Office of Graduate Studies
University of South Florida
Tampa, Florida

CERTIFICATE OF APPROVAL

Master's Thesis

This is to certify that the Master's Thesis of

MARK ALLEN SATTERFIELD

with a major in Computer Engineering has been approved
for the thesis requirement on December 8, 2000
for the Master of Science in Computer Engineering degree.

Examining Committee:

_____
Major Professor: Rafael Perez, Ph.D.

_____
Member: Eugene Fink, Ph.D.

_____
Member: Dimitri Goldgof, Ph.D.

APPLYING NEURAL NETWORKS TO INVESTIGATE ELECTRICAL POWER PLANT COOLING-WATER DISCHARGE TEMPERATURE

by

MARK ALLEN SATTERFIELD

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Computer Engineering
Department of Computer Science and Engineering
College of Engineering
University of South Florida

December 2000

Major Professor: Rafael Perez, Ph.D.

To my parents Katharina and James, and recognizing God, the creator of all things.

Acknowledgements

I wish to express my gratitude to Dr. Rafael Perez for his continued support through the many changes in research domains. I likewise extend a special gratitude to William Culbreth at Tampa Electric Company for his generous and patient teaching about the problem at hand, and for his attention to detail in the many e-mails we exchanged. I also thank my committee members, Dr. Eugene Fink and Dr. Dimitri Goldgof, for their pledge in reviewing this work.

Table of Contents

List of Tables

List of Figures

List of Acronyms

AE	Average Error
AI	Average Increase
ATI	Average Temperature Increase
BB#	Big Bend unit identifier
CNC	Cascaded Node Count
CV	Coefficient of Variation
EPA	Environmental Protection Agency
ETD	Equal Time Delay
HMT	Highly Massaged Testing
IS	In Sample
IWP	Inlet Water Pipe
IWT	Inlet Water Temperature
ME	Maximum Error
MI	Maximum Increase
MID	RMS Midrange Error
MSTD	Multi-Staggered Time Delay neural model
(n)	Number of Models
OC	Outlet Canal
OCT	Outlet Canal Temperature
OCTD	Outlet Canal Temperature Difference
OOS	Out of Sample
OT	Overall Testing
OWP	Outlet Water Pipe
OWT	Outlet Water Temperature
PI	Plant Information
RMS	Root Mean Squared

SCO	Scale Data Only
SS	Neural Model Set Size
STD	Staggered Time Delay
TC	Trained Clean
TD	Trained Dirty
Te	Test Set
TECO	Tampa Electric Company
TN	Trained Natural
Tr	Training Set

APPLYING NEURAL NETWORKS TO INVESTIGATE ELECTRICAL POWER

PLANT COOLING-WATER DISCHARGE TEMPERATURE

by

MARK ALLEN SATTERFIELD

An Abstract

of a thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Computer Engineering
Department of Computer Science and Engineering
College of Engineering
University of South Florida

December 2000

Major Professor: Rafael Perez, Ph.D.

This thesis investigates neural computing applied to electrical power plant cooling water system forecasting. The target system is a coal-fired, base-load electrical power generation facility owned and operated by the Tampa Electric Company in Apollo Beach, Florida. During the process of converting chemical energy contained in coal to heat energy and finally to electrical energy, excess heat energy is created and must be dissipated. The excess heat dissipation occurs by way of Tampa Bay, a large body of salt water directly navigable to the Gulf of Mexico.

The Environmental Protection Agency has established water temperature discharge guidelines and restrictions in order to protect the surrounding habitat. Current plant operations dictate unwritten heuristic approaches used to reduce the opportunity for temperature violations. The value in this work is twofold. First, the study attempts to uncover features that affect the water temperature as it leaves the Big Bend facility. These features, once identified, are then used to understand the characteristics of the plant as they relate to the water temperature discharge.

Neural computing techniques are used in this study because most of the variables exhibit interactions that are difficult to discern and categorize through other approaches. The pattern recognition facility of neural computing is especially important to this investigation. Intermediate neural network architectures are designed for proving specific feature interactions, and these models contribute to the final system architecture mostly by demonstrating preprocessing requirements.

The resulting artificial neural network architecture and associated preprocessing accommodates time delay features and produces results to within 1.0 degree Fahrenheit. The final architecture is a Cascade Correlation hidden layer model, composed of 28 input nodes and 1 output node.

Abstract Approved: _____

Major Professor: Rafael Perez, Ph.D.
Professor, Department of Computer Science and Engineering

Date Approved: _____

1 Introduction

This thesis investigates neural computing applied to electrical power plant cooling water system forecasting. The target system is a coal-fired, base-load electrical power generation facility owned and operated by Tampa Electric Company in Apollo Beach, Florida. The facility is called "Big Bend."

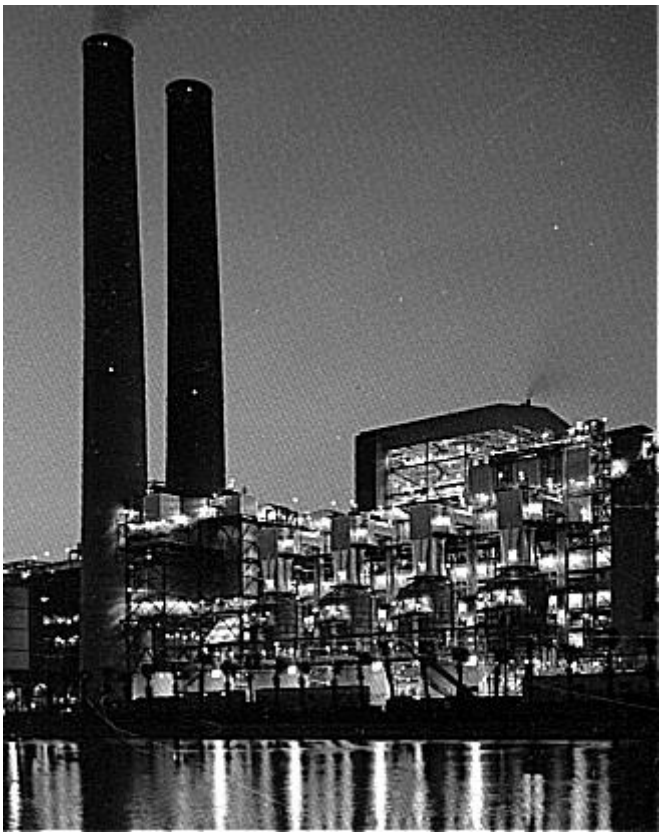

Figure A. Tampa Electric Big Bend Facility

1.1 Motivation

During the conversion of chemical energy contained in coal to heat energy and finally to electrical energy, excess heat energy is created and must be dissipated. At this facility known as Big Bend, the excess heat is dissipated by way of a very large natural

body of salt water called Tampa Bay. Tampa Bay is directly navigable to the Gulf of Mexico.

The Environmental Protection Agency has established water temperature discharge restrictions in order to protect the surrounding habitat. Tampa Electric Company faces fines and financial penalties if the water discharged into Tampa Bay exceeds the EPA established temperature limits.

1.2 Goals and Objectives

Current plant operations dictate unwritten heuristic approaches used to reduce the opportunity for temperature violations. The value in this work is twofold.

- First, the study attempts to uncover features that affect the water temperature that leaves the Big Bend facility. Several rapid prototype neural computing architectures are created to investigate feature interaction and to demonstrate preprocessing requirements.
- Once identified, these features are used to understand the characteristics of the plant as they relate to the water temperature discharge.

1.3 Methods

Neural computing techniques are used in this study because most of the variables exhibit interactions that are difficult to discern and categorize through other approaches. The pattern recognition facility of neural computing is especially important to this investigation.

Data mining begins by decomposing the large physical plant into elementary components, with ambition to determine effectual (or feature) variables. Individual components have a reduced number of influencing variables, so the decomposition process reduces the opportunity for introducing ineffectual variables that could corrupt and confuse the neural solution. In other words, inappropriate variables can be avoided more easily due to the decomposition and the resulting smaller sub-problem scope. Significant and required preprocessing steps are established during the decomposition process.

Neural computing is used as a pattern recognition engine during the feature selection process to assist in determining if an identified feature contributes to the

physical model. If adding the feature in question to a neural system positively influences the predictive capacity of the model, then the feature is probably effectual. If the feature has no effect or negatively influences the model, then the feature cannot be used in its presented form, but still may be influential and eventually integrated to the model with appropriate preprocessing. For example, this was the case with the Inlet Water Temperature feature, which is integrated into the final neural model through preprocessing.

The iterative approach of reducing the system continues until the necessary and sufficient features are discovered. The decomposition process then reverses and the prototype models and associated preprocessing are integrated and assembled into the final neural system architecture.

1.4 Organization

This work is separated into three parts, composed of seven chapters.

Part 1 includes four chapters. Chapter 1 is this introduction. Chapter 2 introduces neural networks, describes various network architectures, and details concepts especially suited for this study such as preprocessing and variable selection. Chapter 3 introduces the generic Power Company and describes power cycles as used in a coal-fired power plant. Chapter 4 integrates the generic power plant and neural computing by explaining the applicable details regarding Big Bend, reviews the EPA restrictions that purpose this thesis, enumerates Big Bend data collection variables of interest, and explains influential aspects of the neural network tool used in this exercise.

Part 2 includes two chapters. Chapter 5 investigates Feature Selection, enumerates time delay driven feature selection, and details preprocessing requirements. Chapter 6 investigates an offline approach to discovering various system effects.

Part 3 includes one chapter. Chapter 7 concludes this work by offering analysis and discussion of the problem, proposes ideas for future work in this domain, and offers insights and proposals for future neural computing tool design.

2 Introduction to Neural Computing

Neural computing is a branch of artificial intelligence research based on the paradigm of what is considered the human brain computational element, the biological neuron. Advocates hold fast to three significant tenants towards neural computing possibilities:

- Learning and self-adjustment opportunities in that retraining is inherent since the algorithm itself is the training foundation,
- Generalization, or the ability to overcome both noise and the "literal-mindedness" of a traditional algorithm, and
- Abstraction to idealized prototypes based on imperfect inputs [Wasserman, 2].

The solution methodologies designed and implemented through neural networks offer separation between knowledge (or data, which is the excitation properties and their relation to system output) and control (or the algorithms designed to learn those relationships). Structured programming design rules equally support the separation between data and control.[1]

In as much as neural network models are appropriate for fluid and algorithmically evasive problems, neural models are typically not well suited to performing even basic arithmetic functions. That said, human reasoning itself is largely heuristic based, and heuristics is of great benefit when dealing with never observed, inconsistent, noisy, and otherwise invalid information in which traditional algorithms are simply unable to cope. Neural networks have an interesting ability to approximate results for many systems, and as such have the potential to capture non-linear relationships that are not easily discerned by human researchers [Harrald, 40]. In essence, neural computing truly shines where traditional algorithmic solutions are simply not known [Giarratano, 52].

[1] See [Schach] or other course material on structured software design for details on separation between data and logic.

2.1 The Biological Foundation for the Artificial Neuron

Biological neurons are composed of three intertwined units called the cell body, the dendrite, and the axon.[2] The cell body is the control mechanism for the neuron, the dendrite delivers input values to the cell body, and the axon delivers a firing pulse to adjoining neuron dendrites. The cell body receives two kinds of inputs through the dendrite, some of which are excitatory and some of which are inhibitory. Interestingly enough, each biological neuron offers only "fire" and "rest" responses to the downstream neurons, based on its own input trigger levels.

The brain is composed of input, output, and hidden neuron layers.[3] The input layer consists of those neurons that receive triggering information from outside the brain, and the output layer consists of those neurons that deliver trigger information to outside the brain. The hidden layers are the interconnecting neurons that bond the inbound input layer neurons and the outbound output layer neurons. There are innumerable hidden layers within the brain, yet each hidden layer functions exactly like the others. By computing standards, the biological neuron network is massively parallel since all of the neuron threads execute simultaneously.

Similar to their biological cousins, artificial neurons are simple processing units, interconnected through weighted links where the sum of all input links causes the output link to exhibit varying levels of excitation.

---

[2] This is not intended to be an exhaustive study of the biology of neurons, but instead a presentation to bridge the study into that of the artificial neuron. For a more complete investigation, the reader is referred to course or reference material on neurons, such as [Roediger].

[3] In actuality, the brain is but one analogy to representing biological neurons. Although biological neurons are concentrated in the brain, neurons exist throughout the body as sensory neurons, motor neurons, and interneurons. However, for this discussion, any contained unit is an acceptable course of study. The brain was selected because of the paradigm that "thinking" and "learning" occurs within the brain. The body itself is an equal analogy.

2.2 History, Dark Ages, and Rebirth

Neural computing was birthed near the end of the 1940s, through a proposal by D. O. Hebb described as a learning law [Wasserman, 24, 212; McClelland, 83; Perez, 2-3]. Hebb suggested that a network of simple computational units and interconnecting weighted links could be configured to encourage learning capabilities. (See Figure B, "Single Layer Perceptron Network.")

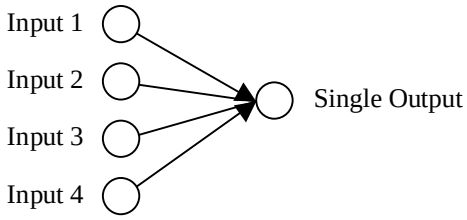

Figure B. Single Layer Perceptron Network

Through the 1960s, neural computing continued in an effort to create devices capable of learning. The early years continued with great advancements, but simultaneously presented astounding obstacles due to the very nature of learning itself. Learning requires generalization and extraction, and these activities in tern require exceedingly great computational power – power that was not available at that time. In addition, computers are founded on an expectation that results are error free – an inherent difficulty with the concept of learning through generalization since generalization is non-deterministic and results are possibly inconsistent [Wasserman, 9]. (For example, what one calls a "dark hue of red," another calls "a light hue of burgundy.")

There was an additional – and significant – difficulty with learning systems, and that was the configuration of the learning network itself. Most learning algorithms at the time modeled what is now known as single layer networks[4], where the input layer is directly connected to the output layer.[5] These single-layer "perceptron" networks were

[4] The single layer network design was adopted in part because no learning or training algorithms for multiple layer networks was known. See also [Wasserman, 30].

[5] According to [Wasserman, 46], literature is inconsistent in conclusively defining "number of layers." Some authors refer to the layers of neurons, while others refer to layers of weights. This text adopts the weight layer count definition.

mathematically proven (by Minsky in 1969) to be unable to represent certain functions. The representation restriction is that the problem space must be linearly separable in order to be represented (and thus learned) by a neural network. That is, single layer neural networks are restricted to problem domains that are linearly separable; those problems that are not linearly separable cannot be solved using single layer neural networks. This restriction may not at first appear critical in a two input binary model, since 14 of the 16 binary two-input functions can be trained [Giarratano, 53; Wasserman, 29; Koza, 515]. However, as the dimensions (n) and input patterns (P) increase, the number of linearly separable functions across the problem domain approaches zero, as dictated in the following formula by Nilsson [Perez, 2-25].

$$\text{Linearly Separable}(P, n) = 2 * \Sigma \left[(P-1)! / ((P-1-i)!\, i!) \right] \quad \text{for } P > n$$

$$= 2^P \quad \text{for } P \leq n$$

n = Input Dimensions

P = Total Number of Possible Input Patterns

i = 0 to n - 1

By example, in a two input, binary valued system n = 2, and $P = 2^2 = 4$, so LS(P, n) = LS(4, 2) = 2 * (1 + 3 + 3) = 2 * 7 = 14, as expected. There are $2^4 = 16$ functions in a two input, binary valued system; therefore, 87.5% are linearly separable. However, as the dimensions increase, the percentage of linearly separable functions quickly diminishes. For a three input, binary valued system n = 3 and $P = 2^3 = 8$, so LS(P, n) = LS(8, 3) = 2 * (1 + 7 + 21 + 35) = 2 * 63 = 126 linearly separable functions. There are $2^8 = 256$ functions in a three input, binary valued system; therefore, 49% are linearly separable.

As a near direct result of these obstacles, neural computing was set aside in lieu of more traditional and deterministic computing models [Giarratano, 53].

Two events came together in the early 1980s that generated great interest in neural computing. First, a theoretically sound training algorithm appeared for multi-layer networks [Wasserman, 36, 43]. Next, Hopfield contributed theoretical foundations on approximating exponential order (or combinatorially explosive) problems such as the

Traveling Salesman Problem in constant time [Giarratano, 53; Wasserman, 106].[6] The possibility of rapidly approximating a known NP-Complete problem in much less than exponential time was intriguing, and these events revitalized the neural computing paradigm.

2.3 Network Organization

The early days of neural computing introduced perceptrons (through Hebb). Perceptrons are interconnected systems of computational units. These computational units multiply each input X_i by an individual associated weight W_i, and sum the results into a single value.

$$\text{Node Value} = 1 \text{ if } \Sigma [X_i * W_i] > \text{threshold value}$$
$$= 0 \text{ otherwise}$$
$$i \text{ over all input links}$$

The perceptron output is one if and only if the summed weighted input value (Node Value) is greater than a predetermined threshold; otherwise, the unit output is zero. In general, perceptron systems consist of a single layer of non-recurrent (or feed-forward) neurons connected to the input layer through weighted links, although more complicated networks often bear the same name.[7]

In the late 1950s, Rosenblatt continued Hebb's theoretical neuron work with more complicated perceptron networks and training algorithms [Wasserman, 216]. By 1962, Rosenblatt proved that the Hebbian network could learn anything that it could represent. The original perceptron networks were binary input/output networks. Windrow-Hoff furthered the early perceptron networks by adding continuous outputs and multi-output structures.

A strong argument for multi-layer networks is that multi-layer networks are able to recognize and learn non-linearly separable problems [Koza, 515]. These multi-layer

---

[6] See also [Koza; Hopcroft; Sedgewick; Wasserman] for a more exhaustive study of the Traveling Salesman Problem, NP Completeness, and heuristic approaches to approximating NP-Complete problems in polynomial time.

[7] For an excellent in-depth study of Perceptrons, the reader is referred to the original manuscripts or textbooks such as [Wasserman].

networks, just like their single-layer counterparts, are composed of neurons with activation functions and interconnected weights. The addition is that multi-layer networks have hidden layers of nodes interconnecting the input and output layers.

Neuron activation functions can maintain many forms, most of which tend to normalize the neuron output [Wasserman, 14].[8] In fact, non-linear activation functions must exist to increase the computational power of multi-layer networks. In other words, an equivalent single-layer network can replace every multi-layer network that maintains purely linear activation functions [Wasserman, 19; Perez, 2-7, 4-2]. Therefore, multi-layer networks with linear activation functions remain affected by the limitations demonstrated by Minsky.

Furthermore, studies have demonstrated that the non-linear gain associated with these activation functions help to alleviate noise saturation between nodes, and encourage the network to respond appropriately to wide ranges in input excitation levels [Wasserman, 15].

In summary, neural computing of today is a more generalized web connected collection of (possibly) multiple hidden layer computational units. In addition, the units may be cyclically connected (or "with feedback," known as recurrent), and the unit outbound value may take on continuous values (as prescribed by Windrow-Hoff). Current configurations for most notable neural architectures are three layer, with one input layer, one output layer, and one hidden layer, in either a feed-forward or recurrent design [Koza, 515].

2.4 Roadblocks

Even now, neural computing is not without significant roadblocks. For example, the very nature of forecasting based on network models instead of algebraic (or formula based) solutions lends itself to unpredictable results. In addition, due to the large design of many network models, exhaustive testing is at least impractical if not impossible.

---

[8] For example, sigmoid (centered between 0 and 1) and tanh (centered between –1 and +1) functions may be used as the activation function. Signal Hebbian Learning involves non-linear transfer functions; Differential Hebbian Learning involves time delayed transfer functions.

Further, networks are traditionally unable to explain their own possibly unexpected and incorrect results [Giarratano, 55], and there remains a somewhat unspoken expectation that computers and computer based models should be error free. Over all these, training time may be quite extended for large problems, requiring cutting edge hardware months or years to find appropriate solutions, and these solutions are not yet tested [Hammerstrom, June 1993, 32].

With these considerable roadblocks, network users may deem the neural approach unacceptable before recognizing and thoroughly understanding neural benefits. It is worth repeating that neural approaches are especially beneficial where traditional solutions are not easily recognized.

2.5 Selecting Critical Data

The input layer to the biological neuron network is continually stimulated, and the brain must minimize unrelated data as it attempts to determine significant patterns. Unfortunately for us, the brain is not always successful at discarding insignificant patterns, such as that observed with Pavlov's experiment in animal feeding behaviors [Fink, 8].[9]

In this experiment, Pavlov managed to induce salivation in a dog by the ringing of a bell. In order to teach this response to the dog, Pavlov rang a bell and then immediately introduced food substance to the dog under test. Eventually, the dog "learned" to salivate at the sound of the bell alone, attaching the sound to expected nourishment. In the same manner, the dog was conditioned to expect a shock when presented with a higher pitched bell. As the experiment progressed, the bell tones converged to one another. Eventually, as the tones became non-discernable, the dog exhibited behavior consistent with neurosis. Put simply, "when we cannot distinguish between stimuli, we become confused" [Fink, 10].

The same thing happens within an artificial neural network. Here, the engineer is often the culprit in introducing unwanted and unwarranted input variables into the

[9] For additional information on conditioned inappropriate responses and neurosis, see [Fink]. For a brief investigation of representational meaning and consciousness, see [Barnaby].

artificial neural network, just as Pavlov is credited (or is the culprit, depending on one's vantage point) in introducing unacceptable variables into the canine's natural neural network. Therefore, a paramount aspect of selecting appropriate data is not selecting inappropriate data.

A neural network "learns" based on the data presented during training sessions. If the network is excited with a barrage of unrelated and ineffectual information during training sessions, the network may self-suppose and learn patterns that simply do not exist.

For example, say that a car mechanic is informed that an engine experiences hard starting every other time the vehicle is started. This of course is highly trainable data, especially that the data is binary and appears probabilistic (there are no other cases – either "yes" it is hard to start, or "no" it is not hard to start). The neural network engineer, being uneducated about the problem domain yet highly educated at network models and data sets, receives the data and demonstrates 100% effectiveness at solving this problem.

