International Journal of
Emerging Research in Management &Technology
ISSN: 2278-9359 (Volume-6, Issue-4)

Research Article

April 2017

Artificial Intelligence in Incident Detection

Kostandina Veljanovska*
Department of Intelligent Systems, Faculty of ICT, University "St. Kliment Ohridski"
Bitola, Republic of Macedonia

Abstract—

The aim of this paper is to prove the ability of artificial intelligence technique - neural network to respond to variable real-time traffic conditions and adapt while performing incident detection. The algorithm of neural network is compared to conventional algorithm. The effectiveness of the algorithms were measured by several measures: correct classification, detection rate and false alarm rate. The results are promising, proving that the neural network algorithm is capable for incident detection on the freeway corridor and also suitable for implementation in the intelligent system as a part of freeway management system.

Keywords— ***artificial intelligence (AI), intelligent system (IS), incident detection, neural network, incident management system***

I. INTRODUCTION

Republic of Macedonia, is at the starting point on the way to safer roads regarding designing intelligent transportation system (ITS) and incident management system and its implementation in traffic management centers. This situation was crucial for starting a research connected with the functional aspects of the intelligent systems for traffic incident management system. In order to support traffic incident management system designing an intelligent infrastructure is needed.

Automatic incident detection algorithm research was done as the key element of every traffic incident management system. Research included one conventional algorithm which uses McMaster methodology and one advanced algorithm in the field of artificial intelligence which uses Multi - Layer Feed - Forward Neural Network with Back Propagation methodology.

There are many efforts in this field such as: hybrid neuro-fuzzy system [1], different types of neural networks as incident detection algorithms [2, 3]. Incident management is the most needed and from the safety point of view the most useful freeway management system function [4, 5]. This research was conducted to prove the ability of NN as freeway incident detection algorithm and to make a starting point of research on implementation of intelligent transportation system via incident management system in the Republic of Macedonia.

II. ARTIFICIAL INTELLIGENCE IN INCIDENT MANAGEMENT SYSTEM

Freeway congestion due to the occurrence of incidents is a major cause of traffic delays. Mitigation of such delays through rapid and reliable incident detection is clearly a vital traffic management objective. Automated incident detection systems are the basic elements of a freeway traffic management systems that rely on Automated Incident Detection Algorithms (AIDA). The purpose of this paper was to develop automatic incident detection (AID) algorithms using Multi Layer Multi Feed Forward with Back Propagation Neural Network. The algorithm's efficiency in detecting incidents was compared to one conventional algorithm - McMaster Algorithm.

In order to test the algorithms, two sets of data were provided. One set of training data while the other set consisted of validating data. Both incident detection algorithms were supposed to be trained on the training data and the control components obtained were tested on the validating data which provided the outputs to substantiate the practicality of these aids.

The core components of these data were the occupancy and volume count data with reference to the time in thirty second intervals and location when the incident was detected by the loop detectors and when they were reported by the reporting agency. These data were part of a large collection from different locations and time slots.

A. Neural Network Design

Neural networks that work with supervision require prior inputs and outputs to configure the system or force it to learn the pattern within the data for validating other data based on the same logic. In our case, we had a neural network with four inputs in one layer namely volume and occupancy at upstream loop detector and volume and occupancy at the downstream loop detector. Network consists of one hidden layer of two neurons, and one output layer consisting of one output neuron indicating whether incident was detected or not.

A number of commercially available software were consulted and primary training of the network was done on them. These include Qwiknet, EasyNN and Neurosolutions. Though these software produced relevant outputs, it was decided to use Excel to form our own customized version of a neural network, as all these software were basically "black boxes" with virtually no information with regards to the actual working of the network.

The core components of processing are the nodes and links that form the network (Fig.1).

