University
of
St Andrews

Third Pactical-CS5011

Reasoning with Uncertainty- Bayesian Networks

Submitted by:

180030908

Submission Date: November 18, 2019

1. Introduction:

In the given report, I have designed Singly connected and multiple connected Bayesian networks. Other than that, I have implemented Variable Elimination Algorithm. After a brief introduction about the topic, I will describe more in detail about what decisions I made during my design and how I implemented the algorithm. I will use screenshots to demonstrate that as well.

Lastly, I have tried to attempt the advanced part of the algorithm as well by defining a self-made scenario and then demonstrating its network through Bayesian Network.

Bayesian Network:

These are graphical models that make use of probability to make decisions about uncertain situations and problems. *Bayesian networks aim to model conditional dependence, and therefore causation, by representing conditional dependence by edges in a directed graph.* [1]

With these relationship between nodes, one can make inference about certain nodes (random variables) in the graph through different factors.

In the report, I have shown Profiling query as well which occurs as a result of Markov Blanket 's process. *A Markov blanket defines the boundaries of a system (e.g. a cell or a multi-cellular organism) in a statistical sense. It is a statistical partitioning of a system into internal states and external states, where the blanket itself consists of the states that separate the two* [6].

Instructions to Run the File from AIspace tool:

Simply load the file 'BN1.xml' or 'BN2.xml' or 'BN3_Advanced.xml' from the file explorer. Make an observation on evidence node and then do toggle monitoring on the query variable node to see its probabilities and to see the effect of the evidence state on its probabilities.

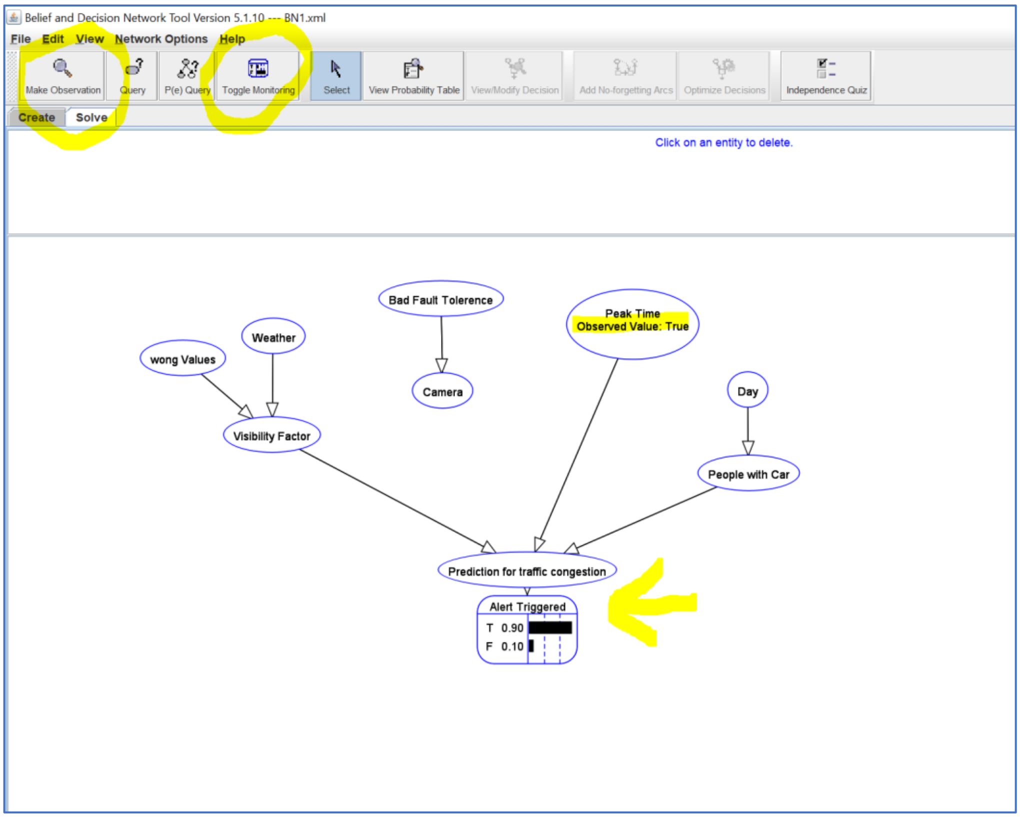


Figure 1: Running files on AISPACE Tool

Instructions to Run the Program:

In command prompt, path of the folder should be given in which A3main.java file is present. After that following two command should be given:

1. javac A3main.java
2. java A3main

In this program, I am not passing any parameters in the arguments. User have to give the value of query variable, evidence, their Boolean state to the 'VariableElimination' function of the code. If user wishes to pass no evidence, just pass null to that and for its state, it can be true or false as it won't affect the result in that scenario.

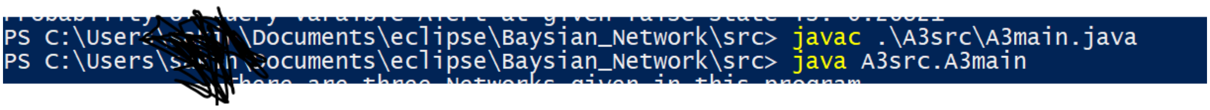


Figure 2: Commands to execute the program

```
VariableElimination(BaysianNetwork.Alert,false,BaysianNetwork.PeakTime,false);
```

```
VariableElimination(BaysianNetwork.Alert,false,null,false);
```

Figure 3: Adding parameters in function on A3main Class

```
PS C:\Users\[redacted]\Documents\eclipse\Baysian_Network\src> javac .\A3src\A3main.java
PS C:\Users\[redacted]\Documents\eclipse\Baysian_Network\src> java A3src.A3main
----------------There are three Networks given in this program---------------------
In the 'VariableElimination' function, give query X variable and evidence along with their boolean values--------------
-----------------------------------------------------------
Select from A,B,C,D if you want to use Simple Chain Network
-----------------------------------------------------------
Select from Fire,Alarm,Smoke,leaving,report if you want to use Fire Alarm Example
-----------------------------------------------------------
Select from PeakTime,PredictionForTrafficCongestion,Alert if you want to use Traffic Congestion Example
--------Values in the network list before reversal----------
Alert
PredictionForTrafficCongestion
PeakTime
----------Values in the network list after reversal----------
PeakTime
PredictionForTrafficCongestion
Alert
The first node to be eliminated now: PeakTime
value at T: 0.37
value at F: 0.63
Values at child :PredictionForTrafficCongestion:---------------

1.0

0.0

0.7

0.3
After multiplication with parent node state, child values become: TT: 0.37 ,TF: 0.0 ,FT: 0.44099999999999995 ,FF 0.189

After addition child values become at T, F: 0.8109999999999999 , 0.189
 New probabilities of the PredictionForTrafficCongestion node after normalization at true and false states are:
0.8109999999999999 , 0.189
Probability of Query Varaible PredictionForTrafficCongestion at given false state is: 0.189
The first node to be eliminated now: PredictionForTrafficCongestion
value at T: 0.8109999999999999
value at F: 0.189
Values at child :Alert:---------------

0.9

0.1

0.01

0.99
After multiplication with parent node state, child values become: TT: 0.7299 ,TF: 0.0811 ,FT: 0.00189 ,FF 0.18711

After addition child values become at T, F: 0.7317899999999999 , 0.26821
 New probabilities of the Alert node after normalization at true and false states are:
0.7317899999999999 , 0.26821
Probability of Query Varaible Alert at given false state is: 0.26821
PS C:\Users\[redacted]\Documents\eclipse\Baysian_Network\src>
```

Figure 4: Output of the program

2. <u>Design and Implementation</u>

Part 1: Design of Bayesian Networks
Singly Connected Network:

Figure 1 shows the singly connected network. A singly connected network is one where between two nodes, there is always one path [3]. In the first part of the assignment, I designed two networks using AIspace tool. Given problem states that we need to design a solution for predicting the risk of traffic congestion at the entrance of Meridonia City Hospital. Different identified domain variables (nodes) and relationship between them is shown in Figure 1.

1. There is a visibility sensor that's value is either high or low and it is affected by the state of the weather. Also, these values of the senor can be affected by wrong

values. Hence the parent nodes of the visibility sensor are weather and wrong value node.

I assumed that the chances of having all four kinds of weather is the same.

Probability Table for Weather

	P(Weather=sunny)	P(Weather=overcast)	P(Weather=raining)	P(Weather=snowing)
Prior Probability	0.25	0.25	0.25	0.25

No observed value for this node.

OK

For wrong values, there is a chance of 5 percent so its probability in table is 0.05.

Probability Table for wong Values

	P(wong Values=T)	P(wong Values=F)
Prior Probability	0.05	0.95

No observed value for this node.

OK

Now probabilities of these two tables affect the probability and result of visibility sensor. In the image below, you can see that if weather is sunny and error value is present, there are 95% chances that sensor will show high visibility other wise with sunny and no error value, the visibility level of sensor is high. From the table, you can see the same with all other weathers as well. Like for snowing, if there are wrong values, visibility probability is 10% and without error values is 15%, showing a difference of 5% in values.

I connected visibility node with prediction for traffic congestion node as its effecting the result of the prediction.

Probability Table for Visibility Factor

Weather	wong Values	P(Visibility Factor=High)	P(Visibility Factor=Low)
sunny	T	0.95	0.05
sunny	F	1.0	0.0
overcast	T	0.55	0.45
overcast	F	0.6	0.4
raining	T	0.25	0.75
raining	F	0.3	0.7
snowing	T	0.1	0.9
snowing	F	0.15	0.85

No observed value for this node.

OK

2. Timings of the hospital effects the traffic flow in hospital as well. As on peak times, there is more crowd then other time. In the specifications, it was given that peak

times of hospital are between 6-10 am and 5-10pm. I calculated the hours between these periods and divided them with total 24 hours to find the probability of having a peak time in hospital. Below image shows that

Probability Table for Peak Time

	P(Peak Time=True)	P(Peak Time=False)
Prior Probability	0.37	0.63

Observed value : True

OK

3. Week days and weekends effect the number of cars in the area as well. So first I calculated the probability of having a week day and weekend. After that I directed it towards number of people with car in that area. The probability distribution is shown below.