However, the 100% effectiveness may be based on analysis that is 100% wrong or variables that are 100% ineffectual. It may be instead that this car is used daily for one round trip between home and a grocery store. With this additional information, the hard starting may have absolutely nothing to do with "every other time," as dictated by the expert (car owner). Instead, it may simply be based on one of many instantaneous conditions such as:

- Cold or hot engine (choke),
- Cool or warm ambient air temperature (air charge temperature),
- Fuel system primed or not primed (vapor lock), or even
- Direction of the resting vehicle such as uphill or downhill (carburetor float level).

Therefore, in this scenario, the "every other time" history data may be unrelated and ineffectual, and even poses a difficulty in properly training the network since the problem has now been deemed "solved with 100% accuracy."

Neural computing is similar to most evaluation-oriented disciplines in that a significant step towards proper analysis includes determining critical and pertinent data. Selecting critical data (also known as data mining) is an interesting paradigm in neural

computing since superfluous data will introduce an additional opportunity to learning illogical, invalid, or otherwise wrong patterns. Often, the results continue to pass through unnoticed and offer correct (or nearly correct) solutions, until such time that the inappropriate and ineffectual stimuli unexpectedly changes. At that time, there may be no external validation step involved, and the invalid result is presented to the network user. In this, realize that illogical learning is simply not a good thing. With this warning comes an important tenet: In most applications, it is important that the neural engineer have experience in the target domain, and the domain expert have some experience with neural computing [Hammerstrom, June 1993, 29].

2.6 Data Preprocessing

A grand difficulty in designing appropriate neural network systems is representing the data appropriately, and describing how the presentation mimics the natural world [Winston, 252]. Creating the data sets is complicated and costly. Data preprocessing applied to the data sets involves manipulating input data into trainable patterns, as well as removing outliers and other anomalies [Hammerstrom, June 1993, 51].

For example, it may be that the instantaneous barometric value has no effect on predicting a coming storm; however, the rise or fall of barometric pressure is significant in storm forecasting.[10] There may also exist time delay driven information. For example, an earthquake might cause no immediate water damage; however, a tidal wave generally follows earthquake activity [Infoplease]. This type of cause-effect time delayed relationship is sometimes difficult to recognize yet the neural network training scenarios must appropriately analyze the delta time configurations.

Similarly, scaling may be required. For example, in stock market prediction algorithms, the absolute stock price may present no real and fundamental information, yet when observed in a logarithmic scale the stock price measure delivers percentage change information.

[10] Rising barometric readings usually indicate improving weather conditions; falling barometric readings usually indicate declining weather conditions. See [Hurricane] or other meteorological publications.

In summary, acquiring data is one thing; making it useful is still another [Hammerstrom, June 1993, 30].

2.7 Network Training and Weight Selection

To reiterate, neural networks are not programmed in a traditional manner, but instead are configured to learn based on a learning algorithm and presentation (or excitation) data. In fact, the network itself "knows" nothing until trained. Although the required training may consume extended amounts of time, the resulting network can determine solutions very quickly [Giarratano, 51].

2.7.1 Training Set Size

The training set size is important for the network to learn training patterns, yet encourage the network to remain flexible in generalizing onto unobserved patterns. An often-used rule of thumb is that five to ten training patterns are required for each interconnection weight [Hammerstrom, July 1993, 51]. Therefore, if a network consists of four input variables, ten hidden nodes with bias inputs, and one output variable, the network is composed of (4x10 + 1x10 + 10x1) = 60 weights, and requires 300 to 600 training patterns. Generalization difficulties surface when too few training patterns are applied. The difficulty is that with too few patterns (e.g., a large network with few training patterns), a single network node may be devoted to learning each training pattern. In this reduced case, there is no generalization at all [Yuan and Fine].

2.7.2 Training Algorithms

There exists more than a dozen known learning algorithms [Giarratano, 51], with one high level categorization being divided between "supervised learning" and "unsupervised learning." Supervised training methods use the target output during weight selection, and the network learns how to consider that output. Unsupervised (or reinforcement) training methods use a good/no-good indicator [Holst, 7]. In general, unsupervised learning is any learning in which the target output is unknown to the network during training, and the network learns patterns that are similar to one another. Supervised learning, on the other hand, is any learning in which known target output values can be compared to the network derived output results during training sessions.

Hebbian learning is part of the unsupervised class (sometimes called correlation learning) and the learning structure requires neither a teacher nor the expected output pattern [Wasserman, 212]. Hebb's theory on adopting the correlation paradigm rested on the expectations that the biological networks (e.g., the brain) do not require a teacher, but instead self-learn patterns. In other words, unsupervised learning involves only three entities: Two neurons, and one interconnecting synapse.

Most training paradigms presuppose the network architecture. For example, the number of layers, the number of elements in each layer, and the output functions must be pre-designed and static [Koza, 516]. Some more recent systems build the architecture while training [NeuralSoft]. The goal, though, remains the same: Develop a network weight architecture that can recognize observed excitation data, and generalize to unknown and unobserved excitation data.

2.7.3 Cascade Correlation

One such dynamic architecture is the Cascade Correlation network proposed by Fahlman at Carnegie Mellon University. The Cascade Correlation algorithm does not require that the network architecture is pre-supposed, but instead the architecture itself is built while training with additional hidden layer nodes and additional weighted interconnections. Figure C, "Cascade Correlation 4x0x2," shows the first stage of a Cascade Correlation network.

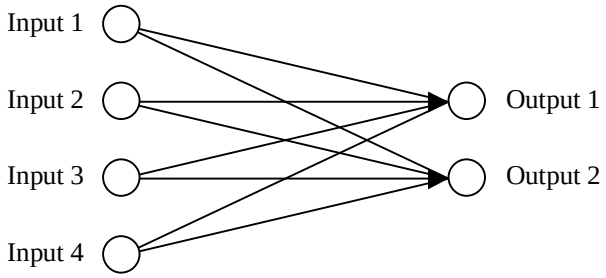

Figure C. Cascade Correlation 4x0x2

The Cascade Correlation algorithm builds a forward-looking fully connected mesh, where each additional node is connected to all existing nodes including the input and output nodes. The "forward-looking fully connected" identification simply means that only the hidden nodes are fully connected; in other words, the input nodes are not

fully connected to each other, and the output nodes are not fully connected to each other. Figure D, "Cascade Correlation 4x1x2," demonstrates the network with one cascaded node. The heavy links are additional from the nominal case. Note the dissimilarity with a traditional hidden layer architecture, where there are typically no direct connections from input nodes to output nodes.

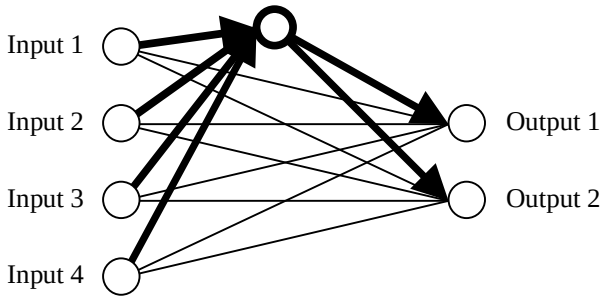

Figure D. Cascade Correlation 4x1x2

Further deviations exist from the traditional hidden layer network as the number of internal nodes increase. The Cascade Correlation network builds a forward-looking mesh. Figure E, "Cascade Correlation 4x2x2," demonstrates the network with two cascaded nodes. Heavy links are the additional links from the 1 internal node structure. Note the "mesh connector" from the existing hidden node to the newly added node.

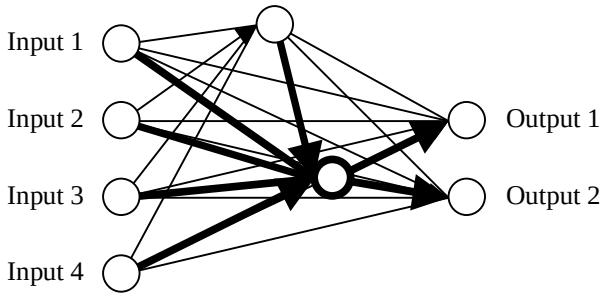

Figure E. Cascade Correlation 4x2x2

The dynamic network architecture relieves some pressure about correctly designing the network, since the network training mechanism may add hidden processing elements and connections as required.

3 Introduction to Power Company

This chapter discusses power companies in general, and specifically how power is generated at a coal-fired, base-load power generation facility.

3.1 Consumers and Power Demand

Not long ago, heating was fulfilled by natural gas, cooking was accomplished over wood burning stoves, and lighting was offered by kerosene lamps. These options are now often offset by equivalent electrically driven appliances.

Power companies are consumer driven. Without the consumer demanding electrical energy for air conditioning, street lighting, heating, cooking, and even computers, power companies would run idle. Luckily for power companies, consumers enjoy the modern conveniences offered by demand driven electrical energy. Consumers contract with a local power company to deliver energy on demand.

3.2 Power Transmission Overview

Power is transmitted to the consumer through physical power lines webbed within all metropolitan areas, and nearly all rural areas. In fact, there is a government mandate that promises every consumer electrical power.[11]

There are three independent power grids within the United States. These grids connect all power generation facilities (power sources) and consumers (power sinks). The only requirement is that power consumed (or removed from the grid) must equal power generated (or introduced to the grid).

$$\Sigma \text{ Sink} = \Sigma \text{ Source}$$

[11] The Rural Electrification Act of 1936 was an emergency program issued by Roosevelt through Executive Order in 1935. Congress passed the act in 1936. The act guaranteed full electrification of America's rural areas.

3.3 Power Generation Overview

In that power removed from the grid must equal power introduced to the grid, each power company must manage the power source effectively. Power generation takes many forms with very abstract level definitions including base-load and peak-load generation.[12] Each form offers unique advantages, and at the same time introduces unique disadvantages.

Base-load generation units include coal, hydroelectric, and nuclear; peak-load generation units include gas turbine and diesel. Base-load production is significantly less expensive than peak-load production, and concurrently presents significant and unique challenges. Base-load units are comparatively very expensive to build, and take considerable time to build (including extensive environmental impact studies and fees).

During times of generation outages (low source) and excessive peak loads (high sink), a power company will opt for one of two methods to meet the overage demands. One is to reduce the power sink (or power removed from the grid). This is performed with Load Management devices installed at customer premises (such as heating and air conditioning switches, water heater switches, and pool pump switches). Rolling brownouts of varying degrees also helps to reduce the sink demands; of course, rolling brownouts do not help the public relations associated with the particular Power Company.

The alternative is to increase power introduced to the grid. Once the generators are operating at peak capacity, the only method left to increase grid source is to purchase power from other power companies. The industry term for buying and selling power between wholesale accounts is wheeling [Central]. Wheeling is generally the most expensive alternative to increasing grid source, especially unplanned wheeling.

During planned outages, such as maintenance outages, a power company will contract to buy power from another company for a set price. Just as in most contract situations, if the purchase is planned instead of emergency or last minute situations,

---

[12] Although power is also supplied by customer owned and operated co-generation units, the definitions are avoided to not divert focus. Study of co-generation is left to the reader if desired.

negotiations to reduce purchase price are possible. However, unplanned excessive peak loads are more expensive to accommodate since the Power Company purchasing the power does not have liberal amounts of time to negotiate the contract.

3.4 Coal Fired Base Load Power Generation

This study investigates a base-load, coal-fired plant operating under non-maintenance outage conditions. We will investigate Coal to Electricity below.[13]

A power generator is called a power cycle, and operates the opposite of a heat pump. Heat pumps use work to produce thermal energy, while power cycles use thermal energy to produce work.

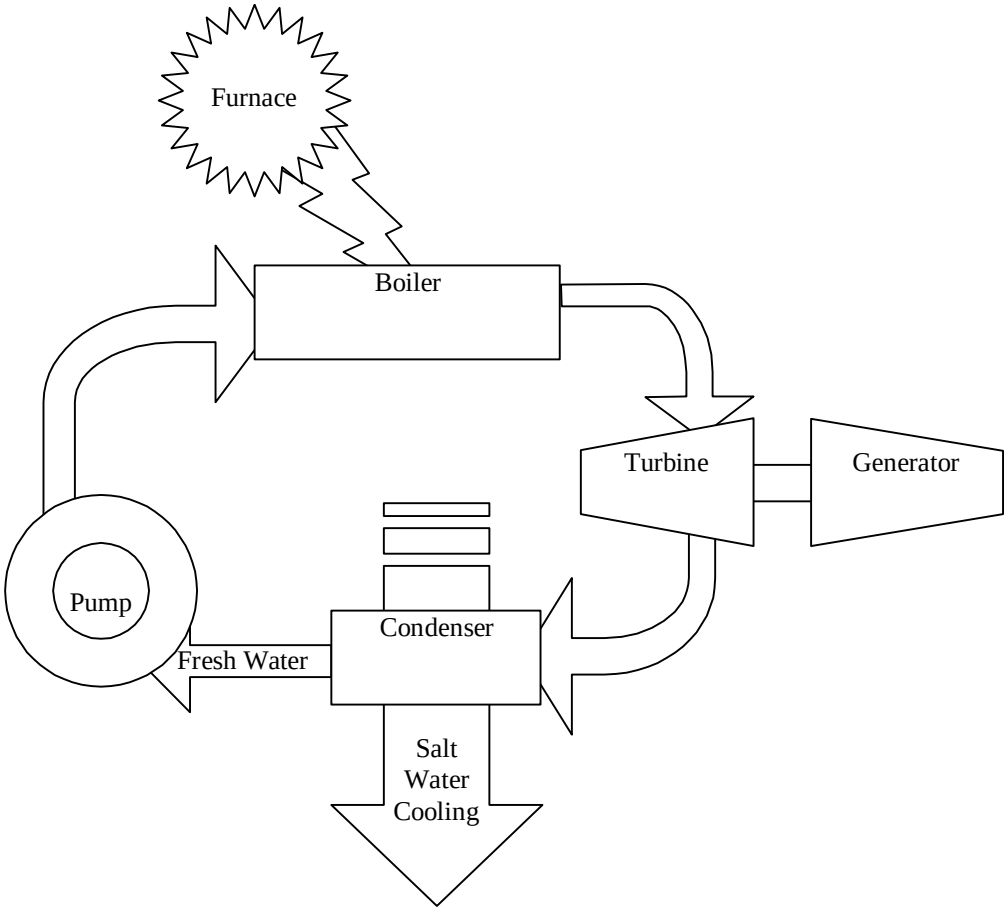

Figure F. Power Cycle Boiler Design

[13] See [Busot] or other fundamental thermodynamic text for a foundation to this study.

Coal to Electricity works as follows (see Figure F, "Power Cycle Boiler Design"):

- FURNACE: The furnace converts the coal based chemical energy into heat energy.
- BOILER: The boiler is the heat exchanger that transfers the heat energy to the FRESHWATER. The FRESHWATER is phase changed to steam since the temperature is increased above boiling point. Note that the FRESHWATER is encased in a closed loop system.
- TURBINE: The FRESHWATER steam (with significant heat energy) is converted into mechanical energy by the turbine (see also step PUMP). This transfer of energy reduces the steam energy and thereby reduces the FRESHWATER temperature. In an idealized situation, the TURBINE system would use all heat energy within the steam.
- CONDENSER: The FRESHWATER steam (and remaining heat energy) enters the condenser heat exchanger, where the excessive energy is removed from the FRESHWATER steam by the SALTWATERCOOLING. The steam temperature is reduced below the boiling point, and the steam is phase changed back into water. Note especially that the heat introduced to the FRESHWATER at BOILER must be removed from the FRESHWATER between TURBINE and CONDENSER, to effect a phase change from steam back into water.
- PUMP: The FRESHWATER is pressurized by a water pump deriving energy from TURBINE or a separate turbine on the high side.
- GOTO BOILER: The high-pressure water enters the boiler heat exchanger, and the cycle starts anew.

The purpose of the condenser is to phase change the steam back into water, since it requires exceedingly great energy to pressurize steam. Instead, the liquid water is pressurized and reintroduced to the system. Raising the pressure of a liquid through the pump requires much less mechanical energy than is extracted by the turbines, and this difference is the mechanical energy that is applied to the electrical generator.

At Step TURBINE, two things occur:

- The turbine (or a derivative) spins the PUMP.
- The excess work (lots of it) spins the generator, where mechanical energy is finally converted into electrical energy.

The turbine/generator pair is very heavy and offers large inertial energy; the pair acts as a flywheel to accommodate instantaneous increased load demands. The feedback loop from the turbine/generator to the furnace/boiler identifies when changes in energy are required. Energy generation changes are only required during sustained increased or decreased load requirements.

4 Problem Analysis/Investigating this Power Plant

This chapter introduces the generation facility under study, introduces the motivations for studying outlet temperature graphs, and explains data collection opportunities inherent at this plant.

4.1 Plant Configuration

The Tampa Electric Company Big Bend power generation plant is located on Tampa Bay. There are four base load coal fired power generators, each with a maximum generation capacity in excess of 400 megawatts. Big Bend offers a total coal-fired generation capacity of over 1800 megawatts.

The generators are built on a manmade peninsula finger between two canals. One canal is the cooling water inlet canal, while the other is the cooling water outlet canal. Water is retrieved from the "cold side" inlet canal, passes through the condenser (see Section 3.4, "Coal Fired Base Load Power Generation,") and is expelled into the "hot side" outlet canal. Both canals open into Tampa Bay. Figure G, "Big Bend Satellite Image," offers a satellite image of the plant configuration [United]. The Big Bend plant is highlighted by a rectangle. Tampa Bay is to the left (west). The plant is oriented with inlet canal towards the top (north) of the plant flowing from left to right (west to east) and the outlet canal toward the bottom (south) flowing from right to left (east to west).

Figure G. Big Bend Satellite Image

4.2 Restrictions with EPA

There are artificial limits to the maximum power production at the TECO Big Bend generation facility.[14] One of the artificial limits faced is an Environmental Protection Agency (or EPA) limit on the outbound cooling water temperature that leaves the outlet canal.

The facility uses the waters from Tampa Bay for generator cooling, and consequently the bay water temperature is increased. Water temperature fluctuations affect the surrounding habitat; excessive water temperature increases may irreparably damage the habitat. In an effort to reduce the destructive effects on plant and animal life, the EPA establishes guidelines to which the generation facility must adhere or else face significant fines and penalties.

[14] The word "artificial" as used here implies that the production facility and power units are physically capable of increased capacity but are operating under reduced capacity due to other constraints.

Among other regulations and restrictions, the EPA specifies that the hot side outlet (or exhaust) water into Tampa Bay must not exceed 108 degrees Fahrenheit.

4.3 PI

Plant wide data-collection is stored through a commercial packaged called PI (a mnemonic for Plant Information). The PI system stores information obtained from hundreds of sensors, some of which are redundantly configured. PI offers an automated Microsoft Excel based data query mechanism. Once the data is linked into an Excel document, typical Excel data manipulation, massaging, and graphing techniques can occur.

One benefit of this automated approach to data collection is that the opportunity for data collection discrepancies due to transcription and time offset errors is reduced because the same data is recorded at regular time slices. Most sensors are configured in redundant arrays so that sensor failures can be determined and isolated. Some data collection points are not required for Federal review and are not maintained to the same accuracy as other sensors.

4.4 Collection Points

Of the hundreds of sensor data collection points, the following sensor points are of special interest to this study. See Figure H, "Sensor Orientation," for a visual representation of the sensor placements.

- Inlet Water Temperature (IWT). The IWT measurement occurs within the Inlet Water Pipe (IWP), between the Inlet Canal and the Condenser. There are several redundant IWT sensor probes within each IWP. There are four IWP, one for each generator. The IWT measurement is regulatory, and therefore the sensors are maintained to within ½ of 1 degree Fahrenheit accuracy.
- Load. There are four Load measurements, one for each generator. The Load measurement is not regulatory, but the probes and measurement equipment are maintained for plant operations. The Load measurement passes through analog conversions that introduce some noise and reduce accuracy.
- Outlet Water Temperature (OWT). The OWT measurement occurs within the Outlet Water Pipe (OWP), between the Condenser and the Outlet Canal. There are several

redundant OWT sensor probes within each OWP. There are four OWP, one for each generator. The OWT measurement is not regulatory, and therefore the probes are not maintained. It is expected that the sensor measurements remain within a 2 degree Fahrenheit accuracy band.

- Tide. The Tide measurement occurs in the Outlet Water Canal. The measurement is useful only in a relative scale, and is not calibrated to any standard. The Tide measurement is not regulatory, and therefore the sensor is not maintained.
- Outlet Canal Temperature (OCT). The OCT is measured at the end of the Outlet Canal, as the canal water is discharged into Tampa Bay. There are several redundant OCT sensor probes at the demarcation point between the canal and the bay. The OCT measurement is regulatory, and therefore the sensors are maintained to within ½ of 1 degree Fahrenheit accuracy.

