, 21, 30, 41, 42, and 43 (see Table 3). Of these, input codes 2, 5, 6, 7, 10, 14, and 17 stood out as more important than input codes 20, 21, 30, 41, 42, and 43. An examination of Table 3 reveals that of the eight most important of the selected features, seven derive from the float voltages (input codes 2, 5, 6, 7, 10, and 14) and the other is the specific gravity feature t_3 (input code 17). The least important selected features all derive from the water added maintenance task. None of the selected features were derived from the electrolyte level maintenance task. The reasons for the relative importance of the features will be discussed later.

3.2. Neural network optimization

The other aspects of optimized neural network design to be explored included: the overall architecture (number of

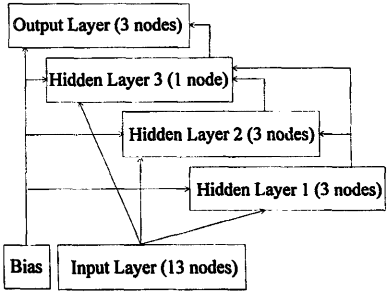

Fig. 4. Architecture determined to produce the best performing neural networks for this study.

hidden layers, number of hidden nodes within hidden layers, and their interconnections); variations in epoch length, and the input scaling ranges. A systematic study was conducted to optimize the basic architecture. The number of hidden layers was varied from one to three, with the number of nodes in each varying from one to four. Connections between layers were tried with only adjacent layers connected and with prior layers feeding into some or all subsequent layers. The outcome of this investigation yielded an architecture, depicted in Fig. 4, that produced promising results. The inputs were the selected features described earlier. The bias was input to all nodes in all layers. There were three hidden layers. The first hidden layer consisted of three nodes with inputs from all of the input layer nodes except nodes 20 and 21 (see Table 3). The second hidden layer consisted of three hidden nodes with inputs from the first hidden layer and all nodes in the input layer. The third hidden layer consisted of one node with inputs from all nodes in the input layer, and the first and second hidden layers. Each hidden layer node used the hyperbolic tangent transfer function. The output layer consisted of three nodes with the only input to each being from the third hidden layer node (and the bias).

Having an adequate architecture in hand, the neural network performance was fine tuned by systematically investigating the epoch size and input scaling ranges. Epochs were varied from 5 to 175 training cycles. Input ranges had the form of [− 1, + 1] or [0, 1] in which ranges within these forms included such variations as [−0.8, +0.8] and [0.2, 0.8]. Variations of these parameters produced neural networks that were more specialized in their abilities and are discussed later.

3.3. Three-class optimization

Since the problem was defined in terms of three classes (high (class 1), low (class 2), and medium (class 3) performance), the most straightforward approach was to create a three-class optimized neural network. However, since not all classifications and mis-classifications have the same significance, the neural network's confusion matrices must be interpreted with this in mind. Table 5 presents the confusion matrices for several neural networks that produced the best classifications for three-class optimization.

Inspection of the confusion matrices in Table 5 reveals that the manner in which mis-classifications occurred was not by chance, but that the neural networks were actually finding decision regions based on the input patterns. Classes 1 and 2 were separated from one another more than either were from

Table 5
Selected confusion matrices (percent classified), overall percent classification accuracies for 3-class neural network test and training sets [a]

NN ID [b]	PC [c]	Test set Confusion matrix (%) True class				Training set Confusion matrix (%) True class			
		Class 1	Class 2	Class 3	OCA [d]	Class 1	Class 2	Class 3	OCA [d]
2	1	75.0	0.0	21.1	77.4	88.2	3.0	27.7	72.0
	2	0.0	95.7	21.1		0.0	84.9	26.2	
	3	25.0	4.3	57.8		11.8	12.1	46.1	
3	1	80.0	0.0	21.1	77.4	94.1	6.1	27.7	74.7
	2	0.0	91.3	21.1		0.0	84.8	23.1	
	3	20.0	8.7	57.8		5.9	9.1	49.2	
4	1	80.0	4.3	21.1	79.0	88.2	7.6	35.4	69.2
	2	0.0	95.7	21.1		0.0	83.3	24.6	
	3	20.0	0.0	57.8		11.8	9.1	40.0	
7	1	100	8.7	31.6	80.6	100	9.1	49.2	68.1
	2	0	87.0	15.8		0	83.3	23.1	
	3	0	4.4	52.6		0	7.6	27.7	

[a] Neural network parameters: (2) 14900 training cycles, epoch = 65, [−0.95, +0.95] input scaling; (3) 15100 training cycles, epoch = 51, [− 1, + 1] input scaling; (4) 9300 training cycles, epoch = 50, [− 1.2, + 1.2] input scaling; (7) 9300 training cycles, epoch = 25, [−0.9, +0.9] input scaling.
[b] NN ID: Neural network identification.
[c] PC: Neural network predicted class.
[d] OCA: Overall classification accuracy.

class 3 as indicated by the percentages of mis-classifications. That is, most mis-classifications occurred between class 3 and the other classes.