Probability Table for Day

	P(Day=Holiday)	P(Day=Working)
Prior Probability	0.22	0.78

No observed value for this node.

OK

Probability Table for People with Car

Day	P(People with Car=T)	P(People with Car=F)
Holiday	0.3	0.7
Working	0.6	0.4

No observed value for this node.

OK

4. There is a camera is placed at entrance of hospital as well to indicate high or low traffic flow. It has a probability of having error values as well just like visibility sensor. I did not connect the camera node with any other node in singly connected network as its mentioned in specification camera that it is not affecting the results. The probability of camera with parent bad fault tolerance node is shown below

Probability Table for Bad Fault Tolerence

	P(Bad Fault Tolerence=T)	P(Bad Fault Tolerence=F)
Prior Probability	0.05	0.95

No observed value for this node.

OK

Probability Table for Camera

Bad Fault Tolerence	P(Camera=High)	P(Camera=Low)
T	0.45	0.55
F	0.5	0.5

No observed value for this node.

OK

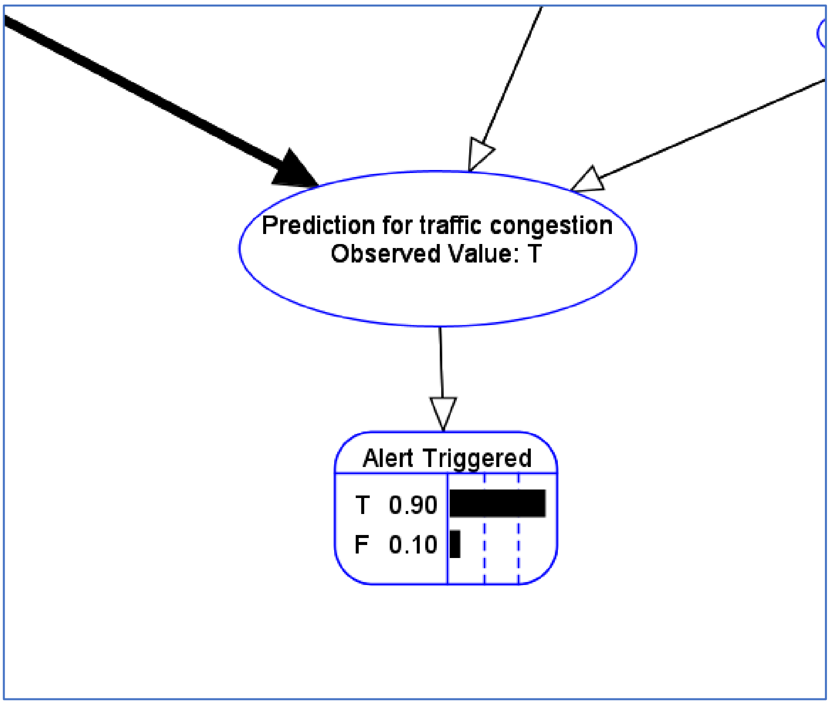
5. The alert will only be called if there is a prediction of traffic congestion but only 90% of the times. So, in the below observation will show that even when traffic congestion is true, there is 10% that alert will be false.

Probability Table for Alert Triggered

Prediction for traffic congestion	P(Alert Triggered=T)	P(Alert Triggered=F)
T	0.9	0.1
F	0.1	0.9

No observed value for this node.

OK



Queries for Singly connected network:

Diagnostic query:

The direction of this query is from child node to parent node. That is, we know the effect and want to predict about the cause of that effect. So, in this query, I set the visibility factor's value as true and I want to see the probability of different types of weather with that.

Following [2], I can write this query as:

P(Holiday| T): probability of holiday when there is prediction about traffic congestion

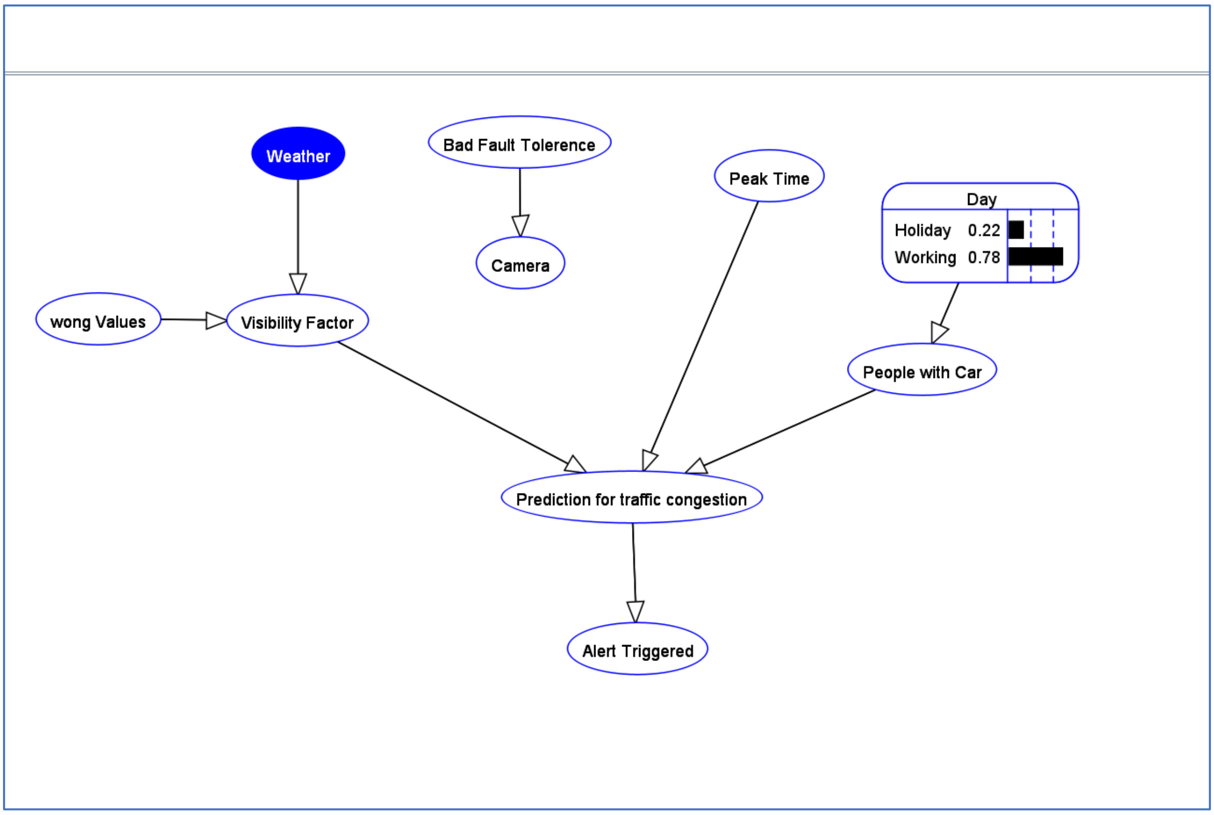


Figure 4: Problem investigation without observation

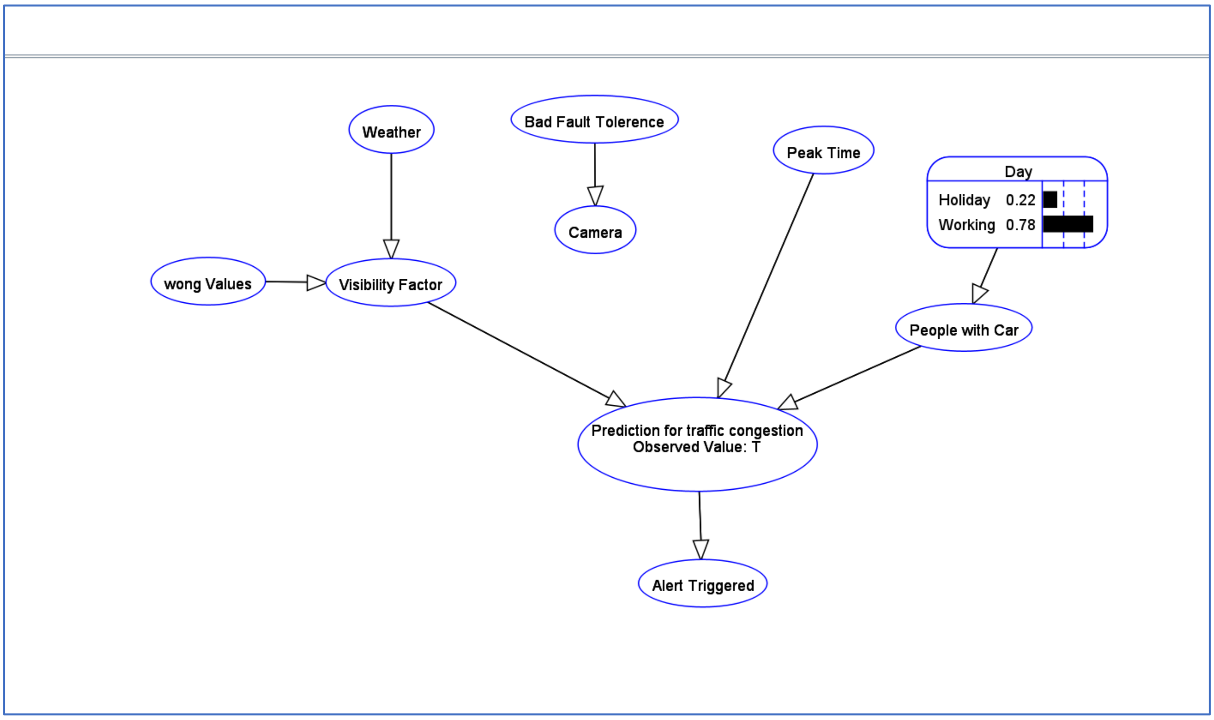


Figure 5: Problem investigation with observation

Predictive query:

It is opposite of diagnostic query, where we move in direction from parent to child. As here we know probability of the cause and we want to predict the probability of the effect.

