

Development of a Java Applet for Generating Trucks from Freight Data

By

Jack Klodzinski, Ph.D.*

Transportation Systems Research Engineer and ITS Lab Manager
Department of Civil and Environmental Engineering
University of Central Florida
P.O. Box 162450 Orlando, Florida 32816-2450
Phone (407) 823-4552, Fax (407) 823-3315
E-mail: jklozdzin@mail.ucf.edu

Ahmad Al-Daraiseh

Expert Systems Engineer and Web page developer, Ph.D. Candidate
Siemens Westinghouse/UCF
Orlando, Florida
E-mail: aldarais@cs.ucf.edu

Michael Georgiopoulos, Ph.D.,

Professor, EE Program
School of Electrical Engineering and Computer Science
University of Central Florida, Orlando, FL 32816
Phone: (407) 823-5338
E-Mail: michaelg@mail.ucf.edu

and

Haitham M. Al-Deek, Ph.D., P.E.

Professor of Engineering and
Director of Transportation Systems Institute (TSI)
Department of Civil and Environmental Engineering
University of Central Florida
P.O. Box 162450, Orlando, Florida 32816-2450
Phone (407) 823-2988
E-mail: haldeek@mail.ucf.edu

Submitted for Presentation and Publication

July 31st, 2003

Transportation Research Board
83rd Annual Meeting

4,518 Words in text, 5 Figures

Total Word Count: 5,768

* Corresponding Author

ABSTRACT

As freight transportation becomes a more significant concern for the future, the ability to accurately estimate truck trips generated by freight activity at an intermodal facility is important for transportation engineering and planning. Previous truck trip generation models that utilize vessel freight data were developed by the University of Central Florida Transportation Systems Institute (UCF-TSI) and proven statistically to accurately determine the number of trucks generated at a seaport. In order to apply these previously developed models more efficiently, a Java Applet was developed to execute a selected Artificial Neural Network (ANN) port model and a trainable ANN port model. This applet provides the user with an easy to understand interface for entering the data and executing the models. One is the previously developed Port Everglades ANN model and the other is a hybrid of the Port Everglades model that allows the user to re-train the ANN model prior to execution for desired data. Though the “trainable” model requires additional re-training with new data, the added complexity comes at the benefit of producing a network with higher degree of accuracy. This applet equips the user with two models and therefore has expanded the capabilities of the previously limited Port Everglades ANN Truck Trip Generation Model. The developed Java Applet can be expanded in the future to include more ANN models and thus more flexibility based on the type of freight data available. This successful adaptation of the ANN model into a Java Applet sets the foundation for further applications.

Key words: freight data, intermodal, trip generation model, heavy trucks, neural networks, applet

INTRODUCTION

Truck traffic can have a significant impact on the operational capabilities of a highway to service the traffic. In order to accurately accommodate the growing traffic on our nation's highways, it is important to provide adequate service for the growing volume of truck traffic generated by freight activity. Simulation modeling can be utilized to visualize existing conditions on a road network and analysis of modeling output used to determine necessary improvements. An intermodal facility usually has significant freight activity and should be considered a special generator of traffic, of which a significant percentage is traffic due to heavy trucks. Truck trip generation models can be utilized for estimating the number of trucks at one of these special generators.

Previously, the University of Central Florida's Transportation Systems Institute (UCF-TSI) developed truck trip generation models using artificial neural networks (ANN) for six of Florida's major seaports (1-5). Each of these models were successfully validated at the 95% confidence level and tested for applications such as short-term forecasting. The models were all developed using the backpropagation neural network architecture.

An ANN model is a network of nodes connected by unidirectional communication channels (connections), which carry numerical data (weights). In some of the popular neural network models (such as the back-propagation ANN) the nodes are organized in layers. A typical back-propagation ANN model consists of three layers of nodes, the input layer (where the inputs or independent variables of the model are applied), the output layer (where the outputs or dependent model variables are produced), and a hidden layer of nodes (where no inputs are directly applied and no outputs are produced, but simply the inputs are appropriately modified prior to their application to the output layer of nodes). A back-propagation ANN with two hidden layers of nodes is shown in Figure 1.

The backpropagation ANN owes its name (a more appropriate name is a multi-layer perceptron architecture, trained with the back-propagation algorithm) to the algorithm used to train its interconnection weights. During the backpropagation of a multi-layer perceptron architecture, an input vector is applied to the network and it propagates forward through its hidden layer to the output layer and produces an output, referred to as the "actual output". This output is compared with the desired output, corresponding to the applied input. This comparison allows the calculation of an "output error", committed by the network. This output error propagated backwards (back-propagated) through the network adjusting the network's weights as it passes through its layers. After all weight adjustments are made for a particular input/desired output vector the next in sequence input/desired output training vector is presented to the network. This process continues until the error function for all training input/desired output pairs is less than a preset tolerance. At this point, the training mode is turned off preventing the further changing of the weights.

There is a vast body of literature on neural network applications to transportation planning, (for a few examples see 6-9). However, except for Al-Deek's research, freight demand models, using ANN to generate truck volumes have not been implemented before. Furthermore, their potential for forecasting future trips was investigated and established (2,3). From Al-Deek's previous research, the Port Everglades ANN model was selected for use in developing a java applet to use for estimating truck trips from freight activity.

