

Inference with Bayesian Network and Causality

Problem statement 1:

Given a dataset. Identify the anomalies, find the feature causing the anomalous behavior and predict the correct values for that feature.

Problem statement 2:

There is an existing alarm tree. Simulate the same thing in a Bayesian network and find the root cause of a problem given certain evidence in particular nodes.

Problem statement 3:

Create a Causal Bayesian network and make interventional and counterfactual queries to infer important insights.

Why Bayesian network models?

First and foremost, we needed a concrete class of models on which to demonstrate and apply our ideas. Second, we wanted to use probability theory as our foundation. Our use of probability theory comes from necessity: most AI application domains involve uncertainty, with which we need to deal with explicitly and from the start in a principled way. Finally, we are interested in deriving cause-effect relationships from data. Even though many classes of models can be used to represent uncertain domains—like decision trees, artificial neural networks etc. only in the Bayesian network literature do we find claims of being able to represent and learn directed causal relationships.

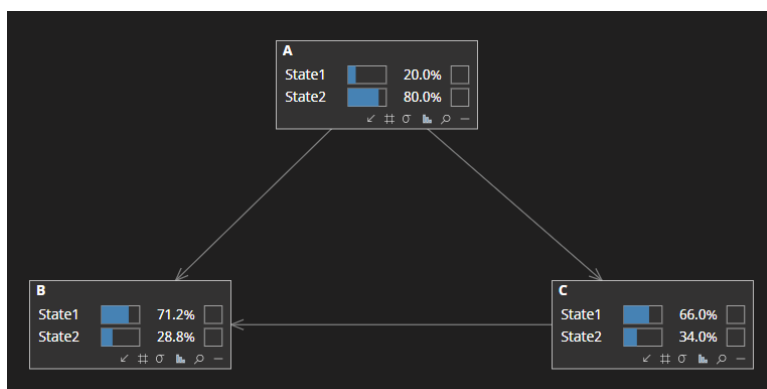
They can be used for a wide range of tasks including reasoning, anomaly detection, diagnostics, causal prediction.

Preliminaries

Bayesian network:

A Bayesian network is a graphical representation of a probability distribution over a set of variables. It consists of two parts: the directed acyclic graph and a set of probability distributions one for each node conditioned on the parent.

Example of a discrete network



A node can represent one variable or many variables and can be discrete, continuous and functional.

Here A, B and C are the discrete variables having two states each and the arrows represent direct dependencies. The edges may or may not be causal.

The conditional probabilities for each of the variable are given

P(A)

A = State1 ▾	A = State2 ▾
0.2	0.8

P(B|A,C)

A ▾	C ▾	B = State1 ▾	B = State2 ▾
State1	State1	0.9	0.1
State1	State2	0.3	0.7
State2	State1	0.8	0.2
State2	State2	0.5	0.5

P(C|A)

A ▾	C = State1 ▾	C = State2 ▾
State1	0.9	0.1
State2	0.6	0.4

The Bayesian network represent the joint probability distribution of the domain

$$\Pr(X_1, X_2, \dots, X_n) = \prod_{i=1}^n \Pr(X_i | \mathbf{Pa}_i).$$

Where Pa is the set containing the parents of X in the Bayesian network.

The probability distribution for A is the prior belief when nothing else is known.

Distribution for C is dependent on A and is calculated using this

$$P(C=State1) = P(C=State1 | A=State1) P(A=State1) + P(C=State1 | A=State2) P(A=State2)$$

Similarly, distribution on B is dependent on A and C both and is calculated using this

$$\begin{aligned} P(B=State1) = & P(B=State1 | A=State1, C=State1) P(A=State1) P(C=State1) + \\ & P(B=State1 | A=State1, C=State2) P(A=State1) P(C=State2) + \\ & P(B=State1 | A=State2, C=State1) P(A=State2) P(C=State1) + \\ & P(B=State1 | A=State2, C=State2) P(A=State2) P(C=State2) + \end{aligned}$$

Linear Gaussian BNs are based on continuous variables which are assumed to follow Gaussian distributions. Hybrid BNs support both discrete and continuous distributions. It does not generally allow continuous variables

to be parents of discrete ones. Sometimes it is useful to discretize continuous data, generating a discrete variable, where each state represents a continuous interval.

When a link does not exist between two nodes, this does not mean that they are completely independent, as they may be connected via other nodes. They may however become dependent or independent depending on the evidence that is set on other nodes. Evidence is the information we know about a variable. If we are 100% certain about the information it is called the hard evidence otherwise it is called the soft evidence.

During inferences or predictions in the Bayesian network it uses the Bayes theorem which states that

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

Where:

- $P(A|B)$ is the conditional probability of event A occurring given that event B has occurred.
- $P(B|A)$ is the conditional probability of event B occurring given that event A has occurred.
- $P(A)$ is the prior probability of event A occurring.
- $P(B)$ is the prior probability of event B occurring.

The links and the conditional probabilities tables can be set manually using expert opinion. Otherwise, there are Structural learning algorithms for Bayesian networks, which can automatically determine the required links from data, this fall into two main classes. The first class is constraint-based methods that eliminate and orientate edges based on a series of conditional independence tests. The second class, score-based methods, represent a traditional machine learning approach where the aim is to search over different graphs maximising an objective function. The graph that maximises the objective function is returned as the preferred graph.

Once the structure has been defined (i.e. nodes and links), a Bayesian network requires a probability distribution to be assigned to each node. Parameter learning is the process of using data to learn the distributions of a Bayesian network and it uses the maximum likelihood estimation procedure.

Problem statement 1

Given the dataset. Identify the anomalies, find the feature causing the anomalous behavior and predict the correct values for that feature.

The dataset was generated by a power plant and had continuous data though out. For the example purpose here, I take small sample from there with few features. It consists of 1450 observations without any missing value.

Anomaly detection:

Anomaly detection algorithms can be used in different health monitoring systems as well as in the preprocessing step to remove any anomalous point. Anomalous points are outliers or rare. Keeping them in our model make our inferences wrong. An observation becomes anomalous when one or more sensors give incorrect result. Here our goal is to predict the accurate value instead.

I work here with a small train and test data and for verification purpose I introduce anomalies manually and do the same work I have done in the project.