Following [2], I can write this query as:

P(T| Holiday): probability of prediction about traffic congestion on a holiday.

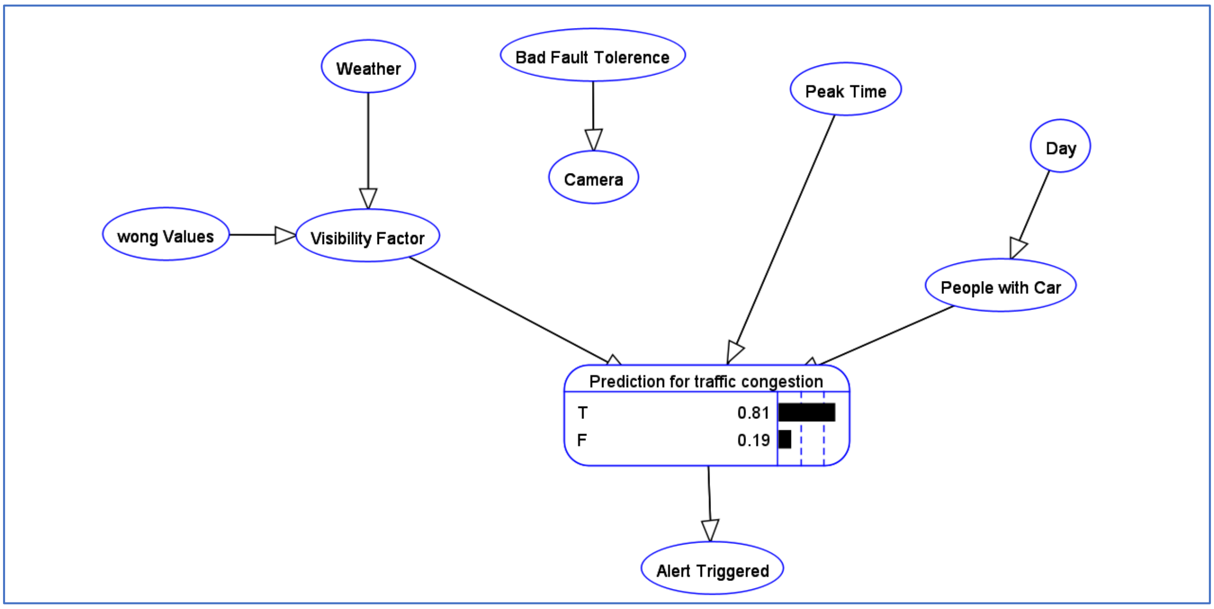


Figure 5: Problem investigation with no observation

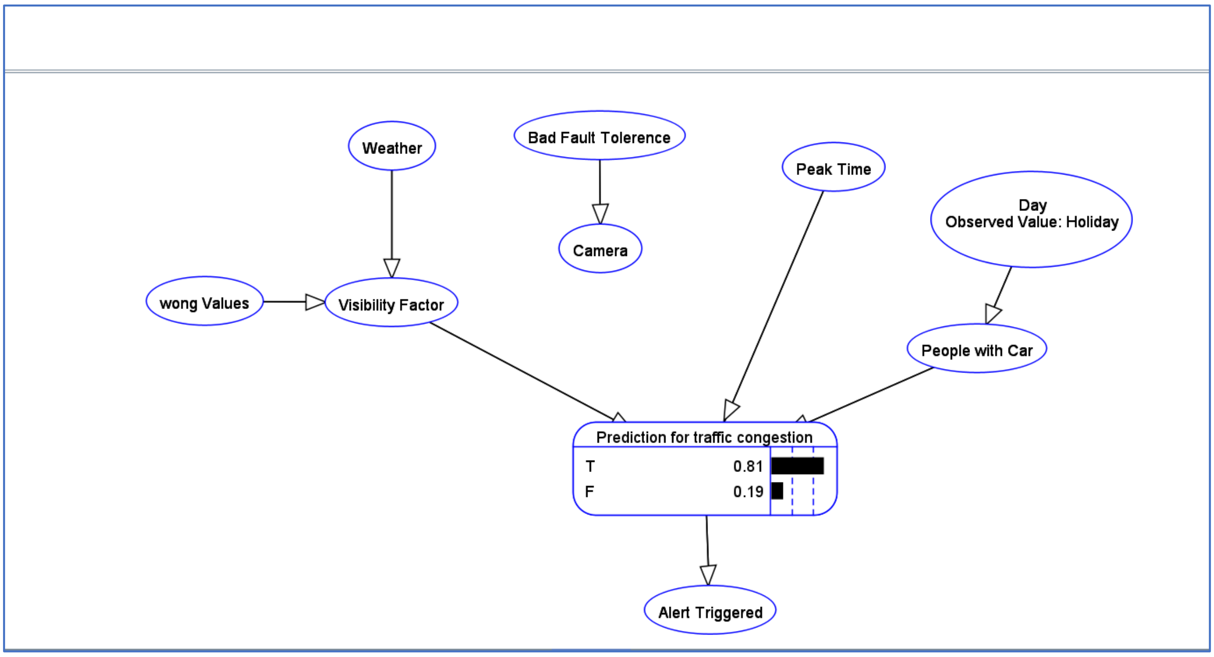


Figure 6: Problem investigation with observation

Profiling query:

In profiling query, we set a state for the query variable and then see its impact on its parent nodes, child nodes, parents of the child nodes and all ancestor nodes.

So here the query is while prediction of traffic congestion is true, what is the effect of probability on the child nodes, parent nodes and parents of child nodes. This is done by applying Markov's Blanket concept.

Writing it in formal way, it can be like that:

P(Visibility Factor ,Alert Triggered ,Peak Time, People with Car | Traffic Congestion Prediction)

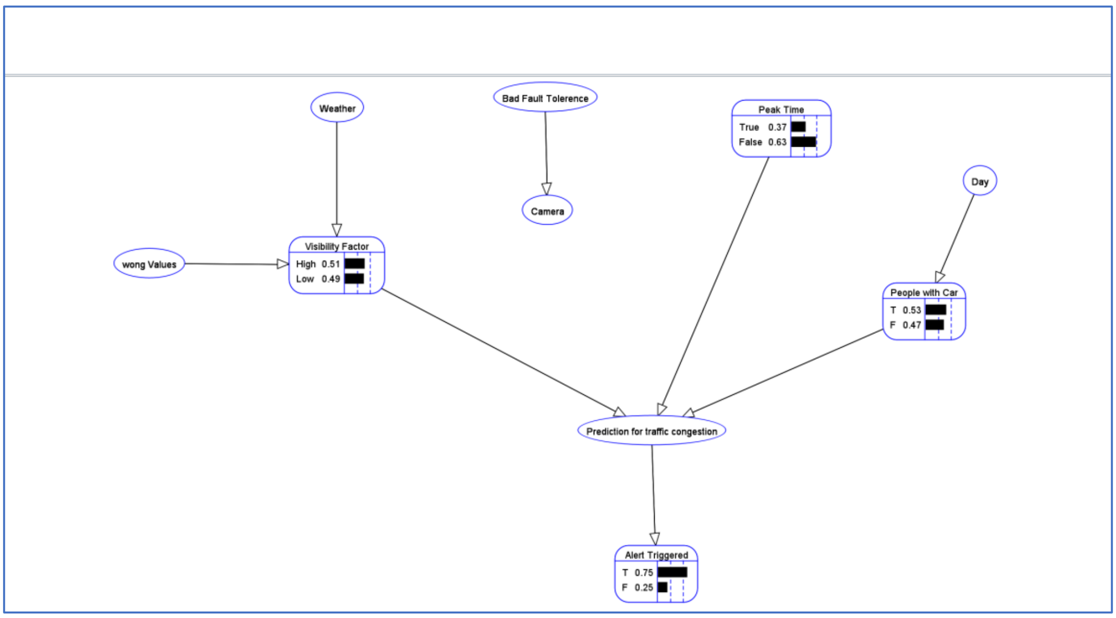


Figure76: Without Observation

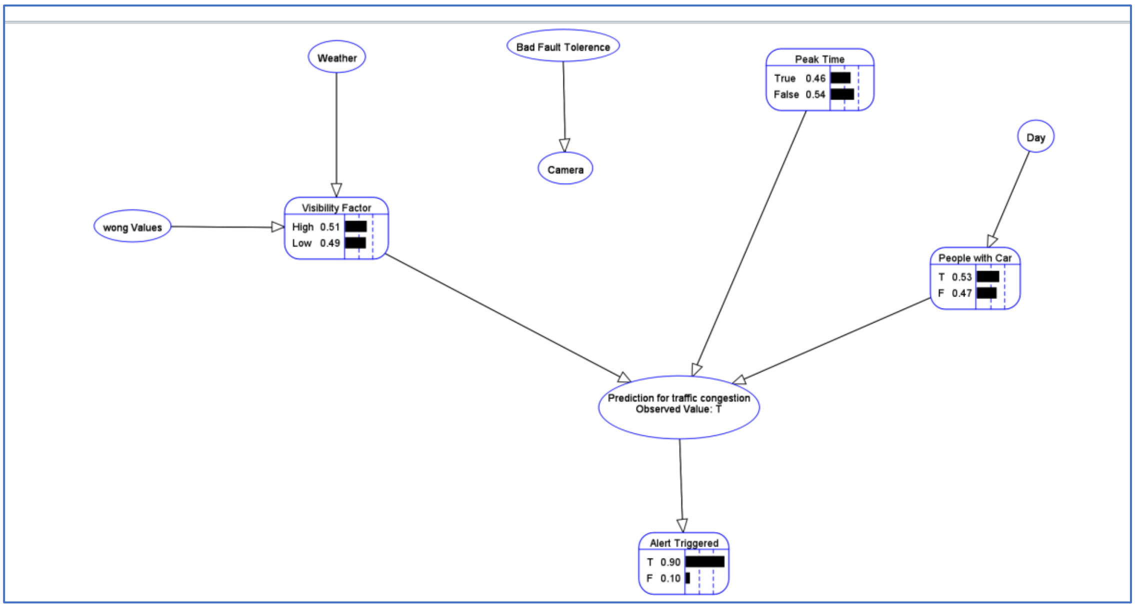


Figure 8: With Observation

Multi Connected Network:

Unlike singly connected network, in multi connected network, there can be more than one paths between two nodes. In the given problem, based on my assumptions, I have added different connections and paths between the nodes. Below image shows that I have added three more connections, one from visibility factor to camera as I think Visibility factor will affect the result of the camera which will then affect result of traffic congestion prediction. Other I think, peak time will affect the number of people with car as well as at busy periods, more cars will be in hospital area. Therefore, I added a connection from Peak time towards People with Car node.