Port Everglades continues to capture an expanding market of containerized cargo as part of its growing market of international freight transportation. Port Everglades was ranked 30 out of the top 100 leading U.S. Ports in 2001 handling 21.9 million short tons (10). It is the 2nd

highest volume port for Florida (11). Figure 2 is a layout of the Port Everglades and the identified access roads.

Some of the port's recognizable characteristics are:

- Significant import of petroleum in terms of overall volume (one of the largest non-refineries in the southern United States).
- Exported petroleum is insignificant, there is virtually none.
- No rail activity currently on port property. All freight traffic in and out of the port is by truck.
- High containerized cargo activity and more development is planned based on this freight category.
- Three main access roads to the port's freight terminals that can accommodate heavy trucks (Eller Dr., Spangler Blvd., Eisenhower Blvd.)

Along with petroleum as one of Port Everglades' bulk commodities, the port also handles cement, gypsum and scrap metal. Some of the higher volume break-bulk commodities are building materials, steel products, lumber, and secondary fiber. Some of the containerized cargo products are tile, leather goods, coffee, paper products, auto parts and furniture. There are a number of automobile storage yards around the port as well. Many used vehicles are shipped out which may also generate significant volumes of trucks during periods of high export due to the limited number of vehicles that can be transported by one truck.

LITERATURE REVIEW

In 1991 Sun Microsystems funded an internal research project with a code-name Green (12). This project resulted in a new programming language which is based on the C and C++ language. The language was given the name "OAK", but later it was found that there was already a programming language called OAK. In a local coffee place, the name "JAVA" was suggested and since then the JAVA name has identified this language.

When introduced in a conference in 1995, JAVA generated overwhelming attention. This was due to the many great features the language has such as:

1. purely object oriented
2. platform independence
3. security and exception handling

The language was initially designed for commercial reasons and it utilizes the use of the World Wide Web through special programs called applets.

An applet is a program written in the JavaTM programming language that can be included in an HTML (Web code) page, much in the same way an image is included. When the Java technology is used, it enables the browser to view a page that contains an applet. The applet's code is transferred to the user's system and executed by the browser's Java Virtual Machine. Applets have useful features such as:

1. no installation issues
2. very secure, no need to worry about the client machine being damaged
3. since an applet is embedded in an HTML file, the documentations are shown at the same time the applet is.

Java applets were thought to have possible applications in some transportation areas such as small modeling applications. One such application may be for specific types of trip generation modeling. Those models which do not have proprietary code or if the applet

developer can obtain the code, can be revised and coded for use in a Java applet such as an ANN model.

Al-Deek has utilized ANN to develop a number of truck trip generation models for several seaports in Florida (2-5). In particular, Al-Deek et al. developed a methodology for modeling truck traffic on a road network that was generated by an intermodal facilities freight activity (5). This methodology utilized vessel freight data and freight-hauling heavy truck movements to execute a network micro simulation model. The developed networks had significant freight activity due to the proximity of high trafficked seaports. ANN truck trip generation models were used successfully for estimating the truck volumes on the local roads. These models are prime candidates for implementation as Java applets. As we emphasized above, an applet is a Java program that runs virtually on any platform in a web browser without the need to modify any code. Since an applet uses the powerful features of the JAVA language and can run on virtually any platform it became the first choice of implementing complex problems on the web.

DEVELOPMENT OF THE TRUCK TRIP GENERATION (TTG) JAVA APPLET

Utilizing a java developed applet for the purpose of executing an ANN model provides significant flexibility to the user who is interested in producing an estimate for the number of trucks generated by freight activity from an intermodal facility. Previous research by Al-Deek et al. has shown the usefulness of utilizing a trip generation model for such a task, especially compared to the previous data collection effort required to develop the model (4,5). This research, conducted for the Florida Department of Transportation, involved a tremendous amount of data collection prior to completing the modeling of the defined road networks. The ANN models for the Port of Tampa and Port Canaveral were successfully applied for determining the truck volumes based on the freight activity at the port. This prior research showed what a substantial quantity of data was necessary to complete the modeling effort (2,4,5). Not only is there significant data collection involved with building an ANN trip generation model for a seaport, but also the importance of including truck traffic when modeling an area with freight activity is essential.

The previously developed ANN truck trip generation models were created using the MATLAB computer software; MATLAB software is necessary in order to run the models (2,13). The developed java applet eliminates the requirement of needing the MATLAB software for executing a model. Running a MATLAB executed ANN model is somewhat more tedious than using a model in a java applet. Furthermore, the applet has an easy to understand windows interface. This interface allows the user to copy and paste the data directly into the applet in order to run a developed ANN model such as a port truck trip generation model.

The applet was developed using JBuilder 5.0. JBuilder is a Java SDK (software development kit) distributed by Borland software company. This program allows the user to build Java applications and applets. It also allows the user to build web applications such as servlets and java server pages. This software package is easy to code, debug and run.

Currently the TTG Java Applet includes two components, a trained ANN trip generation model and a trainable ANN trip generation model based upon the previously developed Port Everglades Port Truck Trip Generation Model. The trained model is the actual Everglades Port Truck Trip Generation Model modified to run in the applet (2,4). It requires only the independent variables for successful execution. The model in the applet accurately duplicates the original ANN model developed utilizing the capabilities offered by the MATLAB software package and

MATLAB's neural network toolbox. The trainable ANN model application is based upon the Port Everglades ANN model structure. The data provided to the trainable model is usually available from the port authority in a readable form, which can be easily copied and pasted into the applet window. The data that is provided to the trainable model belongs to three different categories: (a) training data that consists of a list of input (independent) output (dependent) variables to train the model (b) validation data that consists of a separate list of input (independent), output (dependent) variables to validate the accuracy of the trained models, and (c) a test set consisting of a separate list of input (independent) variables to run the model and obtain its resulting output values.