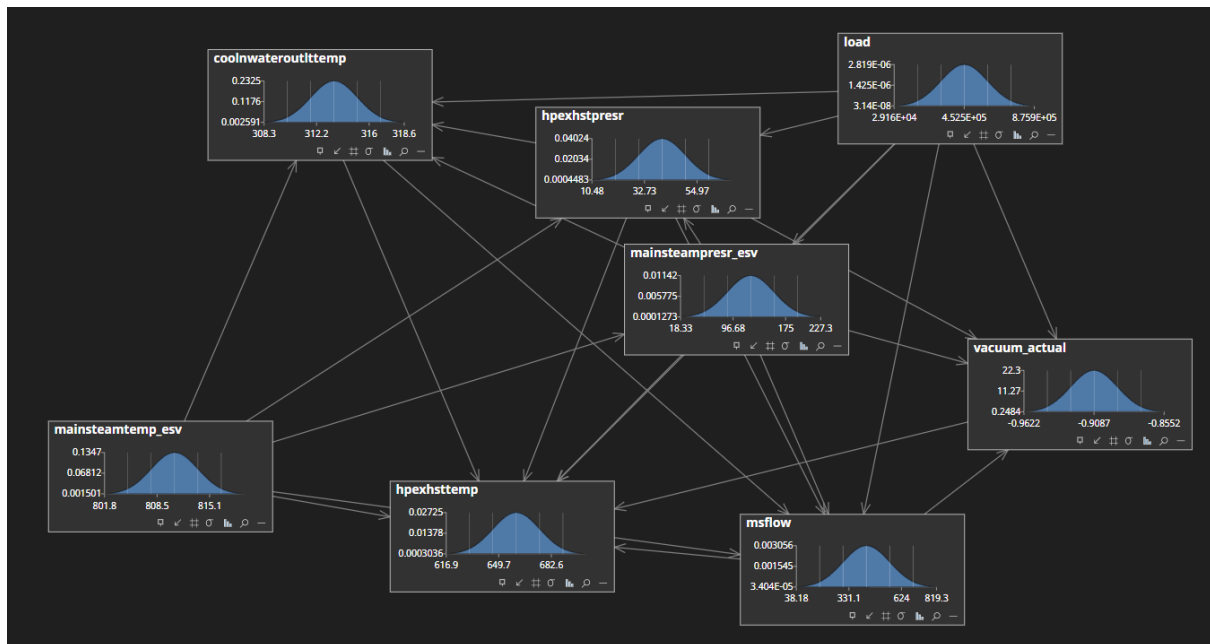
Bayesian networks are well suited for anomaly detection, because they can handle high dimensional data which humans find difficulties to interpret. While some anomalies are clearly visible by plotting individual variables, often anomalies are far more subtle, and are based on the interaction of many variables.

The dataset:

id	msflow	load	hpexhstpres	hpexhsttemp	mainsteampres_esv	mainsteamtemp_esv	vacuum_actual	coolnwaterflow	coolnwaterouttlpres	coolnwaterouttltemp	coolnwaterinletpres
1	336.2827	348111.6	33.048695	666.456	97.394905	803.13696	-0.921247	20833	0.98675	312.25037	1.98675
2	339.233	352044.8	33.291237	666.65515	98.29624	804.36176	-0.920859	20833	0.98675	312.2989	1.98675
3	339.2195	351610	33.27886	665.1304	98.15573	802.4936	-0.92085	20833	0.98675	312.29105	1.98675
4	339.7499	352272.2	33.320694	664.99396	98.306244	802.50854	-0.920787	20833	0.98675	312.30255	1.98675
5	340.679	354330	33.415264	668.0803	98.84829	806.57056	-0.920674	20833	0.98675	312.31763	1.98675
6	338.9581	352118.3	33.27828	668.26807	98.34775	806.2306	-0.920906	20833	0.98675	312.29443	1.98675
7	337.3937	349943.2	33.149197	667.8274	97.84337	805.1789	-0.921094	20833	0.98675	312.27	1.98675
8	339.1733	351780.9	33.282898	665.98517	98.2224	803.51855	-0.920892	20833	0.98675	312.2911	1.98675
9	340.6779	353348.9	33.395645	664.48413	98.54547	802.1827	-0.920732	20833	0.98675	312.31177	1.98675
10	341.1914	354470.2	33.441864	666.14044	98.837204	804.366	-0.920637	20833	0.98675	312.3213	1.98675
11	341.61	355058.2	33.47675	666.28	98.97675	804.68	-0.92055	20833	0.98675	312.33	1.98675
12	342.0674	355073.5	33.50658	664.1122	98.92703	802.1841	-0.92055	20833	0.98675	312.33	1.98675
13	343.1359	357503.8	33.614323	667.83563	99.56239	807.06934	-0.920354	20833	0.98675	312.34955	1.98675
14	343.3174	357146.3	33.6167	665.67316	99.43292	804.4951	-0.920361	20833	0.98675	312.34888	1.98675
15	340.05	353075.7	33.35675	666.51	98.52675	804.45	-0.92075	20833	0.98675	312.31	1.98675
16	339.8055	352986.1	33.338413	667.3536	98.52675	805.39746	-0.92075	20833	0.98675	312.3039	1.98675
17	343.3588	357732.8	33.62747	667.62573	99.61115	806.8839	-0.920294	20833	0.98675	312.35468	1.98675
18	344.5944	359304	33.724537	667.417	99.96756	807.0378	-0.920182	20833	0.98675	312.37357	1.98675
19	344.5034	358645.3	33.70925	665.4565	99.775375	804.61597	-0.920203	20833	0.98675	312.36945	1.98675
20	342.9993	356846.6	33.592186	666.13214	99.381516	804.94836	-0.920386	20833	0.98675	312.34644	1.98675

The model:

Variables having same values throughout is ignored since they do not depict uncertainty in its occurring. Search and Score algorithm was used in Structural learning since it is sufficient to only consider correlation while doing the task. While doing parameter learning Relevance tree algorithm was used. Hence, we get the model like this where the arrows have no causal meaning.



We see how the variables are dependent on each other. We can condition on one variable and study the updated probability distributions on rest of the variables to get valuable insights. We used the normal procedure to build the model just because we know that the data with which we are creating the model is normal and does not contain any anomalous points. Else, we had to use some complex model like a Mixture model using Clustering algorithm to first find the anomalies using in-sample anomaly detection technique and then eliminate the anomalous points from the train data. After that we can build this model from normal points in the train dataset.

Since we don't have any anomalous point in the train data, we create this model using basic structural and parameter learning algorithms.