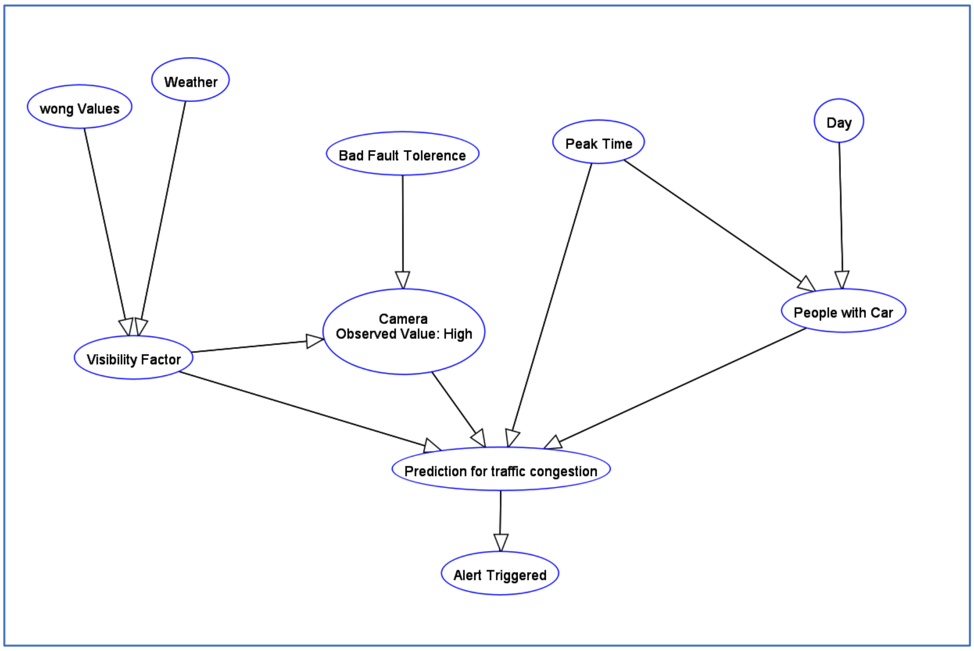


Figure 9: Multi- Connected Network

Now with the addition of these new connections, the probability distribution of the following nodes will be updated like that:

Probability Table for Prediction for traffic congestion

Peak Time	People with Car	Camera	Visibility Factor	P(Prediction for traffic congestion=T)	P(Prediction for traffic congestion=F)
True	T	High	High	1.0	0.0
True	T	High	Low	0.8	0.2
True	T	Low	High	1.0	0.0
True	T	Low	Low	0.8	0.2
True	F	High	High	0.6	0.4
True	F	High	Low	0.6	0.4
True	F	Low	High	0.6	0.4
True	F	Low	Low	0.6	0.4
False	T	High	High	0.7	0.3
False	T	High	Low	0.7	0.3
False	T	Low	High	0.7	0.3
False	T	Low	Low	0.7	0.3
False	F	High	High	0.2	0.8
False	F	High	Low	0.1	0.9
False	F	Low	High	0.2	0.8
False	F	Low	Low	0.1	0.9

No observed value for this node.

OK

Probability Table for Camera

Visibility Factor	Bad Fault Tolerence	P(Camera=High)	P(Camera=Low)
High	T	0.45	0.55
High	F	0.5	0.5
Low	T	0.45	0.55
Low	F	0.5	0.5

Observed value : High

OK

Probability Table for People with Car

Peak Time	Day	P(People with Car=T)	P(People with Car=F)
True	Holiday	0.3	0.7
True	Working	0.6	0.4
False	Holiday	0.3	0.7
False	Working	0.6	0.4

No observed value for this node.

OK

Predictive Query:

Here I will use the same query that I used for Singly connected network to see the difference in the result of probabilities.

P(T| Holiday): probability of prediction about traffic congestion on a holiday.

If we compare results of below image with the one from singly connected network, we see that the probabilities of having a traffic congestion in multi connected network is reduced now it is influenced by the result of peak time as well.

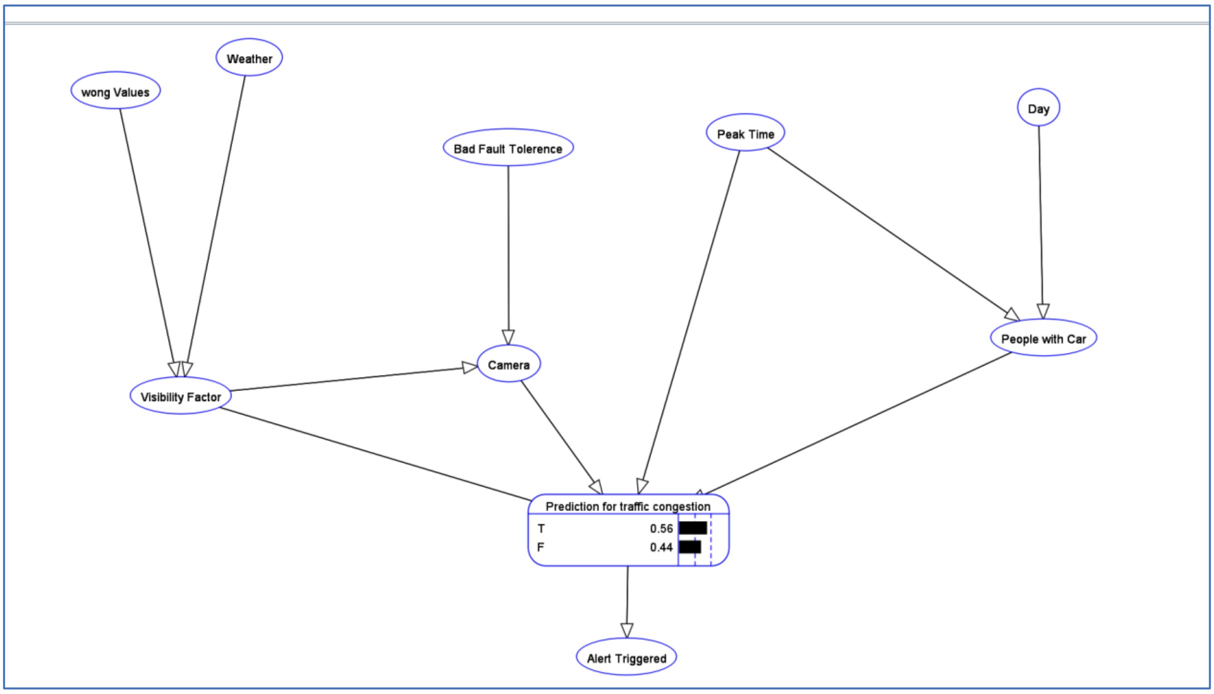


Figure 10: Without Observation

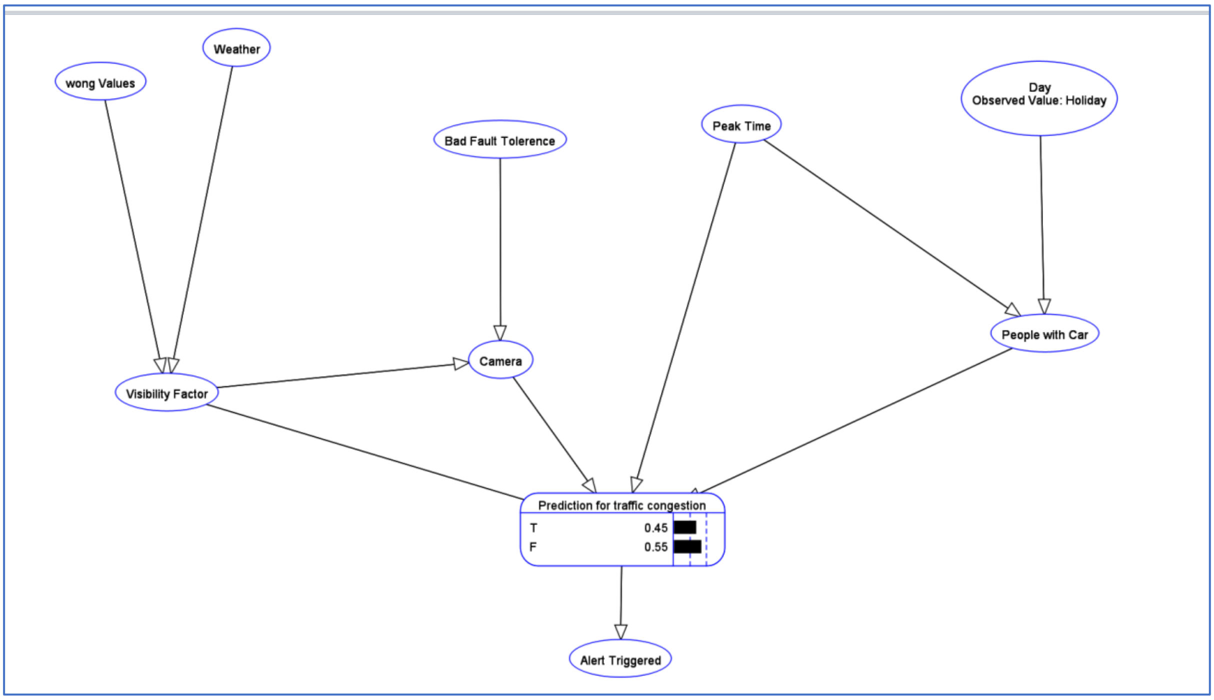


Figure 11: With observation

Diagnostic query:

P(Holiday| T): probability of holiday when there is prediction about traffic congestion. Again, as compared to single connected network, here prediction for having a working day increased as its being influenced by the probabilities of the traffic congestion as well which was different in case of singly connected network.