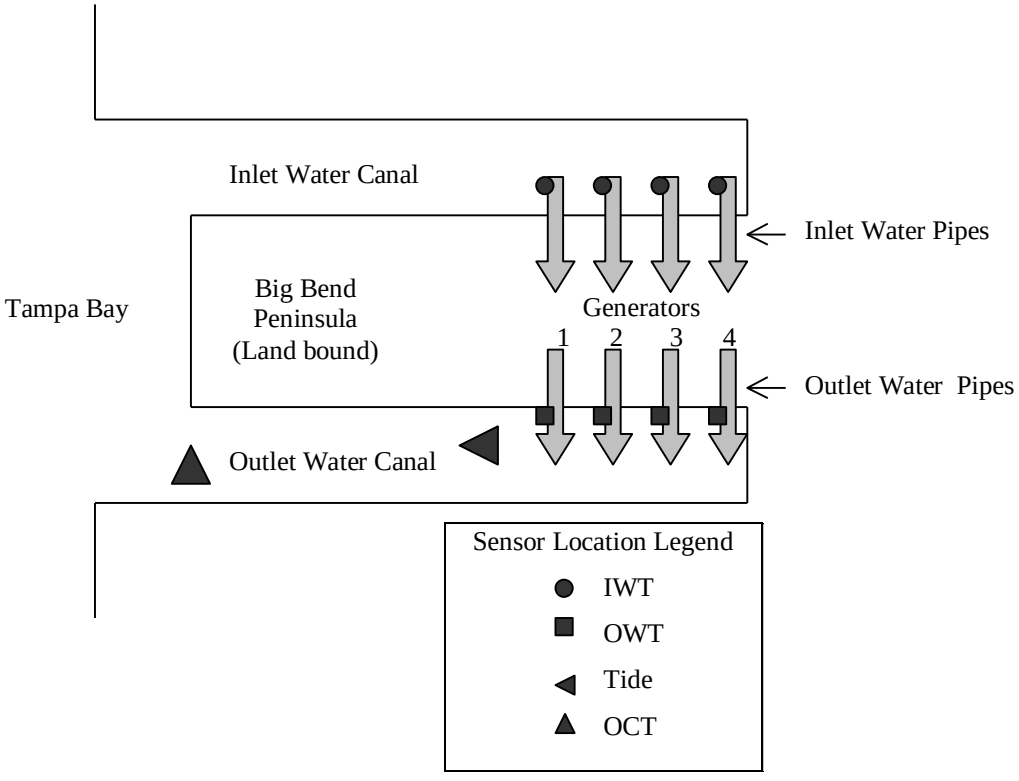

Figure H. Sensor Orientation and Location

4.5 Predict

This study uses a neural network software packaged called Predict manufactured by NeuralSoft. Predict builds models based on the Cascade Correlation paradigm. Predict is a configurable neural modeling tool with many configuration options, such as:

- Two network types (one designed for clean data, the other designed for noisy data),
- Several selectable node-to-node transfer functions,
- Various selectable error calculation functions, and
- Multiple preprocessing options such as data transformations and variable selection capability.

4.5.1 Predict Options

The NeuralSoft documentation suggests using the Kalman Filter when dealing with noisy data. The data collected at Big Bend represents very noisy data; the data was collected from a real-world structure with many influencing features and observed sensor failures.

The Input Variable Selection option uses a Genetic Algorithm to find the most appropriate subsets of the full input variable set, thereby reducing the active input variable count. The discard rules became especially prevalent and troublesome during the MSTD described in Section 5.5, "Effects of Load to Temperature Time Delay." Since a goal of this study is to determine effectual variables, it is important that the input variables remain those selected by the neural computing engineer. Identifying the models as No Variable Selection forces the inclusion of all variables into the network model.

The Data Transformation Level shapes continuous values through non-linear transformations. Selecting Scale Data Only is suggested when preprocessing is completed externally to the network tool, as was the case for these studies.

The remaining variables are left as default.

4.5.2 Working Data Set Selection

The Predict software engine requires identifying several data sets. Two sets are used while training the network.

The primary training data set is composed of two independent but related subsets. The Training Data Set is used for explicit weight calculations during the training cycles.

The Testing Data Set is used to test the weight calculations performed as a result of the Training set. These two sets are part of the iterative process of building the neural network.

The Validation Set is separated from the Training Data Set and Testing Data Set, and is used to perform in-sample testing.

The Manual Set is selected independently of the primary set, and is used to perform out-of-sample testing. The distinction between the sets will be further expressed in Chapter 5, "Determining Effectual Factors – Feature Selection and Preprocessing."

5 Determining Effectual Factors – Feature Selection and Preprocessing

This chapter covers data mining, variable selection, and data transfer functions and preprocessing requirements as they apply to data presentation. The chapter concludes by establishing a practical and predictive neural architecture.

As with most neural network studies, determining the critical data and manipulating that data into trainable sets is in itself a significant issue. A traditional neural computing study may allow all data mining to be handled by the resident expert engineers while the neural network scientist can simply accept this data and determine patterns. However, in most cases (as was the situation with this study) an iterative approach to data mining is embraced.

In the case of Big Bend, the initial data collection included plant-wide maximum load per day, average inlet water temperature per day, and the outlet water temperature per hour. These first models required identification and segregation of weekend, weekday, and holiday data, and required identification of seasons that influenced the shape of the hourly curves due to load effects. The twelve models (three daily divisions by four seasons) were then somewhat arbitrarily selected since the Tampa, Florida (demographics for the TECO Big Bend customer base) winter day is often more reminiscent of a summer day. For example, during a hot and humid winter day, customers demand air conditioning loads instead of heating loads. The magnitude of managing the quite diverse group of models is a significant task in itself since the final model selection remains a human heuristic. Specifically, questions such as if a particular day will be a warm winter day or a cool summer day must be answered by a human operator.

The studies in this chapter comprise an evaluation of system wide Features, where the neural network tool is used to discover patterns in the data. While disassembling the plant, time driven variables such as the Outlet Canal and temperature driven variables

such as the Inlet Water Temperature were determined. The system was eventually reduced to four independent generators and an outlet canal.

Each study influenced the others to the point that each successive understanding required revisiting previous studies. By example, the effects of Load in Section 5.1 were not fully understood until Section 5.4, "Effects of Pump Status" and Section 5.5, "Effects of Time Granularity" were visited. Similarly, the effects of pump status in Section 5.4 were not recognizable until Section 5.1, "Effects of Load" determined that there was a correlation between Load and Temperature. The interaction between the studies is captured in this chapter; the iterative steps and backtracking are largely left as an exercise due to the volume of data required and the limited usefulness to understanding the final neural model architecture.

Each section of this chapter is intended to prove a different concept. Section 5.8, "Evaluating Performance," culminates the work by demonstrating the effectiveness of the Feature Selection process and producing evidence that the resulting neural models are able to both generalize across similar feature values and to abstract through noisy data. Generalization and abstraction capabilities are established by measuring the features against statistically separated out-of-sample data sets. Section 5.9, "Feature Selection Summary," describes the effectual features for the power plant.

The neural network results presented in this work are evaluated based on the following variables:

- Number of Runs (n): The number of validation sets represented by the statistical descriptions.
- Cascaded Node Count (CNC): The number of cascaded nodes for each of the (n) models built for this section. The number of cascaded nodes reflects the linearity discovered among the input variables to the output variables and represents the complexity of the mapping function (e.g., more cascaded nodes results from a more complex map).
- Training Set Size (Tr-SS): The number of records actively used by the weight selection process within the network engine. Unless explicitly stated, the Training set selection is automated through a random number generator built within the neural modeling tool.

- Test Set Size (Te-SS): The number of records used by the network algorithm for network evaluation purposes. Unless explicitly stated, the Test set selection is automated through a random number generator built within the neural modeling tool.
- Validation Set Size (V-SS): The number of records isolated from the network during training and testing. The Validation set is always composed of data that is kept independent from the training and testing sets. In the case of same-month (in-sample) data, the Validation set is selected as the remaining records after Test and Training set selection. In the case of disparate-month (out-of-sample) data, the Validation set is composed of all non-error flagged data for the particular month. The Validation set is evaluated for each of (n) networks.
- Average Error (AE): The mean of the average absolute errors obtained by the (n) models built for this section. The average absolute error of each model is the mean of the absolute difference between each computed value and the corresponding measured value for that record. The AE is a measure of the accuracy with which the network algorithm is able to evaluate the function of input variables.[15]
- Maximum Error (ME): The maximum error sustained by any of the (n) models built. Different from the Average Error which reduces comparison dependencies on single outliers; the ME identifies the most significant outlier among the (n) models.[16]
- Coefficient of Variation (CV): The sample coefficient of variation of the Average Error for the (n) model test sets. The CV reflects the precision (or repeatability) with which the (n) neural networks mimic one another (i.e., the precision among all model results), and reflects neither the accuracy nor precision of the models to the expected value. A large coefficient of variation is indicative of a non-repeatable process.

---

[15] Accuracy is a measure of how closely a computed value agrees with the true value. Precision is a measure of how closely the computed values agree with one another. A highly precise model may be poorly accurate [Chapra].

[16] By example, one network posted a 0.286 degree Fahrenheit Average Error, and a 2.88 degree Fahrenheit Maximum Error. However, the top five outliers were [2.88, 2.62, 2.23, 2.07, 1.99]; removing these five of 2000 records (or 1/4 of one percent) resulted in a Maximum Error of less than 1 degree Fahrenheit.

However, a poorly accurate set of (n) networks may still be precise to one another and demonstrate a low CV. This is different than the RMS Midrange Error (MID) and Average Error (AE), each of which indicate the accuracy with which the calculated output value reflects the true value.

- RMS Midrange Error (MID): The midrange Root Mean Squared error among the (n) testing sets. The midrange is calculated as the average of the largest and the smallest RMS error found among the (n) models. The RMS error indicates the accuracy with which the network algorithm is able to evaluate the function of input variables. This is different than the Coefficient of Variation (CV), which is the accuracy with which each network corresponds to one another.

5.1 Effects of Load

The Big Bend power system is composed of four individual and autonomous coal-fired generation units. Individual neural network models were built for each generator in an effort to recognize load effects on water temperature. The September 1999 data set was selected through random number generator.

Figure I, "Load and Temperature Increase Reflected Over Time," reflects Minute granularity collected data within Big Bend Unit One (BB1). The Load is supplied by BB1, and the Temperature Increase is the temperature difference between the Inlet Water Temperature and the Outlet Water Temperature. Note that there appears to be a significant correlation between the Load Increase and the Temperature Increase. There is not a time delay because the water velocity is quite large (although unmeasured).

The graph includes a pump outage instance with a corresponding load reduction to less than 250 Megawatts, as discussed in Section 5.4, "Effects of Pump Outage." The pump outage was excluded from network training and subsequent validation.

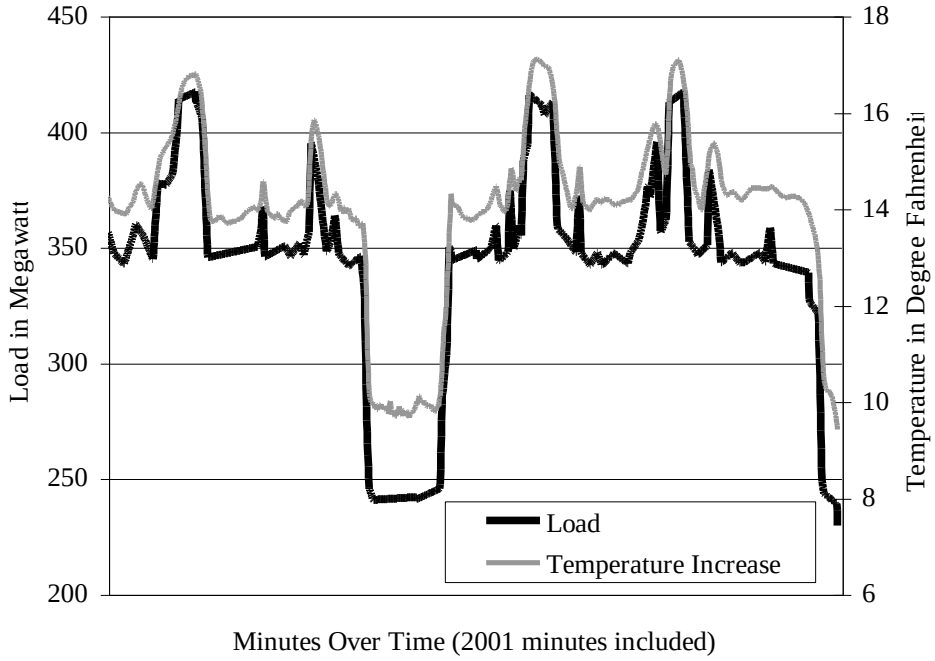

Figure I. Load and Temperature Increase Reflected Over Time

Neural training produced models that were able to generate results to within an Average Error of less than 1 degree Fahrenheit for generator BB1, as depicted in Table A, "Effects of Load." (See also Appendix B, "Independent Generator Analysis.")

Table A. Effects of Load

Data Set	AE	ME	CV	MID	SS
	degree F	degree F	-	degree F	-
Test	0.200	0.902	7.5	0.275	184
IS Validation	0.203	0.921	14.3	0.276	68
	430	Train Set Size (Tr-SS)			
	5	(n)			
0, 0, 0, 0, 2		Cascaded Node Count (CNC)			

The input variable was Generator Load; the network output variable was Temperature Increase (one input, one output). Figure J, "Cascade Correlation 1x2x1," demonstrates a sample single input, single output cascaded network with two internal nodes.

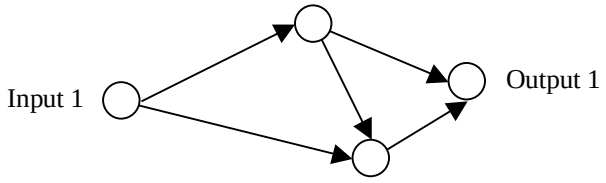

Figure J. Cascade Correlation 1x2x1

In summary, Load is an important variable for system wide modeling. Furthermore, no-load (or very low load) generator situations may be deceptive. For example, if a generator trips or goes out of circuit, thereby supplying no load, significant heat must be removed by the condenser cooling water.[17] Therefore, situations of low load are considered noise, and are removed from training and validation sets.

5.2 Effects of Generator Selection

Each generator is a large physical system with many moving parts and friction points. In addition, the generators at Big Bend are of various vintages dating from 1970 through 1985.

Neural analysis revealed that each generator requires an independent forecasting model to accurately predict water temperature based on load. For example, Table B, "Effects of Generator A Train on Generator B Validation" demonstrates that a BB1 trained model cannot generalize to the BB4 load to temperature patterns, nor can a BB3 trained model generalize to BB4 requirements. The input variable was Load; the output variable was Temperature Increase (one input, one output). Figure J, "Cascade Correlation 1x2x1," demonstrates an example cascaded network with 1 input, 1 output, and 2 internal nodes.

[17] In fact, all heat must be removed by the condenser if the generator goes out of service, since no heat is removed by the generator turbine itself. See Section 3.4, "Coal Fired Base Load Power Generation."

Table B. Effects of Generator A Train on Generator B Validation

BB1 Train, BB4 Validation			CNC = (0, 1, 1, 2)		
Data Set	AE	ME	CV	MID	SS
	degree F	degree F	-	degree F	-
BB1 IS	0.209	1.381	6.7	0.317	215
BB4 OOS	1.153	14.823	8.7	1.977	1102
	1364	Train Set Size (Tr-SS)			
	585	Test Set Size (Te-SS)			
	4	(n)			

BB3 Train BB4 Validation			CNC = (0, 0, 0, 0)		
Data Set	AE	ME	CV	MID	SS
	degree F	degree F	-	degree F	-
BB3 IS	0.331	1.340	24.1	0.412	286
BB4 OOS	2.256	4.050	8.8	2.310	1102
	1808	Train Set Size (Tr-SS)			
	755	Test Set Size (Te-SS)			
	4	(n)			

The BB3 unit exhibits irregular load patterns with swings between 440 Megawatt and 450 Megawatt, etc. These patterns are possibly due to sensor problems. Although BB3 demonstrated a significantly noisy load reading, the BB3 neural models were able to abstract through the noisy load input variable, demonstrating in-sample Average Errors of 3/10ths of 1 degree. See Figure K, "BB3 Load Noise," for an image of BB3 load based noise. Generalizing to out-of-sample data for individual generators is presented in Section 5.3, "Effects of Inlet Water Temperature."

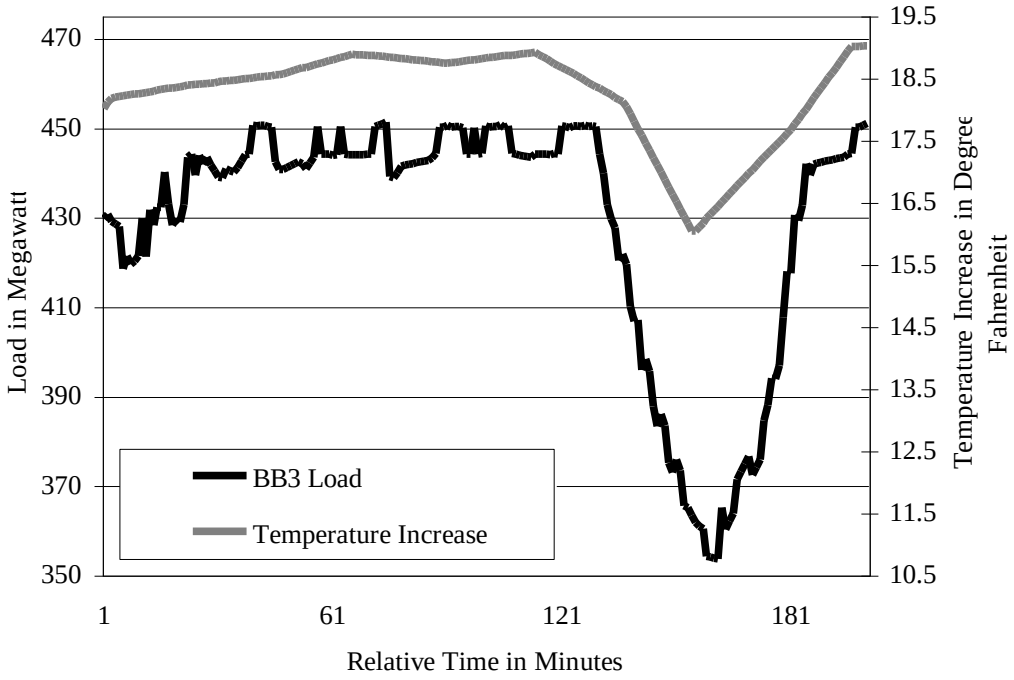

Figure K. BB3 Load Noise

Exhaustively testing and validating each of the 12 generator combinations is not necessary, since the finding that there exists at least two generators that require independent analysis is sufficient cause to always require independent analysis. In summary, each generator must be modeled separately in order to produce the most effective system-wide models.

5.3 Effects of Inlet Water Temperature

The condenser cooling water is the salt water that travels between the Inlet Canal and the Outlet Canal, and is measured as the Inlet Water Temperature and the Outlet Water Temperature, respectively. The condenser cooling water passes over a water conductive radiator to remove heat from the closed-loop turbine-water system. By analogy, the condenser is similar to a radiator in a car, the turbine water like the closed-loop antifreeze system, and the condenser cooling water like the air that passes over the radiator. The desire is to reduce the turbine steam temperature to produce a phase change to liquid water; a direct result is that the condenser cooling water absorbs that heat. See Section 3.4, "Coal Fired Base Load Power Generation," and Figure F, "Power Cycle Boiler Design," for a review of condenser and generator function.

Prototype networks were trained with various data in order to confirm the non-effect of absolute Inlet Water Temperature.[18] The data represented in Table C, "IWT September to February," demonstrates that networks trained with September data perform to within an Average Error differential of less than ½ of 1 degree Fahrenheit on out-of-sample target data in February. The data represented in Table D, "IWT February to September," demonstrates that networks trained with February data perform within an Average Error differential of less than ½ of 1 degree Fahrenheit on out-of-sample target data in September. The September 1999 "Summer" and February 1999 "Winter" month selections were determined through random number generator. September IWT ranged [87, 90] degrees Fahrenheit, while February IWT ranged [70, 75] degrees Fahrenheit, a difference of from 12 to 20 degrees Fahrenheit.

Table C. IWT September to February

Data Set	AE	ME	CV	RE	SS
	degree F	degree F	-	degree F	-
Sep IS	0.394	2.960	13.9	0.562	553
Feb OOS	0.479	1.530	1.5	0.588	1000
	3362	Train Set Size (Tr-SS)			
	1442	Test Set Size (Te-SS)			
	5	(n)			
	0, 0, 0, 1, 1	Cascaded Node Count (CNC)			

Following the pattern discovered in Section 5.1, "Effects of Load," each neural network was constructed with a single input variable as generator load and a single output variable as the difference between IWT and OWT. (See Figure J, "Cascade Correlation 1x2x1.") The models were built with considerably larger training sets to avoid over-fitting the data and to reduce the impact of outlier (noisy) populations. The Maximum Error for the September in-sample validation (Table C, "IWT September to February," identified as "Sep IS") and the September out-of-sample validation (Table D, "IWT February to September," identified as "Sep OOS") indicate difficulties within the September data. The September data difficulties are revisited in Section 5.6, "Effects of Load to Temperature Time Delay," and Section 5.8, "Evaluating Performance."

[18] The heat capacity per gram of liquid water, or the specific heat, is constant. Specific heat is the heat required to raise the temperature of 1 gram of a substance 1 degree Centigrade. For additional information, see a general chemistry book such as [Brady].

Table D. IWT February to September

Data Set	AE	ME	CV	RE	SS
	degree F	degree F	-	degree F	-
Feb IS	0.415	1.750	5.1	0.559	500
Sep OOS	0.646	4.550	4.7	0.921	1000
	3650	Train Set Size (Tr-SS)			
	1350	Test Set Size (Te-SS)			
	5	(n)			
0, 0, 1, 1, 2		Cascaded Node Count (CNC)			

These models demonstrate a correlation between the in-sample validation set selected within the modeling month and an out-of-sample validation set selected from a month with a significantly different IWT. Since the correlation between the errors is within an average of less than ½ of 1 degree Fahrenheit, the seasonal effect of Inlet Water Temperature is considered negligible. The September validation Average Error rests within 1/10th of 1 degree Fahrenheit of the February manual (0.394 compared to 0.479 degree Fahrenheit). The February validation Average Error rests within ¼ of 1 degree Fahrenheit of the September manual (0.415 compared to 0.646 degree Fahrenheit). Removing the absolute Inlet Water Temperature from the input data accomplishes three significant goals:

- It eliminates the necessity of seasonal network training (i.e., the absolute Inlet Water Temperature has no significant impact on the network performance),
- It effectively reduces the input variable count by one, and
- It removes the burden of finding a representative sample of Inlet Water Temperature conditions.

In summary, the Inlet Water Temperature is important as a relative measure to Outlet Water Temperature. Preprocessing calculations to create the Inlet Water Temperature to Outlet Water Temperature difference will accommodate the final solutions.

