Evolving Artificial Neural Systems for Short-Term Power System Load Forecasting

Camille Koch
North Carolina A & T University
ckoch@aggies.ncat.edu

Joshua Murphy
North Carolina A & T University
jtmurph1@aggies.ncat.edu

Charles Winley, Jr.
North Carolina A & T University
cwinley@aggies.ncat.edu

Ali Osareh
North Carolina A & T University
osareh@ncat.edu

Gary Lebby
North Carolina A & T University
lebby@ncat.edu

Abstract

Short-term load forecasting (STLF) is integral to the overall power system operation inasmuch as day-to-day planning requires adequate estimation of load demands for scheduling purposes. This need for STLF is a matter of system dynamics and physics (i.e., spinning reserves cannot be allocated instantaneously), since it may take a few hours to a few days to bring a cold unit into operation. Poor load forecasts result in either underscheduling or overscheduling of costly quick-start units that impact the economic operation of power systems. Previous research by the authors and the state-of-the-art as proposed by others relied upon static models using least squares regression, function decomposition, and artificial neural networks applied to historical power system load data to perform STLF [4, 6-8]. The authors propose applying a technique referred to as EHNSML (evolved hierarchical neural systems machine learning) adapted from the works of Eddahech and King et al. [2, 3] in an effort to form a more robust STLF.

Introduction

The modeling of power system loads (PSL) is highly sensitive to the referenced time scale, and it may include day of the week effects and holiday effects [5]. With respect to the time scale, the PSL model (or forecast) may be short-term—hours to weeks; medium-term—three to five years; long-term—greater than five years. The modeling and forecasting time scales under consideration for this research encompass short-term modeling. Within this time scale, the STLF adequately addresses problems involving unit commitment, scheduling, operating reserve, operation planning, optimization of energy contracts, fuel management, and revenue.

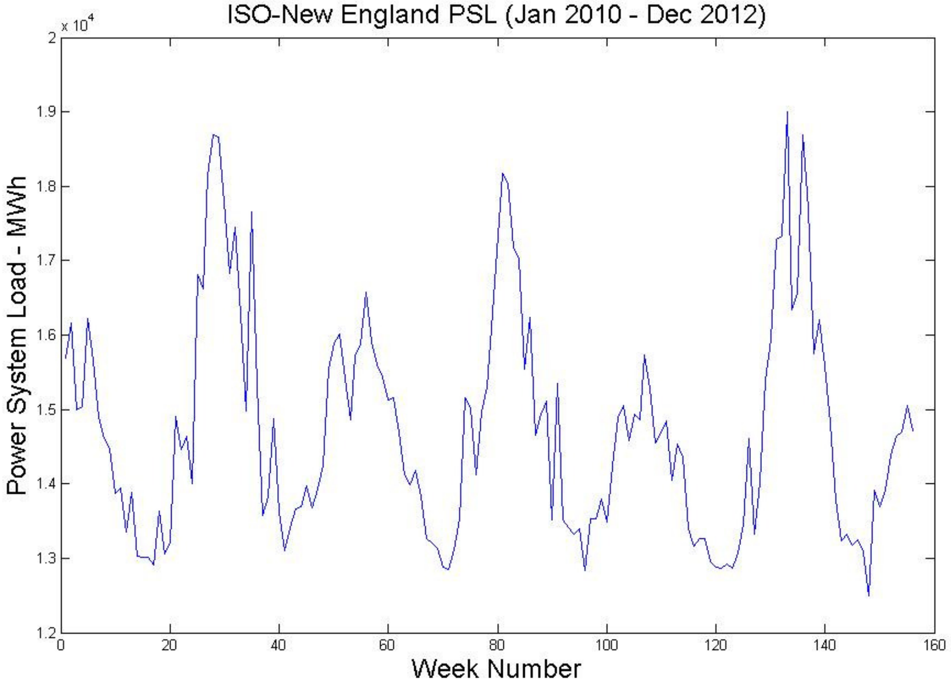

Figure 1. Weekly average power system load on an annual time scale for ISO-New England (Jan. 2010-Dec. 2012)

The basic power system load model (derived as a modified Wien automatic system planning package and the Lebby GENESIS package) that is most universally utilized by power system planners is

$$\boldsymbol{PSL = BL + GL + SL + WSL + \epsilon_n} \tag{1}$$

where *BL* represents a base load, *GL* represents the growth rate load, *SL* represents the normal seasonal variation in the load, *WSL* represents the non-seasonal weather sensitive load, and ϵ_n represents other effects that are not accounted for in the model, including random noise [1, 5].

The various models used in this paper consist of least squares regression (LSR), functional decomposition, and radial basis function general regression neural network (RBFGRNN). The functional decomposition method involves removing trends from the data one by one in order to model the system. It is more complex than the LSR, which performs computations based on the past trends of the data set provided. The method is common and relatively easy to perform. The RBFGRNN is a neural network that accepts input features of the power system load data and calculates the best weights to model the power system load data. The weights are then used for forecasting future loads.

Functional Decomposition

As explained previously, the basic power system load model consists of a base load, a growth rate, a seasonal load, a non-seasonal weather-sensitive load, and other effects. The functional decomposition is the oldest method of the three methods mentioned that will be used to model and forecast power system loads. The first two years of the data, shown in Figure 2, will be used predict the third year of power system loads.