Loglikelihood:

If the result of learning is a model that does not contain information about the anomalous data, we have a model which represents normal behaviour. We can use this model to see how likely it is that unseen data could have been generated by this model. This tells us how anomalous the unseen data is. The lower the value, more anomalous the point is. The log-likelihood is simply the log of the probability density function (pdf) for the Bayesian network where the evidence from the test data is set.

So, if this is our test data

id	msflow	load	hpexhstpresr	hpexhsttemp	mainsteamtemp_esv	mainsteamtemp_esv	vacuum_actual	coolnwaterflow	coolnwateroutltemp	coolnwateroutltemp	coolnwaterinletpresr
1	341.4391	354478.56	33.465096	624.9394	85.80901	890.99005	-0.920567	20833	0.98675	312.32443	1.98675
2	340.05	353075.72	33.35675	666.51	98.52675	804.45	-0.92075	20833	0.98675	312.31	1.98675
3	321.6425	353950.88	13.495197	669.38745	99.24731	808.47003	-0.920529	20833	0.98675	312.33212	1.98675
4	500.5283	582638.4	48.263454	639.39124	101.59135	811.0224	-0.895572	20833	0.98675	317.8622	1.98675
5	665.8549	702004.06	48.087704	639.68164	185.91634	820.93677	-0.875732	20833	0.98675	316.59384	1.98675
6	341.77	357024.44	33.52675	672.71	99.56675	812.56	-0.92045	20833	0.98675	312.34	1.98675
7	666.5712	702926.75	58.143204	639.8918	186.12991	811.1326	-0.875581	20833	0.98675	316.6073	1.98675
8	340.87	355360.72	33.44675	670.97	99.14675	810.15	-0.92065	20833	0.98675	312.32	1.98675
9	366.4749	399950.56	36.27123	663.67487	109.17682	845.21924	-0.916128	20833	0.98675	300.81223	1.98675
10	614.366	663823.75	54.322308	641.3253	173.48164	816.1234	-0.883759	20833	0.98675	315.9191	1.98675

And we want to check which observations are anomalous we check the loglikelihood of each of the observations.

LogLikelihood
-504694.6161372801
0.08520753264552461
-26605695.2102583
-6307485.327566361
-7915399.930632953
-0.09320415678760163
-0.21411535525492909
0.8527366577162407
-12703037.747496322
-11.380699947552552

Looking at this we can conclude that points 1,3,4,5,9 are anomalies.

Retracted Loglikelihood:

When the anomaly score detects anomalous behaviour, we are usually then interested in diagnosing the cause of that anomaly. We can do this by testing the anomaly score without evidence set on a subset of particular variables in the Bayesian network. Below is the retracted loglikelihood for each of the feature for all of the observations in the test data.

Msflow

LogLikelihood	RetractedLogLikelihood
-504694.6161372801	-172098.10734522005
0.08520753264552461	-1.7260951950481758
-26605695.2102583	-2937058.2093678084
-6307485.327566361	-1585023.7349228095
-7915399.930632953	-749251.6423552231
-0.09320415678760163	-0.5475646165553112
-0.21411535525492909	-1.680187066543768
0.8527366577162407	-0.7846906675691869
-12703037.747496322	-12135764.685024686
-11.380699947552552	-12.728368643425503

Load

LogLikelihood	RetractedLogLikelihood
-504694.6161372801	-308443.8843631649
0.08520753264552461	6.615379560063854
-26605695.2102583	-9160399.51438414
-6307485.327566361	-623330.7768212381
-7915399.930632953	-2574696.520087518
-0.09320415678760163	8.683436100533427
-0.21411535525492909	6.432078945207918
0.8527366577162407	6.9205845187815225
-12703037.747496322	-10929997.845021082
-11.380699947552552	-4.80340161375327

hpexhstpresr

LogLikelihood	RetractedLogLikelihood
-504694.6161372801	-172134.50001321113
0.08520753264552461	-4.789515902264551
-26605695.2102583	-11549.618087301465
-6307485.327566361	-2456413.855020457
-7915399.930632953	-761.644940772843
-0.09320415678760163	-4.04696258909803
-0.21411535525492909	-4.7194366569623405
0.8527366577162407	-2.7705304777861137
-12703037.747496322	-12693726.746781483
-11.380699947552552	-16.20858654346445

hpexhsttemp

LogLikelihood	RetractedLogLikelihood
-504694.6161372801	-84234.30120747369
0.08520753264552461	-1.2514226088279505
-26605695.2102583	-2952706.9404331413
-6307485.327566361	-1674640.1657032727
-7915399.930632953	-791310.1054296407
-0.09320415678760163	-0.49609233916092066
-0.21411535525492909	-1.2393827778402633
0.8527366577162407	0.2213800297882762
-12703037.747496322	-11524495.9480487
-11.380699947552552	-12.488151197017707

mainsteampresr_esv

LogLikelihood	RetractedLogLikelihood
-504694.6161372801	-360234.8459857629
0.08520753264552461	-2.577833158979267
-26605695.2102583	-7119870.433729063
-6307485.327566361	-806326.108269522
-7915399.930632953	-2075844.6584989908
-0.09320415678760163	-0.6180110601798265
-0.21411535525492909	-2.536285959681212
0.8527366577162407	-1.6619731624731813
-12703037.747496322	-11923585.882302158
-11.380699947552552	-14.21992792245318

mainsteamtemp_esv

LogLikelihood	RetractedLogLikelihood
-504694.6161372801	-54912.28274524005
0.08520753264552461	-0.9477527497040099
-26605695.2102583	-3724261.505941121
-6307485.327566361	-1985946.3881736181
-7915399.930632953	-996773.5976020304
-0.09320415678760163	-0.4291439671414974
-0.21411535525492909	-1.0477803526939304
0.8527366577162407	0.35990954144776577
-12703037.747496322	-11498407.846861118
-11.380699947552552	-11.875236119197396

vacuum actual

LogLikelihood	RetractedLogLikelihood
-504694.6161372801	-461339.10462690325
0.08520753264552461	-9.243867516049097
-26605695.2102583	-26557759.172380872
-6307485.327566361	-5841725.732123311
-7915399.930632953	-7893462.365310222
-0.09320415678760163	-8.239634035352015
-0.21411535525492909	-9.721948724536617
0.8527366577162407	-8.389372365456842
-12703037.747496322	-12687468.425602866
-11.380699947552552	-20.596996781962268

coolnwateroutlttemp

LogLikelihood	RetractedLogLikelihood
-504694.6161372801	-469045.6542227228
0.08520753264552461	-4.6083207798529
-26605695.2102583	-26605652.84499698
-6307485.327566361	-4380858.845005432
-7915399.930632953	-7912638.773564874
-0.09320415678760163	-4.403626054107655
-0.21411535525492909	-4.92079959376288
0.8527366577162407	-3.58971449161438
-12703037.747496322	-8321.622798125623
-11.380699947552552	-16.470440441254325