5.4 Effects of Pump Status

Figure L, "Pump Status as Reflected in Cooling Water Temperature," exposes the Inlet Water Temperature to Outlet Water Temperature difference during pump outage conditions. At time 60, Pump 1 is taken out of service. The immediate impact is a significant increase in water temperature. Since there are two pumps, the condition is

accompanied by a 50% reduction in water volume. Near time 110, Pump status conditions change. In this case, at time 105, Pump 1 is officially brought in service, and at time 112, Pump 2 is taken out of service. At time 154, Pump 2 is officially brought in service. Note that the pumps appear restored to service about 10 minutes before the official Pump 1 status ON at time 105, and Pump 2 status ON at time 154, revealed by a sharp decrease in temperature. Then, between times 154 and 176, a gradual reduction in temperature occurs, after which time no residual effects remain.

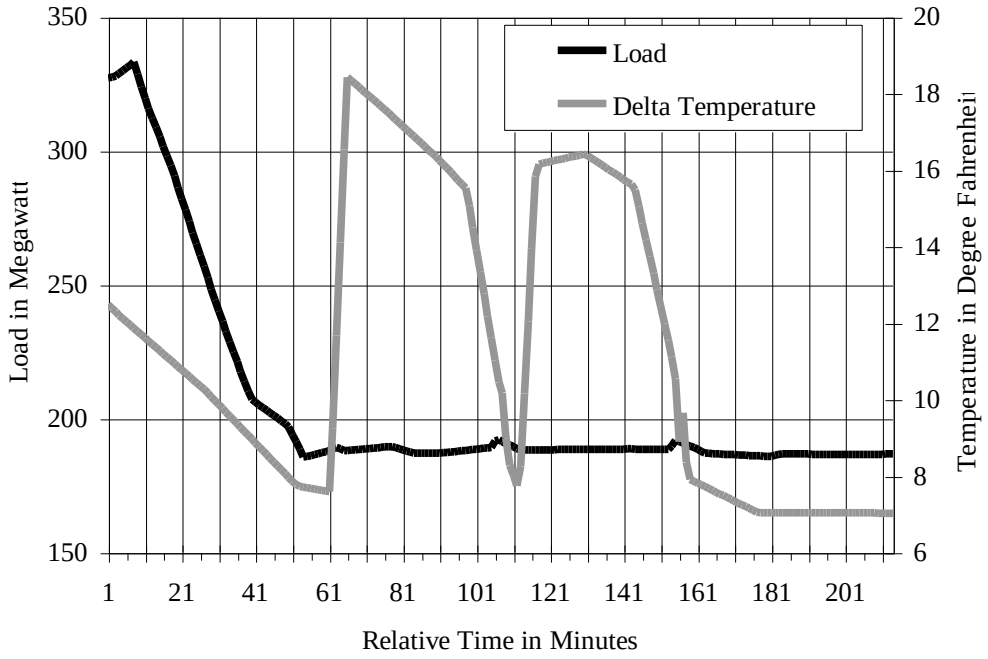

Figure L. Pump Status as Reflected in Cooling Water Temperature

Pump maintenance outages are typically scheduled during off-peak times. Furthermore, pump outage conditions can be recognized as a systematic and predictable noise. Since this study is especially devoted to identifying peak water temperatures during normal operation, removing the pump outage data will enable the neural prediction models to examine in-service conditions more closely.

In summary, pump status has a significant effect on evaluating water temperature based on load. Pump outages are handled in the system wide neural architecture through preprocessing removal. Section 5.6, "Effects of Load to Temperature Time Delay," demonstrates a nominal Load to Outlet Canal Temperature delay of two hours and

residual effects that linger for at least an hour longer. Therefore, pump outage preprocessing must remove at least three hours of data to be remotely effective.

5.5 Effects of Time Granularity

The time granularity is herein defined as the segment of time between successive records. The data-collection time granularity influences the problem in many ways. The benefits of a large granularity include:

- Capturing more variable range with each record since the time slice observes additional variations (i.e., reduces the required number of data records), and
- Reducing exposure to individually noisy variables since the time slice observes a larger averaged segment (i.e., "forced" abstraction through averaging noisy data.)

The benefit of a small granularity include:

- Increasing the ability to observe small variations in input variables (i.e., generalizing), and
- Allowing the neural network to isolate outliers (i.e., "natural" neural computing abstraction by presenting noisy data).

Neural models built on single generators revealed that 10-minute time slices resulted in significantly larger errors compared to 1-minute time slices (Average Error, Maximum Error, Coefficient of Variation, and RMS-MID). The networks were built on single generator data with one input variable (Generator Load) and one output variable (Temperature Difference, or OWT subtracted by IWT; see Figure J, "Cascade Correlation 1x2x1."). The 10-minute model results are depicted Table E, "Ten Minute Time Granularity." The 1-minute model results depicted in Table F, "One Minute Time Granularity" are duplicated from Section 5.1, "Effects of Load" for completeness. Figure M, "Analyzing Time Granularity Graph," visually depicts the error measurements for time granularity.

Table E. Ten Minute Time Granularity

Data Set	AE	ME	CV	MID	SS
	degree F	degree F	-	degree F	-
Test	1.414	8.314	22.6	2.074	567
IS Validation	1.359	8.234	18.4	1.911	209
	1321	Train Set Size (Tr-SS)			
	5	(n)			
	0, 0, 2, 3, 9	Cascaded Node Count (CNC)			

Table F. One Minute Time Granularity

Data Set	AE	ME	CV	MID	SS
	degree F	degree F	-	degree F	-
Test	0.200	0.902	7.5	0.275	184
IS Validation	0.203	0.921	14.3	0.276	68
	430	Train Set Size (Tr-SS)			
	5	(n)			
0, 0, 0, 0, 2		Cascaded Node Count (CNC)			

In summary, time granularity is important. The generator under study experienced a maximum load increase of 55 Megawatts over a ten-minute period within the data under investigation, and a maximum load decrease of 93 Megawatts over the same training data set. The significant load swings affected the neural network models. Therefore, network models must be built to accommodate the benefits of large time granularity, and at the same time must result in both trainable and useful networks. In these studies, one-minute granularity is adopted unless otherwise noted.

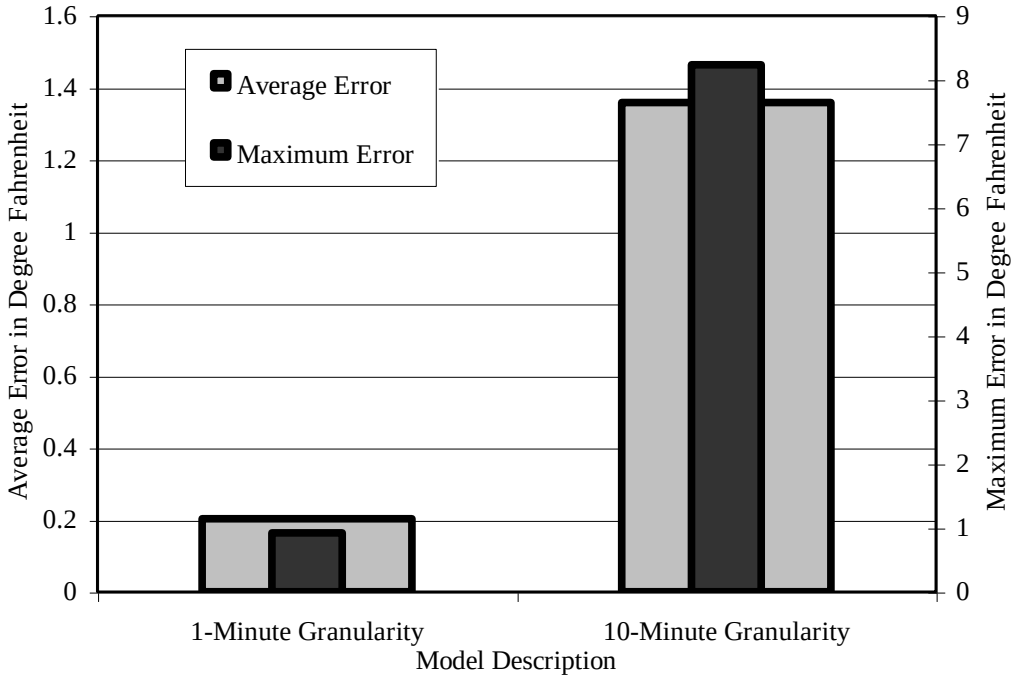

Figure M. Analyzing Time Granularity Graph

5.6 Effects of Load to Temperature Time Delay

This section is composed of several building block parts that culminate into a final proposed neural architecture. A summary of this section is included as Section 5.6.4, "Time Delay Summary."

The Big Bend plant is built on a manufactured peninsula. One side of the peninsula is used as the cooling water Inlet Canal, while the other side is the cooling water Outlet Canal.

The plant Outlet Canal is integrated as four inlets or tributaries and a single outlet or mouth. Each inlet identifies with a separate one of four generators. The inlets integrate at one end of the Outlet Canal and collectively flow through the canal towards the Outlet Canal Temperature sensors and eventually into the waters of Tampa Bay. During the transition through the Outlet Canal, the water mixes and radiantly cools.

Figure N, "Outlet Canal," is a satellite image of the Big Bend Outlet Canal. The image represents an area that is 1200 yards wide and 800 yards high. As demonstrated in the figure, there is about 800 yards between the Generator 1 Outlet Tube (designated by the right-most white vertical bar) and the Outlet Canal Temperature probes (designated by the left-most white vertical bar).

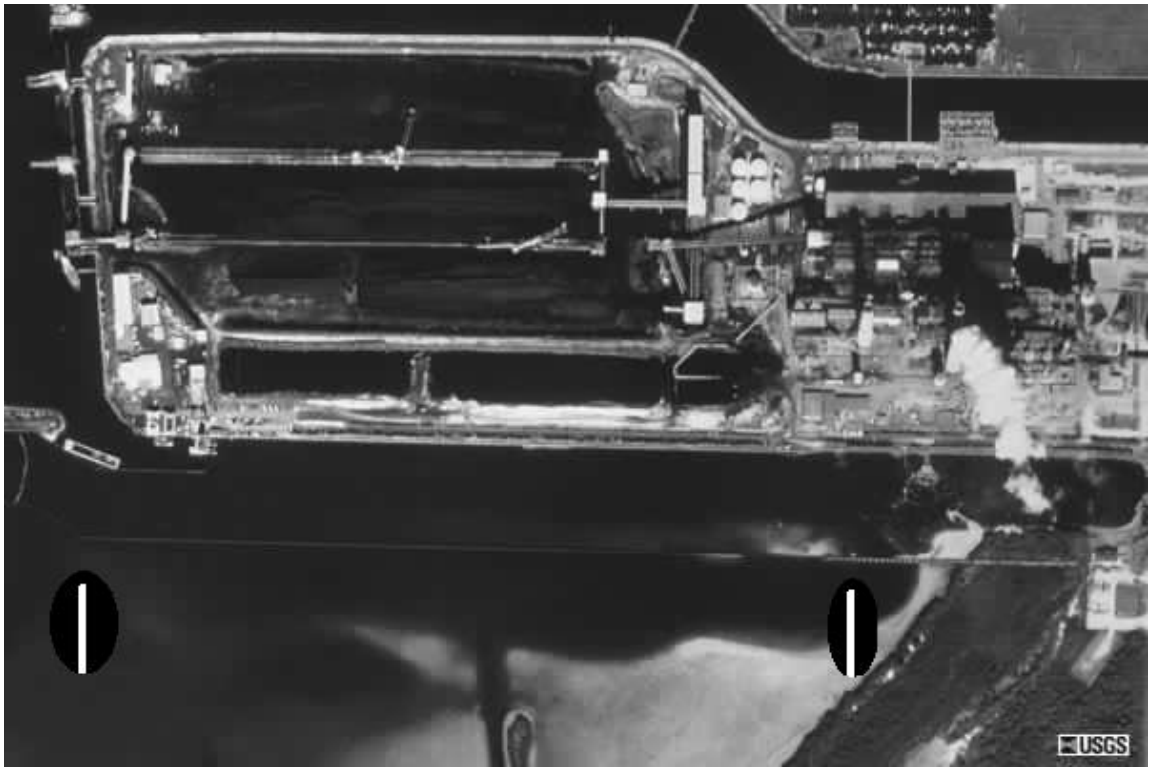

Figure N. Outlet Canal[19]

Although there exists undetermined quantities in the Outlet Canal delay equation, it is obvious that there does exist a delay between the Load and the corresponding outlet canal water temperature (OCT). Unfortunately, flow rate measurement is not part of the Big Bend plant data-collection system. Attempts to measure flow rate based on surface movement (i.e., a floating log or other object) are not possible due to wind, tide, and other environmental effects.

Neural computing is designed to recognize patterns in input data sets. However, when the target system introduces time delayed data, the neural computing engine architecture is designed to accept data that has been time delay preprocessed. The processing must occur off-line, before the data is introduced to the neural engine. In studying the outlet canal, various time delay accommodations were attempted in order to find a "best fit" for the data. The strategy is that the "best fit" time delay scenarios describe the best time delay values for representing the system.

[19] This infrared image is courtesy of United States Department of Geological Survey, through Microsoft Terra Server [United]. Latitude 27N 47' 39", Longitude 82W 24' 18".

The "best fit" scenarios were made more difficult due to the nature of the collected data. Much of the data was determined quite noisy even after significant preprocessing removal. A desired data graph is reflected in Figure O, "Regular Loading and Temperature Increase Curve."

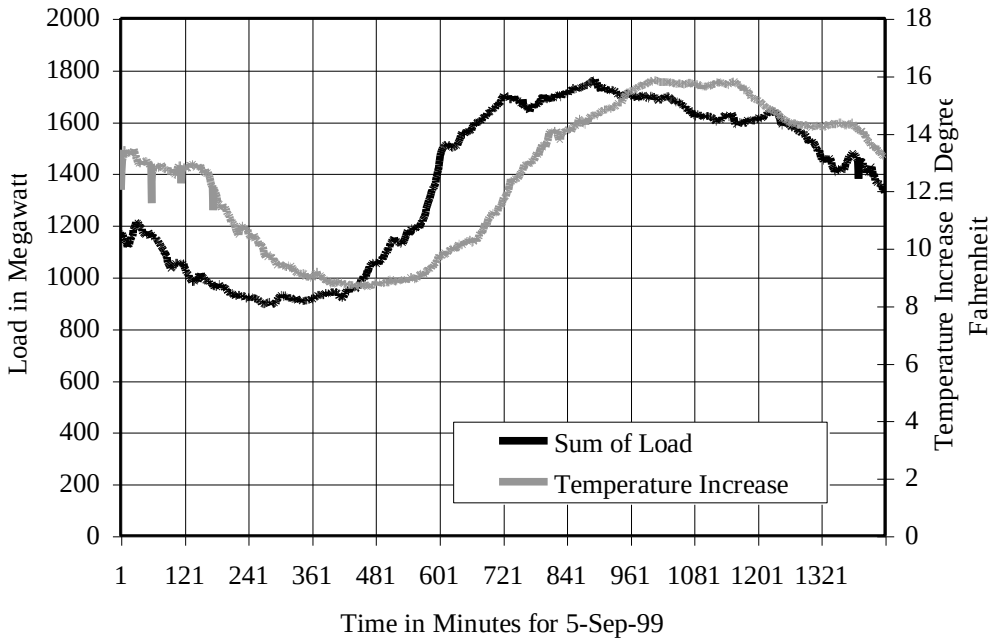

Figure O. Regular Loading and Temperature Increase Curve

The graph depicted in Figure P, "Regular Loading and Temperature Increase Shifted 120 Minutes," depicts the temperature delayed 120 minutes from the point at which a certain load is experienced. Note that the Load and the Temperature Increase curve visually correspond to one another with only slight irregularities.

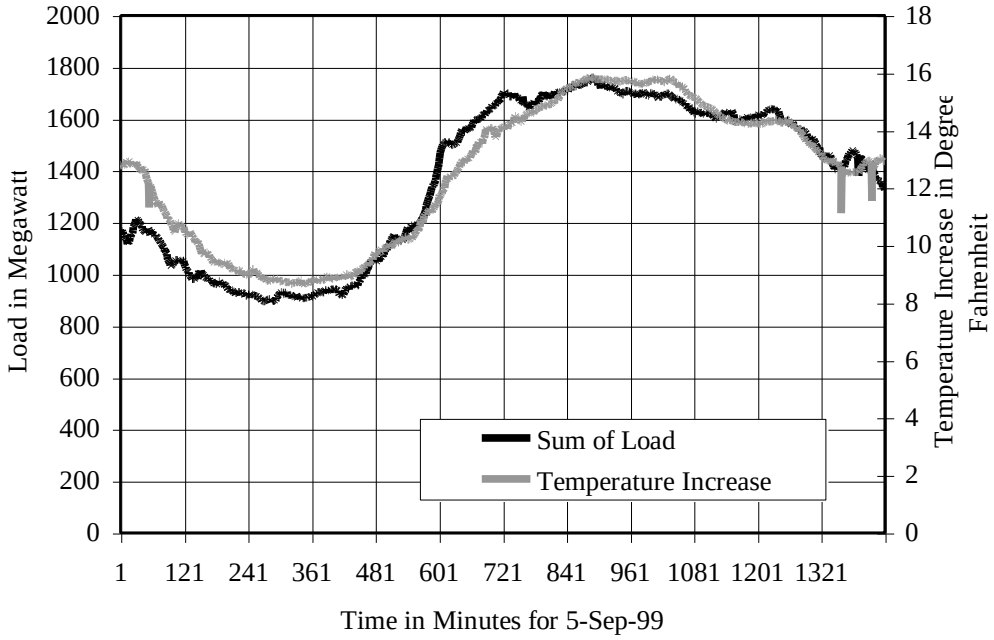

Figure P. Regular Loading and Temperature Increase Shifted 120 Minutes

Figure Q, "Irregular Loading and Temperature Increase Curve," reflects two of many noisy and difficult to train days for the September training data. The source of the noise is attributed to sensor failures.

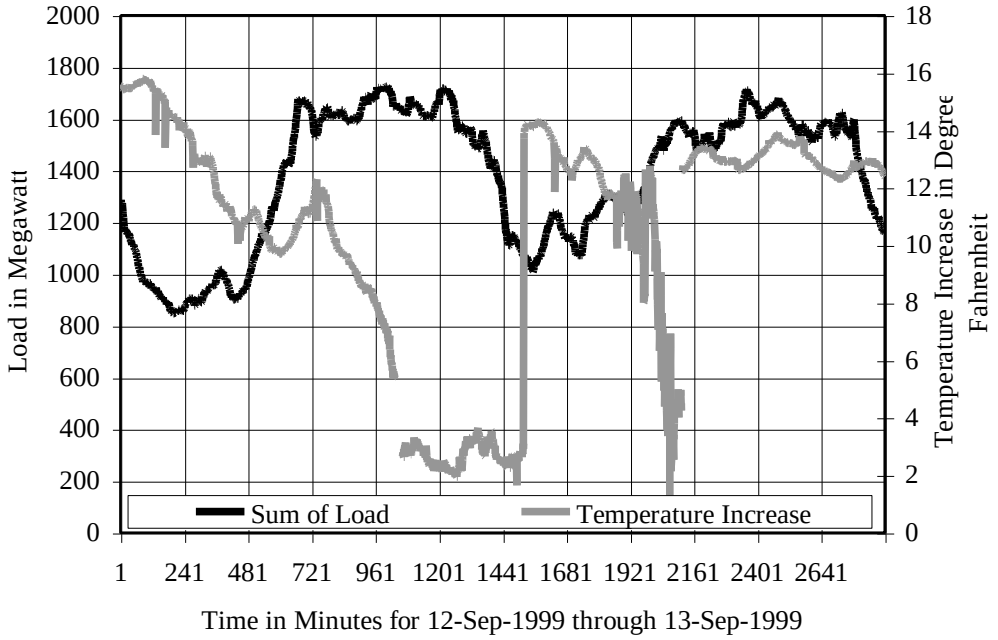

Figure Q. Irregular Loading and Temperature Increase Curve for September

Even after removing easily observed difficult days, models trained with September 1999 data were inconclusive for deriving time delay due to the nature of the error measurements found among models. The models built with time delay from 60 minutes to 240 minutes following the theory described in Section 5.6.1, "Equal Time Delay" resulted in comparably equal Validation Set Average Errors. The Average Errors for the September 1999 data are depicted in Figure R, "September Equal Time Delay Validation Sets." The goal was to determine "best fit" training based on error measurements; Figure R demonstrates no "best-fit" from 60 to 240 minutes. As a direct result of this inability to abstract through the significant noise in the September 1999 data set, an alternate month was randomly selected.

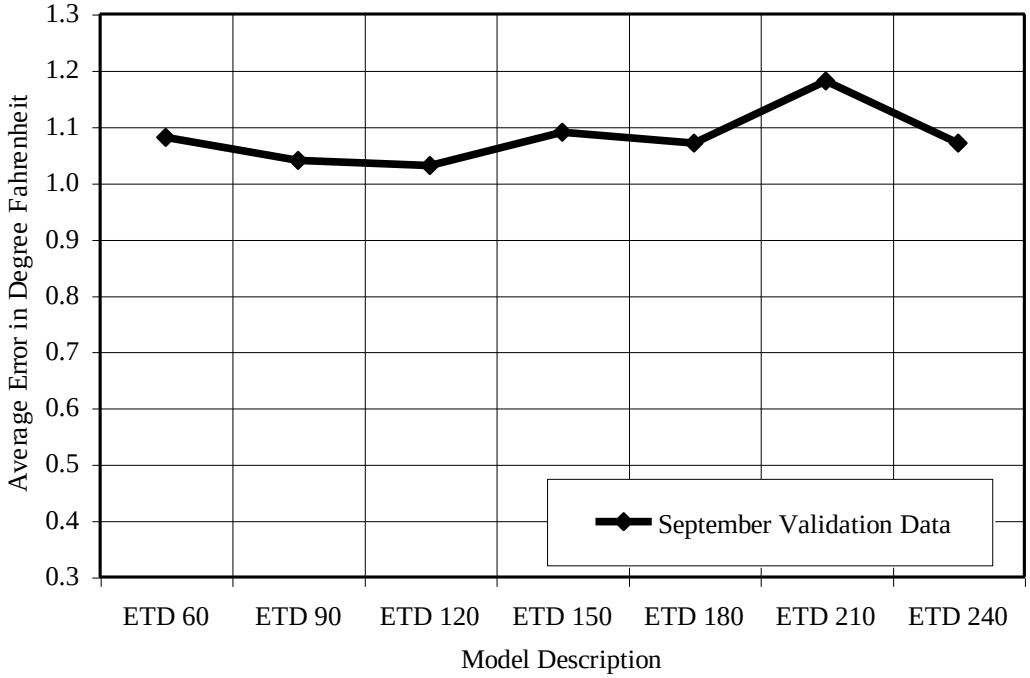

Figure R. September Equal Time Delay Validation Sets

The following sections investigate appropriate time delay values between the load (increasing water temperature) to the water temperature observed at the Outlet Canal Temperature sensors. The models that follow are constructed with February 1999 data, selected by random number generator.