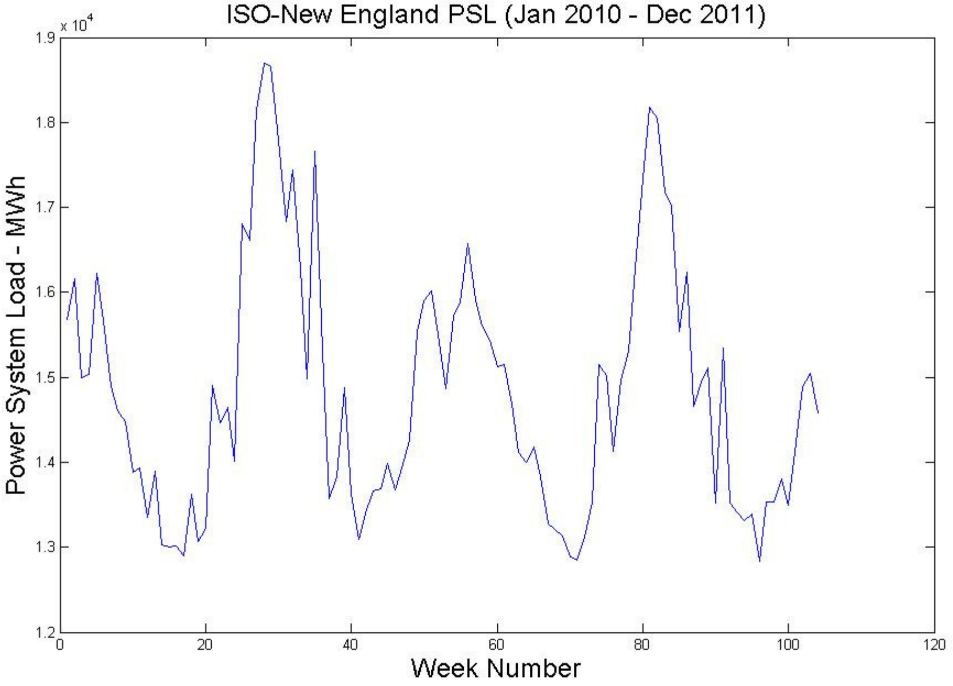

Figure 2. Weekly average power system load on an annual time scale for ISO-New England (Jan. 2010-Dec. 2011)

These two years of data will be used to determine the components of the power system load model. The first three components of the power system load will be factored out (base, growth rate, and seasonal load). Because other weather-related information (non-seasonal) is unknown, what is left over after factoring out these three components will be considered the modeling error.

The first component is the base, which can be determined by simply taking the average of all the data. The formula for the base and growth rate is given below where the two constants b_0 and b_1 are the base and growth rate respectively. The week number is represented by the letter w. With the base known to be the average of all the data, the averge growth rate can be calculated.

$$PSL = b_0 + b_1 * w \qquad (2)$$

Using the entire training data set (first two years of weekly power system load data), the base was found to be 15047 MWh and the growth rate was found to be -3.9924 MWh/week.

Figure 3 shows the PSL data with the base just calculated. The first residual (PSL minus the base) is shown in Figure 4 with the growth rate.

