NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

BAYESIAN NETWORKS FOR AUTOMATED EQUIPMENT DIAGNOSTIC SUPPORT

by

Zachary C. Polson

June 2021

Thesis Advisor: Jefferson Huang
Second Reader: Peter A. Nesbitt

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE				*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.				
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** June 2021	**3. REPORT TYPE AND DATES COVERED** Master's thesis		
4. TITLE AND SUBTITLE BAYESIAN NETWORKS FOR AUTOMATED EQUIPMENT DIAGNOSTIC SUPPORT				**5. FUNDING NUMBERS**
6. AUTHOR(S) Zachary C. Polson				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000				**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A				**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.				**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

As combat vehicles and other legacy systems age and are required to perform additional capabilities on increasingly remote battlefields, the Marines responsible for them currently lack tools necessary to diagnose and fix these critical assets independent from higher echelon corrective maintenance service support. For a Light Armored Reconnaissance detachment conducting distributed maritime operations and tasked with providing organic precision fires, small unit leaders and maintainers are responsible for performing all levels of diagnostics with minimal direct support, a situation that threatens expeditionary advanced base operations when vehicles inevitably fail. At the operator level, current troubleshooting procedures are primitive and fail to capitalize on recent breakthroughs in computation and causal reasoning algorithms. An automated program driven by a causal Bayesian network allows the maintainer to input observed symptoms into a model that directs their attention to the most probable causes of failure. Expert knowledge, Bayesian learning techniques, and automated reasoning are applied to determine network structure, model parameters, and the degree to which various symptoms affect output. When linked to a user interface, the maintainer can now quickly and accurately diagnose a degraded system from a handheld device, hundreds of nautical miles from the nearest maintenance bay.

14. SUBJECT TERMS causal Bayesian networks, Bayesian learning, R, bnlearn, Shiny, probabilistic graphical models, data analysis			**15. NUMBER OF PAGES** 59
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

BAYESIAN NETWORKS FOR AUTOMATED EQUIPMENT DIAGNOSTIC SUPPORT

Zachary C. Polson
Captain, United States Marine Corps
BBA, University of Wisconsin - Eau Claire, 2010

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
June 2021

Approved by: Jefferson Huang
Advisor

Peter A. Nesbitt
Second Reader

W. Matthew Carlyle
Chair, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

As combat vehicles and other legacy systems age and are required to perform additional capabilities on increasingly remote battlefields, the Marines responsible for them currently lack tools necessary to diagnose and fix these critical assets independent from higher echelon corrective maintenance service support. For a Light Armored Reconnaissance detachment conducting distributed maritime operations and tasked with providing organic precision fires, small unit leaders and maintainers are responsible for performing all levels of diagnostics with minimal direct support, a situation that threatens expeditionary advanced base operations when vehicles inevitably fail. At the operator level, current troubleshooting procedures are primitive and fail to capitalize on recent breakthroughs in computation and causal reasoning algorithms. An automated program driven by a causal Bayesian network allows the maintainer to input observed symptoms into a model that directs their attention to the most probable causes of failure. Expert knowledge, Bayesian learning techniques, and automated reasoning are applied to determine network structure, model parameters, and the degree to which various symptoms affect output. When linked to a user interface, the maintainer can now quickly and accurately diagnose a degraded system from a handheld device, hundreds of nautical miles from the nearest maintenance bay.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

BIC Bayesian information criterion

BLT battalion landing team

C5ISRT command, control, communications, computers, combat systems, intelligence, surveillance, reconnaissance, and targeting

CBN causal Bayesian network

CPG Commandant's Planning Guidance

CPT conditional probability table

DAG directed acyclic graph

DES discrete event simulation

DMO distributed maritime operations

EABO expeditionary advanced base operations

GCSS Global Combat Support System

GUI graphical user interface

LAR light armored reconnaissance

LAV light armored vehicle

LCT littoral combat team

MCSC Marine Corps Systems Command

MEU Marine expeditionary unit

MLE maximum likelihood estimate

NPS Naval Postgraduate School

NCO	non-commissioned officer
OS	operations and support
OPF	organic precision fires
TM	technical manual
TM EABO	Tentative Manual for Expeditionary Advanced Base Operations
USMC	U.S. Marine Corps