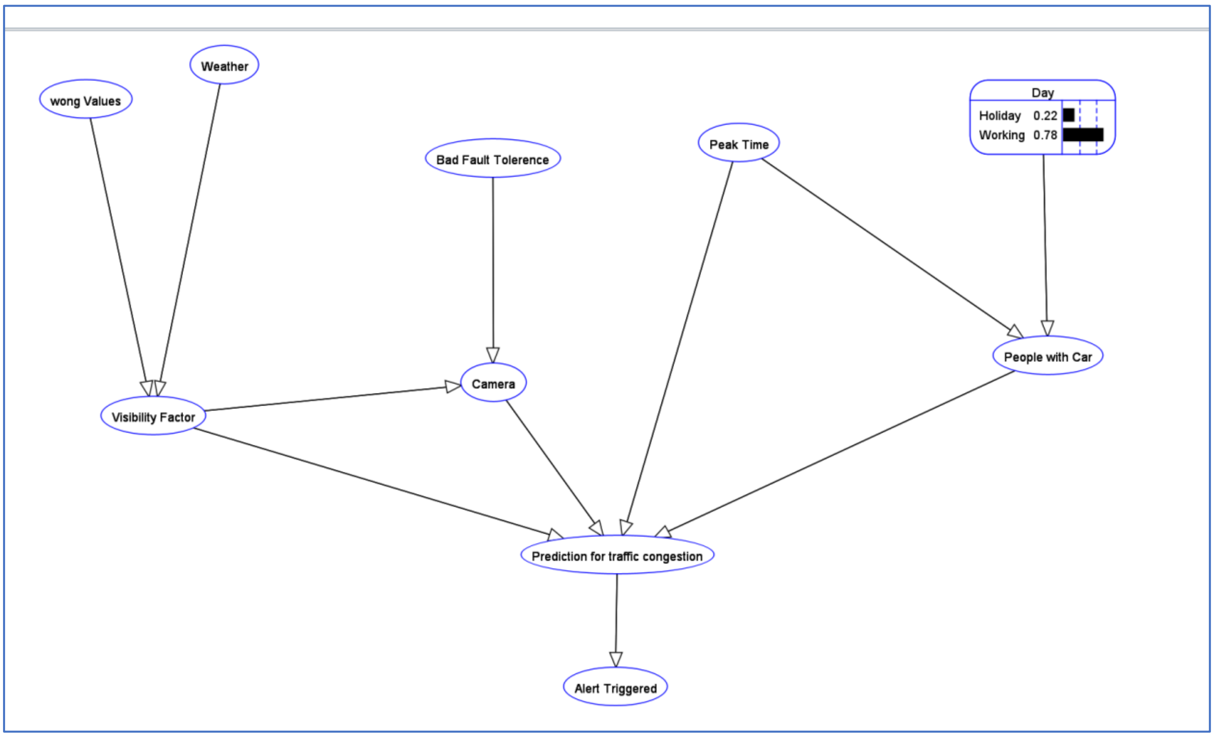


Figure 12: Without Observation

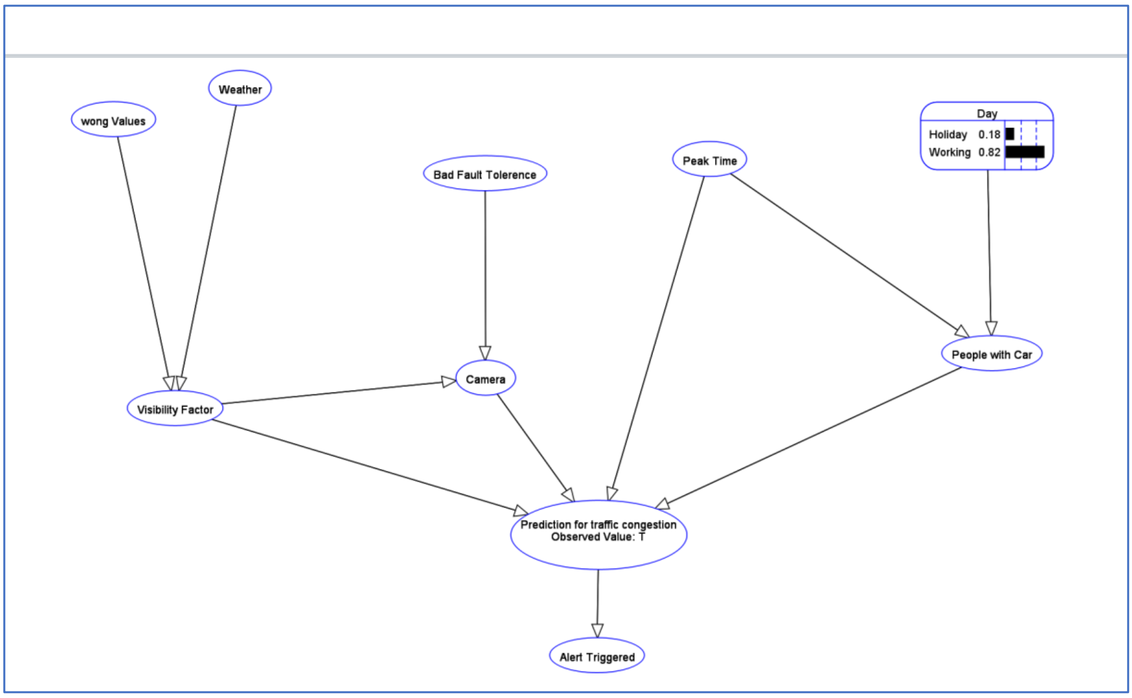


Figure 13: With Observation

Profiling Query:

P(Visibility Factor ,Alert Triggered ,Peak Time, People with Car, Camera| Traffic Congestion Prediction)

This means that we need to find probability of all children, parents, children parent's nodes when the probability of query variable 'Traffic Congestion Prediction' is true.

This query and its results are shown below which are different from singly connected network. It is because the probability of people with car is now influenced by the peak time and now camera results are influencing the results of the traffic congestion.