Examining the confusion matrices and overall percent classification accuracies in Table 5 reveals that the training set typically had lower values than the test set. This is due to the NeuralWorks Professional II/PLUS® 'Save Best' function, which uses the performance of the test set and not the training set as the criterion for saving the neural network to disk. At a certain number of training cycles over-training begins to occur: the test set's results become poorer as the training set's results begin to improve. In general, to have confidence in the overall performance of a neural network on unknown patterns, it is necessary for the training set to have comparable or better results than the test set. Neural network 3 in Table 5 meets this criterion the best.

3.4. Two-class optimization

An alternative classification approach involving a two-class distribution, originally studied by Li [13], was also explored for neural network analysis. This consisted of breaking the overall classification problem into two. For one problem, class 1 cells were to be separated from a class consisting of both class 2 and class 3 cells. The other problem was to separate class 2 cells from a class consisting of both class 1 and class 3 cells.

The same architecture and inputs as used in the three-class optimization were used in these experiments. However, the output layer only consisted of one node, since this is all that is required in a two-class problem.

3.4.1. Class 1 versus classes 2 and 3

The results from this study exhibited a correlation between high classification accuracy for class 1 and the number of mis-classifications from classes 2 and 3. This was the same trend observed for the three-class study. The difference here is that information is lost regarding what cells are being mis-classified as class 1. In the three-class evaluation relatively high amounts of mis-classifications by the other two classes as class 1 may be acceptable if, for example, the amount of class 2 cells identified as class 1 is very small. However, by grouping class 2 and 3 together, the origins of the mis-classifications are unknown making the results less informative than those from the three-class study.

3.4.2. Class 2 versus classes 1 and 3

The results of this evaluation were more promising than the preceding study. Table 6 presents the results of the two best neural networks. For neural network number 14 potentially 92.3% of the class 2 cells could be identified and removed from a battery with 13.0% of non-class-2 cells being removed. The training set's results corresponded fairly well to the test set's results, indicating that the neural network would probably apply in a general manner. However, these results are still not as good as those achieved in the three-class study. This is because roughly twice as many cells belong to the combined classes 1 and 3 as belong to class 2. The 13% of non-class-2 cells of neural network number 14 (Table 6) is, in actual number of cells, larger than those indicated by the 15.8% in neural network number 7 (Table 5). Thus the actual numbers of classifications and mis-classifications should be considered when comparing results with Tables 5 and 6. (The needed breakdowns are provided in Tables 7 and 8.)

3.5. Single-class optimization within a three-class neural network classifier

Another strategy was explored to maximize the classification accuracies. The approach taken was to create neural networks which specialized in classifying one class. The neural network would concentrate on achieving the highest classification accuracy for one class at a time with the fewest number of false positives from the other two classes. The classification accuracy of the other two classes relative to one

Table 6
Confusion matrices (percent classified) and overall percent classification accuracies for 2-class neural network test and training sets: class 2 versus classes 1 and 3 [a]

NN ID [b]	PC [c]	Test set Confusion matrix (%) True class			Training set Confusion matrix (%) True class		
		2	1 and 3	OCA [d]	2	1 and 3	OCA [d]
13	2	79.5	8.7	87.1	76.7	12.1	84.1
	1 & 3	20.5	91.3		23.3	87.8	
14	2	92.3	13.0	88.7	90.5	18.2	85.2
	1 & 3	7.7	87.0		9.5	81.8	

[a] Neural network parameters: (13) 18000 training cycles, epoch = 15, [−1, +1] input scaling; (14) 20565 training cycles, epoch = 15, [−0.8, +0.8] input scaling.
[b] NN ID: Neural network identification.
[c] PC: Neural network predicted class.
[d] OCA: Overall classification accuracy.