1) w15= Weight from Node 1 to Node 5
2) w16= Weight from Node 1 to Node 6
3) w25= Weight from Node 2 to Node 5
4) w26= Weight from Node 2 to Node 6
5) w35= Weight from Node 3 to Node 5
6) w36= Weight from Node 3 to Node 6
7) w45= Weight from Node 4 to Node 5
8) w46= Weight from Node 4 to Node 6
9) wB5= Weight from Bias Node to Node 5
10) wB6= Weight from Bias Node to Node 6
11) w**15= Adjusted Weight from Node 1 to Node 5
12) w**16= Adjusted Weight from Node 1 to Node 6
13) w**25= Adjusted Weight from Node 2 to Node 5
14) w**26= Adjusted Weight from Node 2 to Node 6
15) w**35= Adjusted Weight from Node 3 to Node 5
16) w**36= Adjusted Weight from Node 3 to Node 6
17) w**45= Adjusted Weight from Node 4 to Node 5
18) w**46= Adjusted Weight from Node 4 to Node 6
19) w**B5= Adjusted Weight from Bias Node to Node 5
20) w**B6= Adjusted Weight from Bias Node to Node 6
21) X5= Activation Function for Node 5
22) X6= Activation Function for Node 6
23) X7= Activation Function for Node 7
24) Y5= Output Values for Node 5
25) Y6= Output values for Node 6
26) Y7= Output value for Node 7 (Final Output)
27) ∂ 5= Change in error to be back propagated
28) ∂ 6= Change in error to be back propagated
29) ∂ 7= Change in error to be back propagated

Fig. 1. The core components of the NN created for the algorithm

The complete network can be seen in Fig.2.

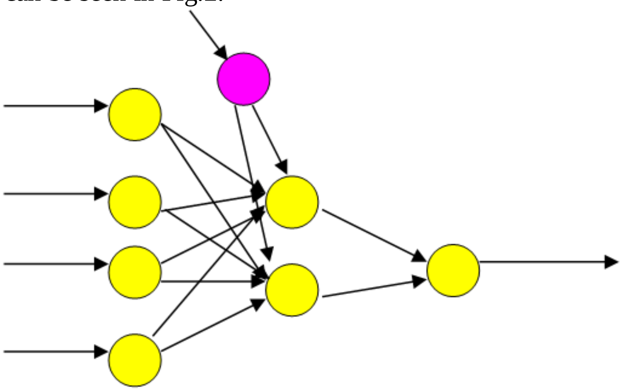

Fig.2 Neural Network created for the Algorithm

The input data that has been used is normalized to reduce the effect of local pull and once the outputs have been finalized, these outputs are once again de-normalized (scaled back). NN algorithm is given as follows: Four inputs were taken which included volume and occupancy counts for both upstream and downstream loop detectors. These inputs were normalized using:

$$2*((x_i-x_{min})/(x_{max}-x_{min}))-1$$

We took average of input and output neuron to get a tentative value for the hidden layer's number of neurons. (in this case (4+1)/2=2.5 or take 2). After that these neurons were connected together and add a Bias neuron. Random weights were generated to initialize the system by putting these on all the links between inputs, bias and between hidden and output layers. Output value for activation function Y for hidden neurons to be used for back propagation was calculated using:

$$Y_j=\sum wij * xi + wb * xb$$

In this case xb and wb are the bias node input and bias node weight respectively. We took bias input constantly as -1. The next step involved the calculation for activation function values used by the hidden layer neurons. This was done using:

$$\mathbf{X_j=1/(1+exp(-Y_j))}$$

Then $\mathbf{Y_k}$ was calculated: $\mathbf{Y_k} = \sum xj \,\mathbf{*w_{jk} + x_b*w_b}$ and also compute $\mathbf{X_k=1/(1+exp(-Y_k))}$.

Change in weights or delta was calculated in final output value which shows either incident or non-incident. This is given by:

$$\partial k = xk(1 - xk) * (T - xk)$$

where T is the given output or target value. Then ∂j was calculated for all the hidden nodes or neurons by using the formula:

$$\partial j = xj(1 - xj) * (\partial k * xj)$$

Finally compute change in the weights and also calculate the adjusted weights by using the formula:

$$\mathbf{wij=wij + \alpha*}(\partial j * xi)$$

where α is the learning rate for the network.