The Port Everglades ANN truck trip generation model was developed from 8 months of field data collection (2,4). The Port of Everglades had average daily truck volumes of over 3,000 trucks per day. The vessel freight activity was provided by the port authority and included 22,129 daily vessel cargo import and export records for year 2000. This represented over 3900 vessel calls.

From the collected data, the following is a list of independent variables which includes storage variables to take into account imports and exports prior to and following the day where output is desired:

- Daily Imported Containers (-2)
- Daily Imported Containers (-1)
- Daily Imported Containers (0)
- Monthly Average Imported Barrels
- Daily Imported Tonnage (-3)
- Daily Imported Tonnage (-2)
- Daily Imported Tonnage (-1)
- Daily Imported Tonnage (0)
- Daily Exported Containers (0)
- Daily Exported Containers (+1)
- Daily Exported Containers (+2)
- Daily Exported Containers (+3)
- Daily Exported Tonnage (0)
- Saturday (1 if Saturday else 0)
- Sunday (1 if Sunday else 0)

The numbers in parenthesis after the variable indicate the number of days considered for storage on site at the port that may influence the number of daily trucks for the specified commodity categories. These are considered "lead" and "lag" variables. The model was trained and validated statistically at the 95% confidence level. A diagram of the ANN model structure is provided in Figure 3.

Both TTG ANN Models in the TTG applet have benefits and drawbacks. The Port Everglades ANN model that is already trained has been established to produce statistically accurate output at the 95% confidence level (2). Furthermore, it has been proven to be applicable for short-term forecasting. It does not require additional modifications after its initial design. If it is executed with input data it will produce outputs that represent predictions of the number of inbound and outbound trucks. Nevertheless, the accuracy of the predictions from the trained ANN model in the TTG applet may be compromised if the characteristics of the port have changed significantly since the development of the model. For instance, if the type of freight commodities or classes used to define one of the independent freight variables has had a

significant change that would influence the generation of trucks. Another example is if the operations of the port have changed the turnaround time of certain commodities by increasing or decreasing the storage time.

The trainable ANN model in the TTG applet has the benefit of versatility for modeling changes in the freight activity or port operations that may have occurred since the development of the original Port Everglades ANN model. However, training and validation of this model is necessary using data sets that include values for both the independent and the dependent variables. Furthermore, acquiring accurate results from training a neural network requires, at times, extensive and time-consuming experimentation, a task for which JAVA applet implementations are not very efficient in doing. The trade-off between the two ANN models in the TTG applet is efficiency versus accuracy. The trained model is efficient and accurate as long as the characteristics of the port do not change significantly. The only data required for the trained model are the values for the independent variables. Using these values, the trained model will provide the predicted values for the output variables. On the other hand, the trainable model provides us with more accurate results especially if the characteristics of the port change with time, and additional model retraining is required. But, the increased accuracy of the trainable model comes at the expense of additional computational complexity involved in retraining, and validating the retrained model, prior to its use.

EXECUTING THE TTG APPLET

Everglades Model (Trained Model)

The user must have at least seven days worth of data to execute this model in the applet to achieve one day of output. The data must be in a tabular format as indicated by the table in the applet. The user selects the number of days to estimate. Once this is done, the user copies the data and then it is pasted into the text area beneath the applet. The “paste” button is then used to transfer the data from the text area to the applet table. The user then clicks the “get inbound and outbound estimates” to execute the model and obtain output. The output will appear in the bottom table. The window interface of the TTG applet, described above, is shown in Figure 4.

Create your Own Model (Trainable Model)

The user must first select the number of data points (days) desired in order to properly train the model. Next, the user selects the number of independent variables that the data consists of. The number of epochs or runs (400 is suggested) must also be selected along with the learning factor (0.01 is suggested to begin). The user provides the applet with the input data as described above and then the “build” button is clicked. Once this is completed, an ANN model is created. The results of running the training data in the applet will appear in the bottom table. If the training data is accurate enough the user can proceed to either validating the model with a separate set of data points or using the model to estimate inbound and outbound truck counts for days selected from the previously used data set. If not, the user can either choose to keep the existing weights and train the model for more epochs or can start the training from the beginning. This may be desired since the randomly chosen set of initial weights affects the accuracy of the model. If the “validation” button is selected, an “error” value is calculated for the inbound and outbound estimates. The window interface of the TTG applet, described above, is shown in Figure 5.

APPLICATIONS OF THE TTG APPLET

Truck trip generation models produce pertinent information for transportation computer modelers, especially when the areas being simulated include a significant freight activity center. Microscopic simulation models such as CORridor SIMulation (CORSIM) and Verkehr In Staedten SIMulation (VISSIM) use the percentage of trucks as model input, and some models can actually be provided with a separate data set consisting only of trucks as part of the model's input (14). The UCF-TSI recently investigated road networks connected to selected Florida seaports and the research results concluded the significant impact trucks can have on the network traffic (5). Due to the difficulty in collecting necessary truck data, especially when there are known high volumes, truck trip generation models provide the modeler with a toll for accurately estimating truck percentages in areas such as an intermodal facility with freight activity.