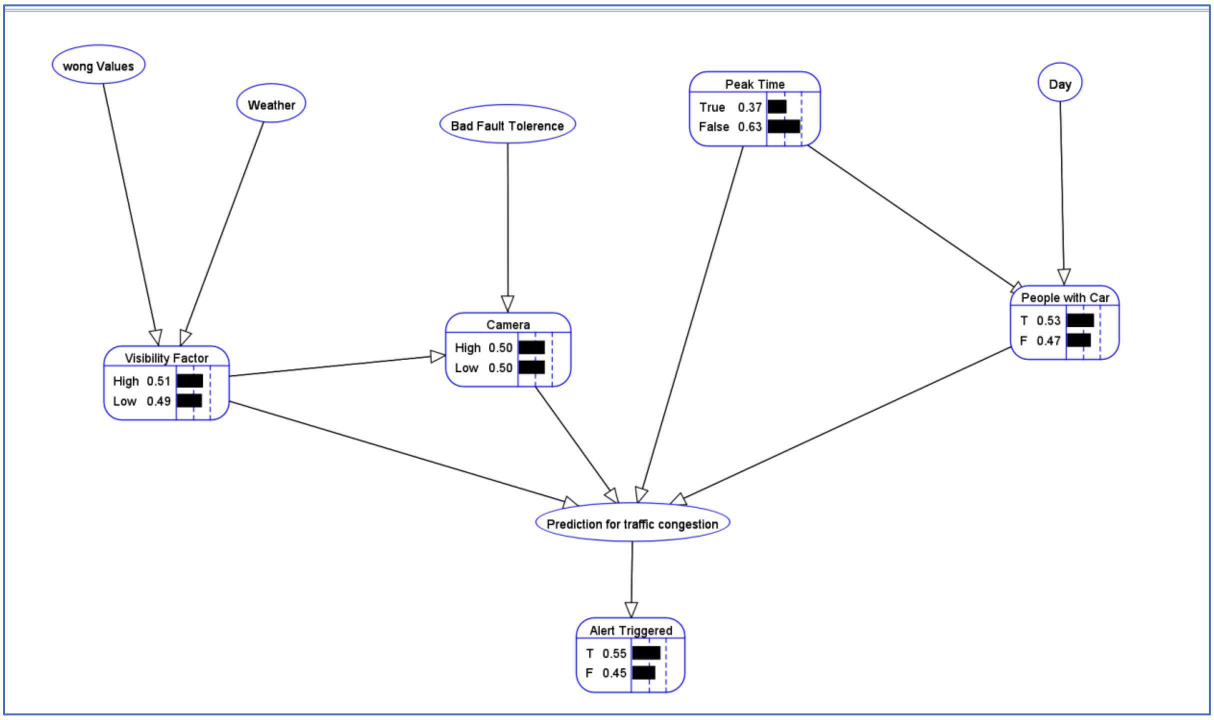


Figure 14: Without Observation

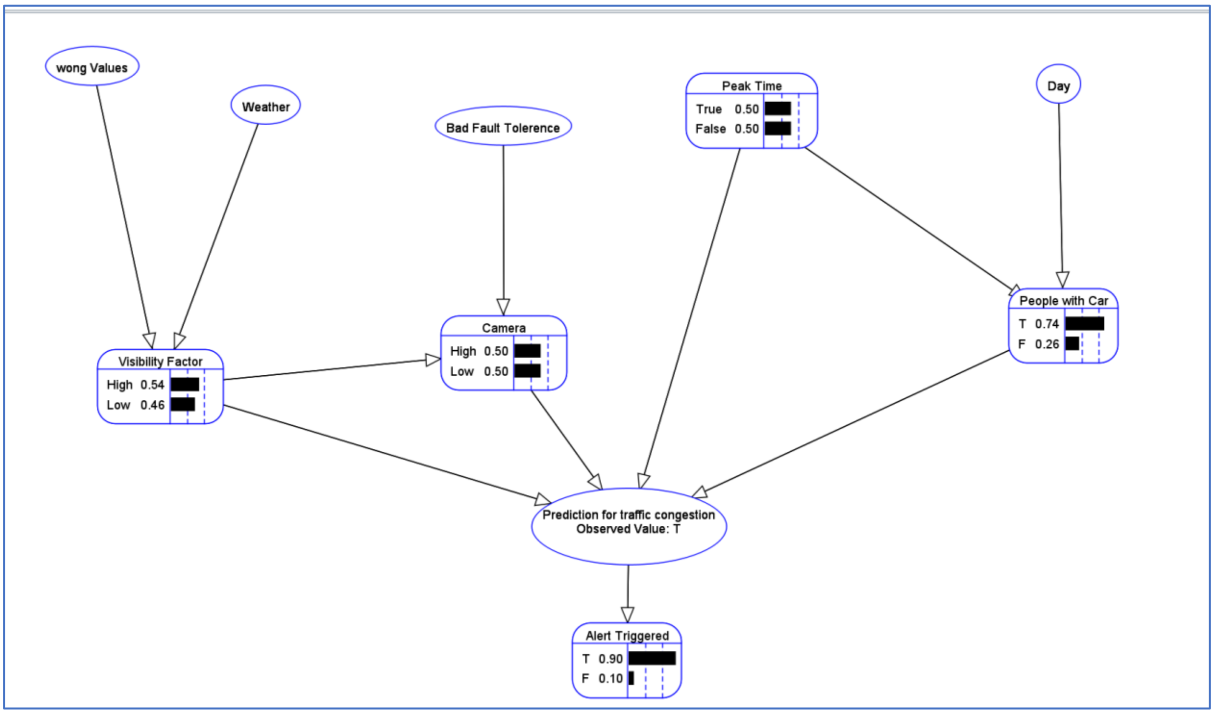


Figure 15: With observation

Part 2: Implementation of Variable Elimination Algorithm

In this part, I implemented a solution for the agent to construct a Bayesian network and make inferences using the variable elimination algorithm. Before starting development, I designed my solution with PEAS model. However, this model is not specifically targeted on agent but more on the given problem's scenario.

PEAS Model:

Problem: Construct a Bayesian Network and make interferences using variable elimination algorithm.

Agent: Software Program

Performance: This determines the level of correctness of the final probabilities of the query variable that the agent will infer through the algorithm. It will be verified through running the same query in AISPACE tool.

Environment: Entrance of Meridonia City Hospital, Cars, Weather, Traffic

Actuators: Alert

Sensors: Camera, Visibility Sensor

Implementation Strategy:

For the implementation, I used the following steps of the Variable Elimination Algorithm from lecture slides which is shown in Figure 16.

Variable elimination (VE) is a simple and general exact inference algorithm in probabilistic graphical models, such as Bayesian networks and Markov random fields [2].

```
function ELIMINATION-ASK(X, e, bn) returns a distribution over X
   inputs: X, the query variable
           e, evidence specified as an event
           bn, a belief network specifying joint distribution P(X_1, ..., X_n)
   factors ← []; vars ← REVERSE(VARS[bn])
   for each var in vars do
       factors ← [MAKE-FACTOR(var, e)|factors]
       if var is a hidden variable then factors ← SUM-OUT(var, factors)
   return NORMALIZE(POINTWISE-PRODUCT(factors))
```

Figure 16: Variable Elimination Algorithm

First, I developed the algorithm for queries without evidence, after that I added evidence value into the program as well. Steps of the algorithm are following:

1. Inputs are query node, its state (true or false), evidence and its state (true or false)

2. Add the parent and ancestors of query node to a list (until evidence node if it given, otherwise until the starting node of that network if there is no network).
3. Reverse the list and take the first element of the list,
4. Perform the *sum-out function of that node with its child node's probability.
5. Assign the child with the new probability and move to the child of that node
6. Repeat step 4 and 5 until you reach query variable node.
7. Normalize the new probability of the query variable node.

In *sum-out function, probability of the parent node at True state is first multiplied with those probability of the child nodes where parent node probability is true. Similarly, probability of the parent node at False state is then multiplied with those probability of the child nodes where parent node probability is false. True probabilities are then joined together through addition and false probabilities are also joined together to generate two new true and false probability of the child node. Parent node is eliminated at that point.

Creation of Parent Node and Child nodes:

I use Java class for creating nodes and child nodes. In the constructor of every node, I pass the reference to its parent node, a string array that contains probability distribution table, name of the node. For the first node, I passed null.

Use of Data Structures:

I used Array List to store the nodes and then use collections function to reverse the elements of the list so that path from parent node or evidence node to query variable can be followed.

Code Structure:
In the code, I have three classes. One is main class, one is the node class and one is Bayesian network class where all network data (nodes and their probabilities) are hardcoded. Also, I have defined the array list in this class as well.

Part 3: Define a problem involving reasoning with evidence and uncertainty

For this part, I would like to take medical uncertainty problem. I collected the data for factors from [4][5]. I am making use of some random people in this example with data based on UK. Anne's mother was suffering from diabetes 2. Anne's maternal grandmother suffered from diabetes2 as well. Now Anne want to find out the probability of her child getting diabetes 2.

Different factors that affect the diabetes 2 are listed below:

- an age of 45 years or older have 25% chances of getting the diabetes 2
- chances of obesity increase in men is 68% and women is 59%
- There are 15% chances of getting diabetes if one of the parents has diabetes
- Between 5% and 10% of women between 15 and 44 have polycystic ovary syndrome (PCOS)

Based on the above given factors, I created the Bayesian Network as shown below. Figure 17 shows different domain variables and relationship between them.

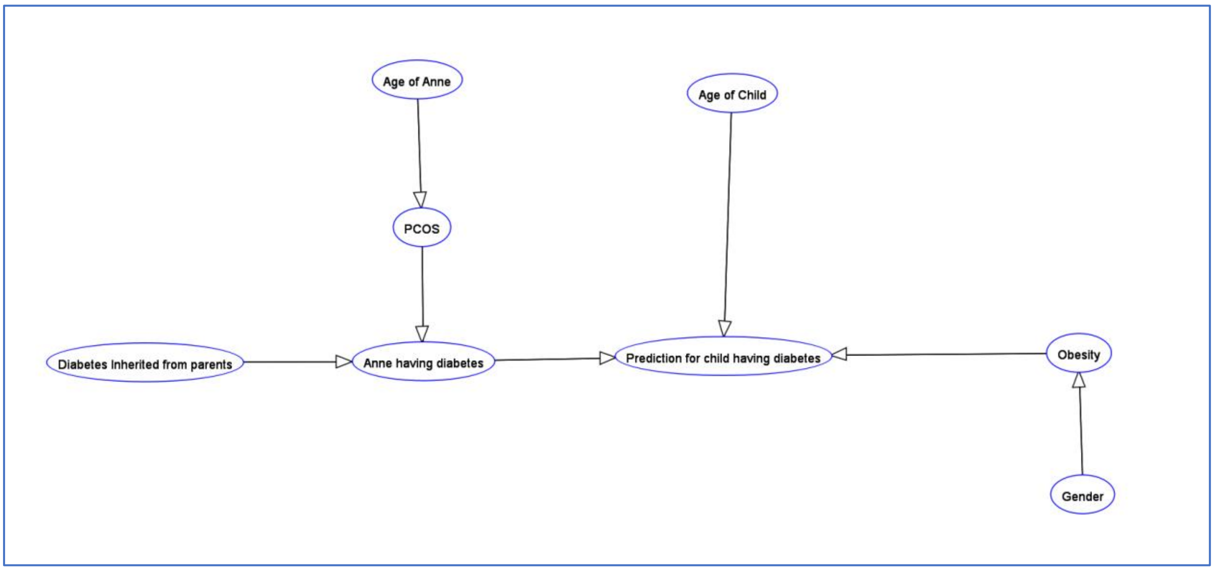


Figure 17: Bayesian Network for Diabetes Diagnosis

The Table below shows how age affects chances of having PCOS.

Probability Table for PCOS

Age of Anne	P(PCOS=T)	P(PCOS=F)
Greater than 30	0.02	0.98
less than 30	0.1	0.9

No observed value for this node.

OK

The table below shows how Diabetes inherited from parents and PCOS is increasing the chance of Anne to suffer from diabetes as it will have an affect on her child's chance of having diabetes.

Probability Table for Anne having diabetes

Diabetes Inherited from parents	PCOS	P(Anne having diabetes=T)	P(Anne having diabetes=F)
T	T	0.25	0.75
T	F	0.15	0.85
F	T	0.1	0.9
F	F	0.02	0.98

No observed value for this node.

OK

The Table below shows how gender changes the probability of having obesity.

Probability Table for Obesity

Gender	P(Obesity=T)	P(Obesity=F)
Male	0.68	0.32
Female	0.59	0.41

No observed value for this node.