This model has 4 inputs, neurons in the hidden layer and 1 output. A bias node was also added to initialize a threshold function at 1. Our data set was normalized to reduce effect of weight pull and an initial set of randomly generated weights were produced to start the iteration process. Weight randomness was between 0 and 1. The learning rate was taken to be 0.3 so that a better descent could be achieved on the function surface. A macro in Excel was developed to run the several iterations to change the weights and plug them into the next set of data to achieve learning.

A section of about 4000 lines was used from this training data which was again scaled back and output in the form of incident and non-incident (close to 1 and 0) was compared with the validating dataset. This data was also normalized and final outputs were scaled back to original values. A cutoff value was taken at 0.85 to chart the outputs. This was based on observing the trend in data. Any value above 0.85 means that an incident was detected and anything lower means not. We could conclude that our learned data followed the same proximity and indeed locate incidents at most cases of data.

B. McMaster Incident Detection Algorithm Design

This algorithm detects incidents in two distinct phases:

1. Detect the existence of traffic congestion, and
2. Determine the cause of congestion.

For any given detector station, congestion is detected when occupancy and volume readings rise above established thresholds. The cause of congestion is then determined based on readings from the adjacent downstream station. In essence, the cause of congestion is deemed to be a capacity reducing incident if the volume and occupancy readings from the downstream station are sufficiently low. Otherwise, the congestion is deemed to be of the recurrent variety, which arises when traffic demand exceeds freeway capacity.

The core of the McMaster algorithm is based on the logic that traffic downstream of a permanent bottleneck differs from that downstream of an incident – caused (or temporary) bottleneck. Traffic operations are classified into one of four possible traffic states on the basis of two variables: volume and occupancy. These variables are obtained from electronic detectors, located along the freeway. Occupancy, a measure of concentration, is defined as the percentage of time a detector is occupied by a vehicle (or vehicles) during the reporting interval.

First step in this development process was to prepare a template according to which the algorithm can classify data. The procedure for creating the algorithm was classical procedure of McMaster algorithm. The pick period was taken to be from 6:00 to 9:30 am. The maximum un-congested occupancy, defined as **OCMAX** was found to be **0.2**. It was found that as expected, volume – occupancy pairs for the un-congested period tend to cluster tightly about the LUD (lower bound un-congested data) line. The lower bound of the cluster can be established as:

$$Flow = a*(occupancy)^2+b*(occupancy)+c$$

Parameters for the LUD line which best suits the template, adjusted by manual inspection were found to be as follows:

A = - 151   B = 146 C = - 1

$\mathbf{V_{crit}}$ defined as minimum discharge volume, downstream of a road during peak incident free hour was found to be **14**. After the template was prepared, a program was developed in Java.

III. ANALYSIS OF THE RESULTS

From total number of data provided, with processing only 447 were skipped due to loop detector malfunctioning. Total number of data that are taken into consideration is 20918. For 6894 report says there is an incident, and for 14024 report says there is no incident. In the Table 1 (bellow) results that were obtained are shown. The parameters a, b, c for the LUD curve were already adjusted.

Table 1. Results for McMaster Algorithm for different parameters

		Det inc	**% det inc**	**FAR**	**FAR%**	**det nor**	**%det nor**	**correct class**	**% correct cl.**
Vcrit=12	**Ocrit=0.25**	732	10.62	231	1.65	13793	98.35	14525	69.44
Vcrit=12	**Ocrit=0.2**	797	12.35	303	2.16	13721	98.20	14518	69.12

Vcrit=13	**Ocrit=0.25**	1063	15.41	422	3.01	13602	96.99	14665	70.1
Vcrit=13	**Ocrit=0.2**	1154	16.74	532	3.79	13492	96.21	14646	70.02
Vcrit=14	**Ocrit=0.25**	1441	20.90	725	5.17	13299	94.83	14740	70.47
Vcrit=14	**Ocrit=0.2**	1523	22.09	824	5.88	13200	94.12	14723	70.38
Vcrit=15	**Ocrit=0.25**	1842	26.72	1211	8.64	12813	91.36	14655	70.06
Vcrit=15	**Ocrit=0.2**	1891	27.43	1258	8.97	12766	91.03	14657	70.07

As the best performing set of parameters was selected: **Vcrit=14, Ocrit=0.2, A=-151, B=146, C=-1.** Selection of the parameters is dependent on the goal of the detection. If higher detection rate was selected, FAR also goes up. It can be seen that there is no big difference between Vcrit =14 and Ocrit=0.2 and Vcrit =14 and Ocrit=0.25. The results are shown in the Table 2.