This TTG java applet allows users to more efficiently utilize their time for the simulation modeling instead of lengthy data collection for a small portion of data (percent of trucks). Previous research on freight movement, access routes traveled by truck drivers, and other simulation modeling may also be able to utilize this TTG Java applet (15, 16, 17). If a network with a seaport is being modeled for a projected year of growth, one of the ANN models can be executed to determine estimated truck traffic based on a forecast of the freight activity of the target year. A seaport may want to do a quick study of the impact on increasing freight activity at one of their terminals. The ANN model will provide a quick estimate of expected truck traffic. This will offer some insight on whether the current roads will have the necessary capacity to handle the growth in truck traffic.

Another possible application is to determine the volume of trucks from freight for use with a modal split model. A number of modal split models have been developed over the years to investigate the modal split based on variables related to the movement of freight. The volume of trucks hauling freight (percent of freight shipped by truck mode) is sometimes a necessary component of a modal split model such as Herendeen documented in the developed regression model (18). The ANN trip generation model can provide rapid results for truck volumes when it is necessary to run a simulation model and test scenarios with various levels of freight activity at a special generator.

CONCLUSIONS

The successful development of java applets for both a trained ANN and trainable ANN model can prove to be a versatile tool for professionals that utilize simulation modeling for analyzing and solving transportation problems. The successful application of an ANN trip generation model in a java applet form shows the ability to continue and expand upon the applet functionality to the limit where it could generate very accurate and complex models and could be used as a general NN modeling tool. Modifications such as including additional ANN models from other ports would expand the versatility by encompassing ports with characteristics different from Port Everglades (see Figures 4 and 5).

The benefits from the TTG Java Applet include the efficiency of this tool because of its ability to run in any windows environment that has internet web browser software. Most applets like this one we developed have a user friendly interface. There is also no high investment cost necessary to run the model unlike software packages such as MATLAB. The applet can also contain more than one model thus providing the user with various alternatives when estimating truck traffic from an intermodal facility.

Furthermore, the applet can also be enhanced in order to produce visualizations, such as graphical analysis of model output (19). Additionally, it can be utilized to visually examine a sequence of runs during a training exercise. Other improvements may be the addition of more algorithms for the trainable model, which expands the user's ability to choose the most accurate model from a suite of models examined during the training process. The number of independent variables used as inputs to the model can also be expanded. However, the more variables used in training, the longer it could take to train the model. If too many variables are included, this could ultimately result in a reduction to the accuracy.

With the change in perspective of national security and identification of possible targets such as seaports, especially those with high containerized cargo volumes, this concept of a java applet that includes selected truck trip generation models may have considerable applicability. The Port Everglades for instance has high containerized cargo traffic. The current applet may be utilized to determine the number of trucks generated by this containerized cargo activity and provide the interested government agencies with this data so the required staff and equipment are available to accommodate implemented security measures which may find it necessary to check these trucks prior to departing those container terminals. Of course if the modeling of heavy trucks will be the focus, then more data may be necessary to reach the desired degree of accuracy and the execution of the applet may not be sufficient.

Though the output from a java applet may not be the most statistically accurate means of generating data for determining the volume of trucks, it provides a reasonable estimate that is useful for computer simulations when a percentage of truck traffic is necessary for input to a simulation model. A Java Applet is very easy to access and very easy to use. It greatly reduces the necessary data for producing the truck percentage value of a local road. The value from the applet described herein could be utilized in a simulation model with the confidence that it was generated by freight data, which has been proven to have a direct relationship to the number of trucks generated by an intermodal facility such as a seaport.

ACKNOWLEDGEMENT

The work of Michael Georgiopoulos and Ahmad Al-Daraiseh is partially supported by the National Foundation grant CRCDC-0203446

REFERENCES

1. Al-Deek, H. M., G. Johnson, A. Mohamed, and A. El-Maghraby. Truck Trip Generation Models for Seaports with Container/Trailer Operation, *Transportation Research Record 1719*, TRB, National Research Council, Washington D.C., 2000, pp.1-9.
2. Al-Deek, H. M. (2002). Using Vessel Freight Data to Forecast Heavy Truck Movements at Seaports, *Journal of the Transportation Research Board No. 1804*, TRB, National Research Council, Washington D.C., pp. 217-224.
3. Klodzinski, J. and H. Al-Deek (2003). "Transferability of an Intermodal Freight Transportation Forecasting Model to Major Florida Seaports" Presented at the 82nd Annual TRB Meeting and accepted for publication in the *Journal of the Transportation Research Board*, TRB, National Research Council, Washington D.C.
4. Al-Deek, H. M., Klodzinski, J. G., Jujare, A., and A. ElHelw. *Development of a Statewide Model for Heavy Truck Freight Movement on External Road Networks Connecting with Florida Ports, Phase-II*. Transportation Systems Institute, Department of Civil and Environmental Engineering, University of Central Florida. Final Report.