Executive Summary

This study seeks to demonstrate the viability of Causal Bayesian Network (CBN) models to support the diagnosis of degraded systems and equipment to inform troubleshooting and repair decisions. Operating beyond the reach of higher echelon support, small unit leaders of littoral combat team (LCT) attachments will be solely responsible for maintaining equipment that is vital for mission success and enabling expeditionary advanced base operations (EABO). The current approach to corrective maintenance favors centralizing forces around the higher-echelon expertise and cumbersome equipment of battalion level maintenance. Due to the nature of distributed maritime operations (DMO), the decentralized force must provide their leaders, maintainers, and technicians at the first-echelon with every resource possible to persist independently while remaining mission capable.

An automated program using CBN logic can empower first-echelon maintainers to accurately diagnose degraded equipment without the support of cumbersome diagnostic machines or senior-maintainer expertise. Mature, open-source software is available for implementing CBNs that can be rendered in a user-friendly graphical user interface (GUI) application and loaded onto a handheld device or toughbook. The user gathers information about a degraded system, inputs that information into the model, and based on automated reasoning, the model provides the user with the likely root causes of the failure and an associated probability that each potential cause is responsible for the system fault or failure.

Two main approaches can be used to construct CBN structure and determine model parameters. This first is through expert elicitation. A subject matter expert can enumerate all known system components and causal relationships between these components. The second method involves estimating the network structure and parameters using data. This method is preferred when the requisite system data can be readily captured, or when it is too costly to elicit expert support. For this research, three CBN models were built to represent causal dependencies between components of a Light Armored Vehicle (LAV) automotive system. First, an "expert" model was manually constructed by enumerating all components and direct component causal dependencies derived from an LAV technical manual (TM). Next, a "learned" model was built which estimated network structure from data. Lastly, a "hybrid" model was built that resulted from a review and subsequent modification of the estimated network structure of the learned model based on subjective expert reasoning. For all three models, parameters were estimated from the notional dataset used to build the learned and hybrid models. These models are linked to a GUI to demonstrate model functionality. In addition to demonstrating the potential utility of CBNs for augmenting maintainers' diagnostic capabilities, the fact that CBNs can be learned from data indicates a means by which better data collection can be translated into improved maintenance processes. Specifically, if data is available in a format similar to the notional dataset that we employ for constructing the data-driven CBNs, then the methodology presented in this thesis for constructing the learned and hybrid models provides a roadmap to leveraging that data to make improved maintenance decisions.

The potential for using automated processes to support corrective maintenance extends well beyond CBNs. Diagnostic CBN model output can provide inputs for a cost-utility optimization model, determining a sequence of optimal troubleshooting steps and an estimated cost of repair.

While better equipping maintainers to make decisions should result in faster maintenance turnaround, a higher level of material readiness, and a reduction in Class IX repair part costs, further research should be conducted to quantify this benefit to the Marine Corps. Those results can inform the depth and breadth of automated diagnosis implementation across other Marine Corps systems and platforms. While this validation may justify further investment in data-driven automated corrective maintenance processes and divestment of current limited methods outlined in the technical manuals (TMs), this study qualitatively verifies how the CBN can serve as the foundation for translating both expert knowledge and data into a form that is directly applicable to operational decision-making.

CHAPTER 1:
Introduction

1.1 Background

The latest National Defense Strategy directs the services to modernize programs, sustain efforts towards resilient and agile logistics, and "invest broadly in military application of autonomy, artificial intelligence, and machine learning, including rapid application of commercial breakthroughs, to gain competitive military advantages" (Mattis 2018). While the Marine Corps is automating the supply chain process for corrective maintenance and replenishment of class IX repair parts for non-consumable end items, we lack a modern method for supporting tactical-level maintainers who are directly responsible for the resiliency of material system capabilities deployed to support sea control and sea denial operations, maritime domain awareness, forward command, control, communications, computers, combat systems, intelligence, surveillance, reconnaissance, and targeting (C5ISRT) and counter-C5ISRT, and forward sustainment operations.