Table 2. Results of performance McMaster Algorithm

Measurements	**Percentage**
Correct Classification	70.38%
Non Detected Normal Condition Data	94.12%
Detection Rate	22.09%

Performance of the algorithm with this program is shown in the Table 3 (below). False Alarm Rate is defined as number of detected incidents that were actually not incidents.

Table 3
Performance of the McMaster Algorithm

DR – Detection Rate	**FAR – False Alarm Rate**
12.27	1.65
12.35	2.16
15.41	3.01
16.74	3.79
20.9	5.17
22.09	5.88
26.72	8.64
27.43	8.97

The distance between the loop detectors were analyzed. From the visual inspection of the results, it can be seen that:

- for the incidents that are not detected distance between loop detectors is always between 0.2 and 1.3, for most of them between 0.65 and 1.3 miles.
- for the incidents that are detected distance between loop detectors is always between 0.6 and 1.4, for very few of them in the range of 0.3 and 0.65 miles.
- for the false alarms distance between loop detectors is usually between 1.1 and 1.3 and almost never more than 1.3 miles.

Analysing Back Propagation Neural Network algorithm we can say that this approach was quite different than that used by the McMaster algorithm which involved the formation of a template. We calculated the results from the validating data by asking for the following from our output. The results obtained as summarized below:

Table 4. Results of performance NN Algorithm

Measurements	**Percentage**
Correct Classification	66.84%
Non Detected Normal Condition Data	62.48%
Detection Rate	57.98%

We can see from the results that these values seem to be on the lower side but comparing to other software, we see that we get similar results. However the important point to mention here is that a better understanding of the neural process was established due to the worksheet were all changes made to the system are viewable and where the effect of tweaking can be visualized efficiently. The commercially available software though are more powerful and professional in terms of design and output, but we think that our system is a better learning tool and a good basis for incident detection in incident management system.

IV. CONCLUSIONS

Incident management is the most needed and from the safety point of view the most useful freeway management system function. Successful automated detection of incidents in their early stages is vital for formulating effective response strategies. Research of the algorithms for automated incident detection shows that implementing these new technologies in the incident management process, the number of potential accidents could be reduced, damage could be reduced, number of killed and wounded could decrease.

In terms of producing the conventional algorithms for automated incident detection as McMaster algorithm the most important is to have the data for the specific freeway prior to implementation on field. Because of the random nature of incidents it is not possible to determine which incidents are not detectible. Results show that McMaster algorithm performs excellent false alarm rate and acceptable incident detection rate. But, Neural network (NN) algorithm shows much higher incident detection rate and the data for the specific freeway are not necessary before implementation of the algorithm, although prior to implementation it is possible to model the network in order to speed up the detection process.

REFERENCES

[1] S. Srinivasan, et al., *Freeway incident detection using hybrid fuzzy neural net*work, IET Intelligent Transport Systems, Volume 1, Issue 4, 2007, p. 249 –259

[2] Weerasuriya, Sujeeva A., *"An application of artificial neural networks in freeway incident detection" (1998).* Graduate Theses and Dissertations

[3] L. Weiming, et al., *Improved freeway incident detection using neural network based on pulse data of the loop detector*, IEEE *Xplore*, 2004.

[4] S. Bhonsle, et al., *Database Centered Architecture for Traffic Incident Detection, Management and Analysis*, 2000 IEEE Intelligent Transportation Systems Conference Proceedings, Dearborn, MI, USA, 2000

[5] K. Veljanovska, *Intelligent Infrastructure Functional Aspects of the Traffic Incident Management System Support in the Republic of Macedonia*, MSc paper, Technical Faculty Bitola, University St Clement Ohridski, Macedonia, 2002.