FINDING ANOMALIES WITH ARTIFICIAL NEURAL NETWORK

Stanko Stankov[1], Stefan Ivanov[2]
Technical University of Gabrovo

Abstract: *Nowadays all companies and corporations have their own external and internal servers with information that require specialized software for their support and configuration. Sometimes when data is exchanged with other external or internal sources for unwanted reasons data traffic may be different than expected. In this case, artificial neural networks may be used to monitor the traction. Thanks to their ability to learn the artificial neural networks can detect an anomaly in communication traffic.*

KEYWORDS*: ARTIFICIAL NEURAL NETWORKS, COMMUNICATION TRAFFIC ANOMALIES*

Overview

Network Management System (NMS) is an application or set of applications that allow network engineers to manage independent network components. NMS systems identify, configure, monitor, update, and troubleshoot networking devices - both wired and wireless in the corporate network. In addition, they collect and display performance data from each network component that allows engineers to make changes to their needs. Such software allows companies to monitor performance on both their networks and external networks [1].

NMS software is useful for :

- Finding a device on the network
- Monitoring device
- Monitoring network productivity
- Network maintenance
- Smart notifications, adaptive alarms

Using NMS software brings the following benefits:

- Time saving - Each IT provider receives direct access to all data as needed.
- Productivity Growth - Helps monitoring every aspect of networks including software, hardware, and other peripherals. The NMS identifies the problem once it appears to ensure that productivity or loss of data will not be lost.

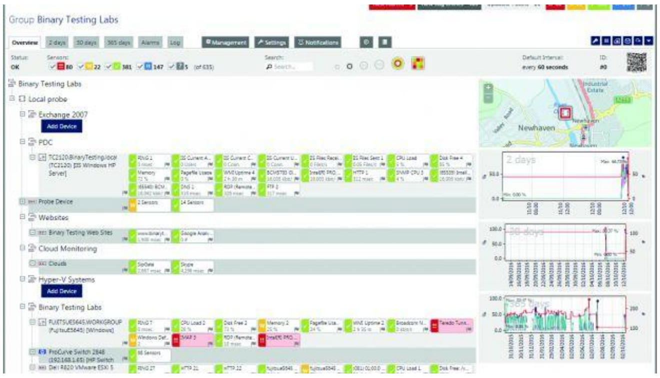


Fig 1. *Picture of a NMS software system*

Predicting events with ANN

One of the most common problems is predicting events, involves understanding the variable value of an element based on previous values and their classification. The purpose of the classifier is to assume the variable (class) by creating a classification model based on the previous values and then using it to predict the next value of the input data. This type of data processing is called controlled learning as the data processing phase is guided by the class variable in the model building.

The power of the neural networks comes from the nonlinearities of the hidden layers when weight correction contributes to the final decision. The ANN model is built after processing the contents of the input data set. Weights associated with interlinked components are constantly changing to achieve high levels of accuracy. These changes are made by the neural networks until they reach the user-defined result.

Describing the problem

An attack on consumer IT infrastructure can lead to an unusually large amount of data passing through the network. One way to protect yourself is to automatically compare the values with historical data based on time and day of the week [2]. The software automatically displays the unusual condition and sends to the administrator a notification to fix the problem.

Realization

Artificial Neural Networks (ANN) are the most common used approach for predicting problems and events [2]. ANN contains interconnected components (neurons) that transform a set of input data into the desired output data. The structure of the ANN is shown in Figure 2

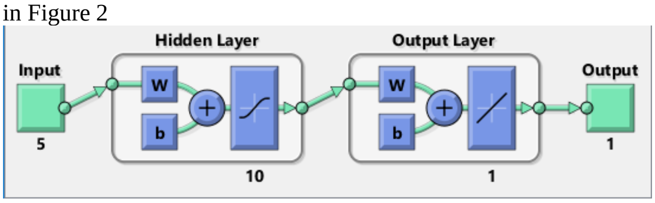


Fig 2. *An example of a ANN structure with 5 input nodes on the input layer, 10 nodes in a hidden layer, and 1 output node in the output layer.*