5.6.1 Equal Time Delay

The single-point, Equal Time Delay models (ETD) capture differences in generator efficiencies by including each of the four generator load measurements as separate neural network input variables, pursuant to Section 5.2, "Effects of Generator Selection." The goal of this ETD section is to use rapid prototype neural models in determining the outlet canal delay.

The delay figures for these models are built based on the shifted loading graph in Figure P, "Regular Loading and Temperature Increase Shifted 120 Minutes." The in-sample results for the Equal Time Delay model are listed in Table G, "Analyzing Equal Time Delay." All models were built with approximately 6500 total records, separated into 10% Validation set (approximately 650 records), 30% of the remaining as Test set (approximately 1750 records), and the remainder as Training set (approximately 4100

records). The models are not trained based on exactly equal training sets since the time shift results in different composed records. After the time shift, records with errors and omissions are removed before being applied to the neural training tool. The preprocessing technique is described in Appendix A, "Data Preprocessing."

Table G. Analyzing Equal Time Delay

ETD 75 Minutes			CNC = (0, 0, 3, 4, 4, 4, 4, 8)		
Data Set	AE	ME	CV	MID	(n)
	degree F	degree F	-	degree F	-
IS Validation	0.506	2.410	9.5	0.638	8
ETD 90 Minutes			CNC = (0, 0, 1, 1, 3, 4, 5, 8)		
	AE	ME	CV	MID	(n)
IS Validation	0.443	1.920	10.0	0.552	8
ETD 105 Minutes			CNC = (1, 1, 1, 2, 2, 4, 6, 7)		
	AE	ME	CV	MID	(n)
IS Validation	0.370	1.680	11.1	0.452	8
ETD 120 Minutes			CNC = (0, 0, 0, 0, 1, 1, 2, 2)		
	AE	ME	CV	MID	(n)
IS Validation	0.372	1.400	4.9	0.468	8
ETD 135 Minutes			CNC = (0, 0, 0, 0, 0, 2, 2, 4)		
	AE	ME	CV	MID	(n)
IS Validation	0.373	1.390	7.2	0.448	8
ETD 150 Minutes			CNC = (0, 0, 0, 2, 2, 2, 2, 8)		
	AE	ME	CV	MID	(n)
IS Validation	0.410	1.560	5.1	0.495	8
ETD 165 Minutes			CNC = (1, 1, 2, 2, 4, 4, 6, 11)		
	AE	ME	CV	MID	(n)
IS Validation	0.444	1.760	6.9	0.544	8

The model architecture is constructed as a 4 input single output cascaded mesh, where each input identifies with one of the 4 generator load readings and the output is the Outlet Canal Temperature differential (OCT – IWT). Figure S, "Cascaded Correlation 4x3x1," demonstrates an example four input, single output cascaded architecture with 3 internal nodes.

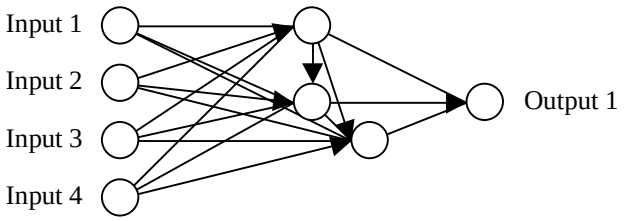

Figure S. Cascaded Correlation 4x3x1

The Equal Time Delay models are executed to determine a delay pattern correlation for the data. The best delay correlation is between 105 and 135 minutes, as demonstrated by Figure T, "Analyzing Equal Time Delay Validation Sets." By 165 minutes delay, the correlation decays to reflect a "less fit" solution.

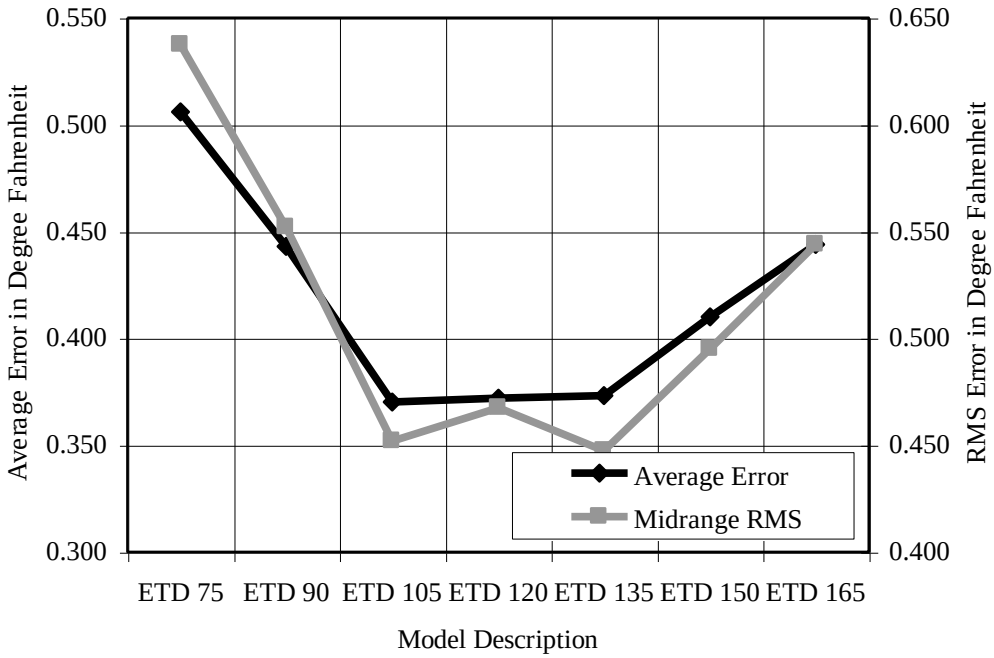

Figure T. Analyzing Equal Time Delay Validation Sets

In summary, the Equal Time Delay models demonstrate that the physical delay caused by the Outlet Canal is between 105 and 135 minutes, and its residual effects quickly decay after this time. The modeling results correspond to the visual results observed in Figure P, "Regular Loading and Temperature Increase Shifted 120 Minutes."

5.6.2 Staggered Time Delay

The Equal Time Delay models capture a significant portion of the delay through the canal. The generator discharge pipes, however, are not coincidental. Instead, the generator discharge pipes are progressively further away from the canal mouth, as demonstrated by the Figure U, "Generator Outlet Water Pipe Discharge." The leftmost white vertical bar indicates the location of the Generator 1 discharge, the next is Generator 2, then Generator 3, and the rightmost vertical bar identifies Generator 4 discharge.

Therefore, each generator Load to Outlet Canal Temperature delay is actually represented by a unique time value. Also note that each generator discharge pipe is effectively a stream (or tributary) flowing into a single river; as each stream merges into the final river, the river flow-rate increases.

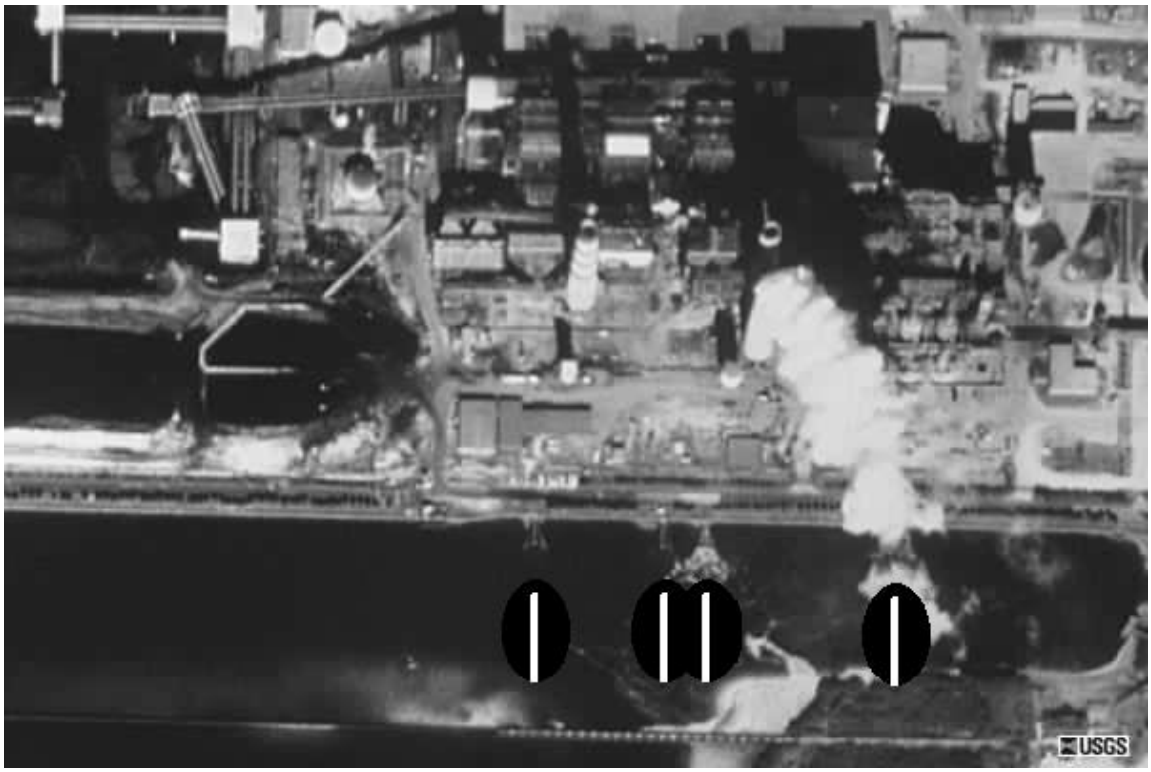

Figure U. Generator Outlet Water Pipe Discharge

Once each outlet pipe discharges the water into the Outlet Canal, the water mixes with the other generator waters. The temperature increase caused by the independent generators cannot be discerned by the time the water flows into Tampa Bay. The water

mixes, and the water tends to a common temperature. For example, if one generator is out of service, there is not a corresponding void in the Outlet Canal Temperature recordings. Instead, if the Cooling Pumps are active during the generator outage, the Outlet Canal Temperature will be reduced. If the Cooling Pumps are inactive during the generator outage, the volume of water introduced to the Outlet Canal will be reduced and cause a reduced flow rate that results in a longer delay period.

Any temperature gradients will tend to disappear over time. As the cooling water travels down the outlet canal to the canal mouth, the stratification between each generator Outlet Water Temperature is reduced due to a mixing effect.

In addition, the water radiantly cools, much as a hot cup of coffee will cool to room temperature given enough time so that both bodies are eventually at the same temperature. There is not enough time for the canal to experience this much temperature change, but the effect is observed between the Outlet Water Temperature discharged into the canal, and the Outlet Canal Temperature measured at the end of the canal.

The mixing effect and various cooling effects cause some difficulty in direct time delay measurements since there is not a one-to-one correspondence between Load and eventual Outlet Canal Temperature. If it were possible to adjust each generator Load separately, then either a spike or reduction in Load could be observed as a spike or decay in the Outlet Canal Temperature; unfortunately, increased system load typically affects all operational units. In addition, during the transfer through the outlet canal, the water loses heat energy to the air and surrounding structures.

The Staggered Time Delay model (STD) attempts to capture the time delay applicable to each generator and thereby extends the Equal Time Delay models. The ETD models used in establishing STD models are listed in Section 5.6.1, Table G, "Analyzing Equal Time Delay," and Figure T, "Analyzing Equal Time Delay Validation Sets." Since the ETD models recognized more "fit" patterns between 105 and 135 minutes, the STD model is built surrounding these delays. The Staggered Time Delay model attempts to formulate effective time delay solutions and capture the delay between the independent Outlet Water Pipes. The results of one Staggered Time Delay solution model with BB1 latency at 110 minutes, BB2 at 115 minutes, BB3 at 120 minutes, and BB4 at 125 minutes is listed Table H, "Analyzing Staggered Time Delay."

Table H. Analyzing Staggered Time Delay

L1(110), L2(115), L3(120), L4(125)					
Data Set	AE	ME	CV	MID	SS
	degree F	degree F	-	degree F	-
Test	0.371	1.500	3.8	0.458	1723
IS Validation	0.369	1.400	3.4	0.461	637
	4019	Train Set Size (Tr-SS)			
	8	(n)			
0, 0, 0, 0, 0, 1, 2, 2		Cascaded Node Count (CNC)			

There are a potentially unlimited number of Staggered Time Delay models. Neural computing techniques in the next section help to avoid an exhaustive test and validation for all time delay models, and this avoidance makes the neural computing paradigm an appropriate investigation for this problem. In summary, the independent Staggered Time Delay models may produce results that are more sophisticated, but at the expense of exhaustively programming and validating each model.

5.6.3 Multi-Staggered Time Delay

The Multiple Point, Staggered Time Delay model (MSTD) attempts to allow the neural software to capture the time delay between Outlet Water Pipes and the bay.

The MSTD model architecture consists of 28 input nodes and a single output node, with 0 or more cascaded internal nodes. The 28 input nodes are comprised of 7 time slots for each of 4 generators. The 7 time slots are positioned at 105, 110, 115, 120, 125, 130, and 135 minutes pursuant to the "best fit" discovered in Section 5.6.2, "Staggered Time Delay." The 4 generators are modeled independently as described in Section 5.2, "Effects of Generator Selection."

As described in Section 2.7.3, "Cascade Correlation," the Cascade Correlation architecture consists of dynamically constructed, forward-looking fully connected input, internal, and output nodes. Figure V, "Cascade Correlation 28x2x1," demonstrates a 28 to 1 cascade correlation architecture with 2 internal nodes. The input nodes are labeled "BB xx - yy," where xx is the generator identifier and yy is the time slot as described previously. The output node is identified as "OCTD" signifying the Outlet Canal Temperature Difference. The OCTD is the Outlet Canal Temperature measured at the Outlet Canal demarcation to Tampa Bay subtracted by the Inlet Water Temperature

measured at the Inlet Water Pipe, or OCT - IWT. See Section 5.3, "Effects of Inlet Water Temperature."

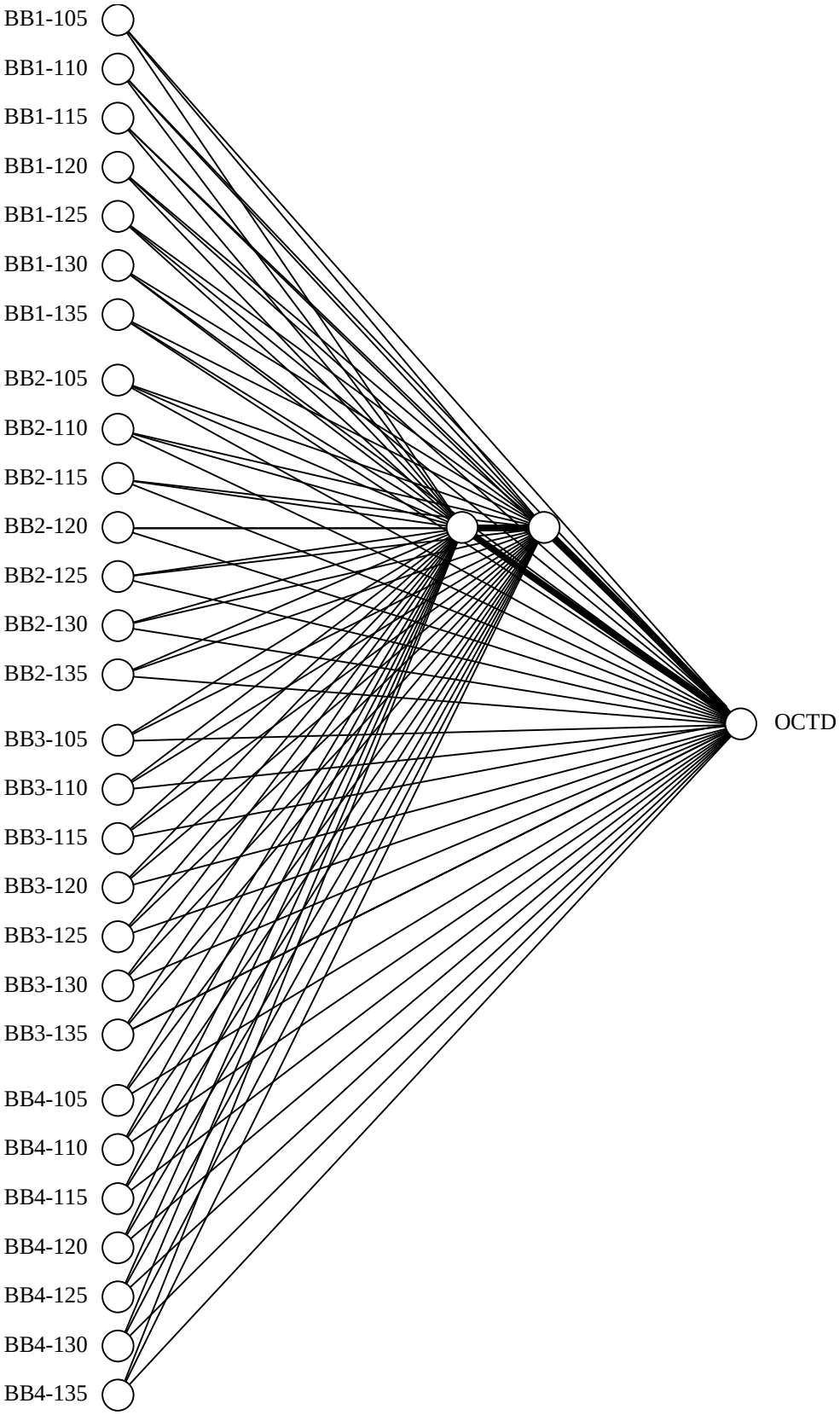

Figure V. Cascade Correlation 28x2x1

The number of weighted links (not including bias weights) contained in a particular 28-1 model architecture is described by the formula:

$$\text{Link Count} = (\text{Input Nodes}) * (\text{Internal Nodes} + 1) + \Sigma\, i$$

where i = 0 to Internal Node Count

Table I, "Model Architecture Link Count," iterates the number of weighted links for several sample architectures based on the internal node count. As discussed in Section 2.7.1, "Training Set Size," the ratio between the training set size and the number of weighted links should be at least 5, and preferably 10 or more if possible. Therefore, a MSTD cascaded network with 10 internal nodes should be trained with no fewer than 1615 records, and preferably more than 3630 records.

Table I. Model Architecture Link Count

Internal Nodes	Weighted Links	Training Set (x5)	Training Set (x10)
0	28	140	280
1	57	285	570
2	87	435	870
3	118	590	1180
5	183	915	1830
10	363	1815	3630
15	568	2840	5680
20	798	3990	7980
30	1333	6665	13330

The Multi-Staggered solution model listed in Table J, "Analyzing Multi-Staggered Time Delay" represents the results obtained after training the 28-1 architecture.

Table J. Analyzing Multi-Staggered Time Delay

28 Input Variables, 1 Output Variable					
Data Set	AE	ME	CV	MID	SS
	degree F	degree F	-	degree F	-
Test	0.309	1.340	13.3	0.397	1672
IS Validation	0.306	1.240	13.8	0.394	618
	3899	Train Set Size (Tr-SS)			
	8	(n)			
0, 0, 0, 2, 4, 5, 8, 8		Cascaded Node Count (CNC)			

The MSTD models required significantly more training time (orders of magnitude) and were able to reduce the Average Error to just over 3/10ths of 1 degree Fahrenheit against in-sample data. A histogram representing the Error Distribution for one model of 6189 data records (or minutes) is contained in Figure W, "MSTD Model

Error Histogram." The distribution shows that 4109 minutes (66%) are predicted to within plus or minus 2/5ths of 1 degree Fahrenheit, and 3725 minutes (60%) are predicted to within plus or minus 1/5th of 1 degree Fahrenheit.

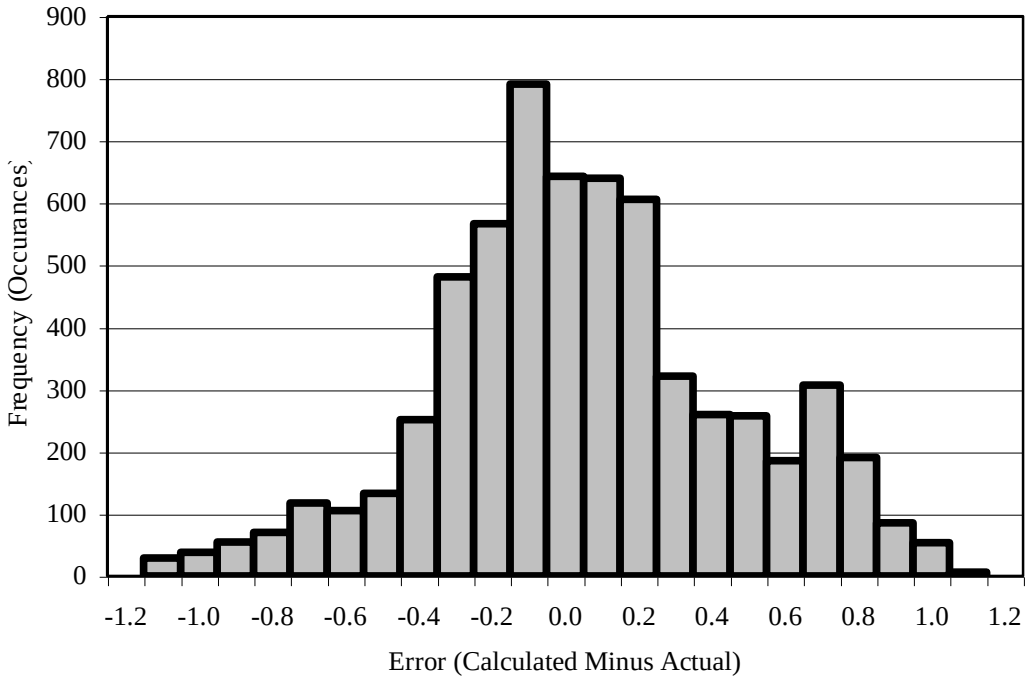

Figure W. MSTD Model Error Histogram

5.6.4 Time Delay Summary

To reiterate, the neural network architecture programming must explicitly accommodate time delay measurements. The way the programming occurs in this study is through preprocessing the data set to explicitly accommodate the delay.