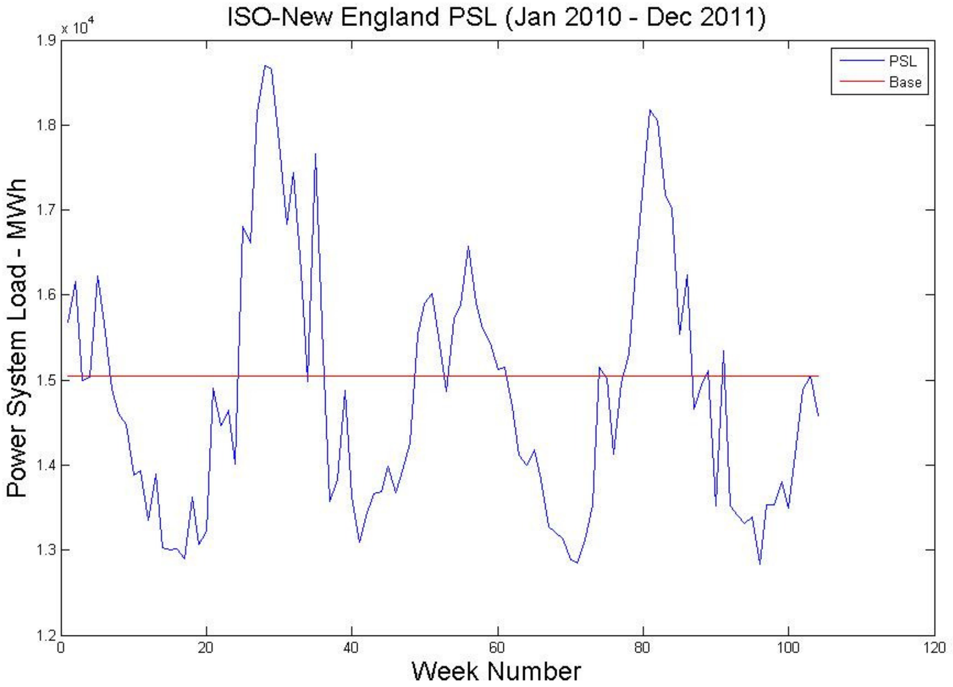

Figure 3. Weekly average power system load with base

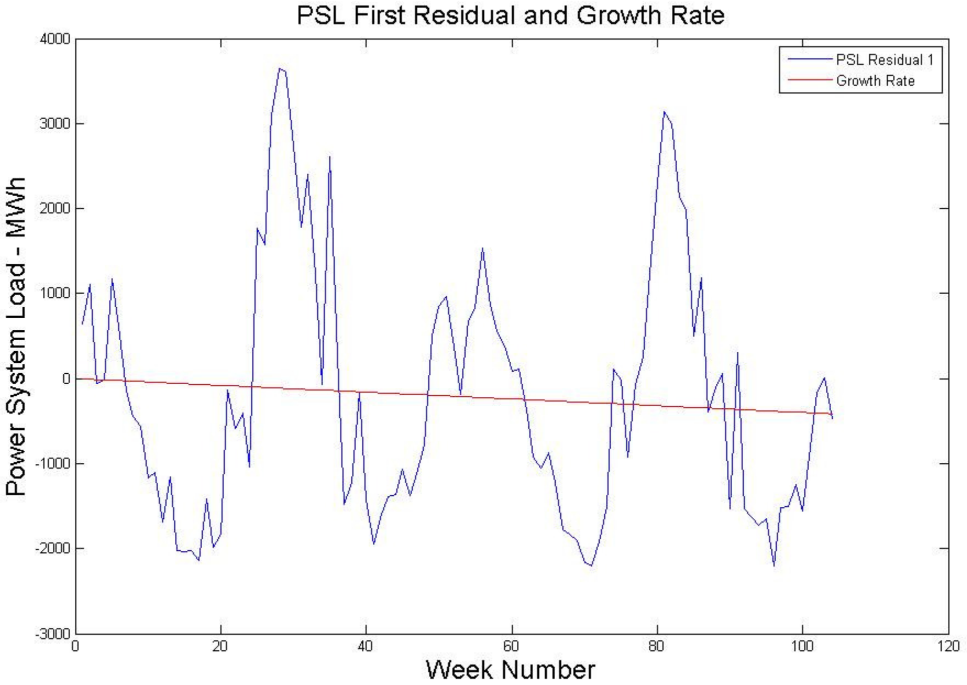

Figure 4. First residual (power system load minus the base) and the growth rate

The growth rate does not have to be determined by using the entire training data set. It can be determined from the peaks only or from the valleys only or from some other portion of the data set. The growth rate as determined from summer peak load data (PSL greater than 16600 MWh in Figure 2) was -1.2537 MWh/week. The growth rate as determined from all valleys (PSL less than 13500 MWh in Figure 2) was 1.7441 MWh/week. The growth rate that ultimately resulted in the smallest testing error (error for predicting the last year of data) was the growth rate obtained from the entire data set (-3.9924 MWh/week).

The second residual is the result of factoring out both the base and the growth rate. Therefore, the second residual is equal to the first residual minus the growth rate. The second residual is shown in Figure 5 along with the first residual to show the difference between the two.

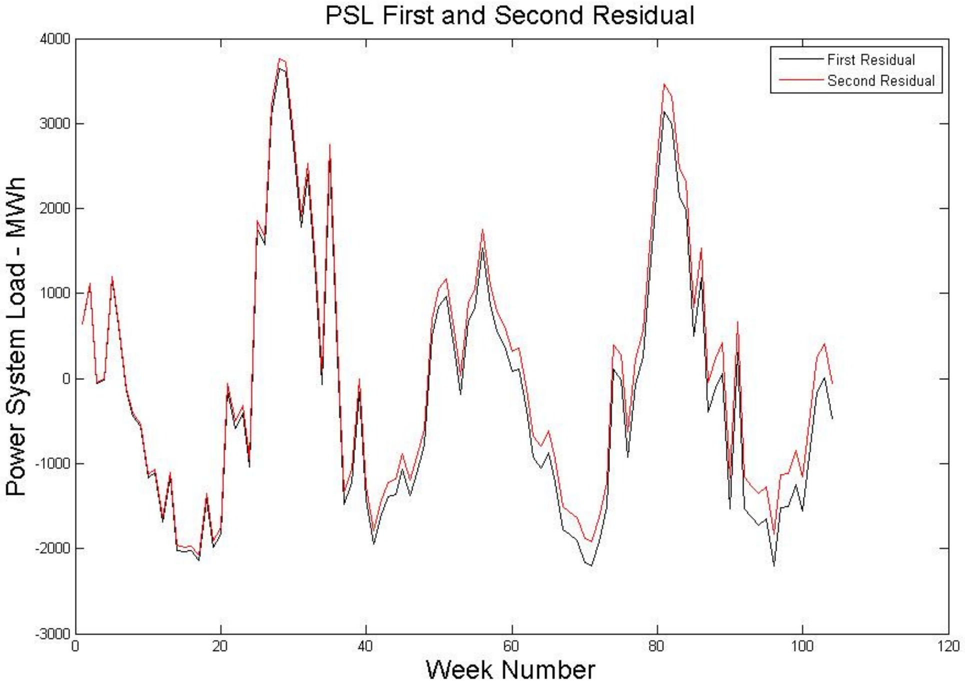

Figure 5. First residual and second residual (first residual minus the growth rate)

The next step is to factor out the seasonal trends. There appears to be a pattern that occurs every 52 weeks and a pattern, one peak, that occurs every 26 weeks. This seasonal trend can be described by the following equation containing sine and cosine components. Only the first and second harmonics of the Fourier decomposition will be used. The constants A_k and B_k are unknown constants, SL stands for seasonal load, and w is the week number.

$$SL(w) = \sum_{k=1}^{2} A_k \cos\left(\frac{2\pi}{52} k * w\right) + B_k \sin\left(\frac{2\pi}{52} k * w\right) \qquad (3)$$

The constants A_k and B_k can be approximated using the following formulas for the two years of training data (104 weeks):

$$A_k = \frac{1}{52} \sum_{w=1}^{104} \cos\left(\frac{2\pi}{52} k * w\right) * Resid2(w) \qquad (4)$$

$$B_k = \frac{1}{52}\sum_{w=1}^{104} \sin\left(\frac{2\pi}{52}k * w\right) * Resid2(w) \quad (5)$$

The constants A_1 and B_1 were found to be -471.2820 and -464.8181, respectively, and the constants A_2 and B_2 were found to be 1119.4 and 1297.2, respectively. Residual 2 with the seasonal load is shown in Figure 6.