OK

The table below shows how final predication about child having diabetes2 will be affected by other connected domain variables:

Probability Table for Prediction for child having diabetes

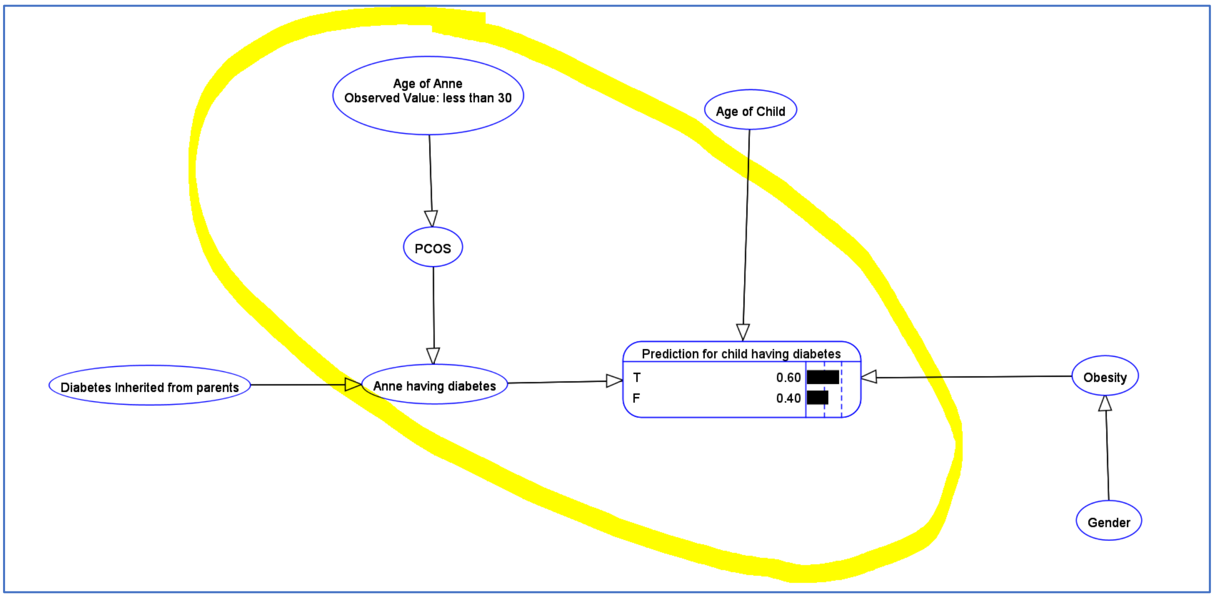
Obesity	Age of Child	Anne having diabetes	P(Prediction for child having diabetes=T)	P(Prediction for child having diabetes=F)
T	>45	T	0.95	0.05
T	>45	F	0.9	0.1
T	<45	T	0.85	0.15
T	<45	F	0.8	0.2
F	>45	T	0.4	0.6
F	>45	F	0.25	0.75
F	<45	T	0.15	0.85
F	<45	F	0.02	0.98

No observed value for this node.

OK

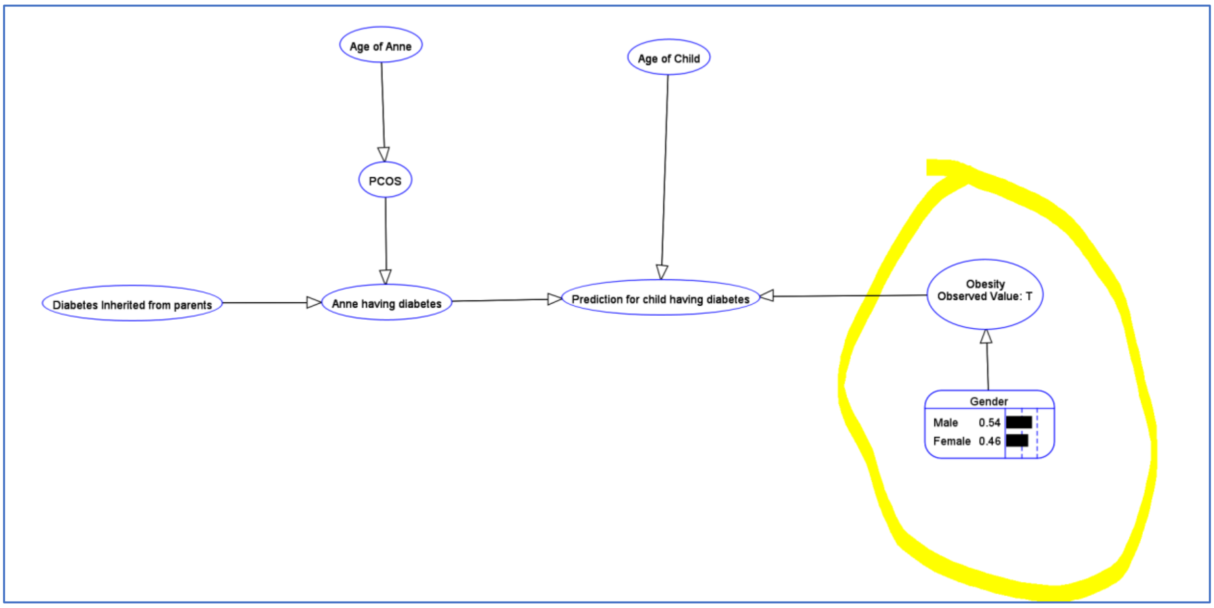
Predictive Query:

P(D|AgeA): Probability of child having diabetes given age of Anne at time of child birth is less than 30. The below image supports the above given factors. As with age less than 30, Anne has more chances of having PCOS and hence more chances of having diabetes herself which leads to child having greater chances of diabetes.



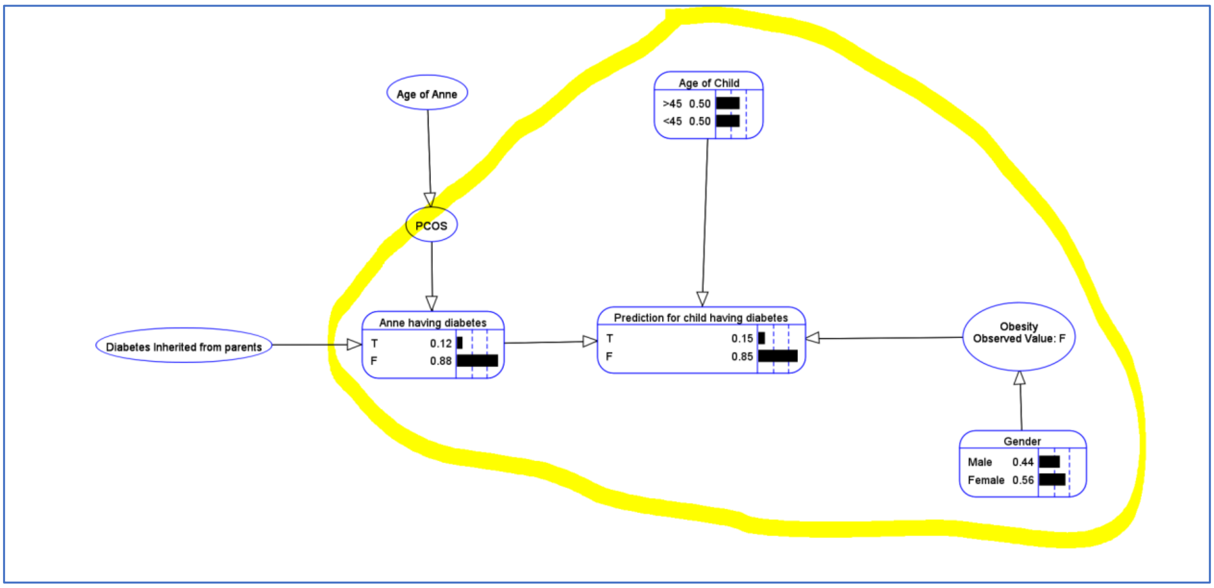
Diagnostic Query:

P(Gender|Obesity): Probability of Gender being female, given Obesity.



Profiling Query:

P(Gender, Prediction of Child having Diabetes, Age of child | ¬Obesity): Given obesity, affect on child nodes, parent node and parent of child node. The highlighted part shows Markov Blanket area.



3. Testing

For testing of the network and algorithm that I have designed in part 1 and part 2 of this assignment respectively, I tested two different predictive queries, one with evidence and one without evidence on AISPACE tool and on my program to check how similar are the results.