- Contract BB-869. Florida Department of Transportation, Tallahassee, Florida, 2001, pp. 1-395.
5. Al-Deek, H. M., J. Klodzinski, A. El-Helw, P. Sarvareddy, and E. Emam. *Development of a Statewide Model For Heavy Truck Freight Movement On External Road Networks Connecting With Florida Ports, Phase III*, Final report, Contract No. BC-355. Florida Department of Transportation, Tallahassee, Florida, August 2002.
 6. Faghri, A.; and J. Hua. Evaluation of Artificial Neural Network Applications in Transportation Engineering. In *Transportation Research Record 1358*, TRB, National Research Council, Washington, D.C., 1992, pp. 17-30.
 7. Ishak, S. and H. Al-Deek. Performance of Automatic ANN-Based Incident Detection on Freeways. *Transportation Engineering Journal of ASCE*, Vol. 125, No. 4, 1999, pp. 281-290.
 8. Haykin, Simon. *Neural Networks, A Comprehensive Foundation*. Macmillan College Publishing Company, New York, 1994.
 9. Hagan, M. T., Demuth, H. B., and M. Beale. *Neural Network Design*. PWS Publishing Company, Boston, 1996.
 10. "Waterborne Commerce of the US", Navigation Data Center, Army Corps of Engineers, <http://www.iwr.usace.army.mil/ndc/wcusnat100.pdf>, (May 2002).
 11. Statistics, Florida Ports Council, <http://www.flaports.org/statistics.htm>, accessed July 2003.
 12. "Introduction to Scientific Programming Online Resources", <http://www.cs.utah.edu/~zachary/isp/resources.html>, accessed June 2003.
 13. MATLAB Student Version. Users' Manual. The Mathworks, Inc., Massachusetts, 1999.
 14. PTV AG Corporation, *VISSIM Ver. 3.5 Manual*, December 2000.
 15. Turnquist, M. A. and G. F. List, "Estimating Truck Travel Patterns in Urban Areas", In *Transportation Research Record 1430*, TRB, National Research Council, Washington D.C., 1994, pp. 1-9.
 16. Corradino Group, *Freight Movement Study*, prepared for Dade County MPO, 1995.
 17. Mahmassani H. and K. F. Abdelghany, Dynamic Trip Assignment-Simulation Model for Intermodal Transportation Networks, *Transportation Research Record 1771*, TRB, National Research Council, Washington D.C., 2001, pp. 52-60.
 18. Herendeen, J.H., Jr., Theoretical Development and Preliminary Testing of a Mathematical Model for Predicting Freight Modal Split." Pennsylvania State University, 1969.
 19. Machine Learning Applets, http://www.seecs.ucf.edu/ml/machine_learning_applets.htm, accessed July 2003.

List of Figures

FIGURE 1 MULTI-LAYER NEURAL NETWORK ARCHITECTURE WITH TWO HIDDEN LAYERS	12
FIGURE 2 PORT EVERGLADES AND ACCESS ROADS	13
FIGURE 3 PORT EVERGLADES ANN MODEL STRUCTURE	14
FIGURE 4 EVERGLADES MODEL (TRAINED) TTG JAVA APPLET WINDOW	15
FIGURE 5 CREATE YOUR OWN MODEL (TRAINABLE) TTG JAVA APPLET WINDOW	16