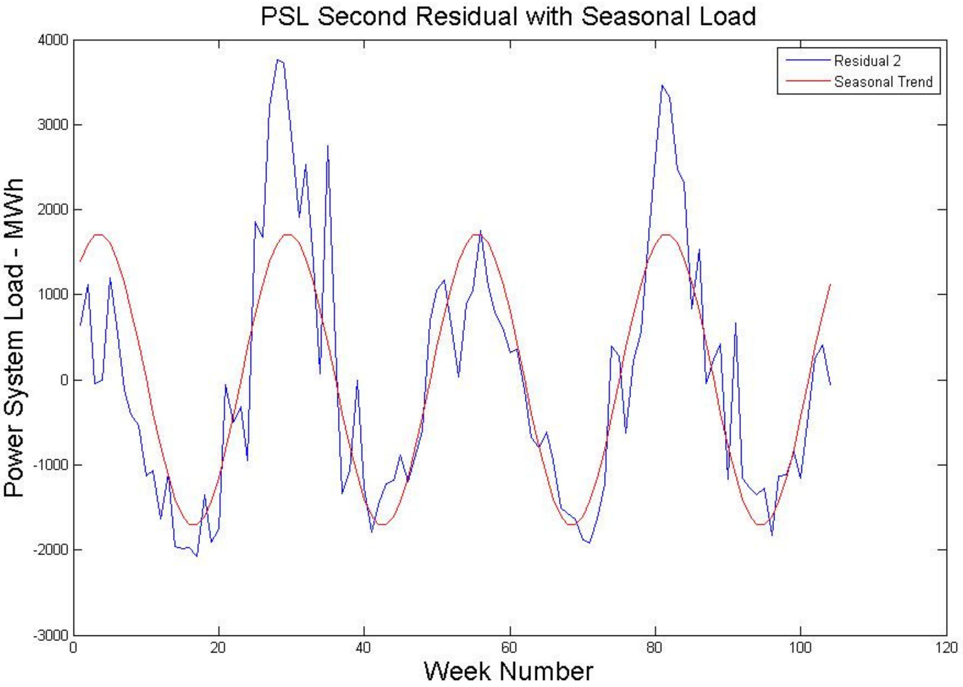

Figure 6. Second residual and the seasonal load trend

The third residual is the second residual minus the seasonal trend, as shown in Figure 7. The mean absolute percent error (MAPE) for the training data set was found to be 3.36% using the formula below. The base is the 15047 MWh found earlier.

$$MAPE = \frac{mean(|Resid3|)}{Base} \quad (6)$$

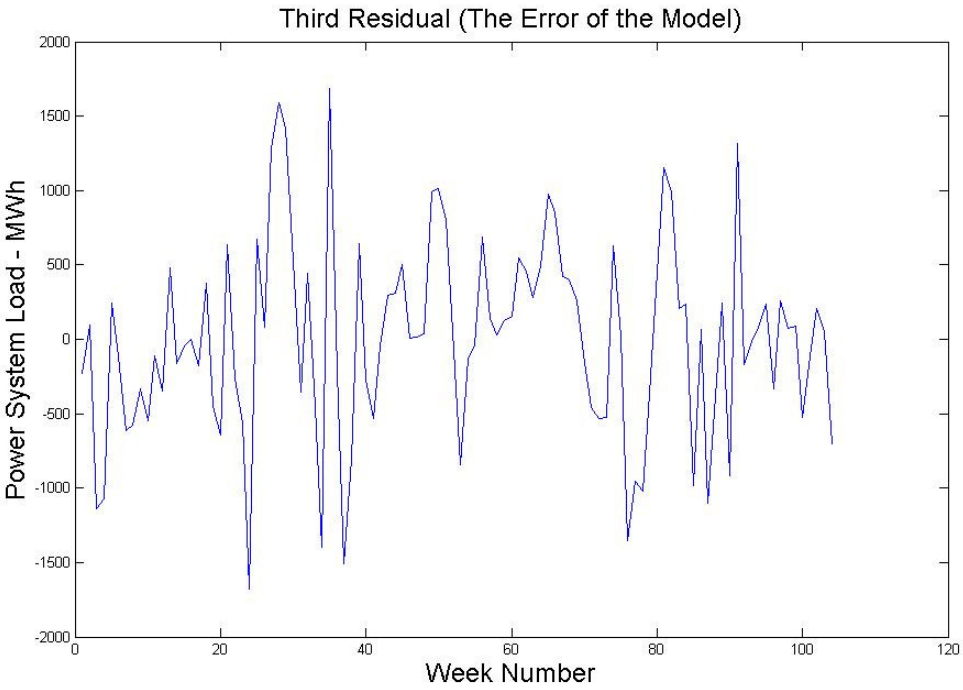

Figure 7. Third residual (error of the model)

Using the constants found (base of 15047, growth rate of -3.9924, A_1, B_1, A_2 and B_2) and applying them to the remaining year (Jan. 2012-Dec. 2012) PSL data, the MAPE error was found to be 3.37 percent. Figure 8 shows the forecasting of the PSL data for the last year.

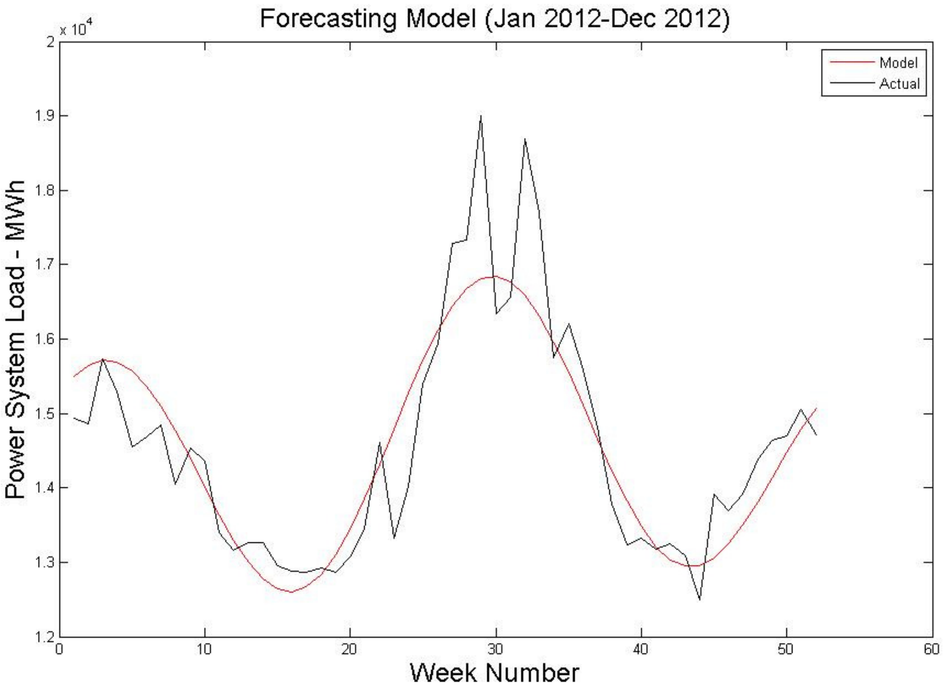

Figure 8. Forecasting model for 2012

Least Squares Regression

LSR involves attempting to accurately fit points along a data curve based on prior points of the data. The least squares regression method calculated the coefficients and the modeled PSL values using the ISO-New England PSL data set. The coefficients were inversely calculated using equations 2 and 3 and the actual PSL data. Some coefficient variables include bias/offset, growth rate, week number, and harmonic coefficients. The modeled PSL values were determined from the calculated coefficients using the same weeks as the functional decomposition method.