Checking each observation at a time we conclude for each anomalous observation the variable giving the largest negative retracted loglikelihood i.e., the variable giving the most impact for the anomalous behavior of the observation is

Point 1: mainsteamtemp_esv

Point 3: hpexhstpresr

Point 4: load

Point 5: hpexhstpresr

Point 9: coolnwateroutlttemp

Hence these are the most contributing factors for their anomalous behavior. If the retracted log likelihood is around 0 then no other factors are there which contributes to their anomalous behavior otherwise there are more factors.

For point 1, removing the effect of mainsteamtemp_esv we get mainsteampresr_esv having largest negative retracted loglikelihood

Msflow		Load		hpexhstpresr	
LogLikelihood	RetractedLogLikelihood	LogLikelihood	RetractedLogLikelihood	LogLikelihood	RetractedLogLikelihood
-54912.28274524005	-28073.794755363964	-54912.28274524005	-13360.147624743544	-54912.28274524005	-37545.96863247336
hpexhsttemp		mainsteampresr_esv		vacuum_actual	
LogLikelihood	RetractedLogLikelihood	LogLikelihood	RetractedLogLikelihood	LogLikelihood	RetractedLogLikelihood
-54912.28274524005	-31050.10499371938	-54912.28274524005	-846.5893118333897	-54912.28274524005	-27268.60732161422
coolnwateroutlttemp					
LogLikelihood	RetractedLogLikelihood				
-54912.28274524005	-54885.847928660834				

Removing the effect of mainsteamtemp_esv, mainsteampresr_esv we get hpexhsttemp having largest negative retracted loglikelihood

<u>Msflow</u>		<u>Load</u>		<u>hpexhstpresr</u>	
LogLikelihood	RetractedLogLikelihood	LogLikelihood	RetractedLogLikelihood	LogLikelihood	RetractedLogLikelihood
-846.5893118335196	-341.7099115459589	-846.5893118335196	-354.99686674896213	-846.5893118335196	-844.920203531654

<u>hpexhsttemp</u>		<u>vacuum_actual</u>		<u>coolnwateroutlttemp</u>	
LogLikelihood	RetractedLogLikelihood	LogLikelihood	RetractedLogLikelihood	LogLikelihood	RetractedLogLikelihood
-846.5893118335196	-1.5383301621770258	-846.5893118335196	-162.8884601002288	-846.5893118335196	-851.5588447164934

For Point 3, removing the effect of hpexhstpresr we get msflow having the largest negative retracted loglikelihood,

<u>Msflowc</u>		<u>Load</u>		<u>hpexhsttemp</u>	
-11549.618087301462	-13.41811450208262	-11549.618087301462	-4331.393668378624	-11549.618087301462	-9810.070496633272

<u>mainsteampresr_esv</u>		<u>mainsteamtemp_esv</u>		<u>vacuum_actual</u>	
-11549.618087301462	-10499.802019462928	-11549.618087301462	-8602.04529315957	-11549.618087301462	-7166.063847610243

<u>coolnwateroutlttemp</u>	
-11549.618087301462	-7480.1825123892495

For Point 4, removing the effect of load we get coolnwateroutlttemp having the largest negative retracted loglikelihood,

<u>Msflow</u>		<u>hpexhstpresr</u>		<u>hpexhsttemp</u>	
-623330.7768216642	-618104.46328083	-623330.7768216642	-623092.8988006379	-623330.7768216642	-556851.6232694561

<u>mainsteampresr_esv</u>		<u>mainsteamtemp_esv</u>		<u>vacuum_actual</u>	
-623330.7768216642	-516502.0818107714	-623330.7768216642	-591243.2176998735	-623330.7768216642	-431613.7960478221

<u>coolnwateroutlttemp</u>	
-623330.7768216642	-318706.9810770656

Removing the effect load, coolnwateroutlttemp we get mainsteampresr_esv having the largest negative retracted loglikelihood

<u>Msflowc</u>		<u>hpexhstpresr</u>		<u>hpexhsttemp</u>	
-318706.9810770656	-294537.07127780566	-318706.9810770656	-177833.9090359379	-318706.9810770656	-237920.92033275677

<u>mainsteampresr_esv</u>		<u>mainsteamtemp_esv</u>		<u>vacuum_actual</u>	
-318706.9810770656	-42467.20942583304	-318706.9810770656	-280648.1183217086	-318706.9810770656	-213717.50534892295

Removing the effect load, coolnwateroutlttemp, mainsteampresr_esv we get msflow having the largest negative retracted loglikelihood

<u>Msflowc</u>		<u>hpexhstpresr</u>		<u>hpexhsttemp</u>	
-42467.20942583304	-2.092690621315082	-42467.20942583304	-2898.3776117146685	-42467.20942583304	-10996.90873008508

<u>mainsteamtemp_esv</u>		<u>vacuum_actual</u>	
-42467.20942583304	-9172.561274370872	-42467.20942583304	-20205.46828458001

For Point 5, removing the effect of Hpexhstpresr we get mainsteamtemp_esv having the largest negative retracted loglikelihood

<u>Msflowc</u>	<u>load</u>	<u>hpexhsttemp</u>
-761.6449407728428	-545.9950055862065	-761.6449407728428
-761.6449407728428	-650.3151300199985	-73.28820257208605
<u>mainsteampresr_esv</u>	<u>mainsteamtemp_esv</u>	<u>vacuum_actual</u>
-761.6449407728428	-762.1301968761186	-761.6449407728428
-761.6449407728428	-6.397812152014437	-716.2454469831554
<u>coolnwateroutlttemp</u>		
-761.6449407728428	-303.0004991268383	

For Point 9, removing the effect of coolnwateroutlttemp we get mainstreamtemp_esv