Table 7
Origin and number of cells forming the combined and randomized training set

Capacity test event	Number of class 1 cells	Number of class 2 cells	Number of class 3 cells
March 1989	13	7	15
April 1990	12	20	15
September 1991	26	39	35
Combined/randomized training set	51	66	65

Table 8
Origin and number of cells forming the combined and randomized test set

Capacity test event	Number of class 1 cells	Number of class 2 cells	Number of class 3 cells
March 1989	4	8	3
April 1990	5	4	6
September 1991	11	11	10
Combined/randomized test set	20	23	19

Table 9
Confusion matrices (percent classified), overall percent classification accuracies for case 1 neural network test and training sets [a]

NN ID [b]	PC [c]	Test set Confusion matrix (%) True class				Training set Confusion matrix (%) True class			
		Class 1	Class 2	Class 3	OCA [d]	Class 1	Class 2	Class 3	OCA [d]
15	1	70.0	0.0	0.0	75.8	66.7	1.5	10.8	73.1
	2	5.0	82.6	26.3		0.0	83.3	23.1	
	3	20.0	17.4	73.7		33.3	12.1	64.6	
16	1	75.0	0.0	5.3	74.2	78.4	4.5	13.8	73.6
	2	5.0	78.3	21.1		0.0	84.8	27.7	
	3	20.0	21.7	68.4		21.6	10.6	58.5	
17	1	80.0	4.4	5.3	67.7	94.1	4.6	15.4	72.0
	2	10.0	69.6	42.1		0.0	74.2	32.3	
	3	10.0	26.1	52.6		5.9	21.2	52.3	
18	1	90.0	4.4	21.1	77.4	90.2	9.1	36.9	68.1
	2	0.0	82.6	21.1		0.0	80.3	24.6	
	3	10.0	13.0	57.9		9.8	10.6	38.5	
19	1	95.0	4.4	31.6	72.6	98.0	4.6	49.2	68.1
	2	0.0	73.9	21.1		0.0	89.4	27.7	
	3	5.0	21.7	47.4		2.0	6.1	23.1	
20 [e]	1	100	8.7	31.6	80.6	100	9.1	49.2	68.1
	2	0	87.0	15.8		0	83.3	23.1	
	3	0	4.4	52.6		0	7.6	27.7	

[a] Neural network parameters: (15) 15600 training cycles, epoch = 40, [−1, +1] input scaling; (16) 15600 training cycles, epoch = 35, [−1, +1] input scaling; (17) 17800 training cycles, epoch = 20, [−1, +1] input scaling; (18) 12300 training cycles, epoch = 25, [−1.1, +1.1] input scaling; (19) 8600 training cycles, epoch = 20, [−1, +1] input scaling, input nodes 41 and 43 not fed into first hidden layer; (20) 9300 training cycles, epoch = 25, [−0.9, +0.9] input scaling.
[b] NN ID: Neural network identification.
[c] PC: Neural network predicted class.
[d] OCA: Overall classification accuracy.
[e] This neural network is identical to neural network 7 in Table 5.

another would be inconsequential. However, since the goal of this investigation was to eliminate low capacity cells (class 2) and accurately separate out high capacity cells (class 1), class 3 was not of importance except as a source of false positives for class 1. The technique was used with some success by Li [13] in her classification study for this battery using maintenance events in connection with the fabrication data (see Table 1).

The classification problem was broken up into two cases. The case 1 objectives were to maximize class 1 classification accuracy but minimize the number of false positives from classes 2 and 3. The case 2 objectives were to maximize class 2 classification accuracy but minimize the number of class 2 cells identified as class 1, and classes 1 and 3 cells identified as class 2. In this work these objectives were accomplished by operator supervision of the training of different neural

Table 10
Confusion matrices (percent classified), overall percent classification accuracies for case 2 neural network test and training sets [a]

NN ID [b]	PC [c]	Test set Confusion matrix (%) True class				Training set Confusion matrix (%) True class			
		Class 1	Class 2	Class 3	OCA [d]	Class 1	Class 2	Class 3	OCA [d]
21	1	60.0	0.0	10.5	74.2	76.5	1.5	18.5	73.1
	2	0.0	87.0	15.8		0.0	86.4	24.6	
	3	40.0	13.0	73.7		23.5	12.1	56.9	
22	1	70.0	0.0	15.8	75.8	80.4	3.0	20.0	74.2
	2	0.0	91.3	21.1		0.0	84.8	21.5	
	3	30.0	8.7	63.2		19.6	12.1	58.5	
23	1	75.0	0.0	21.1	77.4	88.2	3.0	27.7	72.0
	2	0.0	95.7	21.1		0.0	84.9	26.2	
	3	25.0	4.4	57.9		11.8	12.1	46.2	

[a] Neural network parameters: (21) 9400 training cycles, epoch = 70, [−1, +1] input scaling; (22) 15720 training cycles, epoch = 125, [−1, +1] input scaling; (23) 14900 training cycles, epoch = 20, [−0.95, +0.95] input scaling.
[b] NN ID: Neural network identification.
[c] PC: Neural network predicted class.
[d] OCA: overall classification accuracy.

networks. Those most suitable for each of the objectives above were selected out for further study.