Although LSR has closely related to the functional decomposition method, LSR is a unique method. It calculates all the coefficients of the model (base, growth rate, seasonal) all at once without removing any trends. As noticed in the functional decomposition section, various residuals were calculated and subtracted from the data.

The data were separated into training and testing. The training for the LSR consisted of the first two years, which corresponds to the first 104 weeks. The testing data entailed the remaining 52 weeks. The algorithm for this method was programmed to detect the best of four LSR methods: linear, first harmonic, and second harmonic. The linear LSR only involved the base and the growth rate. The first harmonic of the LSR has a base, growth rate, and sine and cosine functions with its respective inside function with k (in Equation 3) referencing the harmonic number. The other LSR harmonics follow the same format for calculating the model PSL.

Figure 9 depicts the calculated LSR model against the actual PSL data set for the first two years. The model for training follows the original curve closely excluding the peaks. The base rate for this training model was 14962 MWh and the growth rate was -2.3817. Coefficients were A_1, A_2, B_1 and B_2 are -472.8928, 1117.8, -438.1892 and 1310.4 respectively.

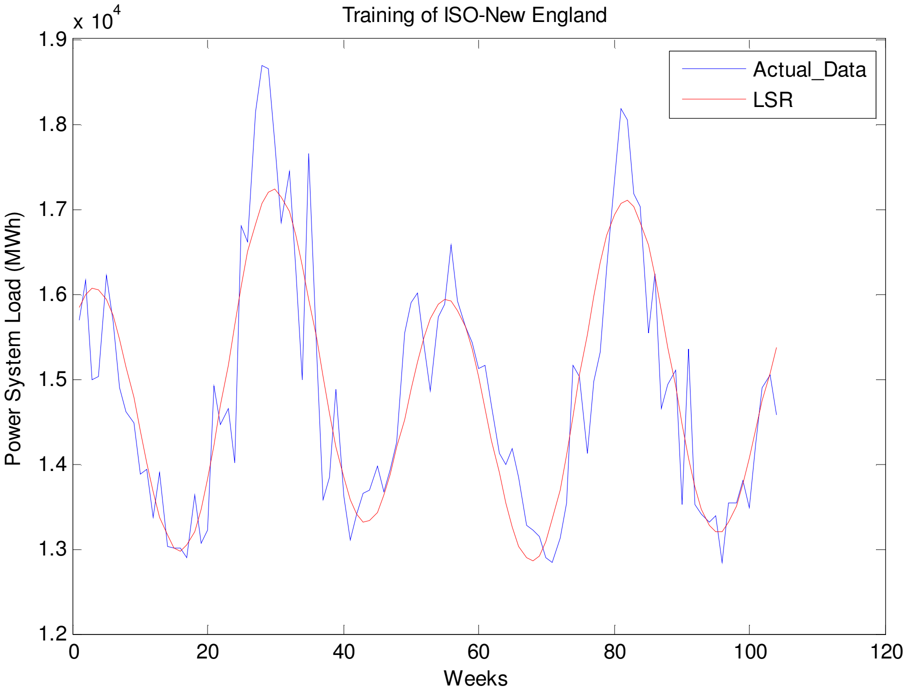

Figure 9. LSR training model and the actual ISO-NE PSL

The training error produced using the above values was 3.31 MAPE. Figure 10 depicts the testing error curve for the remaining weeks or the third year. This error was not significantly greater than the training error at 3.36 MAPE.

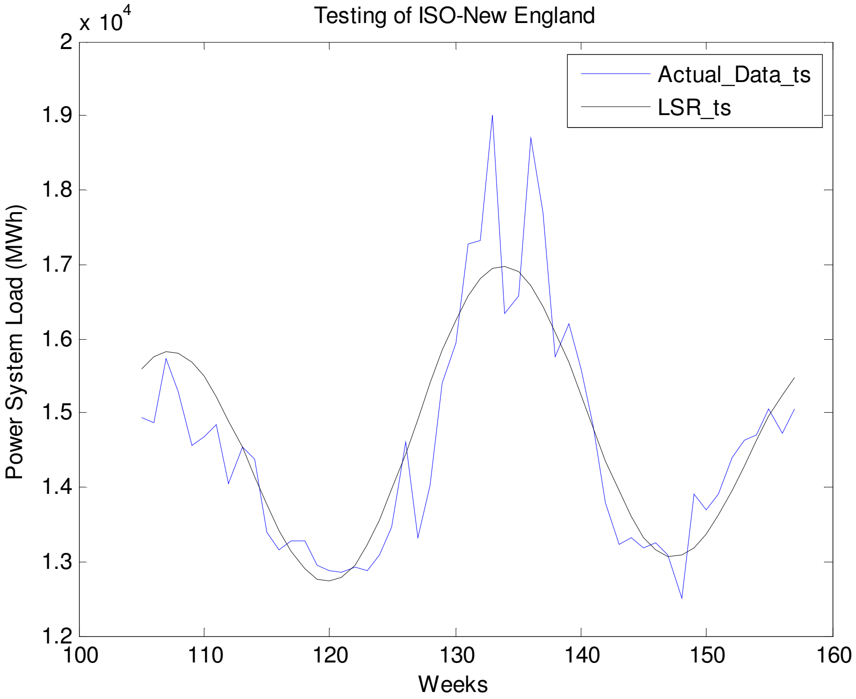

Figure 10. LSR testing model and the actual ISO-NE PSL

RBFGRNN

The radial basis function general regression neural network (RBFGRNN) is the most recently developed approach out of the three methods being used to model and forecast the power system load data. The RBFGRNN is an adaptation of the traditional GRNN developed by Donald Francis Specht. The RBFGRNN is made up of three layers: input, hidden, and output (Figure 11).