The Big Bend system presents a time delay through the outlet canal, fed by four staggered discharge pipes. The situation requires independent generator loads per Section 5.2, "Effects of Generator Selection," and independent time delays for each generator per the staggered discharge tubes.

In summary, the Multi Staggered Time Delay model proved effective at automatic time delay feature selection and eliminated the requirement of preparing, training, and comparing a potentially unlimited number of individual staggered time delay test cases. The attained Average Error is 0.3 degree Fahrenheit as tested against in-sample validation data. Since there is known mixing and heat transfer in the canal, a multi-

staggered model should naturally capture the temperature increase more accurately than any single-point staggered time delay model. The final architecture is 28 input nodes, 1 output node, cascaded mesh, with appropriate preprocessing requirements to accommodate a time delayed Load to Outlet Canal Temperature increase.

5.7 Effects of Tide on System

The Big Bend Inlet Water and Outlet Water canals are not dammed to the waters of Tampa Bay. Consequently, the tide influences the water within the canals. As the tide comes in, the volume increases; as the tide goes out, the volume decreases.

In analyzing the effects of tide on the system, it is important to note that the tide will not directly affect the water temperature. However, the tide is expected to introduce changes to the Section 5.6, "Effects of Load to Temperature Time Delay" equation. When the tide is inbound, the result is a pushback on the canal waters, viewed as an increased time delay. When the tide is outbound, the result is a vacuum effect on the canal waters, viewed as a decreased time delay. Figure X, "Tidal Effect," attempts to visually capture the tidal impact.

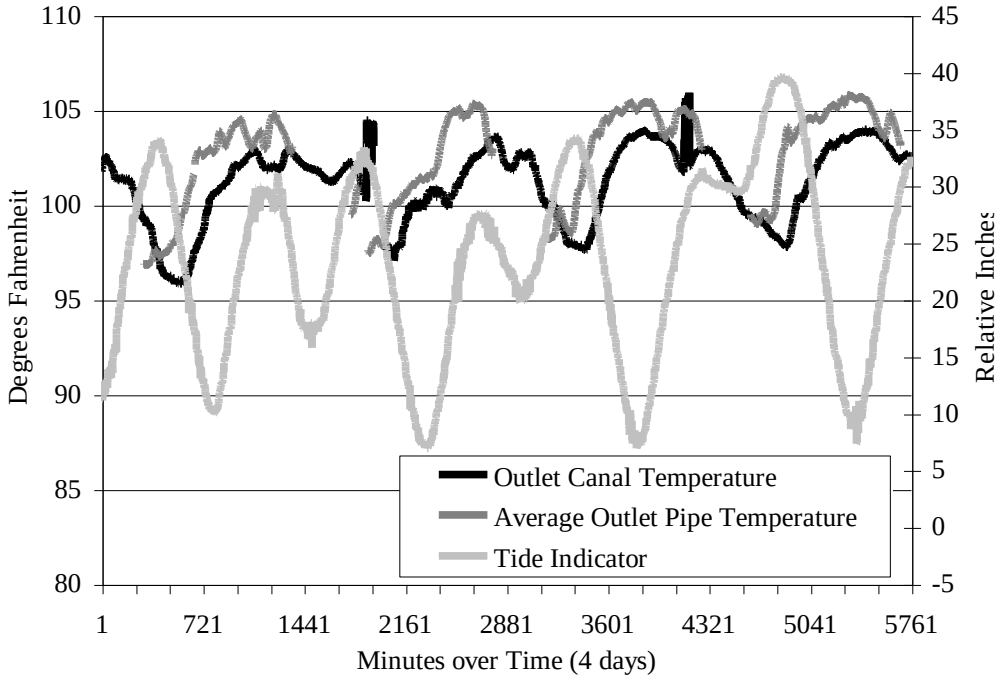

Figure X. Tidal Effect

With the vacuum and pushback recognition, an appropriate data transformation for tidal impact can be investigated. Two tidal effect consequences that can be characterized as changes to the Load to Outlet Canal Temperature time delay are:

- Absolute tide affects the water quantity, and therefore the velocity at which the outlet canal flows. The generator Cooling Water Pumps (i.e., stream source) introduce a steady amount of water to the outlet canal. The canal water volume and velocity are inversely proportional. If the volume is doubled, the velocity is halved (since the stream volume remains constant). However, the outlet canal resting volume is quite large, and therefore the tide has negligible impact on the canal volume.
- Inbound tide affects the canal in a push back manner, and outbound tide affects the canal in a vacuum manner.

The tidal delay effect is difficult to categorize, since there are many influencing variables introduced by the point at which tide has an opportunity to influence the system. Since the non-tide affected Time Delay itself is conjecture built on the MSTD equation, the tide affected Time Delay is double speculation.

The measured and trend-line information for Tide is displayed in Figure Y, "Tidal Noise." The trend-line attempts to smooth the measured curve in order to view the prevailing tidal tendency. The trend-line difference captures the tide as a vector quantity (magnitude and direction) over a 10-minute delay. Even over a thirty-minute moving average with current time mid-point, and a 10-minute temperature differential, the tidal difference measurements remain unpolished.

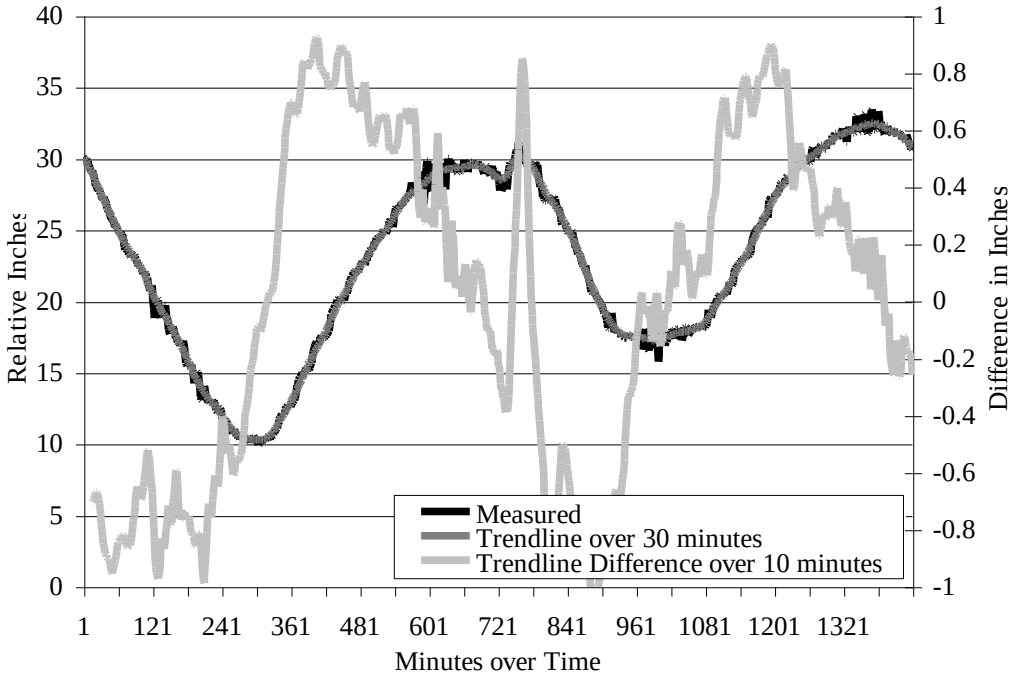

Figure Y. Tidal Noise

The networks were configured as the best-of-breed 28 to 1, Multi-Staggered Time Delay architecture with 28 input nodes and 1 output node, and preprocessing selection of Inbound tide and Outbound tide. The 10-minute temperature differential was selected to reduce the opportunities for wave action and sensor noise appearing as tide. The selection criterion for inbound tide was at least a $1/10^{th}$ of 1-inch increase over 10 minutes. The selection criterion for outbound tide was at least a $1/10^{th}$ of 1-inch decrease over 10 minutes. However, any demarcation of tidal influence should contribute to producing positive results. The expected combination is best generalization and correlation on similarly preprocessed data – in other words, the natural combination of either "outbound" or "inbound" tide should produce the best results.

Table K, "Analyzing Outbound Tide," and Table L, "Analyzing Inbound Tide," demonstrate the results. Figure Z, "Analyzing Tide Graph," offers a visual representation of the average error encountered for each combination.

Table K. Analyzing Outbound Tide

Trained on Outbound					
Data Set	AE	ME	CV	MID	SS
	degree F	degree F	-	degree F	-
Feb 1999 IS Validation	0.264	1.429	9.5	0.337	299
Jun 1999 Outbound	1.876	3.630	5.1	1.905	1539
Sep 1999 Outbound	0.860	3.595	3.0	1.063	3274
Jun 1999 Inbound	2.212	5.057	3.6	2.351	845
Sep 1999 Inbound	0.845	3.771	3.2	1.063	2805
Outbound Overall	1.185	3.630	3.7	1.484	4813
Inbound Overall	1.162	5.057	3.3	1.707	3650
	1889	Train Set Size (Tr-SS)			
	811	Test Set Size (Te-SS)			
	8	(n)			
0, 0, 2, 2, 2, 2, 3, 3		Cascaded Node Count (CNC)			

The neural computing models demonstrate inconclusive results when trained and tested with tidal vector preprocessed data sets. The out-of-sample testing was performed in an effort to confirm the results.

Table L. Analyzing Inbound Tide

Trained on Inbound					
Data Set	AE	ME	CV	MID	SS
	degree F	degree F	-	degree F	-
Feb 1999 IS Validation	0.321	1.331	29.6	0.352	270
Jun 1999 Inbound	1.947	4.656	5.3	2.035	845
Sep 1999 Inbound	0.709	5.276	2.0	0.981	2805
Jun 1999 Outbound	1.723	3.687	8.2	1.768	1539
Sep 1999 Outbound	0.811	3.631	4.1	0.996	3274
Inbound Overall	0.996	5.276	2.7	1.508	3650
Outbound Overall	1.103	3.687	5.4	1.382	4813
	1705	Train Set Size (Tr-SS)			
	732	Test Set Size (Te-SS)			
	8	(n)			
0, 0, 0, 0, 0, 1, 1, 2		Cascaded Node Count (CNC)			

The following list emphasizes the results while comparing Average Error measurements.

- The in-sample Outbound Tide validation set average error is reduced as compared with the overall results obtained in Table J, "Analyzing Multi-Staggered Time Delay" (0.264 degree Fahrenheit versus 0.306 degree Fahrenheit). The in-sample Inbound Tide validation set average error is not as good as the overall results (0.321 degree Fahrenheit versus 0.306 degree Fahrenheit). Inconclusive.
- The Outbound Trained and Inbound Tested set is better than the Outbound/Outbound combination for out-of-sample September 1999 data. This combination is clearly not expected, considering the method of preprocessing invested in the data sets. Inconclusive.
- The Inbound Trained and Outbound Tested set is better than the Inbound/Inbound combination for out-of-sample June 1999 data. Inconclusive.
- The overall Inbound to Outbound and Outbound to Inbound combinations produce better fit than the overall Outbound to Outbound configuration. Inconclusive.

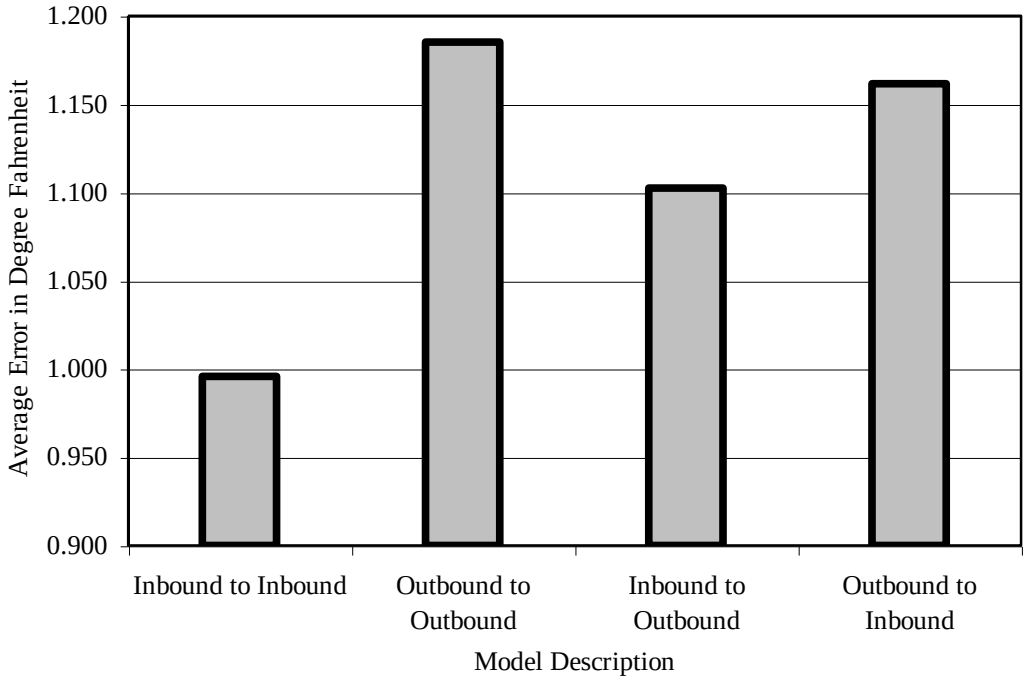

Figure Z. Analyzing Tide Graph

In summary, the influence of tidal vector over time (which represents the tidal direction and velocity) is inconclusive for this system. If there is an influence, the nature of the tidal influence is expected to be Outlet Canal time-delay changes, and not directly related to temperature measurements. The overwhelming measurable and definable system effect remains Time Delay as introduced by the Outlet Canal and as described in Section 5.6, "Effects of Load to Temperature Time Delay."

5.8 Evaluating Performance

This section takes the best of the models described (the Multi-Staggered Time Delay model) and verifies generalization capacity to both in-sample sets and out-of-sample months. The section also investigates the ability for the network to abstract through noise and the effects of highly massaged training and validation sets on the predictive nature of the neural models. The purpose here is to prove that the model can generalize, and that other significant influences do not exist. The various network performance data is reviewed in Section 5.8.4, "Performance Summary."

It is worth mentioning that a month of minute data is composed of 43,200 data points for a 30-day month. The collective month on which the predictive models were built retained far fewer data points after removing sensor failures, pump outages, and other problem data as described in Appendix A, "Data Preprocessing." Each set size count (Training, Testing, and Validation) is listed as part of the model description.

5.8.1 Overall Performance

The MSTD architecture used for the overall performance evaluation is validated across two training data sets. During each evaluation, the training and in-sample validation occurs on one monthly data set, and the models are tested against several out-of-sample validation sets. Table M, "Evaluating February Performance," represents the error measurements for networks trained on February data, as used while detailing Section 5.6, "Effects of Load to Temperature Time Delay," and validated against both in-sample and out-of-sample records. No data pruning of obvious data collection failures occurs, other than as prescribed in Appendix A, "Data Preprocessing."

Table M. Evaluating February Performance

28 Input Variables, 1 Output Variable					
Data Set	AE	ME	CV	MID	SS
	degree F	degree F	-	degree F	-
February 1999 IS	0.300	1.360	14.7	0.383	618
August 1998 OOS	0.721	3.100	9.0	0.846	2288
June 1999 OOS	2.000	5.080	3.3	2.090	2620
September 1999 OOS	0.879	3.970	3.1	1.095	8414
Overall Without June	0.815	3.970	4.9	0.739	11320
Overall Including June	1.038	5.080	4.6	1.236	13940
	3899	Train Set Size (Tr-SS)			
	1672	Test Set Size (Te-SS)			
	8	(n)			
0, 0, 0, 5, 5, 5, 6, 8		Cascaded Node Count (CNC)			

As demonstrated in Table M, the training scenarios generalize to within an Average Error of 1.0 degree Fahrenheit. Significant anomalies exist within the June 1999 data that increase the Average Error to 2.0 degree Fahrenheit for that out-of-sample validation set; without the June 1999 data, the Average Error is reduced to 0.8 degree Fahrenheit. Examining the data revealed that the 2620 members within the June 1999 validation set is composed of eleven time segments. Further, the Load to Temperature Increase is quite irregular, much as Section 5.6, Figure Q, "Irregular Loading and Temperature Increase Curve" entry demonstrated for September 1999 data. See Figure AA, "Irregular Loading and Temperature Increase Curve for June" for a visual depiction of the June 1999 irregular transfer function requirement.

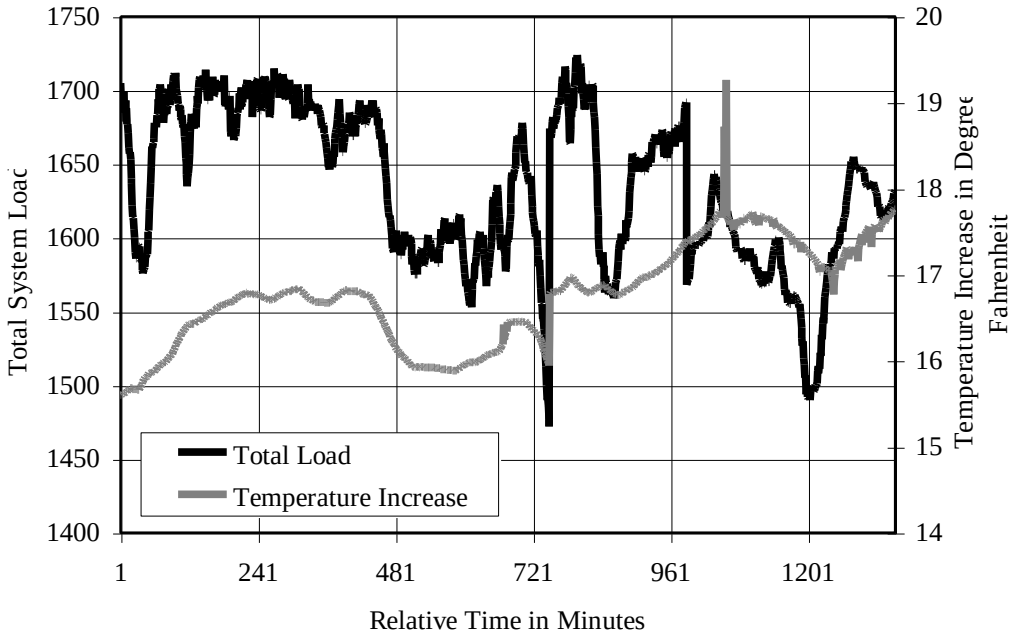

Figure AA. Irregular Loading and Temperature Increase Curve for June

Comparatively, the similarly sized August 1998 validation set is composed of three time segments. Each segment demarcation represents a pump outage, PI data collection error, sensor failure, or other boundary condition. The considerable number of transitions within the June 1999 data, or the underlying issues surrounding these transitions (such as equipment failure) may have contributed to the excessive error conditions. See also Section 6.2, "Sustained Loads," that succinctly demonstrates anomalies at the boundary of training.

Table N, "Evaluating September Performance," represents the error measurements obtained when training networks on September 1999 data and validating against out-of-sample records.

Table N. Evaluating September Performance

28 Input Variables, 1 Output Variable					
Data Set	AE	ME	CV	MID	SS
	degree F	degree F	-	degree F	-
September 1999 IS	0.460	4.979	14.3	0.662	841
August 1998 OOS	0.788	4.034	11.7	0.952	2288
February 1999 OOS	1.014	4.329	9.4	1.297	6189
Overall	0.909	4.979	10.4	0.980	9318
	5301	Train Set Size (Tr-SS)			
	2272	Test Set Size (Te-SS)			
	8	(n)			
6, 9, 11, 11, 17, 21, 28, 29		Cascaded Node Count (CNC)			

The model is able to generalize and abstract to within an Average Error of less than 1.0 degree Fahrenheit. The internal node count increases dramatically for networks trained with known noisy data such as September 1999, when compared to node counts associated with networks trained using cleaner data such as February 1999. The average number of cascaded nodes required to successfully model the clean data is 4, while 17 cascaded nodes are required to model the less clean data.

5.8.2 Highly Massaged Validation Sets

As described in Section 5.6, "Effects of Load to Temperature Time Delay," and in Figure Q, "Irregular Loading and Temperature Increase Curve," there exist data collection failures through a significant portion of each data set. A data set formed by examining each September 1999 day and removing days in which obvious anomalies exist resulted in the performance indicated in Table O, "Evaluating Massaged Performance."

Table O. Evaluating Massaged Performance

28 Input Variables, 1 Output Variable					
Data Set	AE	ME	CV	MID	SS
	degree F	degree F	-	degree F	-
Sep OOS Massaged	0.367	1.933	4.3	0.464	2653
	8	(n)			

The preprocessing removal was performed manually, and was based on day schedules to limit the opportunity of extracting only highly regular minutes. This "day delete" process was adhered to in an attempt to maintain a statistically representative

sample of information for the out-of-sample month.[20] As observed, the Average Error is reduced to less than 4/10ths of 1 degree Fahrenheit, and the Maximum Error is reduced to less than 2 degrees Fahrenheit. The sample size is also reduced, in this case to less than 1/3rd of the original size.

5.8.3 Highly Massaged Training Sets

Continuing with the clean data set models obtained through the methods described in Section 5.8.2, the information in Table P, "Evaluating September (Clean Data) Performance" demonstrates the errors that result when the training data set is pruned.