Table 9 presents a graduation of neural networks for case 1 ranked in order of increasing tolerance for mis-classifications of class 2 and 3 cells. That is, a neural network with the ability to identify a larger number of the high performance cells would require a tolerance for a larger portion of lower performing cells to be included. Examination of the confusion matrices for neural network 17 shows the highest consistency between the training and test sets. Hence, neural network 17, among all of those in Table 9, could probably be deployed in the real world with the most overall confidence. However, if a high tolerance for class 3 cells was acceptable (to such a degree that you may have more class 3 cells than your neural network indicates) then neural networks 19 and 20 could be deployed with confidence since the classification accuracies for all but class 3 correspond well in both the training and test sets.

Table 10 presents the best neural networks specializing in case 2. Based on the how well the confusion matrices correspond in both the training and test sets, neural network 21 seems the best choice to be deployed with the most confidence in the real world. Although neural networks 22 and 23 both have higher test set classification accuracies for class 2, the corresponding training set class 2 classification accuracies do not match well. However, neural networks 22 and 23 show great potential for becoming optimum solutions for the case 2 problem.

In an attempt to identify a consistent set of high performing cells, those cells identified as class 2 (by class 2 specialist neural network 23) were first removed from the test set, and the remaining cells were classified by the class 1 specialist neural network 20. The resulting 'enhanced' confusion matrix (shown below) illustrates a significant improvement by diminishing the number of class 2 cells mis-classified as class 1.

Enhanced confusion matrix (%), neural nets Nos. 20 and 23

Predicted class	True class		
	Class 1	Class 2	Class 3
1	100	4.3	31.6
2	0.0	95.7	21.1
3	0.0	0.0	47.4

3.6. Interpreting the neural network

A major disadvantage that neural networks have, which many multi-variate statistical techniques do not, is interpretation. In least-squares analysis, for example, the slope, intercept, and sign of the correlation coefficient may all have interpretative significance. However, for a neural network, the interpretative significance of the interconnections between nodes and the associated matrix of weights is quite abstract and difficult to explain [36]. However, various aspects of the nature of the input data can be inferred based on what parameters optimized the neural network's performance.

The reason that various features proved more useful than others is believed to lie in the quality of the data from each maintenance task. The float voltages were obtained by well trained technicians under supervision. The specific gravity readings were obtained with a rugged and standardized technique using a hydrometer. The water added task was not recorded as accurately as the float voltages or the specific gravities. Chen and co-workers [10,12] found that features derived from the water added task may potentially be most significant because it reflected the total activity of the cell.

Thus, despite the lower quality of the water added records, its cell performance information content is demonstrated by its being the generator of several of the thirteen features in the selected feature set (see above, Preliminary Investigation section). The electrolyte level readings did not have the precision and consistency as the other maintenance data, and thus may not provide as fine an indicator as needed by a study of this type.

For neural networks, training data appear to resonate at particular epochs. Training with the wrong epoch size can inhibit convergence to an optimum solution. For example, training with too small of an epoch may cause oscillations in the weights. Training with too large of an epoch may cause subtle trends to be lost [37]. Overall best performance tended to occur with epochs between 15 to 65 training cycles. However, in the specialist neural networks in which a single class was optimized within a three-class classifier, the best class 1 classifiers had short epochs and the best class 2 classifiers had long epochs (compare Tables 9 and 10). This indicates that subtle trends in the features may be key to correctly classifying the high performing cells. On the other hand, overall trends in the features seem to contain information related to classifying poor performing cells.

Beyond any preprocessing the user does to the data, the neural network does a linear mapping of the input data within user specified ranges. Ranges such as [0, 1] or [−1, 1] are typical. This mapping allows the network to work with numbers that are within the ranges compatible with the summation and transfer functions of the processing elements. When the inputs are scaled between zero and one the average is 0.5. This type of scaling enhances the effects of average input behavior. A scaling between −1 and +1 provides an average of zero. This type of scaling enhances the effects of deviant input behavior [38]. In all cases, while holding all other variables constant, the input scaling in the [−1, +1] form provided better performing neural networks than when the data was scaled in the [0, 1] range. Typically the [0, 1] scaled inputs had more class 2 cells mis-classified as class 1 and consistently produced lower overall classification accuracies. The implication of this is that the deviant behavior in the features is more significant for determining cell performance than is the average behavior.