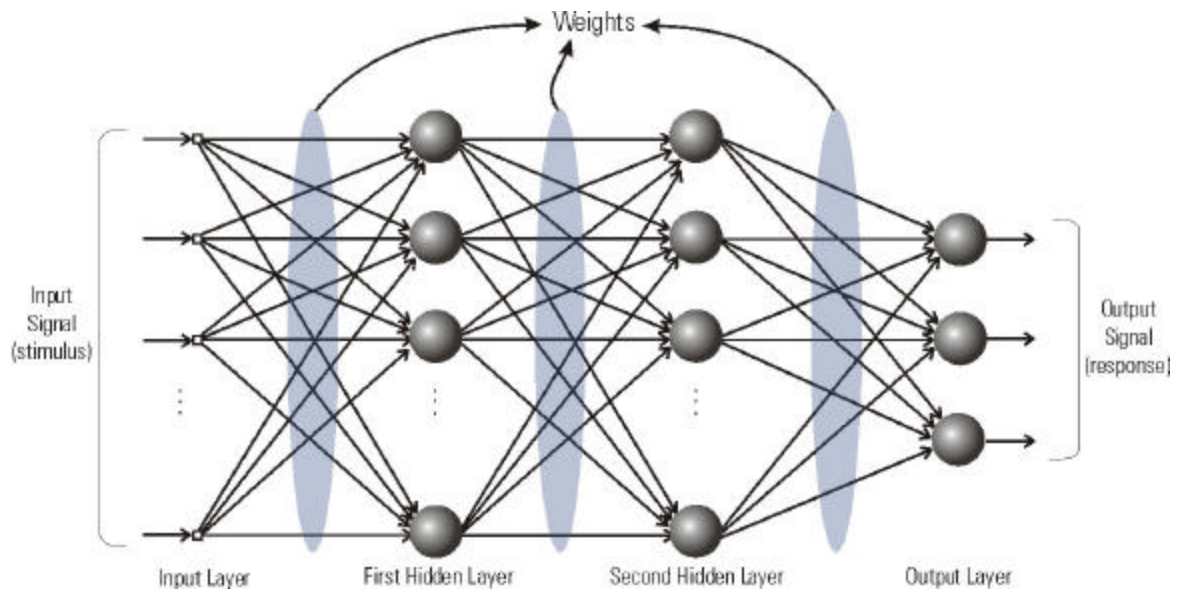


Figure 1 Multi-layer Neural Network Architecture with Two Hidden Layers

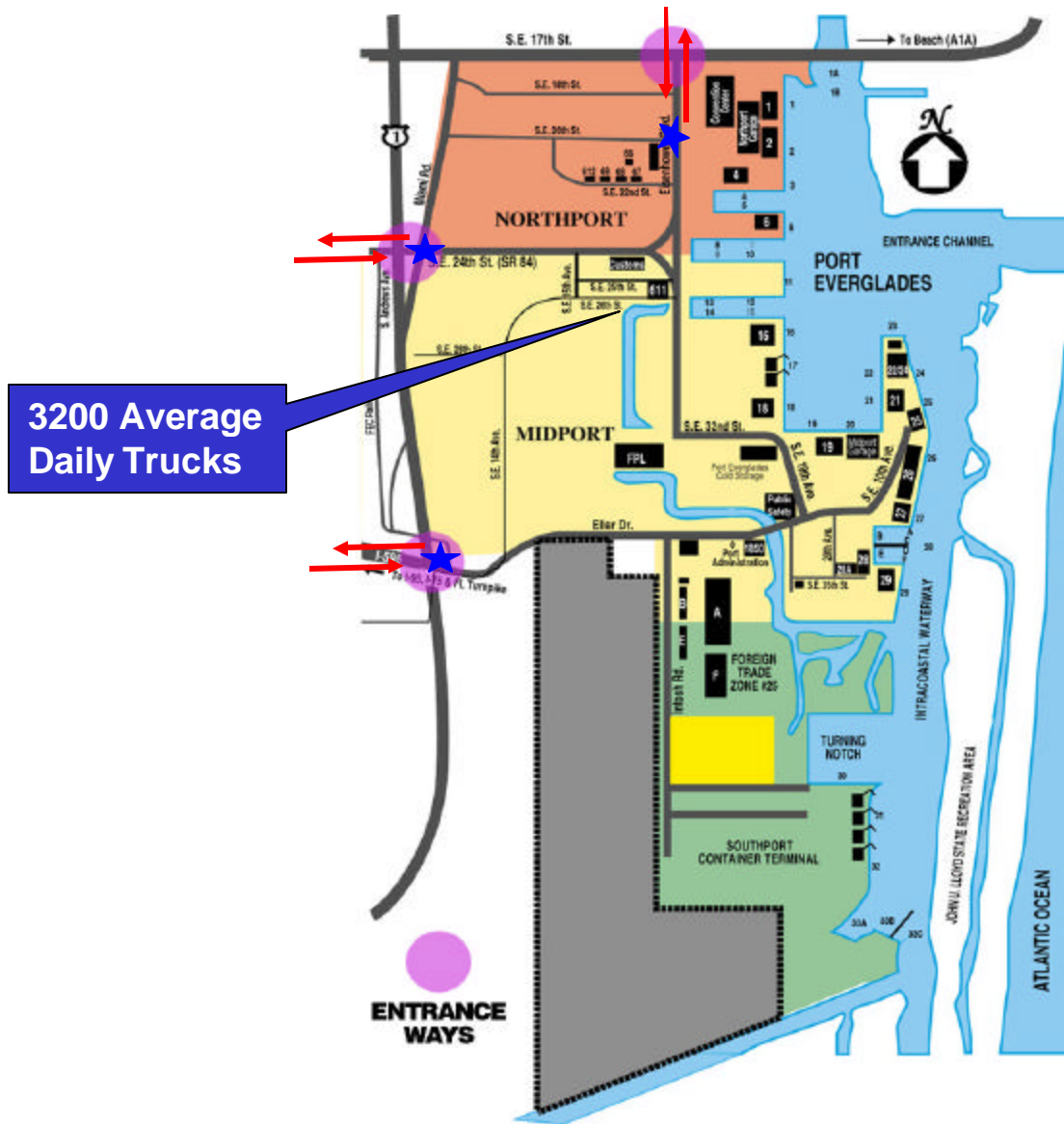