<u>Msflowc</u>	<u>load</u>	<u>Hpexhstpresr</u>
-8321.622798125623	-6306.453844346908	-8321.622798125623
-8321.622798125623	-7619.166516220365	-5056.609978732899
<u>hpexhsttemp</u>	<u>mainsteampresr_esv</u>	<u>mainsteamtemp_esv</u>
-8321.622798125623	-3721.07016177739	-8321.622798125623
-8321.622798125623	-7545.218003130017	-3587.07812700878
<u>vacuum_actual</u>		
-8321.622798125623	-4725.024752192754	

Removing the effect of coolnwateroutlttemp ,mainsteamtemp_esv we get msflow

<u>Msflowc</u>	<u>load</u>	<u>Hpexhstpresr</u>
-3587.07812700878	-5.305559745336077	-3587.07812700878
-3587.07812700878	-1006.3595710000061	-2054.973932638705
<u>hpexhsttemp</u>	<u>mainsteampresr_esv</u>	<u>vacuum_actual</u>
-3587.07812700878	-3570.4752311489056	-3587.07812700878
-3587.07812700878	-2081.799481920097	-460.8119758929072

Therefore, the traced factors for all the observations in the test dataset are

id	msflow	load	hpexhstpresr	hpexhsttemp	mainsteampresr_esv	mainsteamtemp_esv	vacuum_actual	coolnwaterflow	coolnwateroutlttemp	coolnwateroutlttemp	coolnwaterinletpresr
1	341.4391	354478.56	33.465096	624.9394	85.80901	890.99005	-0.920567	20833	0.98675	312.32443	1.98675
2	340.05	353075.72	33.35675	666.51	98.52675	804.45	-0.92075	20833	0.98675	312.31	1.98675
3	321.6425	353950.88	13.495197	669.38745	99.24731	808.47003	-0.920529	20833	0.98675	312.33212	1.98675
4	500.5283	582638.4	48.263454	639.39124	101.59135	811.0224	-0.895572	20833	0.98675	317.8622	1.98675
5	665.8549	702004.06	48.087704	639.68164	185.91634	820.93677	-0.875732	20833	0.98675	316.59384	1.98675
6	341.77	357024.44	33.52675	672.71	99.56675	812.56	-0.92045	20833	0.98675	312.34	1.98675
7	666.5712	702926.75	58.143204	639.8918	186.12991	811.1326	-0.875581	20833	0.98675	316.6073	1.98675
8	340.87	355360.72	33.44675	670.97	99.14675	810.15	-0.92065	20833	0.98675	312.32	1.98675
9	366.4749	399950.56	36.27123	663.67487	109.17682	845.21924	-0.916128	20833	0.98675	300.81123	1.98675
10	614.366	663823.75	54.322308	641.3253	173.48164	816.1234	-0.883759	20833	0.98675	315.9191	1.98675

Thus, we are successful in diagnosing the root cause of the problem as well. Our next aim is to predict the correct values for these places.

Prediction:

We use batch query method because it allows multiple cases to be queried at once. Prediction is the process of calculating a probability distribution over one or more variables whose values we would like to know, given information (evidence) we have about some other variables. The variables we are predicting are known as Output variables, while the variables whose information we are using to make the predictions are known as Input variables. In a Bayesian network any variable can be treated as an output, also any variable can be treated as an input.

Hence after prediction for each of the observations we get

For point 1

id	Predict(hpexhsttemp)	Predict(mainsteampresr_...	Predict(mainsteamtemp_...	hpexhsttemp	mainsteampresr_esv	mainsteamtemp_esv
1	667.8161967582266	98.94296626331905	806.1192254251753	624.9394	85.80901	890.99005

For point 3

id	Predict(hpexhstpresr)	Predict(msflow)	msflow	hpexhstpresr
3	33.47953777535635	340.8946315028727	321.6425	13.495197

For point 4

id	Predict(coolnwat...	Predict(load)	Predict(mainstea...	Predict(msflow)	msflow	load	mainsteampresr_...	coolwateroutlt...
4	314.8679304222034	581532.0376460281	151.57038092819312	533.2897847683033	500.52826	582638.4	101.59135	317.8622

For point 5

id	Predict(hpexhstpresr)	Predict(mainsteamtemp_esv)	hpexhstpresr	mainsteamtemp_esv
5	58.084426007527654	810.7063923294694	48.087704	820.93677

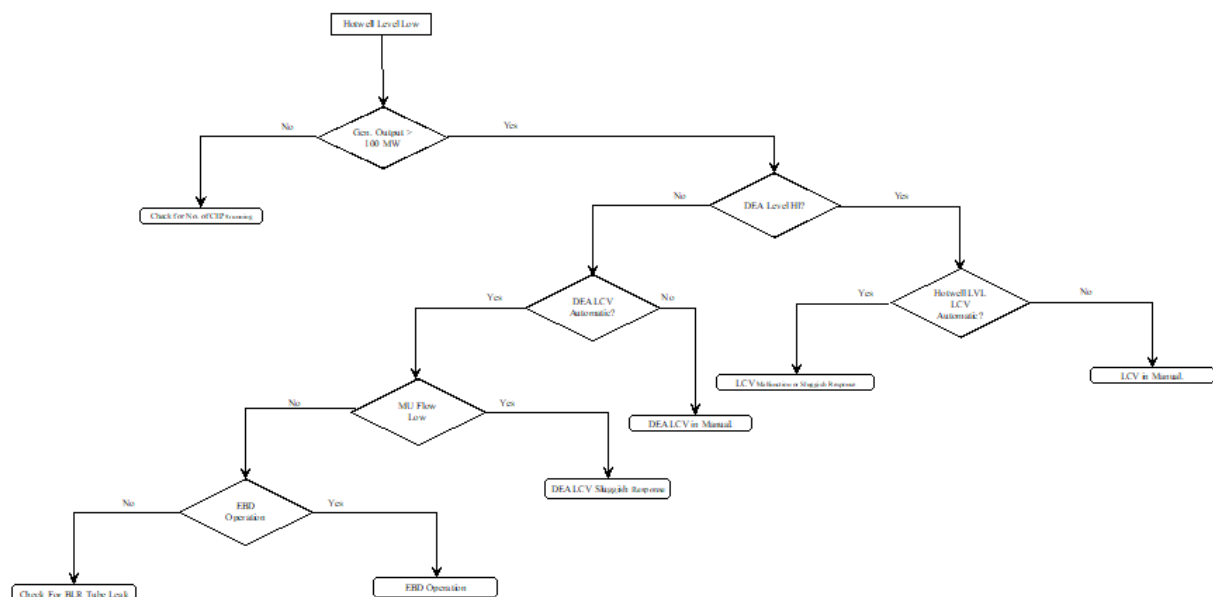
For point 9

id	Predict(coolnwatoutlt...	Predict(mainsteamtemp_...	Predict(msflow)	msflow	mainsteamtemp_esv	coolnwatoutlttemp
9	312.8097313310396	811.5329473376788	376.7347472424151	366.4749	845.21924	300.81223

When a Bayesian network has been built from data, it is common practice to evaluate the performance. Since the variables are continuous, I used the metric R squared to understand how well the model is performing. R-squared, also known as the Coefficient of determination is a standard metric which tells us how well the inputs explain the variance of the output. Its value is between 0 and 1(the closer to 1 is better). For our final model we got a good R-squared value.