4. Conclusions

The results achieved in this study using neural network analysis of time-indexed maintenance events are a major advancement over what had been accomplished in the previous studies of the GNB battery. The maximum overall prediction accuracy achieved in this study was 80.6% (see neural network 7 in Table 5 or neural network 20 in Table 9). The maximum class-specific prediction accuracies achieved were 100% for class 1 (neural network 7 or 20) and 95.7% for class 2 (neural networks 2, 4, and 23). Neural networks optimized for class 3 were not investigated though 73.7% was achieved by two neural networks presented (15 and 21).

The class-specific prediction results obtained with neural network analysis of maintenance data can be compared directly with Li's results [13] using KNN pattern recognition analysis of combined fabrication and maintenance data, summarized in Table 1. A comparison of the best class-specific prediction results for both studies is presented in Table 11. From this summary it is clear that both studies have achieved practical levels of prediction, but the neural network prediction results are somewhat better. For class-1-specific prediction it was possible to select 70% of the high-performing cells, without any false selections from the low-performing cells (using neural net 15). For class-2-specific prediction, it was possible to select 95.7% of the poor-performance cells, with 21.1% of class 3 (intermediate) cells mis-selected, but with none of the class 1 (high-performance) cells mis-selected. By comparison, Li's results showed it was possible to select 50% of the high-performing cells, but with 6.7% mis-selected from lower-performing cells. Only 84.6% of the class 2 cells could be selected, but with only 10% of classes 1 and 3 cells mis-selected.

To assess the impact of these results on the practical management of a battery energy storage facility, consider the value of being able to select a subset of cells which will be among the highest performing cells, with a low to zero probability that lower-performing cells will be selected. Perhaps even more important would be the ability to identify nearly 96% of those cells which will perform poorly. It would be beneficial to rotate in fresh cells, reducing the potential for cell reversals. The cells removed could be scrutinized off-

Table 11
Best class-specific prediction results. Maintenance features (ANN analysis) and combination features (KNN pattern recognition) [13]

a) *Class 1* Source	Class 1 cells (selected %)	Cells mis-selected (% of classes 2 and 3)
ANN		
Maintenance features, (NN15, Table 9)	70.0	0.0
KNN		
Combined features (Table 1)	50.0	6.7
b) *Class 2* Source	Class 2 cells (selected %)	Cells mis-selected (% of classes 1 and 3)
ANN		
Maintenance features (NN23, Table 10)	95.7	21.1 [a]
KNN		
Combined features (Table 1)	84.6	10.0

[a] No class 1 cells mis-selected.

line to identify those intermediate cells which can have continued on-line utilization.

For long strings of cells, such as the GNB battery evaluated in this study, the economic benefit of using routine maintenance events to predict cell performance is very attractive. Conducting capacity tests on a large battery, such as the GNB battery, is expensive and disruptive. The results presented here indicate the feasibility of the routine application of neural networks for performance prediction as part of a maintenance strategy for long-string energy storage systems.

References

[1] C.A. Vincent, F. Bonino, M. Lazzari and B. Scrosati, *Modern Batteries: An Introduction to Electrochemical Power Sources*, E. Arnold, London, 1984, pp. 12–15.

[2] S.P. Perone and P. Symons, *J. Power Sources, 41* (1993) 277–290.

[3] R.N. McClellen and S.L. Deshpande, 500 kWh lead/acid battery for peak shaving energy storage, testing and evaluation, *EPRI-EM3707*, Oct. 1984.

[4] S.P. Perone and W.C. Spindler, *J. Power Sources, 13* (1984) 23-38.

[5] S.P. Perone and W.C. Spindler, in L.J. Pearce (ed.), *Power Sources 10, Proc. 14th Int. Power Sources Symp., Brighton, UK, Sept. 1984.*

[6] S.P. Perone and W.C. Spindler, *Power Sources, 10*, 1985, pp. 507–523.

[7] S.P. Perone and W.C. Spindler, *2nd Int. Conf. Batteries for Utility Energy Storage, Newport Beach, CA, USA, 24–28 July 1989*, Special Technical Publication, Energy Power Research Institute, Palo Alto, July, 1989.