Figure 2 Port Everglades and Access Roads

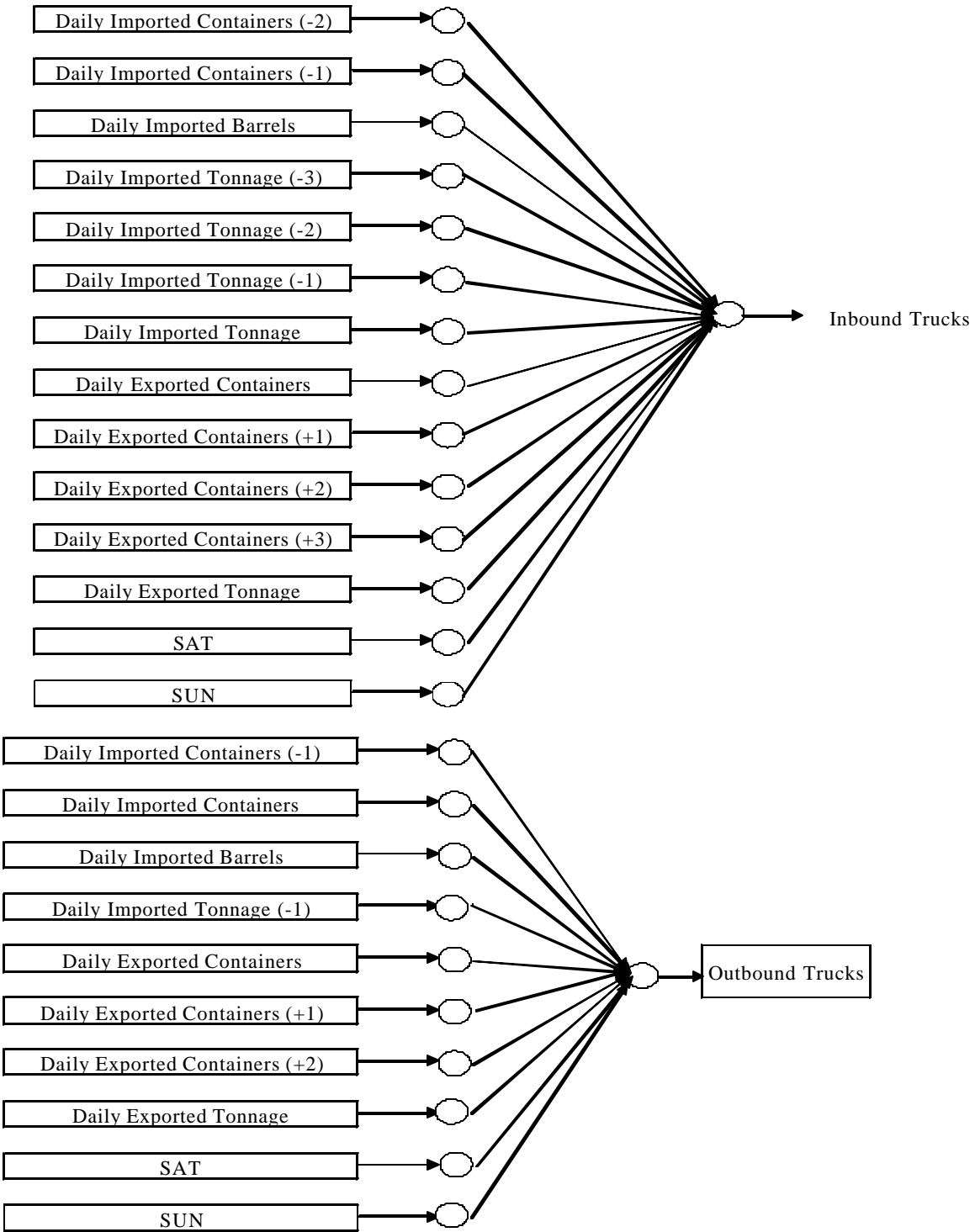


Figure 3 Port Everglades ANN Model Structure

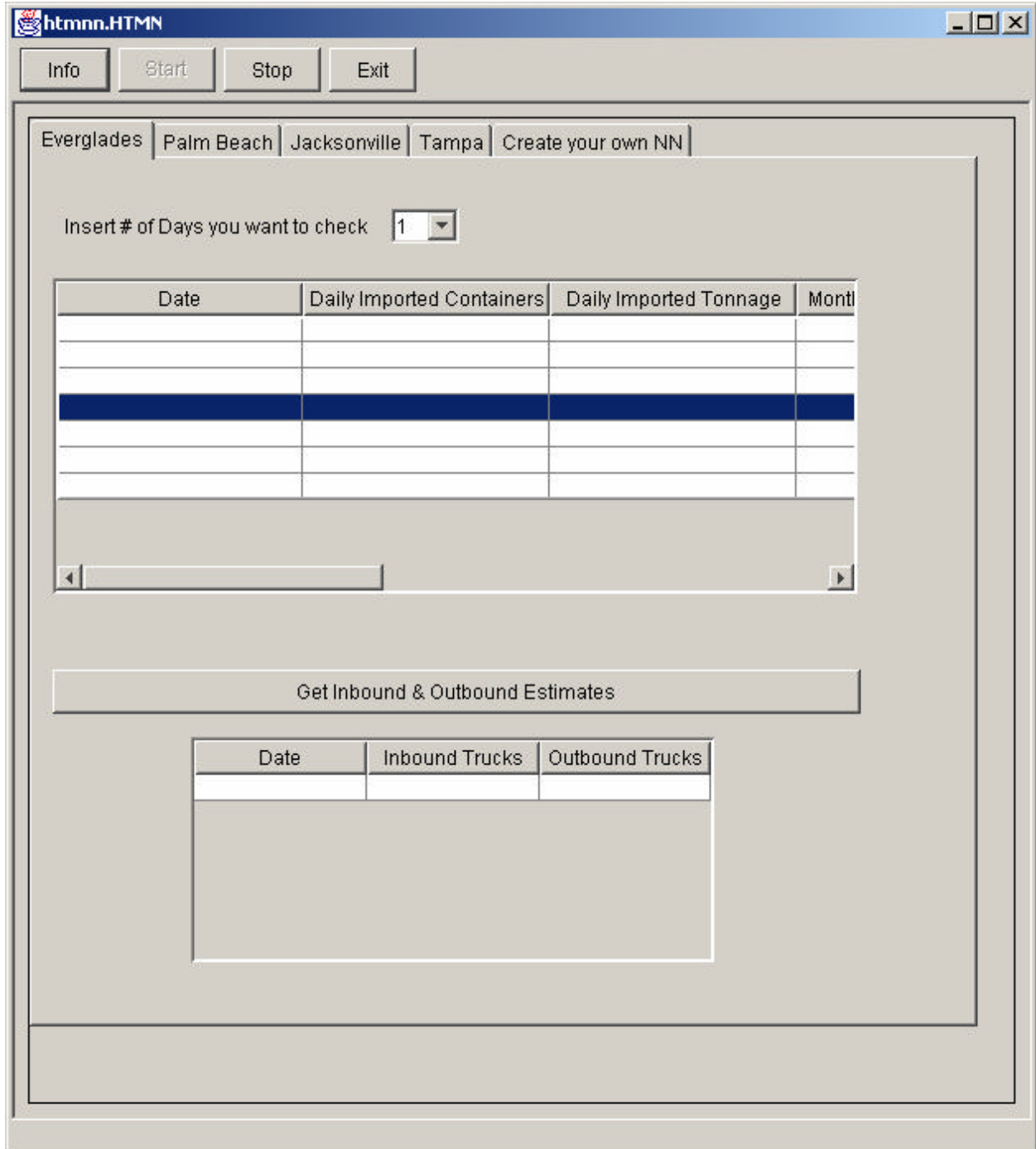