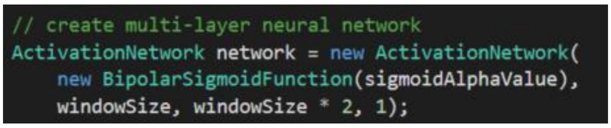
The AForge.NET Framework provides a neural network library that contains a set of classes aimed at creating different types of artificial neural networks and learning to solve specific tasks such as recognition, convergence, forecasting, and so on. The library allows the creation of two categories of artificial neural networks:

- Feed forward neural network with activation function.

This type of networks is represented as by one layer, as by multi layer networks, which don't have recurrent connections - information flaws in these networks from inputs to outputs passing all layers of neural network only one time without doing loops. Neurons of such networks calculate their output by calculating weighted sum of their inputs and passing it to activation function, which value becomes an output of neuron. With the ability to set activation function to use in neural network and configure it size, it is possible to creates different type of networks for different tasks.

- One-layer distance networks.

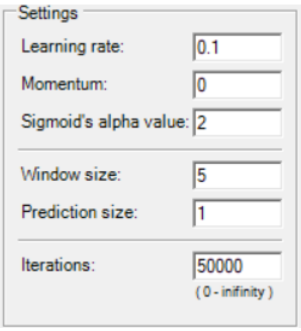
Neurons of this type of networks calculate their output as distance value between neuron's input and its weight - sum of absolute differences.



```
// create multi-layer neural network
ActivationNetwork network = new ActivationNetwork(
    new BipolarSigmoidFunction(sigmoidAlphaValue),
    windowSize, windowSize * 2, 1);
```

Fig. 3 *Creating a multilayer neural network trained with a "Back propagation" algorithm*

For developing a user program for detection of anomalies, abnormal traffic and testing it, the Visual Studio environment and the C # programming language are used. In training neural networks, tests with the following parameters with different values were used for best performance.



Settings

Learning rate: 0.1

Momentum: 0

Sigmoid's alpha value: 2

Window size: 5

Prediction size: 1

Iterations: 50000

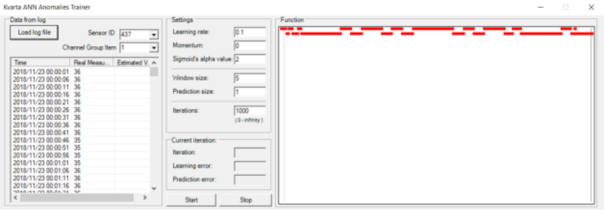
(0 - infinity)

Fig 4. *Neural Network Training Parameters*

Results

Measured values for different types of sensors from the TelecomNMS Network Management System developed by KVARTA were used. Their loading and visualization is shown in Fig 5. The user can experimentally select the desired sensor and set training rules to determine the behavior of the neural network and the parameters for its best training. As the number of iterations increases, the neural network error decreases, ie the neural network is trained. After the training we can see in the table the values that are provided by the neural network how close they are to the real values as well as their visual approximation in the form of the graph.

Firstly, a standalone software was made for developing and testing neural network which was added in Telecon NMS when it was stable.



Kvarta ANN Anomalies Trainer

Data from log

Load log file

Sensor ID 437

Channel Group Item 1

Time	Real Measu...	Estimated V.
2018/11/23 00:00:01	36	
2018/11/23 00:00:06	36	
2018/11/23 00:00:11	36	
2018/11/23 00:00:16	36	
2018/11/23 00:00:21	36	
2018/11/23 00:00:26	36	
2018/11/23 00:00:31	36	
2018/11/23 00:00:36	36	
2018/11/23 00:00:41	36	
2018/11/23 00:00:46	35	
2018/11/23 00:00:51	35	
2018/11/23 00:00:56	35	
2018/11/23 00:01:01	35	
2018/11/23 00:01:06	36	
2018/11/23 00:01:11	36	
2018/11/23 00:01:16	36	