There is an existing alarm tree. Simulate the same thing in a Bayesian network and find the root cause of a problem given certain evidence in particular nodes.

The tree has conditions and the final outcome or the leaf node. These leaf nodes describe what thing we have to check when we get faulty results as inputs. For example, as the leaf nodes we have 'CEP ineffective OR suction trainer choked', 'CEP A motor overload protection' etc.

[illegible]

The diagram is a fault tree for a nuclear reactor system, showing the relationships between various components and their failure probabilities. The components are represented as boxes with their names, states, and probabilities. The connections are represented by lines indicating the flow of failure from one component to another.

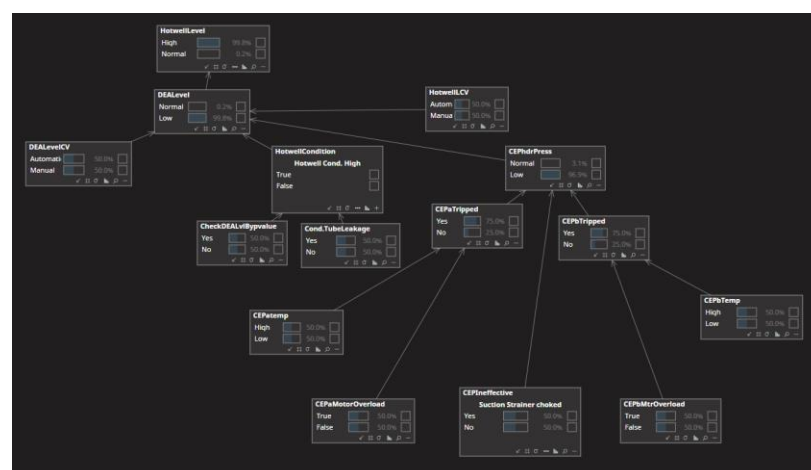
Components and their states/probabilities:

- HotwellLevel**
 - High: 33.3%
 - Low: 33.3%
 - Normal: 33.3%
- DEALLevel**
 - High: 33.3%
 - Low: 33.3%
 - Normal: 33.3%
- HotwellLCV**
 - Automatic: 55.5%
 - Manual: 44.5%
- CEPbdr:Press**
 - Normal: 96.9%
 - Low: 3.1%
- CEPbTripped**
 - Yes: 75.0%
 - No: 25.0%
- CEPbTemp**
 - High: 0.0%
 - Low: 0.0%
- CEPbMotorOverload**
 - True: 50.0%
 - False: 50.0%
- CEPbIneffective**
 - Suction Strainer choked: 50.0%
 - No: 50.0%
- CEPbTripped** (repeated)
 - Yes: 75.0%
 - No: 25.0%
- Cond.TubeLeakage**
 - Yes: 50.0%
 - No: 50.0%
- CheckDEALvlByvalue**
 - Yes: 50.0%
 - No: 50.0%
- HotwellCondition**
 - Hotwell Cond. High: 75.0%
 - False: 25.0%
- DEALLevelCV**
 - Automatic: 75.0%
 - Manual: 25.0%
- MU Flow**
 - Low: 50.0%
 - Normal: 50.0%
- EED Operation**
 - Yes: 50.0%
 - No: 50.0%
- Check for BLR TubeLeak**
 - Yes: 50.0%
 - No: 50.0%
- Deal LCV sluggish response**
 - Yes: 50.0%
 - No: 50.0%
- Gen. Output > 100MW**
 - Yes: 50.0%
 - No: 50.0%
- Check for No. of CEPs runs**
 - Yes: 50.0%
 - No: 50.0%

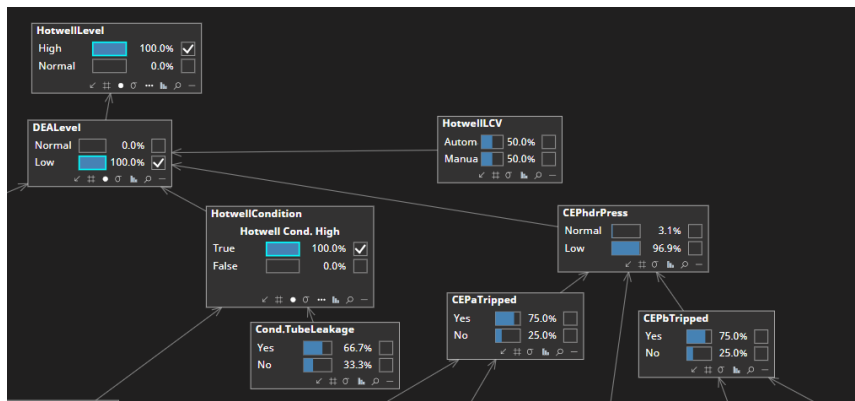
Interconnections:

- HotwellLevel** is connected to **DEALLevel** and **HotwellLCV**.
- DEALLevel** is connected to **HotwellLCV**, **CEPbdr:Press**, and **CEPbTripped**.
- HotwellLCV** is connected to **CEPbdr:Press** and **CEPbTripped**.
- CEPbdr:Press** is connected to **CEPbTripped** and **CEPbTemp**.
- CEPbTripped** is connected to **CEPbTemp** and **CEPbMotorOverload**.
- CEPbTemp** is connected to **CEPbMotorOverload** and **CEPbIneffective**.
- CEPbMotorOverload** is connected to **CEPbIneffective**.
- CEPbIneffective** is connected to **CEPbTripped**.
- Cond.TubeLeakage** is connected to **HotwellCondition** and **CEPbTripped**.
- CheckDEALvlByvalue** is connected to **HotwellCondition** and **DEALLevelCV**.
- HotwellCondition** is connected to **DEALLevelCV** and **CEPbdr:Press**.
- DEALLevelCV** is connected to **MU Flow** and **EED Operation**.
- MU Flow** is connected to **EED Operation** and **Deal LCV sluggish response**.
- EED Operation** is connected to **Check for BLR TubeLeak**.
- Check for BLR TubeLeak** is connected to **Deal LCV sluggish response**.
- Deal LCV sluggish response** is connected to **Gen. Output > 100MW**.
- Gen. Output > 100MW** is connected to **Check for No. of CEPs runs**.
- Check for No. of CEPs runs** is connected to **HotwellLevel**.