Table P. Evaluating September (Clean Data) Performance

28 Input Variables, 1 Output Variable					
Data Set	AE	ME	CV	MID	SS
	degree F	degree F	-	degree F	-
September 1999 IS	0.202	1.073	12.0	0.268	265
August 1998 OOS	0.683	2.650	4.2	0.782	2288
February 1999 OOS	0.909	2.853	6.7	1.089	6189
Overall	0.828	2.853	6.2	0.678	8742
	1671	Train Set Size (Tr-SS)			
	717	Test Set Size (Te-SS)			
	8	(n)			
0, 0, 0, 2, 2, 2, 2, 3		Cascaded Node Count (CNC)			

Compared with Table N, "Evaluating September Performance," the Average Error is reduced by 0.08 degree Fahrenheit (0.909 to 0.828), and the Maximum Error is reduced by more than 2 degree Fahrenheit overall (4.979 to 2.853). Furthermore, the cascaded node count required to capture the Load to Temperature Increase interaction is significantly reduced (average of 2 Cascaded Nodes required instead of 17 Cascaded Nodes.)

5.8.4 Performance Summary

Figure BB, "Performance Summary," demonstrates the errors associated with various forms of Training and Validation sets.

[20] In the Highly Massaged Validation Set, the following September days were pruned: 2, 3, 6, 7, 12, 13, 14, 15, 16, 17, 27, 28, and 29. By removing "days" instead of "records," a statistically acceptable set is expected.

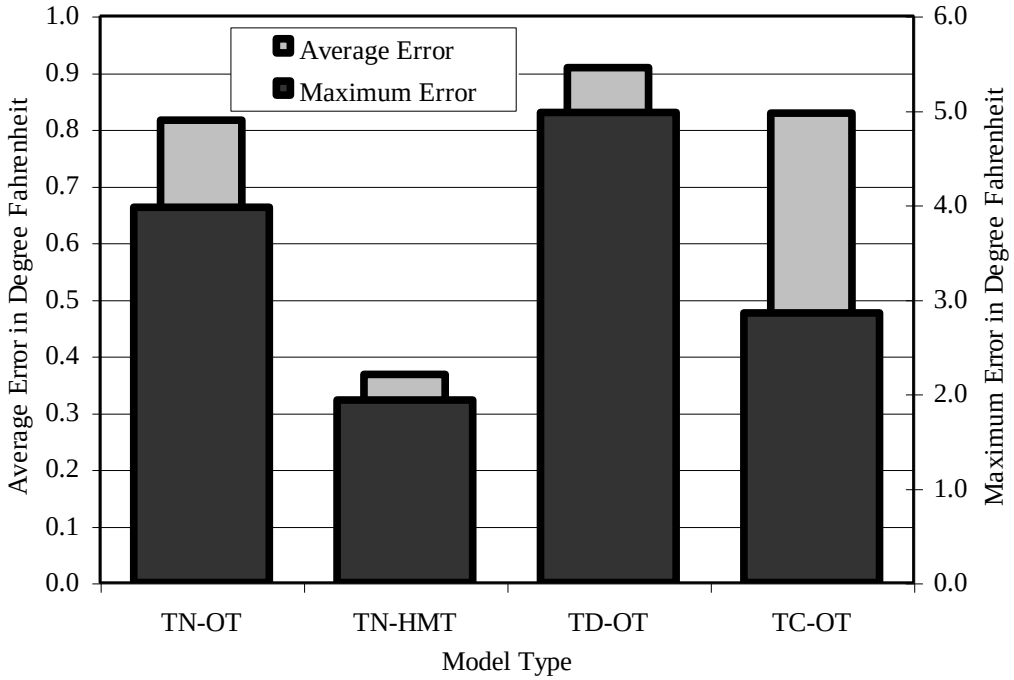

Figure BB. Performance Summary

The models detailed in Figure BB, "Performance Summary," are:

- TN-OT = Trained Natural, Overall Testing. Trained on February 1999 data, Validated against August 1998, February 1999, and September 1999 data.
- TN-HMT = Trained Natural, Highly Massaged Testing. Trained on February 1999 data, Validated against massaged September 1999 data.
- TD-OT = Trained Dirty, Overall Testing. Trained on known irregular September 1999 data, Validated against August 1998, February 1999, and September 1999 data.
- TC-OT = Trained Clean, Overall Testing. Trained on massaged September 1999 data, Validated against August 1998, February 1999, and September 1999 data.

In summary, the MSTD architecture is able to produce models that can predict to within an average error of 1.0 degree Fahrenheit or less, even with the significantly noisy data found within September 1999. This demonstrates that, given enough data, the neural network architecture is able to abstract beyond noisy data. In summary, the data sets are critical to both building respectable training sets and to evaluating the performance of neural models. The models built with cleaner training data are able to abstract more

fluently and generalize more specifically and in a complexity costs of far fewer cascaded nodes.

5.9 Feature Selection Summary

This chapter investigated the system Features that contribute to the Outlet Canal Temperature. The following information is deduced from the analysis:

- Input Data selection is critical to effective abstraction and generalization. Significant errors were discovered within most training sets. To avoid over manipulation and randomly pruning what appears to be "bad data," naturally cleaner months should be used for network training. Many of the months contain a very limited amount of contiguous and useful data and the boundary transitions are expected to contribute negatively to the models.
- Preprocessing is critical to accurate training. Preprocessing includes removing subtle errors such as excessively disparate Inlet Water Temperatures, overt errors such as "Bad Input" tags by the PI system, Pump Outage conditions, and other sensor failures.
- The Inlet Water Temperature is important as a relative value to Outlet Canal Temperature. There is no need to use IWT as an input variable to the models. Therefore, the water-temperature relative measure (OCT minus IWT) is selected as the output variable in the neural modeling paradigm.
- The generator Load is important. The Load is representative of the Temperature Increase experienced by the Condenser Cooling Water.
- The Load for each generator is important. Each generator creates a slightly different Load to Water Temperature pattern. Therefore, the four generator Loads are used as independent input variables to the models.
- The Condenser Cooling Water Pump Status is important. Pump outages significantly affect the Load to Water Temperature pattern, and affect the time delay pattern. Therefore, Pump Outage conditions are removed from the training sets by eliminating all data that could influence the outlet canal.
- The Time Granularity is important. Models built with 10-minute time slices were not able to discern patterns with the underlying data. This is due to the significant

changes in Generator Loading that are possible during a 10-minute time slice. Therefore, 1-minute time slice data is used for system investigation.

- The Load to Temperature Time Delay is important. Each generator discharges cooling water at a different point along a canal that is more than 800 yards long and 100 yards wide. During the flow down river, the cooling water temperature changes due to evaporative cooling, radiative cooling, and mixing with surrounding water. Therefore, the Multi-Staggered Time Delay model is used for system investigation, since the model avoids manual heuristics and posts the best opportunity for automatic (neural computing based) time delay selection.
- The Tidal effect cannot be discerned from the data.

6 Discovering System Effects

This chapter explores influential conditions that affect the Outlet Canal Temperature. It continues by investigating how the OCT responds to increased loading applied to each generator. The models are compared based on:

- Water temperature Average Increase (AI)
- Water temperature Maximum Increase (MI)
- Coefficient of Variation between the (n) model outputs (CV)
- Water temperature Midrange Increase of the AI for all (n) (Midrange)

6.1 Inlet Water Temperature

The Inlet Water Temperature greatly affects the resulting Outlet Canal Temperature. The IWT contribution affects the resulting OCT by introducing a base at which the condenser system begins heating up the cooling water.

The neural models designed as a result of this study are differential-temperature, based on the Inlet Water Temperature. The temperature differential is predicted by the neural network, and this differential is added to the current Inlet Water Temperature to form the forecasted Outlet Canal Temperature.

During periods of reduced Inlet Water Temperature, Big Bend can conceivably satisfy increased load demands. During periods of increased Inlet Water Temperature, Big Bend must throttle back generator production more aggressively to satisfy the maximum Outlet Canal Temperature restrictions.

6.2 Sustained Loads

As expected from Section 5.1, “Effects of Load,” the temperatures that result from increasing load is highly linear. Table Q, “Evaluating Sustained Loads,” and Figure CC, “Evaluating Sustained Loads Graph,” depict the temperature increase expected for various loading configurations. The loading configurations are 800 total Megawatt (four generators at 200 each) through 1800 total Megawatt (four generators at 450 each). ATI-

Base is the Average Temperature Increase over the Base-200 measure, and the ATI-Prev is the Average Temperature Increase over the AI measure of the next lower 200-Megawatt trial.

Table Q. Evaluating Sustained Loads

28 Input Variables, 1 Output Variable							
Load (MW)	AI	MI	CV	Midrange	(n)	ATI-Base	ATI-Prev
	degree F	degree F	-	degree F	-	degree F	degree F
Base 800	8.847	8.956	1.1	8.807	8	-	-
1000	9.644	10.104	2.3	9.767	8	0.797	0.797
1200	11.248	11.744	3.4	11.323	8	2.401	1.604
1400	13.007	13.124	0.7	12.986	8	4.160	1.759
1600	14.508	14.723	0.8	14.559	8	5.661	1.501
1800	14.995	15.164	0.6	15.033	8	6.148	0.487

The temperature costs associated with 50 Megawatts across all four generators (200 Megawatt system increase) is roughly 1.6 degree Fahrenheit not including the boundary measures, or 0.8 degree Fahrenheit per 100 Megawatt system load increase. The calculated Temperature Increase for the boundary conditions at 1000 Megawatt and 1800 Megawatt are an unrealistic 0.642 degree Fahrenheit per 200 Megawatt.

The 200x4 (800 Megawatt) and the 450x4 (1800 Megawatt) boundary situations follow a non-linear format to the remaining data and demonstrate an abstraction limitation of the neural networks. The maximum loading configuration observed by the training algorithm rests with BB1 at 434 Megawatt, BB2 at 434 Megawatt, BB3 at 419 Megawatt, and BB4 at 469 Megawatt. The BB1 through BB3 training units never observe 450 Megawatt loading configuration. The minimum loading configuration of BB1 and BB4 approach 200 at the limit, just as pump outages take effect. The boundary conditions demonstrate that the neural network must be trained with a representative sample of expected load configurations.

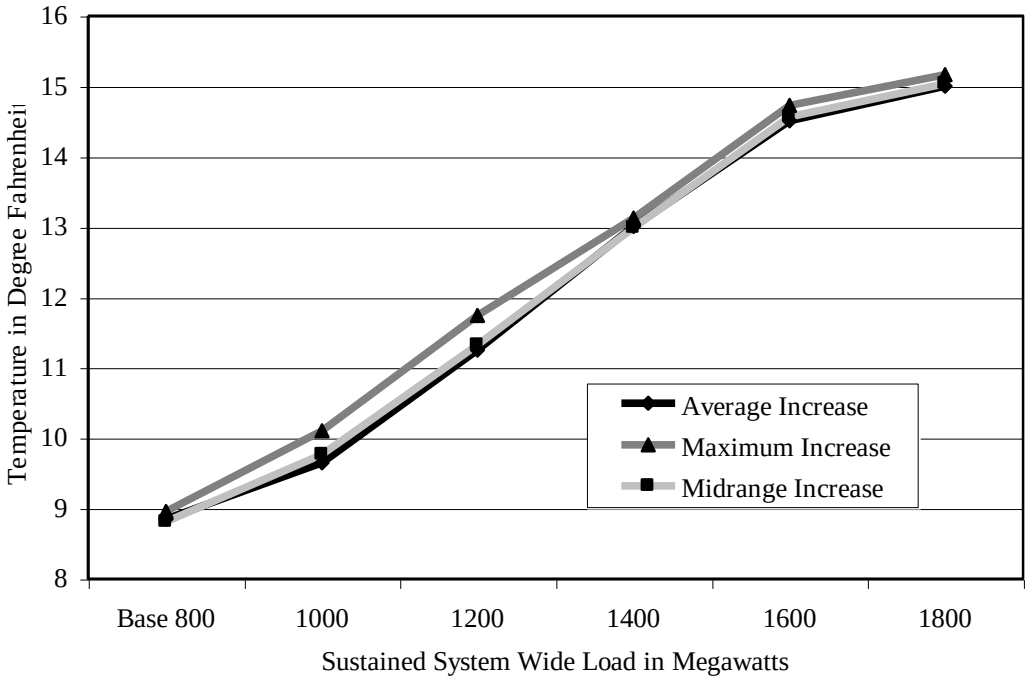

Figure CC. Evaluating Sustained Loads Graph

6.3 Evaluating Individual Generator Production

As described in Section 5.2, "Effects of Generator Selection," the generators are not identical. Table R, "Evaluating Generator Production," and Figure DD, "Evaluating Generator Production Graph," capture the temperature difference results that might be expected under a hypothetical condition of three generators supplying 200 Megawatts each, while the targeted generator supplies 400 Megawatts. The "Base" condition reflects the temperature expected while the four generators are supplying 200 Megawatts each (a system wide load of 800 Megawatts). ATI is the Average Temperature Increase experienced by the Outlet Canal as a result of the high-load generator.

Table R. Evaluating Generator Production

28 Input Variables, 1 Output Variable						
Unit	AI	MI	CV	Midrange	(n)	ATI
	degree F	degree F	-	degree F	-	degree F
Base	8.847	8.956	1.1	8.807	8	-
BB1	9.438	10.212	4.0	9.585	8	0.591
BB2	9.748	10.423	3.9	9.749	8	0.901
BB3	9.964	10.251	2.1	9.907	8	1.117
BB4	9.676	9.836	1.2	9.685	8	0.829

From the data, Big Bend Unit 3 is the most influential unit. The temperature costs associated with the hypothetical situation of increasing Unit 3 load to 400 Megawatts and leaving the other generators at 200 Megawatts is an increase of 1.1 degree Fahrenheit. The same 200 Megawatt load can be supplied by Big Bend Unit 1 with an Average Increase of less than 0.6 degree Fahrenheit. Note that the Average Temperature Increase is affected by fringe values at 200 Megawatt and 400 Megawatt, as described in Section 6.2, "Sustained Loads," so the increased temperature measurements are appropriate only as a relative measure to one another.

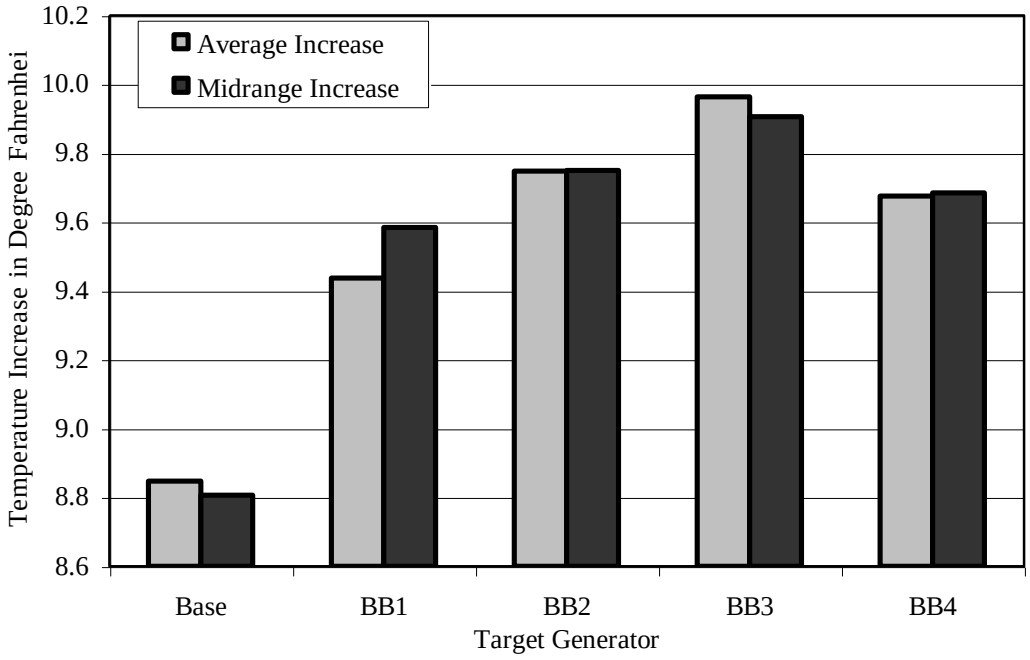

Figure DD. Evaluating Generator Production Graph

6.4 Summary

This chapter investigated the conditions that influence the Outlet canal Temperature. The following information is deduced from the neural computing analysis:

- The Inlet Water Temperature is additive to the eventual Outlet Canal Temperature. In other words, the IWT has no effect on the energy transfer within the condenser, and is simply a "start point" for the temperature change.
- There exists a well-defined linear relationship between load and temperature increase, where the expected temperature increase is 8/10ths of 1 degree Fahrenheit per 100 Megawatt increased system load.
- Big Bend Unit 1 generates load with the least temperature cost while Big Bend Unit 3 generates load with the largest temperature cost.
- The predictive neural network must be trained with data that is representative of the data that may be encountered during forecasting sessions. The networks deliver predictable results while extrapolating between observed data values, but deliver unexpected results when forced to extrapolate beyond observed data values.

7 Conclusions

This chapter concludes the investigation of features and system effects of the power generation facility under review. The conclusion includes analysis and discussion, suggestions for future work in the Power-Plant Cooling Water Discharge Temperature modeling field, and prognosis for neural computing.

7.1 Analysis and Discussion

This paper investigated two prospects of using neural computing tools to describe a time delayed physical system, first identifying the components and features, and then describing the system wide characteristics and effects. The physical system under review is Big Bend in Apollo Beach, owned and operated by the Tampa Electric Company.

As with many artificial intelligence studies, obtaining appropriate and clean data is sometimes difficult and time consuming. With this study, data was freely available but was blemished by incomplete, noisy, and otherwise inaccurate information.

7.1.1 Component Evaluation and Feature Selection

The first phase of the feature selection process involved using neural computing tools to systematically describe and decompose the physical system. With the assistance of neural computing, the system was separated into constituent components such as independent generators and an outlet canal, and Features were discovered through pattern recognition techniques. The components were then reassembled into the original system.

Neural computing and associated pattern recognition techniques are applicable for investigating the system under study because algebraic solutions are not easily discerned. During the process of neurally describing the system, patterns were discovered that were not readily apparent when evaluated with non-neural tools and methods. For example, the Temperature Time Delay equation would have required exhaustive searches of the problem domain in order to arrive at a possibly acceptable solution. In addition, using neural computing tools, the Time Granularity fit was discerned more easily than through

the initial algebraic solutions. The algebraic investigation resulted in an initial architecture involving 1-hour time granularity. It was found, using neural pattern recognition, that even 10-minute granularities were excessively large, and that the network training engine must observe 1-minute or less time granularity for effective abstraction and generalization.

Neural networks were also useful for finding and processing clean training sets, and for discarding sets that exhibit significantly inconsistent behavior. By iterating a "prune, train, validate" algorithm, clean data sets were created such as that for September 1999. In some instances, such as February 1999, the prune sequence was avoided and the entire data set was used intact.

One significant condition in this thesis was to determine appropriate data transformations and other preprocessing requirements. Pattern recognition and "best fit" solutions prescribed an architecture that requires:

- Individual load readings for each of four Big Bend generators.
- Relative Temperature Increase between the Inlet Water Canal and the Outlet Canal Temperature probes, calculated as a preprocessing step.
- Condenser cooling water pump status, and preprocessing removal of pump out of service conditions.
- Outlet Canal time delay of from 105 to 135 minutes measured at a 5-minute interval.
- Input data measured with a 1-minute granularity.

Building predictive networks that are able to generalize to within an Average Error of less than 1.0 degree Fahrenheit for both in-sample and out-of-sample data proved possible. Depending on the "cleanness" of the training data set, the internal node count changes drastically. The neural tool was able to capture the Load to Temperature equation for one "cleaner" training set with an average Cascaded Node Count of less than 4. A known "less clean" training set required an average CNC of 17. Due to the added complexity, training times were also dramatically longer (orders of magnitude) for networks trained with "less clean" training sets.

7.1.2 System Effects

The second phase of the work involved using neural networks as an offline tool for evaluating and describing system effects.

During this phase, the expected Outlet Canal Temperature based on sustained load was calculated with the neural models. As discovered with the offline test suites, there is close to a linear relationship between generation load and Outlet Canal Temperature increase. The cost by way of Temperature Increase is 8/10ths of 1 degree Fahrenheit per 100 Megawatt system load demanded.

Next, each generator was modeled through the canal and the temperature results per generator were determined. Specifically, Big Bend Unit 3 was found the most costly in terms of temperature increase and Big Bend Unit 3 was found the least costly. Big Bend 3 introduces more than 1.1 degree Fahrenheit (calculated) for 200 Megawatt, while BB1 introduces less than 0.6 degree Fahrenheit (calculated) for 200 Megawatt. Although the temperature demands cannot be used as absolutes due to boundary conditions and forecasting beyond observed data, the relative change indicates that Big Bend Unit 1 is much less costly.

It was also discovered that significant non-linearity exists when the network is asked to predict an Outlet Canal Temperature based on Loads that are beyond the boundaries of the training data. For example, if network testing occurs with Load data that is less than the minimum load experienced during training, the network is forced to extrapolate beyond known data. This extrapolation is not delivered as cleanly as extrapolation between learned data. For example, the sustained Loads increase is modeled at approximately 0.8 degree Fahrenheit per 100 Megawatt, but the value is reduced to 0.3 degree Fahrenheit per 100 Megawatt for test sets at the boundary values.

7.1.3 Training and Validation Sets

One of the paramount issues surrounding neural computing and the associated network models is integrating the real-world data into a neural computing trainable collection. By example, a significant source of temperature effect in the power plant example is the Inlet Water Temperature; the Inlet Water Temperature cannot be ignored during the investigation. However, with knowledge of chemistry (specific heat) and

thermodynamics (heat transfer), the Inlet Water Temperature is easily integrated into the neural network input data list by calculating the relative measure against the Outlet Canal Temperature. Reducing the input data list significantly reduces the complexity of the underlying problem, in part by removing the necessity of finding a representative sample of inlet water temperatures in which to train the network.