Figure 4 Everglades Model (Trained) TTG Java Applet Window

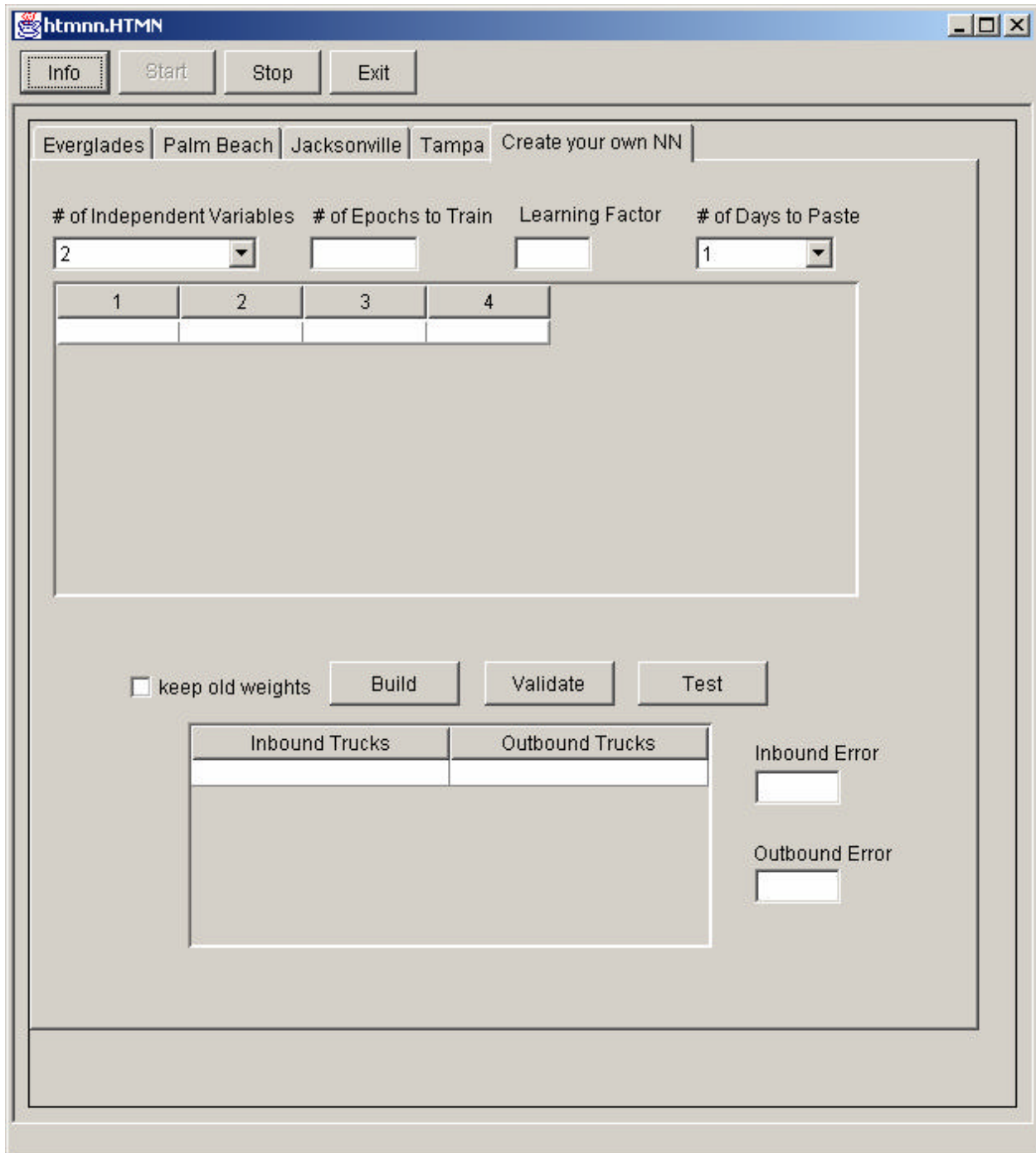


Figure 5 Create Your Own Model (Trainable) TTG Java Applet Window