Let's check on one network to see how it functions



When no evidence is set, we see that all of the variables are in normal state. Now suppose we have some evidence, let's say on three condition node and we get distribution on one leaf node which is not very certain and hence we cannot draw any conclusions from here.



Lets say for HotwellLevel -> High, DEALevel -> Low, Hotwell Condition High -> True

We see that the root cause is Tube Leakage. Since there is a 66.7% chance of tube leakage, we want this uncertainty to get diminished. We use the Value of Information technique which can identify other nodes in the network which if known, will reduce our uncertainty and we can be 100% sure about our conclusions about the network.

Problem statement 3

Create a Causal Bayesian network and make interventional and counterfactual queries to infer important insights.

Correlation and causation:

Correlation/ association is the statistical dependency between two variables. Two types are positive correlation and negative correlation and this is denoted by the Pearson's correlation coefficient 'r'. All of the machine learning models that we use takes this into consideration, but it does not make the machine intelligent like humans.

E.g. – Sleeping with our shoes has positive correlation with having head ache the next day, but this has no meaning we cannot say that sleeping with shoes is a cause of the head ache.

Causation is when one affects the other. B happens only if A happens.

E.g. – I take medicine and I become healthy. Had I not taken the medicine I would have stayed sick. Thus, medicine is the cause and healthy is the effect.

Confounder:

Confounder is the common cause of treatment and effect. In the first example that we considered drinking is the common cause of both sleeping with shoes and having head ache the next day. This is the reason why we have positive correlation between the two variables.

Thus, correlation is the mixture of confounding association and causal association. When we want to study about the causal nature of the problem we have to remove the confounders. This can be done using several techniques like the backdoor adjustment, front door adjustment etc.

Causal model:

This is the Bayesian model where all the edges represent cause and effect relationship.

Individual /Average treatment effect:

Potential outcomes are called potential because they didn't actually happen. Instead, they denote what would have happened in the case some treatment was taken.

Individual treatment effect = $Y(A=1) - Y(A=0)$

Of course, due to the fundamental problem of causal inference, we can never know the individual treatment effect because we only observe one of the potential outcomes. For the time being, let's focus on something easier than estimating the individual treatment effect. Instead, let's focus on the average treatment effect, which is defined as follows.

$$ATE = E[Y_1 - Y_0]$$

Where Y_1 is the output when the treatment is 1 and Y_0 is the output when the treatment is 0.

Another easier quantity to estimate is the average treatment effect on the treated.

$$ATT = E[Y_1 - Y_0 \mid T=1]$$

Association is measured by $E[Y \mid T=1] - E[Y \mid T=0]$

Causation is measured by $E[Y_1 - Y_0]$

Where

$$E[Y|T=1] - E[Y|T=0] = E[Y_1 - Y_0 | T=1] + \{ E[Y_0|T=1] - E[Y_0|T=0] \}$$

Here the first term is the average treatment effect given treated and the second term represents the bias.

I.e., Bias = Affected not given treatment – unaffected not given treatment

The bias is given by how the treated and the control group differ before the treatment, in case neither of them has received the treatment.

Two groups are comparable only if their effects are same when no treatment is given.

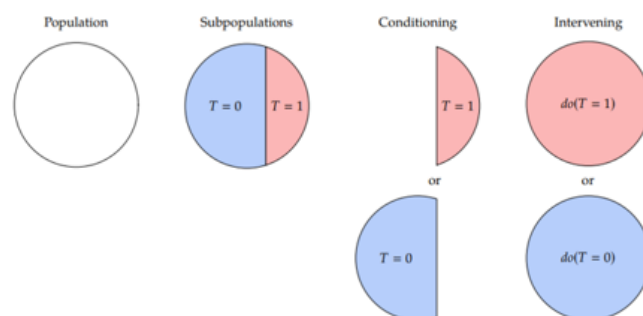
Randomized Control Trials:

Now, we look at the first tool we have to make the bias vanish by randomised experiments. Randomised experiments randomly assign individuals in a population to a treatment or to a control group. The proportion that receives the treatment doesn't have to be 50%. You could have an experiment where only 10% of your samples get the treatment.

Intervention:

Overriding a variable is different to observing its behaviour. When we override, we call it an intervention.

Conditioning on $T=t$ just means that we are restricting our focus to the subset of the population to those who received treatment t . In contrast, an intervention would be to take the whole population and give everyone treatment t .



We denote intervention with the do operator $do(T=t)$

Conditional independence and d-separator:

In a chain junction, $A \rightarrow B \rightarrow C$ controlling for B prevents information about A from getting to C or vice versa.

In a fork, $A \leftarrow B \rightarrow C$ controlling for B prevents information about A from getting to C or vice versa.

In a collider, $A \rightarrow B \leftarrow C$ exactly the opposite holds here. The variables start out independent but if you control for B then information starts flowing.

Conditioning on descendants of a collider also induces association in between the parents of the collider. So, conditioning on the descendant is similar to conditioning on the collider itself.

So whenever two variables are conditionally independent that means that the path between them is 'blocked'

The flow of association is symmetric, whereas the flow of causation is not. Causation only flows in a single direction

Two set of nodes are called d-separated by a set of nodes Z if all the paths between X and Y are blocked by Z.

Backdoor criterion:

This is a method where we control for confounders to find the actual causal effect.