To say again, it is imperative that the neural training algorithm is presented with training data sets that are representative of data that will be used during system forecasting. The neural networks built surrounding this problem domain were able to generalize beyond observed data and to abstract through noisy data. However, the networks proved substantially more effective at modeling with clean data and at extrapolating between (instead of beyond) learned information. Also, excessively inconsistent input data resulted in both significantly more complex network architectures (i.e., more cascaded nodes) and significantly increased training times.

7.2 Future Work

Several opportunities to increase the effectiveness of these studies exist. For example, the Outlet Canal is open air and without covering. The thermal effects presented on the Outlet Canal include heat dissipation into the riverbed, air temperature and humidity levels resulting in evaporative cooling, rainfall, cloud cover, and sun exposure. All of these factors introduce temperature changes to the Outlet Canal water. These factors cannot be investigated at this time since the data collection system at Big Bend does not collect the required variables.

In addition, sensor failures, sensor inaccuracies, and data collection failures have a negative impact on neural operations. Some sensors are redundant; redundant failures do occur. Other collection points (such as Pump Status) are not redundant, and failures will negatively influence the neural model. In addition, the Load measurement collection occurs only after analog converters that introduce noise. More exhaustive failure identification and advanced measurement precision would increase the validity of the data sets. These errors cannot be further exploited at this time since the data collection system has no additional sensors in place.

Long periods of data with neither noise nor errors may positively influence training scenarios by reducing the preprocessing requirements and related boundary

conditions. Unfortunately, long periods with all equipment functioning are difficult to find due to the nature of the machine. Required maintenance outages occur twice weekly or more often for each of the four generators and these outages are on a staggered schedule.

An important feature of this system was time delay driven data within the outlet canal. As demonstrated, neural computing can be used to accurately evaluate the time delay. It is possible to implement a large neural network or genetic algorithm with hundreds of input variables reserved for each generator and assigned to minute or even second granularity data; this may allow more precise time delay deduction. However, this algorithm will require two things not currently available. First, very clean data sets are critical, as described in the previous paragraphs. Second, a very large amount of data must be collected and then presented to the neural tool to accommodate the thousands of weighted links that will result in such a large model.

A useful extension of this work is to include it as part of a multi-level network engine in which the front-end processor is a short-term load forecast machine such as [Yuan]. The study of short-term load forecasting based on region-wide temperature forecasts, cloud cover, and humidity is beyond the scope of the current data-collection system at Big Bend. However, this extension would enable Big Bend staff to monitor for days that approach EPA limitations.

7.3 Neural Networks

Neural computing algorithms have been used in a large variety of problems, including weather forecasting, financial market timing, battlefield management, and playing board games. Neural computing tools have progressed tremendously over the fifty years since Hebb published the first neural algorithm. In the course of time, architecture advances such as dynamic Cascade Correlation architectures, filtering techniques such as those proposed by Kalman and group, and the sheer power of computing devices have been integrated into and brought improvements to neural network tool design that have allowed the neural network engineer to concentrate on integrating the problem domain into a neural paradigm instead of concentrating on the network details such as the number of hidden nodes required to successfully demonstrate a problem.

In other words, neural computing has evolved to where the tools, given appropriate input data, can generally build substantial and conclusive models with little or no required customization. The requirements on the neural engineer are to submit both necessary and sufficient input data to the tool. The issues that remain for the neural computing engineer are *data mining* or collecting data, and *data mapping* or transforming the real-world data into a representation acceptable and appropriate for the neural tool.

This mapping or integration from real-world data to neural paradigm remains a significant source of difficulty. Even now, the neural computing engineer must perform explicit data mining activities, including significant preprocessing and performing various data transformations. In time, as tools become more general and hardware becomes more powerful, there may be a day that the data set preprocessing techniques will also be predominantly tool based. As this comes to age, large quantities of raw data will be introduced into the neural system, and the system itself will select and process appropriate data of even the time delay driven nature. Just as human consciousness allows men to gratefully avoid the neurotic state, the future generations of neural computing tools will build authoritative networks while simultaneously avoiding network neurosis and paralysis.

References

1. Barnaby, Wendy. (March 1995). *Consciousness: Its Place in Contemporary Science.* A briefing document prepared for the Royal Society and the Association of British Science Writers.
2. Brady, James E.; Holum, John R. (1984). *Fundamentals of Chemistry, Second Edition*. New York: John Wiley & Sons.
3. Busot, JC. (1988). *Thermo-I Course Material*. Tampa, Florida: Pro-Copy, Inc.
4. Central Electric Power Cooperative Web Site. (2000). *http://www.cepc.net*.
5. Chapra, Steven C.; Canale, Raymond P. (1985). *Numerical Methods for Engineers*. New York: McGraw-Hill Book Company.
6. Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L. (1990). *Introduction to Algorithms*. Cambridge, Massachusetts: The MIT Press and McGraw-Hill Book Company (joint production-distribution agreement.)
7. Fink, David M.D. (1943). *Release from Nervous Tension*. New York: Simon and Schuster.
8. Giarratano, Joseph, Ph.D.; Riley, Gary. (1989). *Expert Systems: Principles and Programming*. Boston: PWS-KENT Publishing Company.
9. Hammerstrom, Dan. (June 1993). "Neural Networks at Work." *IEEE Spectrum*.
10. Hammerstrom, Dan. (July 1993). "Working with Neural Networks." *IEEE Spectrum*.
11. Harrald, Paul G.; Kamstra, Mark. (Volume 1, Number 1, April 1997). "Evolving Artificial Neural Networks to combine Financial Forecasts." *IEEE Transactions on Evolutionary Computation*.
12. Holst, Anders. (1997). *The Use of a Bayesian Neural Network Model for Classification Tasks*. Dissertation: Royal Institute of Technology, Stockholm, Sweden.
13. Hopcroft, John E.; Ullman, Jeffery D. (1979). *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley Publishing Company, Inc.
14. Hurricane Eye Web Site. (2000). *http://www.hurricane-eye.com*.
15. Illindala, Uday Kiran. (1999). *Using Neural Networks to Predict Blood Type from Spectrophotometer Readings*. Masters of Science Thesis. Tampa, Florida: The University of South Florida, Department of Computer Science and Engineering.
16. InfoPlease Web Site. (2000). *http://www.infoplease.com*.

17. Koza, John R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. The MIT Press.

18. McClelland, James L. (1988). *Explorations in Parallel Distributed Processing*. Cambridge: The MIT Press.

19. NeuralWorks Predict. (1995). *NeuralWorks Predict Help Documentation*. Aspen Technology, Inc.

20. Roediger, Henry L. III. (1984). *Psychology*. Canada: Little, Brown & Company (Canada) Limited.

21. Sedgewick, Robert. (1988). *Algorithms, Second Edition*. Princeton University: Addison-Wesley Publishing Company.

22. Schach, Stephen R. (1990). *Software Engineering*. Homewood, Illinois: Aksen Associates Incorporated Publishers.

23. United States Department of Geological Survey through Microsoft Terraserver. (2000). *http://terraserver.microsoft.com*.

24. Wasserman, Philip D. (1989). *Neural Computing Theory and Practice*. New York: Van Nostrand Reinhold.

25. Welch, Greg; Bishop, Gary. (December 8, 1997). *An Introduction to Kalman Filter*. Department of Computer Science: University of North Carolina at Chapel Hill.

26. Winston, Patrick Henry. (1984). *Artificial Intelligence, Second Edition*. Massachusetts Institute of Technology: Addison-Wesley Publishing Company, Inc.

27. Yuan, Jen-Lun; Fine, Terrence L. (Volume 9, Number 2, March 1998). "Neural-Network Design for Small Training Sets of High Dimension." *IEEE Transactions on Neural Networks*.

Bibliography

1. Department of Transportation. (1975). *Aviation Weather for Pilots and Flight Operations Personnel*. Publication AC 00-6A. Washington, DC: ASA Publication.
2. Harrald, Paul G.; Kamstra, Mark. (April 1997, Volume 1, Number 1). "Evolving Artificial Neural Networks to Combine Financial Forecasts." *IEEE Transactions on Evolutionary Computation*.
3. Jacobson, Ivar. (1992). *Object-Oriented Software Engineering*. Wokingham, England: Addison-Wesley Publishing Company. ACM Press.
4. Johnson-Laird, Philip N. (1988). *The Computer and the Mind*. Cambridge: Harvard University Press.
5. Khadem, M. (February 1993). "Application of the Kohonen Neural-Network Classifier to Season Identification in Short-Term Load Forecasting." Columbus, Ohio: *IEEE Winter Meeting*.
6. Khotanzad, A.; Hwang, R. C., Maratukulam, D. (February 1993). Columbus, Ohio: *IEEE PES Winter Meeting*.
7. Miller, Irwin; Freund, John E. (1985). *Probability and Statistics for Engineers*. Englewood Cliffs: Prentice-Hall, Inc.
8. Swokowski, Earl W. (1979). *Calculus with Analytic Geometry*. Boston: Prindle, Weber & Schmidt.
9. Yuan, J. L. (January 1994). *Neural Networks for Electric Load Forecasting*. Ph.D. Dissertation. Ithaca, NY: Cornell University, School of Electrical Engineering.

Appendices

Appendix A: Data Preprocessing

The final training and testing sets are created as follows:

- Retrieve One-Minute sampled data through the TECO PI Database.
- Remove obvious failures including “Bad Input,” “Shutdown,” “System Error,” and “I/O Timeout.”
- Rename Pump Status STOP conditions to OFF and RUN conditions to ON, in order that all four generators use consistent nomenclature. (Two generators use ON/OFF nomenclature, while the other two use RUN/STOP.)
- Calculate Inlet Water Temperature Average over all generators. The PI system collects Inlet Water Temperature for each of the four generators. At times, the readings were different by more than 2 degrees Fahrenheit. The cause may be stratification of the inlet water within the Inlet Water Canal, but is also indicative of sensor failures. Remove questionable data since the absolute cause was unknown.
- Flag invalid Inlet Water Temperatures (for 360 minutes afterwards) by comparing the entries for each of four generators and removing the outliers. If not at least three entries match to within 2 degrees Fahrenheit, then flag Inlet Water Temperature as invalid.
- Remove invalid Outlet Canal Temperatures that are lower than the average Inlet Water Temperature (an invalid condition).
- Flag Pump Outages by removing Load for 30 minutes before and 360 minutes after the outage. The 30 minutes before accommodates the generator shutdown preparations, and the 360 minutes after accommodates three hours delay doubled (3 hours nominal flow, doubled due to 50% pump flow rate condition.)
- Flag Low Load conditions by removing Load (for 30 minutes before and 360 minutes afterwards) if less than 200 Megawatt. This removes “trip” conditions and accommodates the delay through the canal.

- Convert the input data into an appropriate time delay format.
- Remove records in which any field is flagged. The Flag and Remove must be accomplished in a two step format due to the time-delay format conversion from the previous step.
- Data Set is ready!

Appendix B: Independent Generator Analysis

There are a few items of special interest discerned from the "Load and Temperature Difference Plotted per Minute" graphs depicted on the following pages.

Note that each generator results in a unique load effect graph. For example, Big Bend Unit 3 can generate about 450 Megawatt with a resulting temperature increase of 19 measured degrees Fahrenheit, while Big Bend Unit 4 can supply about the same load at a cost of 17 measured degrees Fahrenheit. This may be due to discrepancies in the measurement devices described in Section 4.3, "PI," and Section 4.4, "Collection Points," but may also be due to different generator efficiencies as described in Section 5.2, "Effects of Generator Selection."

Note that the load and the temperature do not appear directly correlated. Instead, there appears to be additional factors that influence the system. For example, BB1 approximate time 180, Load approaches 450 Megawatt with a measured temperature increase of about 16.5 degrees Fahrenheit, while at time 540, Load rests at 425 Megawatt with a measured temperature increase of 16.6 degrees Fahrenheit. In fact, the Load and Temperature are related through the condenser, as described in Section 3.4, "Coal Fired Base Load Power Generation." Due to feedback loop inefficiencies, the Temperature Increase sometimes lags the Load, while at other times the Temperature Increase leads the Load. For example, BB2 at approximate time 130, Temperature increases before the Load, while at approximate time 230, Load increases before the Temperature.

Note that BB3 has apparent noise on the Load graph. Although the noise appears significant, predictive networks for BB3 demonstrated an average error of less than ½ of 1 degree Fahrenheit for in-sample data, as described in Section 5.2, "Effects of Generator Selection."

Appendix B (Continued)

These four graphs reflect time from 1 September 1999, from 11:02 to 22:23 inclusive. The Inlet Water Temperatures at the pump were recorded as follows: BB1 [86.73-87.38], BB2 [86.84-87.40], BB3 [87.10-87.66], and BB4 [87.47-87.70]. The temperature probe maximum difference between generators is less than 1 degree (.97 degree Fahrenheit). The drift (sometimes much greater than 1 degree) may be due to unresolved maintenance conditions, or to stratification within the Inlet Canal.

Periodic cooling path maintenance schedules are not directly reflected in these graphs. During scheduled cooling path maintenance, one of two cooling water pumps is removed from service. The outage results in much greater cooling water temperatures, but with half the volume.

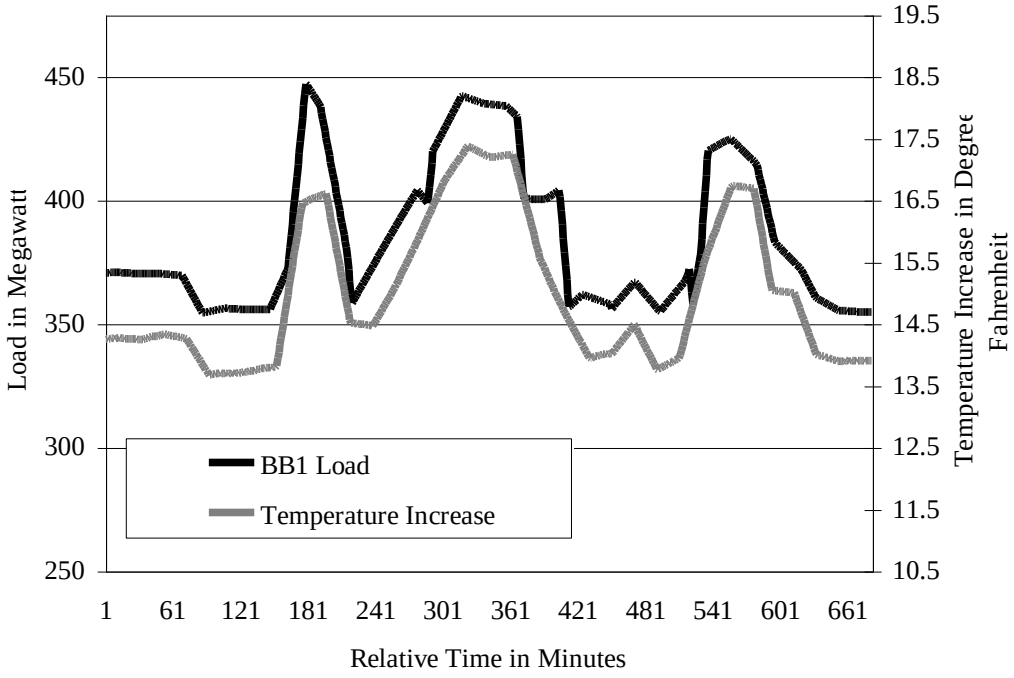

Figure EE. BB1 Load and Temperature Difference Plotted per Minute

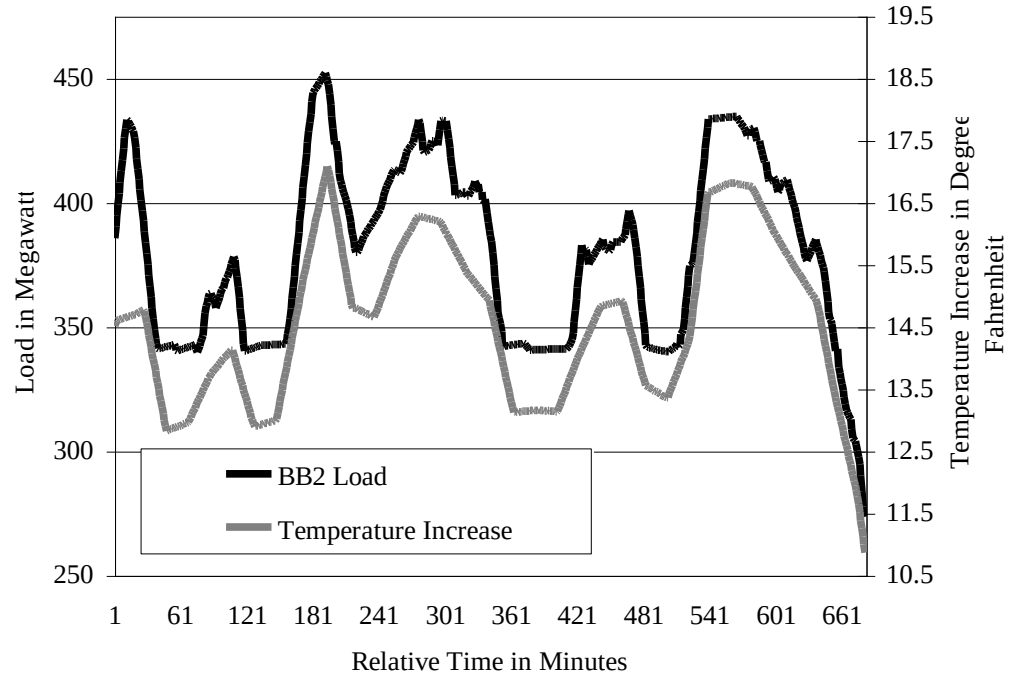

Figure FF. BB2 Load and Temperature Difference Plotted per Minute

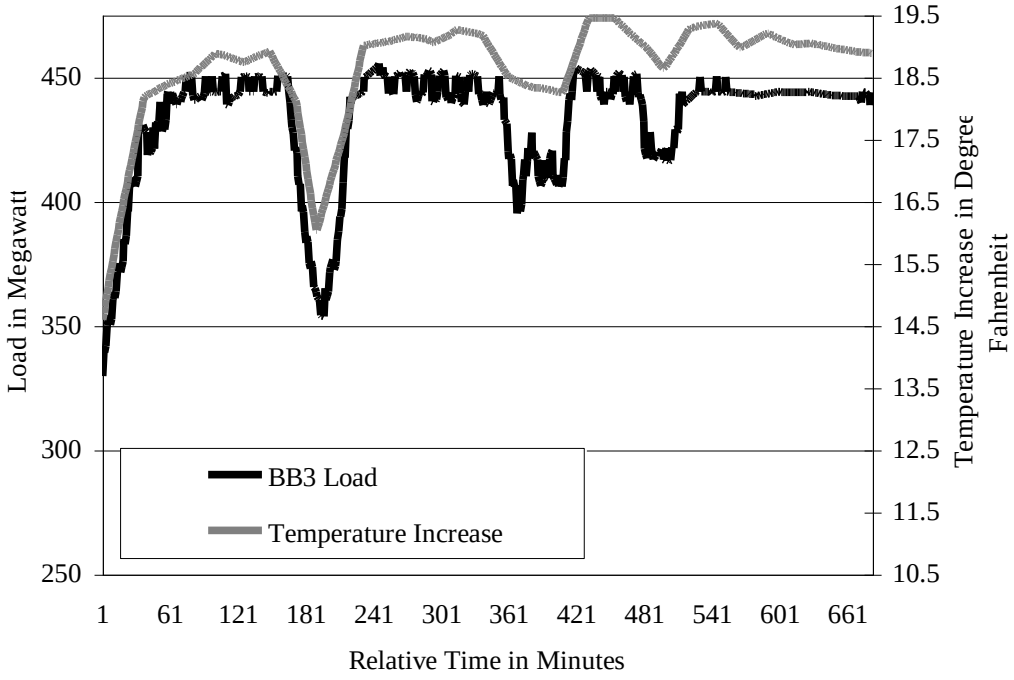

Figure GG. BB3 Load and Temperature Difference Plotted per Minute

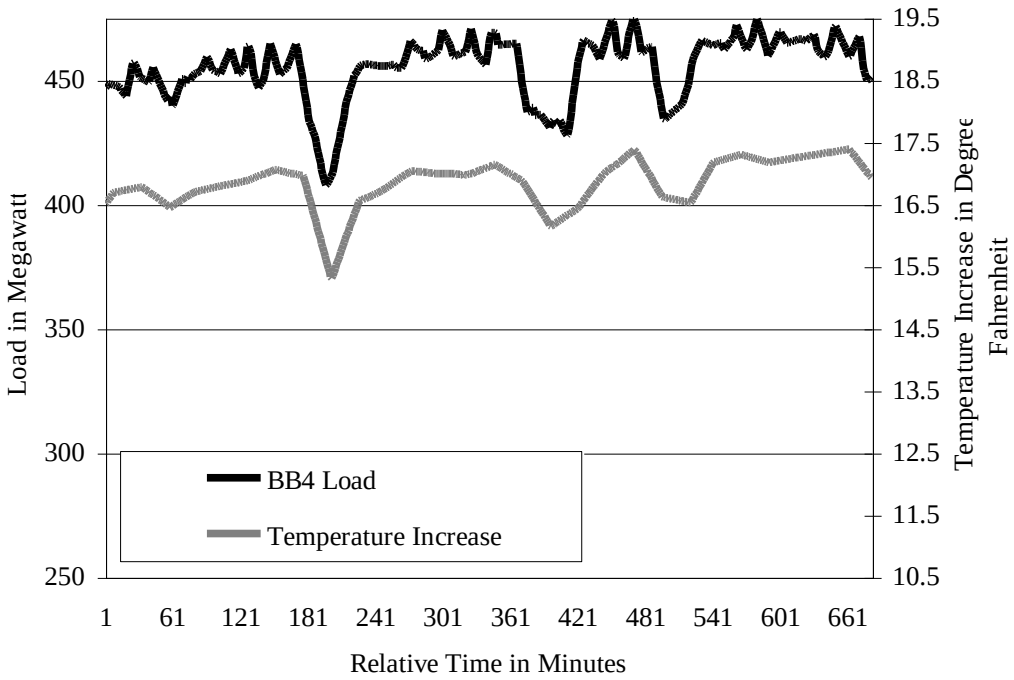

Figure HH. BB4 Load and Temperature Difference Plotted per Minute