For the input layer, a bias, week number, and four previous weeks' PSL data were used to develop the dataset for the training and testing inputs of the neural network. The output is the last column of the data set.

Bias	Week #	PSL(W-4)	PSL(w-3)	PSL(w-2)	PSL(w-1)	PSL

Before training and testing, the RBFGRNN initial centers are developed. This can be done one of two ways: choose centers randomly from events in the training data set or find centers utilizing a clustering algorithm. Both approaches were utilized in determining which option would be used for finding the best initial centers for the network [4].

The RBFGRNN is an adaptation of the traditional GRNN that was developed by Donald Francis Specht (an adaptation of the Nadaraya-Watson kernel regression approximator) [5]. This network is akin to the RBF network in which there is a hidden unit centered at each cluster's center. These RBF units in the hidden layer are called Gaussian displacement units (GDUs) and correspond to kernel functions in the Nadaraya-Watson kernel regression approximator. The output of a GDU can be determined from equation 7.

$$g_\sigma(x_i, c_j) = \exp\left(\frac{\|x_i - c_i\|^2}{2\sigma_j^2}\right) \tag{7}$$

i covers the first event to the t^{th} training event and j covers the first center to the m^{th} center. where x is the input vector, c is the localized centers representing clusters of the input vectors, and σ_j is a constant for all centers that can be estimated by taking half of the largest cluster's spread.

The GDUs transform the input event space into a "filtered" version of the input events space, where the closer a given input is to a centroid the closer the GDU's output tends to unity. The output from the Gaussian displacement layer is fed into a linear regression network in order to map the GDU outputs to target training data. Allowing x to be a set of input vectors and y the corresponding target output a relationship can be established such that a set of weights, w can be found to represent the mathematical connection between the input and output.

$$G = g_\sigma(x_i, c_j) \tag{8}$$

The weight vector is obtained by mapping the "G" space into the output space by using an OLAM.

$$w = (G^T G)^{-1} G^T y \tag{9}$$

If $(G^T G)^{-1}$ cannot be calculated, the pseudo inverse of the Gaussian matrix, G, can be used instead. The output of the RBFGRNN, $\hat{Y}$, can be calculated using equation 10.

$$\hat{Y} = w_0 + \sum_{j=1}^{M} w \cdot g_\sigma(x, c_j) \qquad (10)$$

where w_0 is the weight for the bias input.

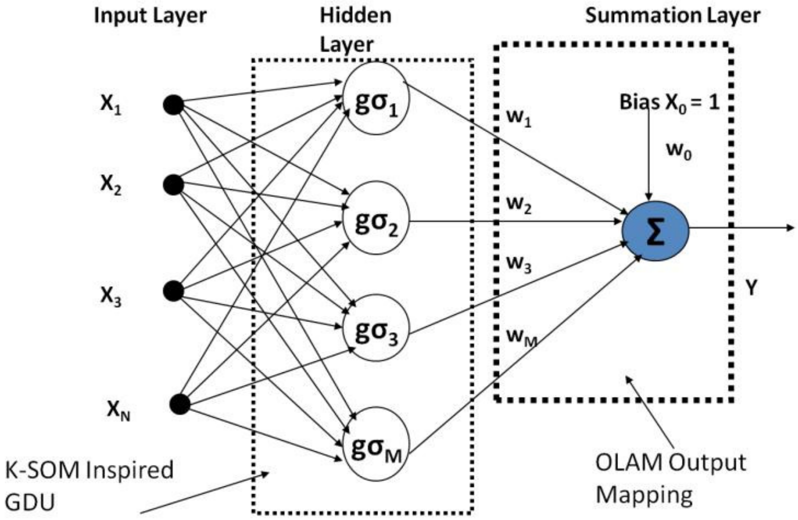

Figure 11. N -to- 1 RBFGRNN internal schematic

EHNSML

The EHNSML considers all three methods and determines the final output based upon the accuracy of each method at different time frames within each year. For the sake of demonstration, each of the three methods is compared against the actual training data set in Figure 12.

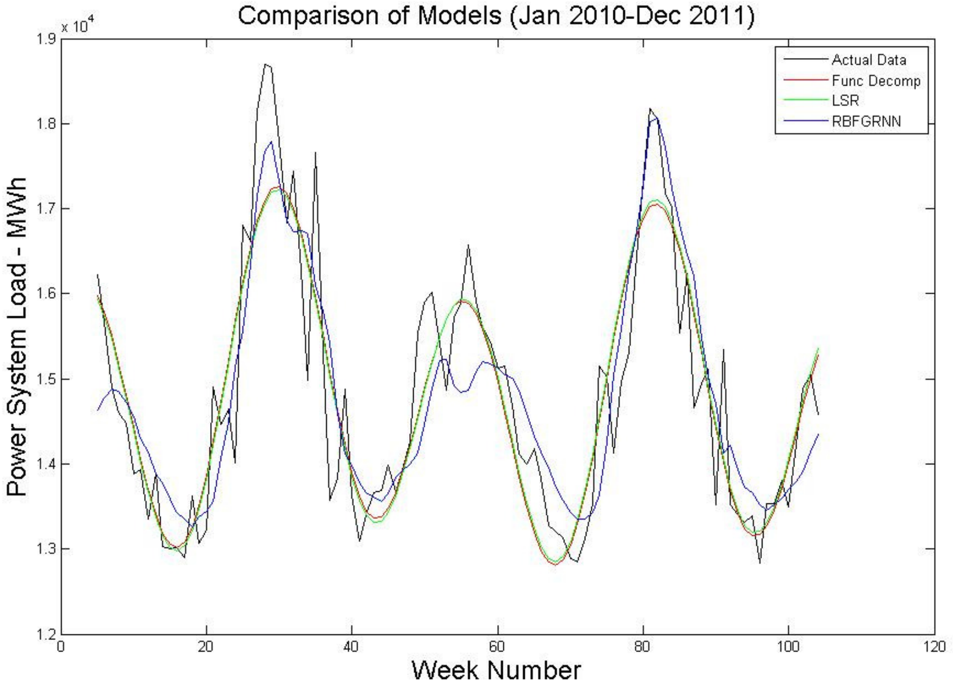

Figure 12. Comparison of functional decomposition, LSR, and RBFGRNN models (Jan. 2010 – Dec. 2011)