Settings

Learning rate: 0.1

Momentum: 0

Sigmoid's alpha value: 2

Window size: 5

Prediction size: 1

Iterations: 1000

(0 - infinity)

Current iteration:

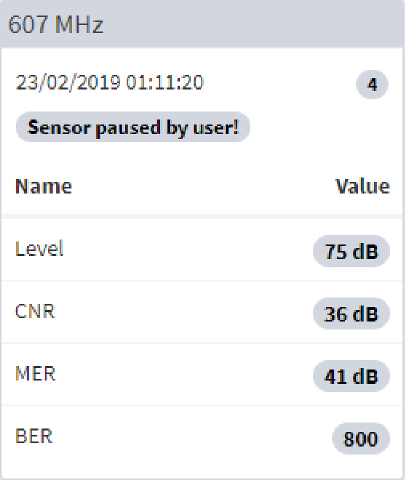
Iteration:

Learning error:

Prediction error:

Fig 5. *Load measured values visually present in software to predict events developed on C # and Aforge Neural Network Library*

Since all sensors have different behaviors once the anomaly detecting software was integrated the predicted values can be viewed on each channel. The user can select a sensor (Fig 6) and by selecting a desired channel based on previous values (minimum values for at least one week) he can view the expected values for the next 7 days.



607 MHz

23/02/2019 01:11:20 4

Sensor paused by user!

Name	Value
Level	75 dB
CNR	36 dB
MER	41 dB
BER	800

Fig 6. *Example of a sensor with 4 channels and their values.*

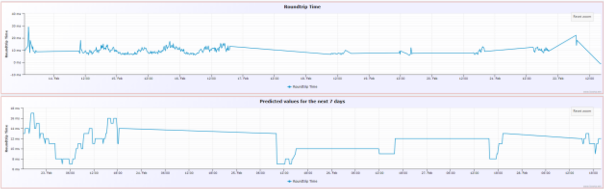


Fig 7. *Sensor channel values for past period and a predicted result for the next 7 days.*

The measured values for the past 60 days are stored in a database, and the user can retrieve the values he wishes to use for training by setting the parameters: start date and time, end date and time, and measurement interval. The displayed values in the database vary depending on past events, resulting in different performance in neural network training. Accordingly, for large variations, there are larger discrepancies between the expected and the real value. Highest efficiency is obtained if, when training the neural network, its initial values are the values that most closely approximate to the expected next values. Figure 8 shows the behavior of the neural network in unpredictable time-varying values.

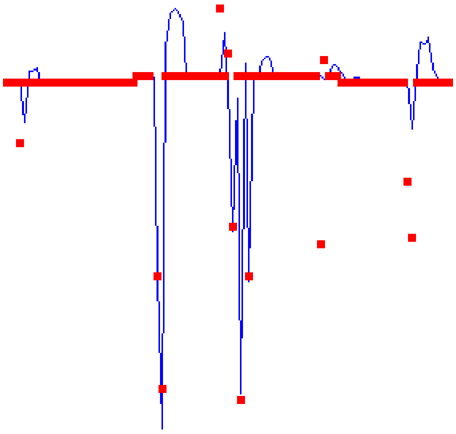


Fig 8. *Neural network training for variable data generated by external factors*

Based on the variations and the database, they can be described as different types of scenarios and the user can receive information at which point of the training what type of scenario is being realized, thus providing a type of training with a neural network teacher to contribute, low operating error.

Conclusion

The development method has been integrated as a feature in Telecom NMS for finding anomalies based on previous events and notifying the administration via email for the channel scenario and has the possibility of future firmware improvement.

References:

[1]https://en.wikipedia.org/wiki/Network_monitoring

[2] Rory P Bunker, Fadi Thabtah, A machine learning framework for sport result prediction– 2017

[3] Monowar H. Bhuyan, Dhruba K. Bhattacharyya, Jugal K. Kalita, Network traffic anomaly detection and prevention - 2017

[4] - https://www.telecomnms.com

[5] - https://www.kvarta.net

[6] - https://www.aforgenet.com