1. P(¬Alert|PeakTime)

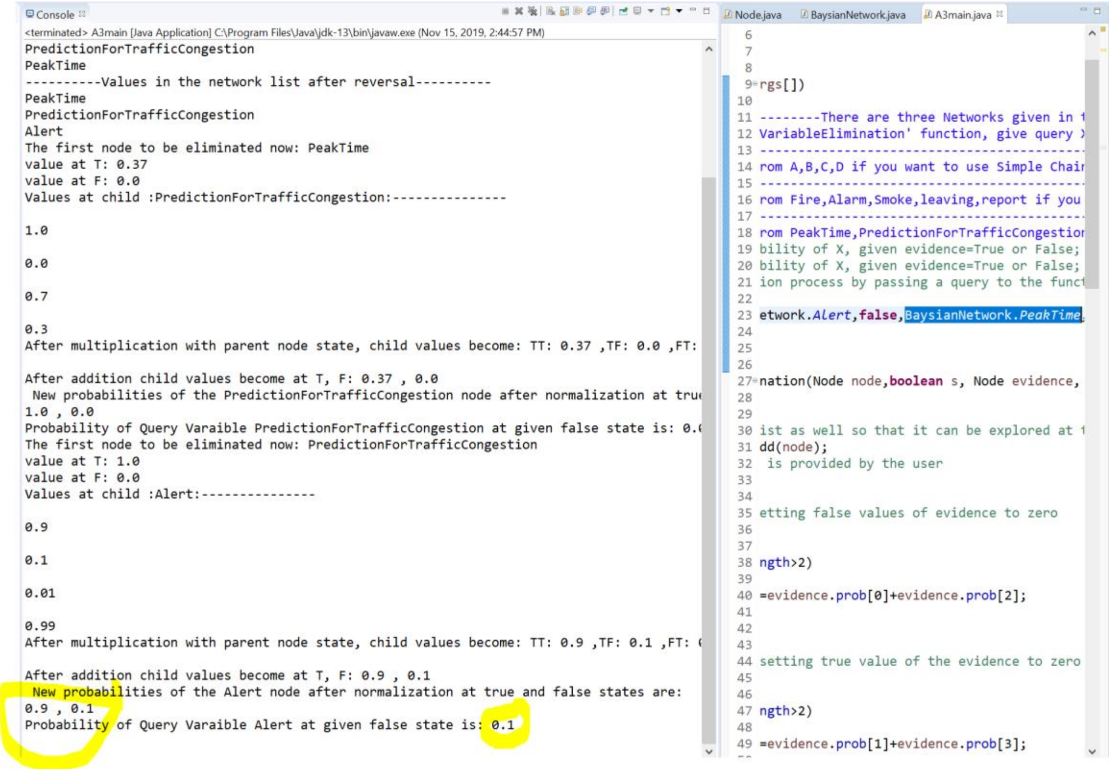


Figure 18: Program's Output

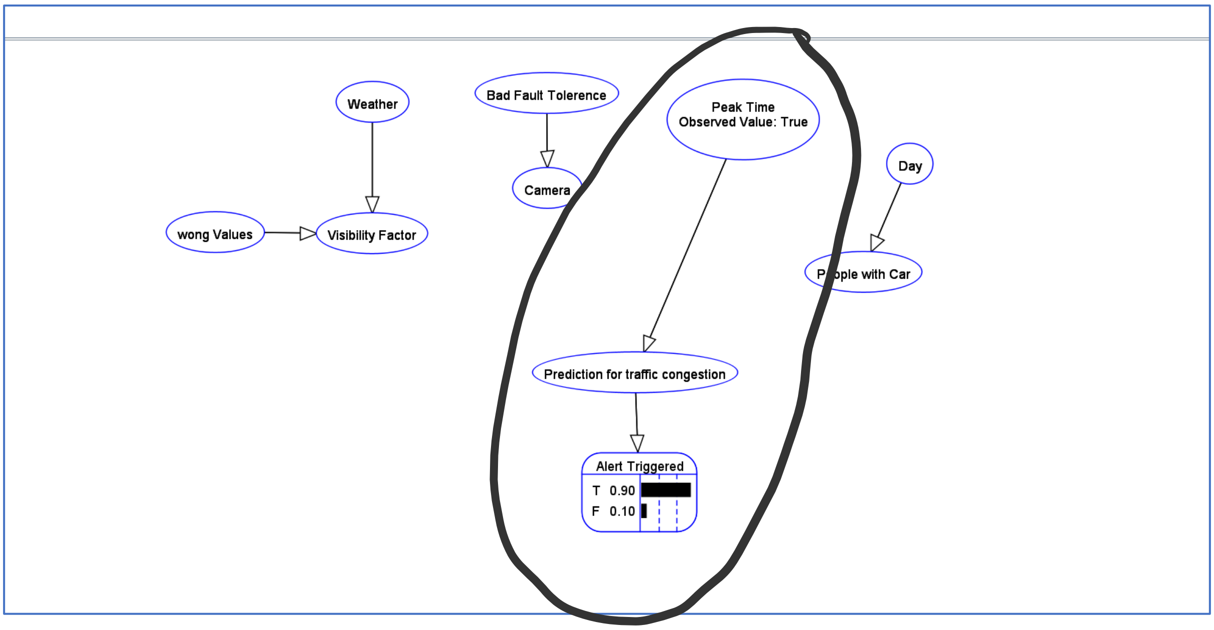


Figure 19: Output from AIspace tool

2. P(¬Alert)

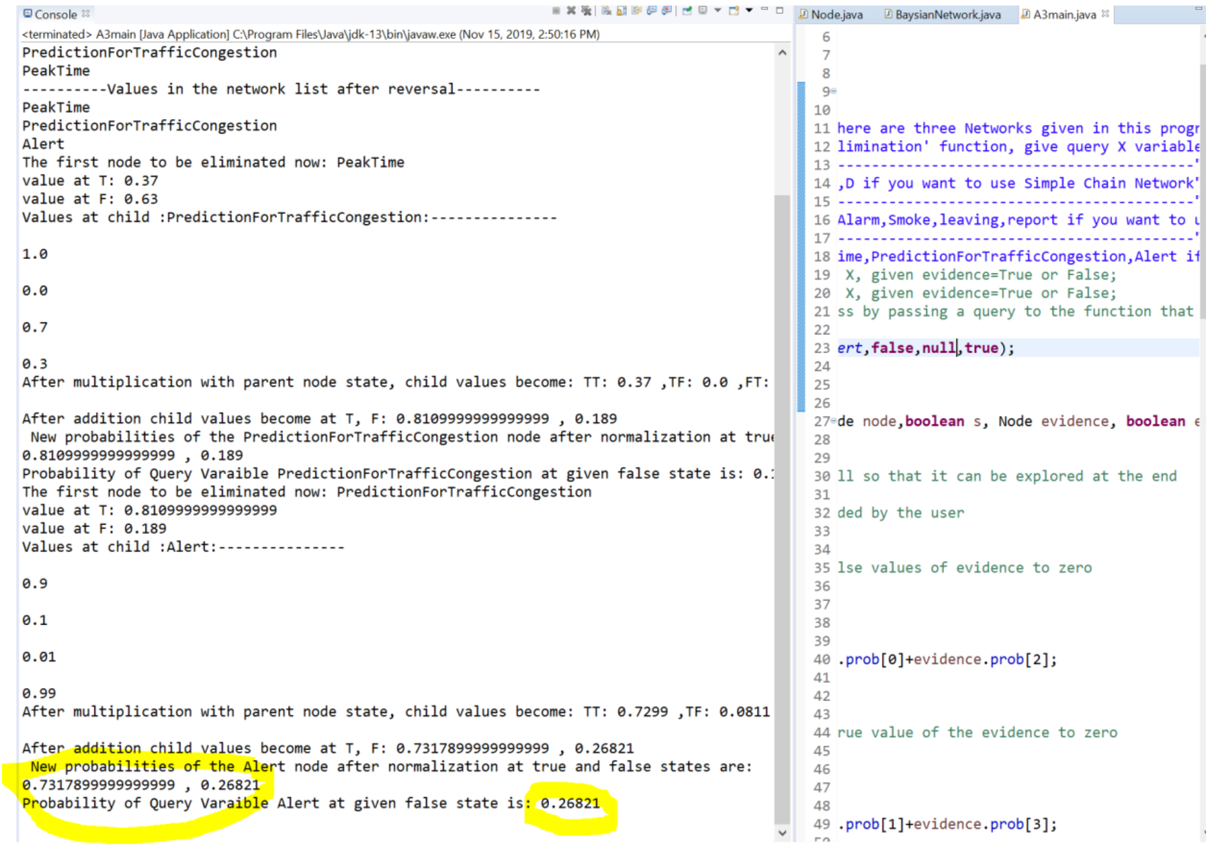


Figure 20: Output from program

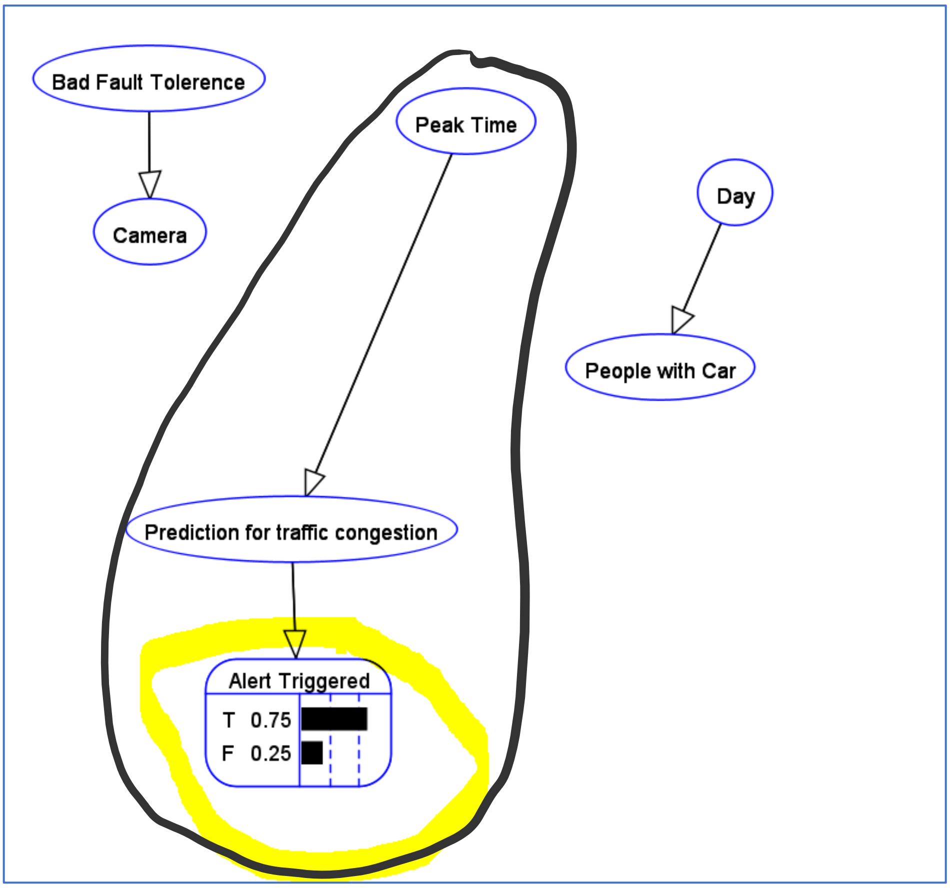


Figure 21: Output from AIspace Tool

Results above shows that the results generated by the tool are almost similar to the one generated by the program.

4. Bibliography

[1] Medium. (2018). Introduction to Bayesian Networks. [online] Available at: https://towardsdatascience.com/introduction-to-bayesian-networks-81031eeed94e [Accessed 11 Nov. 2019].

[2] Smith, A. and Gelfand, A. (1992). Bayesian Statistics without Tears: A Sampling-Resampling Perspective. The American Statistician, 46(2).

[3] Stuart Russell and Peter Norvig. (1995) Artificial Intelligence, A Modern Approach, Prentice Hall.

[4] Diabetes.org.uk. (2019). [online] Available at: https://www.diabetes.org.uk/resources-s3/2019-02/1362B_Facts%20and%20stats%20Update%20Jan%202019_LOW%20RES_EXTERNAL.pdf [Accessed 14 Nov. 2019].

[5] Diabetes. (2019). The UK is the fattest country in Europe. The number of obese adults is forecast to rise by 73% over the next 20 years from to 26 million people, resulting in more than a million extra cases of type 2 diabetes, heart disease and cancer [online] Available at: https://www.diabetes.co.uk/diabetes-and-obesity.html [Accessed 15 Nov. 2019].

[6] Royalsocietypublishing.org. (2018). The Markov blankets of life: autonomy, active inference and the free energy principle | Journal of The Royal Society Interface. [online] Available at: https://royalsocietypublishing.org/doi/full/10.1098/rsif.2017.0792 [Accessed 15 Nov. 2019].