As can be seen in Figure 12, the RBFGRNN clearly models the summer peaks, the highest peaks, with greater accuracy than the other two methods. The other two models, functional decomposition and least squares regression, are very similar, meaning there is very little difference between the constants found using these two methods. That being said, the winter peaks, the lower peaks, and the valleys are better modeled by both the functional decomposition and LSR methods. The EHNSML can take these three outputs and find the weights that will hopefully extract the best features of each of the three methods.

The hierarchal neural network will be trained with 9 input features. These inputs include a bias, week number, four weeks of previous PSL data, and the three outputs from the three modeling methods discussed. All outputs, except the bias, will be normalized for greater accuracy. Two methods for obtaining the weights for the hierarchal neural network will be explored. The first method is to use genetic algorithms to determine the hierarchal neural network weights. The second method utilizes an additional neural network—for this case another RBFGRNN—to find the weights.

Utilizing genetic algorithms as the primary method for obtaining the weights for the hierarchal neural network was dismissed because of the extremely high algorithm complexity. In other words, using genetic algorithms to find the weights for the EHNSML took entirely too long, about 25 times higher than the second method, and the resulting accuracy of the testing output (2012) was not greater than the second method used.

Using a RBFGRNN neural network for the EHNSML was the best choice between the two methods because it did not significantly increase algorithm complexity and consistently found a more reliable solution, meaning the average or median accuracy was greater than the other three methods. Figure 13 shows the actual PSL load for 2012 against the predictions of the hierarchal neural network and the RBFGRNN.

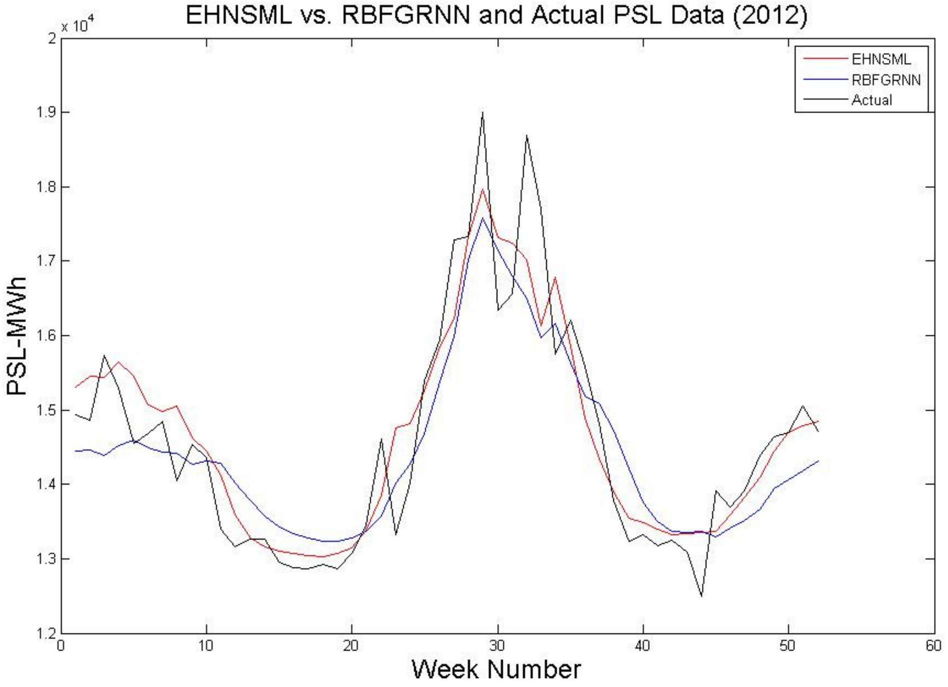

Figure 13. Comparison of EHNSML and RBFGRNN, 2012

Algorithm Complexity

The algorithm complexity, as calculated in Tables 1 and 2 in the next section, are a function of time. The time it took for the functional decomposition and the LSR methods to execute was not dependent upon the number of weights selected but was dependent upon the number of calculations that had to be performed. Also, the RBFGRNN and the EHNSML's algorithm complexity is dependent upon the number of calculations that had to be performed; however, unlike the functional decomposition and LSR methods, the number of calculations depended upon the number of weights or centers that were initially selected by the user. Therefore, the number of centers/weights and data features chosen will affect the time it takes to find a solution. The number of centers chosen for the RBFGRNN was 15 and 10 for the EHNSML

Summary of Results

Utilizing the most promising model methodologies as reported in [3, 5] presented in Table 1, it is shown that the performance of the each method was extremely successful in modeling the PSL of load zones obtained from the ISO-New England from January, 2010, through December, 2012. Tables 1 and 2 show the results of each method used. The second column denotes the median model recognition accuracy, while the last two columns represent the best neural system model of a series of parameter searches that optimize architecture accuracy and the algorithm complexity estimated as the number of calculations with respect to time used in obtaining a solution (less is better).

The results of Table 1 show that each method used in predicting the power system loads for 2012 were relatively successful. Each method was able to obtain about the same "best accuracy." From research, it was expected that the RBFGRNN would have the best accuracy; however, the "median accuracy" for the RBFGRNN exceeded both the functional decomposition and the LSR methods. In this way, the RBFGRNN can be considered the more reliable method. As shown in Figure 12, the RBFGRNN was able to more accurately model the summer peaks but was unable to model the winter peaks and valleys better than the traditional methods of functional decomposition and least squares regression. This is what caused the "best accuracy" of the RBFGRNN to not exceed the other two methods.