[8] S.P. Perone, W.C. Spindler and S. Deshpande, *Proc. 3rd Int. Conf. Batteries for Utility Energy Storage, Kobe, Japan, 18–22 Mar. 1991.*

[9] R. Petesch, *M.S. Thesis*, San Jose State University, Aug. 1992.

[10] P.H. Chen, *M.S. Thesis*, San Jose State University, Aug. 1991.

[11] S.P. Perone, R. Petesch, P.-H. Chen, W.C. Spindler and S.L. Deshpande, *J. Power Sources, 37* (1992) 379–402.

[12] S.P. Perone, P.-H. Chen and W.C. Spindler, *J. Power Sources, 45* (1993) 73–92.

[13] X. Li, *M.S. Thesis*, San Jose State University, Oct. 1994.

[14] W.A Byers and S.P. Perone, *J. Electrochem. Soc., 126* (1979) 720–725.

[15] W.S. McColloch and W. Pitts, *Bull. Math. Biol., 5* (1943) 115–133.

[16] R. Hecht-Nielsen, *Neurocomputing*, Addison-Westley, Reading, MA, 1991, p. 2.

[17] L.M. Fu, *Neural Networks in Computer Intellegence*, McGraw-Hill, New York, 1994, p. 10.

[18] P.A. Jansson, *Anal. Chem., 63* (1991) 357A–362A.

[19] S.M. Weiss and C.A. Kulikowski, *Computer Systems that Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems*, Morgan Kaufmann, San Mateo, CA, 1991, pp. 1–16.

[20] L.M. Fu, *Neural Networks in Computer Intelligence*, McGraw-Hill, New York, 1994, p. 176.

[21] J. Zupan and J. Gasteiger, *Anal. Chim. Acta, 248* (1990) 1–30.

[22] J. Zupan and J. Gasteiger, *Neural Networks for Chemists: An Introduction*, VCH, New York, 1993, p. 42.

[23] R. Hecht-Nielsen, *Neurocomputing*, Addison–Westley, Reading, MA, USA, 1991, p. 22.

[24] J. Zupan and J. Gasteiger, *Neural Networks for Chemists: An Introduction*, VCH, New York, 1993, p. 137.

[25] *Using NeuralWorks: NeuralWorks Professional II/PLUS and NeuralWorks Explore*, NeuralWare, Pittsburgh, PA, 1993, p. 64.

[26] J. Zupan and J. Gasteiger, *Neural Networks for Chemists: An Introduction*, VCH, New York, 1993, p. 143.

[27] D. Miklavic, NeuralWare, personal communication, Jan. 1995.

[28] C.C. Casey Klimasauskas, *Neural Network Development*, Revision 4.4, NeuralWare, Pittsburgh, PA, 1994, p. 243.

[29] C.C. Casey Klimasauskas, *Neural Network Development, Rivision 4.4*, NeuralWare, Pittsburg, PA, p. 42.

[30] J. Moody and J. Utans, *Priciple Architecture Selection for Neural Networks, Neural Information Processing Systems 4*, Morgan Kaufman, San Mateo, CA, 1992, pp. 683–690.

[31] C.C. Casey Klimasauskas, *Neural Networks for Chemists, Revision 4.4*, NeuralWare, Pittsburg, PA, pp. 257, 260–261.

[32] D.E. Rumelhart, G.E. Hinton and R.J. Reynolds, Learning internal representation by error propagation, *Explorations in the Microstructures of Cognition, 1*, MIT Press, Cambridge, MA, 1986.

[33] *NeuralWorks Reference Guide: NeuralWorks Professional II/PLUS and NeuralWorks Explorer*, NeuralWare, Pittsburgh, PA 1994, pp. 126; 132–234.

[34] J. Zupan and J. Gasteiger, *Neural Networks for Chemists: An Introduction*, VCH, New York, 1993, p. 127.

[35] *NeuralWorks Reference Guide, NeuralWorks, Professional II/PLUS* and *NeuralWork Explorer*, NeuralWare, Pittsburg, PA, pp. 227–228.

[36] S.M. Weiss and C.A. Kulikowski, *Computer Systems that Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems*, Morgan Kaufman, San Mateo, CA, 1991, p. 168.

[37] *Using NeuralWorks: NeuralWorks Professional II/PLUS and NeuralWorks Explorer*, NeuralWare, Pittsburg, PA, 1994, p. 152.

[38] C.C. Casey Klimasauskas, *Neural Network Development, Revision 4.4*, NeuralWare, Pittsburg, PA, 1994, p. 123.