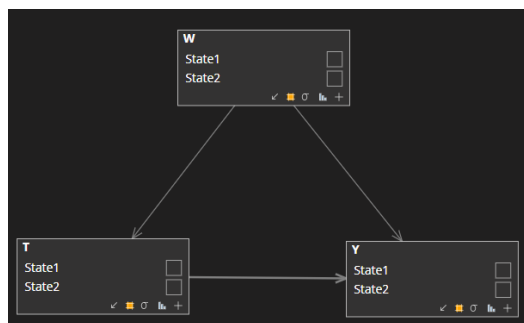
The nondirected unblocked paths from T to Y are known as backdoor paths. And it turns out that if we can block these paths by adjusting (i.e., making it a constant value), we can identify causal quantities. I.e., close all backdoor paths while leaving the front door paths.

A set of variables {Z} satisfies the backdoor criterion relative to an ordered pair of variables (Treatment(T) and Outcome(Y) in a DAG if:

1. No node in {Z} is a descendant of T.
2. {Z} blocks every path between T and Y that contain an arrow into T (called the backdoor path)

Hence the causal effect of T on Y is

$$P(y | do(t)) = \sum P(y | t, w)P(w)$$

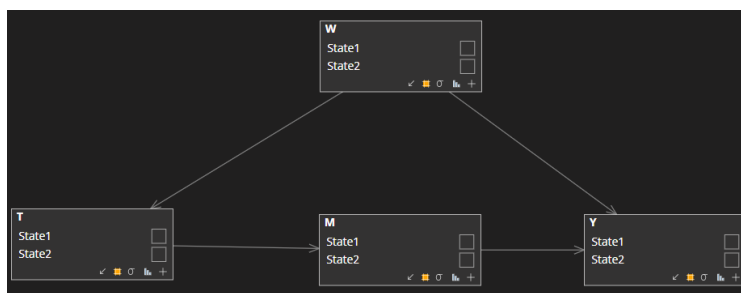


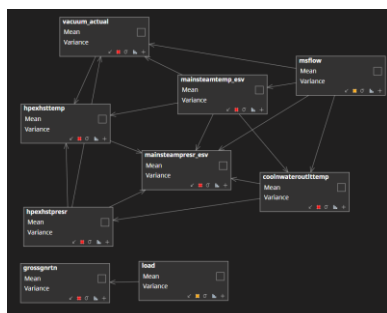
If we don't have data for the conditioning variables then this strategy won't work.

Front door adjustment:

When we cannot collect data on the confounders, we cannot use the backdoor criterion. Instead, we use the front door adjustment method. The steps to perform front door adjustment are

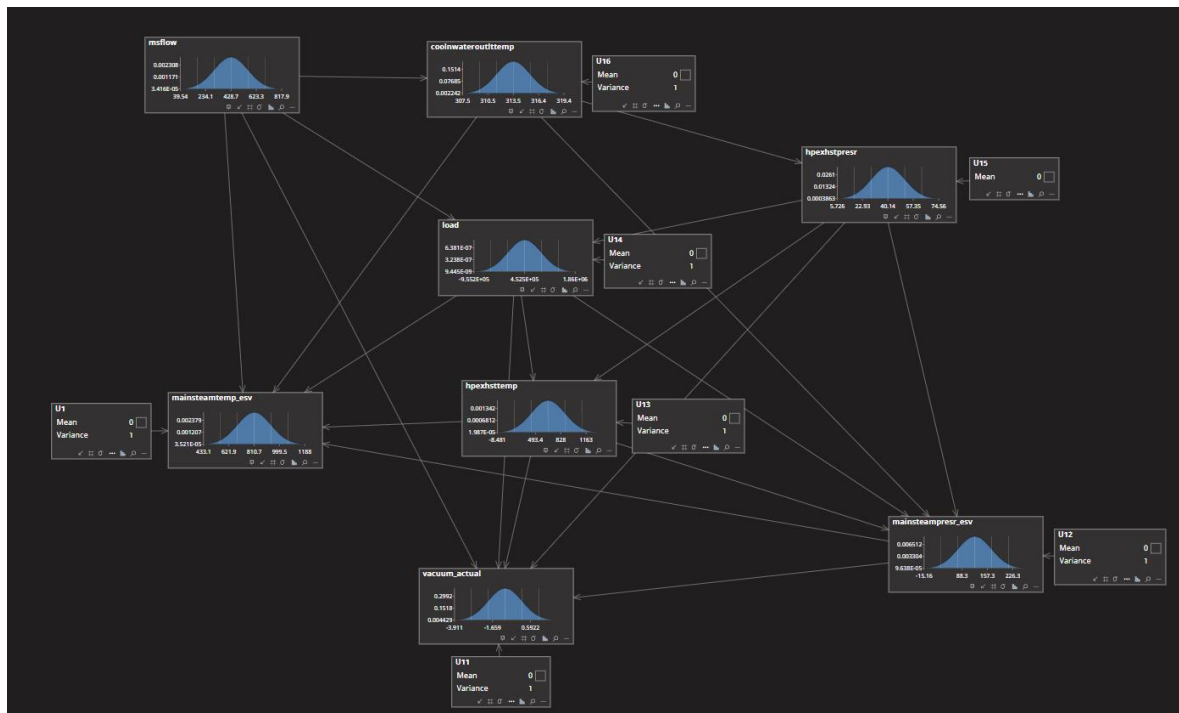
1. Identify the causal effect of T on M.
2. Identify the causal effect of M to Y
3. Combine the above steps to identify the causal effect of T on Y.





We further verify the network by expert opinion and by parameter learning we learn the corresponding conditional probability tables. We can make causal queries using this model .A counterfactual is a hypothetical or "what-if have happened if things would be different”.

Do-calculus offers no way of connecting the information across the different worlds thus we have to make use of the characteristic variable and not just depend on the interventions.



Consider an example to understand the counterfactual queries,

Suppose we have

Caseld	msflow	hpexhstpres	hpexhsttemp	mainsteampres_esv	mainsteamtemp_esv	coolnwateroutltemp	vacuum_actual	load
1	336.28268	33.048695	666.456	97.394905	803.13696	312.25037	-0.921247	348111.62

We want to know under the same situation what would be my load if msflow was 500.

There are three steps to perform such query

1. Abduction – use the data about caseid 1 to estimate its idiosyncratic factors U1, U11, U12, U13, U14, U15, U16 for this case.
2. Action - Use the do-operator to change the model to reflect the counterfactual assumption being made, in this case that it has msflow 500
3. Prediction – Calculate this caseid’s new load using the modified model and the updated information about the exogenous variables U1, U11, U12, U13, U14, U15, U16.

Like this we can get a lot of more information after making queries from this causal network.