Table 1. Performance comparison of functional decomposition, least squares regression, and RBFGRNN on the STLF problem

Method	Median Accuracy(%)	Best Accuracy(%)	Algorithm Complexity
Functional Decomposition	95.6	96.63	911
Least Squares Regression	91.25	96.64	72
RBFGRNN	95.87	96.64	1514

Table 2. Performance comparison of RBFGRNN and EHNSML on the STLF problem

Method	Median Accuracy(%)	Best Accuracy (%)	Algorithm Complexity
RBFGRNN	95.87	96.64	1514
EHNSML	96.33	97.1	1961

Table 2 shows the results of utilizing the most successful models in a hierarchical mixture of experts and to evolve an optimal voting structure from the experts to form an evolved hierarchical neural systems machine learning (EHNSML) using another neural network. Preliminary results show that the EHNSML outperforms the RBFGRNN in accuracy metrics. Because each of the three methods has to complete training before sent to the EHNSML, the algorithm complexity of the EHNSML is in addition to the other three methods. It should be noted, however, that this additional algorithm complexity is not significant when it comes to achieving the additional accuracy.

The additional 0.46% of accuracy may not seem worth the additional time it takes for the EHNSML to compute the network outputs. However, this percentage could reduce the power system load error by up to 90 MWh. With the tight gaps demanded by the power industry, this reduction in the error margin would be worth the extra computational time.

Conclusions

Even though research suggests that the RBFGRNN models PSL data with greater accuracy than traditional methods, the other methods were able to achieve approximately the same accuracy as the neural network. The main reason for this was because of the RBFGRNN's ability to more accurately model the summer peaks, but it failed to perform as well as the traditional methods on the rest of the power system load data. Combining all three methods and their associated outputs to train a more complex evolved hierarchical neural system resulted in more consistent results and better accuracy without a significant increase in algorithm complexity.

References

[1] Buehring, W., Huber, C., & Marques de Souza, J. (1984). Expansion Planning for Electrical Generating Systems—A Guidebook. *International Atomic Energy Agency, Technical Reports Series,#241*. Vienna.

[2] Eddahech, A, Chtourou, S., & Chtourou, M. (2013, January). Hierarchical Neural Networks Based Prediction and Control of Dynamic Reconguration for Multilevel Embedded Systems. *Journal of Systems Architecture*, *59*(1), 48-59.

[3] King, M. C., Lebby, G. L., & Ricanek, K. (2001). A Dialog Control Strategy Using a Hierarchical Controller of Mutually Exclusive Neural Experts. *IASTED International Conference on Artificial Intelligence and Soft Computing*.

[4] Lebby, G. L., Stevenson, K. M., & Shi, G. H. (2003). Power System Load Modeling Using a RBFGRNN with Self Starting Centers. In *PowerCon: Blackout*. Los Alamitos, CA: IEEE.

[5] Lebby, G. L. (1985). *Load Shape Modeling in Southeastern Utility Systems*. (Doctoral thesis). Clemson University, SC.

[6] Miller, S. L., Lebby, G. L., & Osareh, A. R. (2010). Improving the Performance of the Truncated Fourier Series Least Squares (TFSLS) Power System Load Model Using an Artificial Neural Network Paradigm. In *Intelligent Data Engineering and Automated Learning* (368-378). Berlin: Springer.

[7] Taylor, J. W. (2010).Triple Seasonal Methods for Short-Term Electricity Demand Forecasting. *European Journal of Operational Research*, *204*, 139-152.

[8] Workineh, A., Dugda, M., Homaifar, A., & Lebby, G. (2012, July). GMDH and RBFGRNN Networks for Multi-Class Data Classification. *Proceedings of the 2012 International Conference on Artificial Intelligence*, *1*, 216-221.

Biographies

CAMILLE KOCH is currently an electrical engineering master's student at North Carolina A&T State University. She has worked as an adjunct instructor for Guilford Technical Community College in the field of electrical/electronics technology. She has also worked as a systems engineer for the Tennessee Valley Authority. She received her B.S. degree in Nuclear Engineering at the University of Tennessee, Knoxville, and her associate's in

Electrical/Electronics Technology from Guilford Technical Community College. Ms. Koch can be reached at ckoch@aggies.ncat.edu.

JOSHUA MURPHY is currently an electrical engineering master's student at North Carolina A&T State University; his thesis work is in the field of machine intelligence and power associated research (MIPAR). He has held several teaching assistant positions for various courses at the university. He received his B.S. degree in Electrical Engineering from Arkansas State University. Mr. Murphy has interned with companies such as Nissan North America, Thomas and Betts, and Carrier Corporation. Mr. Murphy may be reached at jtmurph1@aggies.ncat.edu.

CHARLES WINLEY, JR. is currently a master's student of electrical engineering at North Carolina A&T State University; his thesis work is focused in the field of machine intelligence and power associated research (MIPAR). He has held several teaching assistant positions for various courses at the university. He received his B.S. degree in Electrical Engineering from North Caolina A&T State University. Mr. Winley has interned with GE Transportation. Mr. Winley may be reached at cwinley@aggies.ncat.edu.

ALI R. OSAREH is currently an adjunct associate professor of Electrical and Computer Engineering at North Carolina Agricultural and Technical State University, specializing in power systems. Dr. Osareh leads industrial automation efforts in BIEES. Dr. Osareh can be reached at osareh@ncat.edu.

GARY L. LEBBY is currently a professor of Electrical and Computer Engineering at North Carolina Agricultural and Technical State University and a senior member of IEEE, specializing in power systems and artificial neural networks. Dr. Lebby is the Director of the Laboratory for Biologically Inspired Engineering and Energy Systems (BIEES). Dr. Lebby can be reached at lebby@ncat.edu.