As the equipment and technology we put into the hands of the warfighter grows more complex, the maintainer's ability to diagnose and repair more complex systems must expand at the same rate. Distributed maritime operations (DMO) and expeditionary advanced base operations (EABO) can function only if we provide small units with enabling capabilities that allow them to be self-sufficient. This requirement is highlighted by the 38th Commandant of the Marine Corps, General David H. Berger, in his Commandant's Planning Guidance (CPG). General Berger (2019) charges leaders at every level to dramatically rethink entrenched processes and service orthodoxy. Equipment, weapons, tools, and processes earmarked as cumbersome and antiquated must be divested of and replaced with capabilities aligned with EABO guiding principles. These ideas and concepts take tangible shape with the release of the Tentative Manual for Expeditionary Advanced Base Operations (TM EABO). In this seminal document, the United States Marine Corps (2021) begins to define this vision of the littoral combat team (LCT) future force construct and inspires a future operating environment where the time and distance required for evacuation reduces the responsiveness of the maintenance system, risks reducing littoral force capability, and threatens the mission.

1.2 Thesis Motivation

1.2.1 Purpose

The purpose of this research is to address deficiencies in human-centric processes for diagnosing and repairing degraded systems and equipment by demonstrating a data-driven automated diagnostic decision support tool capable of streamlining the troubleshooting process.

1.2.2 Defining the Problem

Material readiness deficiencies occur under the broader context of an antiquated model for conducting forward corrective maintenance. These problems are exacerbated in a distributed maritime environment as an attachment to a larger task force (i.e., a Marine expeditionary unit (MEU) or LCT). Currently, the Marine Corps asks attachment commanders to operate under the construct of higher echelon service and support that is not properly outfitted to process their unique corrective maintenance issues. For instance, a regular infantry battalion that forms the nucleus of the battalion landing team (BLT) lacks the experience and resources to fix M777 Howitzer cannons or light armored vehicles (LAVs). Therefore, it is the sole responsibility of the attachment to provide internal corrective maintenance support. In many instances, the material sustainability of a critical supporting attachment will depend solely on the attachment Platoon Commander, Platoon Sergeant, a Non-Commissioned Officer non-commissioned officer (NCO) mechanic or technician, and whatever tools they can carry with them. When breakdowns occur and recovery by higher echelons of support is not an option, these small unit leaders will be forced to decide whether to sanitize and abandon equipment or continue their diagnostic efforts. The current maintenance construct imposed on the future battlefield will inevitably lead to exposed capability gaps resulting in delayed repair, a less agile force, class IX waste, diminished capability, and higher costs.

1.2.3 Current Approach to the Problem

Given this attachment force organization construct, non-routine corrective maintenance problems present a significant challenge. Without a departure from antiquated procedures held over from past conflicts fought in deserts and on large land masses, corrective maintenance efforts at both the tactical and organizational level will continue to weigh on operations and support (OS) life-cycle costs while producing marginal contributions to mission readiness.

The current processes for handling complex mechanical failures beyond the attachment personnel's experience and expertise are defined by the equipment's technical manual (TM). Each TM has an associated level of either 10, 20, 30, or 40. This corresponds with a respective maintenance echelon, where 10 is the lowest (operator level) echelon and 40 is the highest (depot level) echelon. A 10 level TM purposefully lacks the depth and detail to enable somebody at the first echelon to handle in-depth corrective maintenance issues. These TMs contain troubleshooting decision trees for corrective maintenance diagnosis.

The TM approach can rely on subjective and arbitrary prioritization by the maintainer. Degraded equipment may present multiple signs indicative of an issue. That information taken as a whole is likely more valuable than the sum of each piece of information separately, but the TM approach cannot capitalize on this information synergy. Depending on what system fault evidence is initially detected and pursued, a user may go down multiple TM decision tree paths, conduct multiple tests, and possibly replace multiple parts before addressing the true cause of the problem.

Often, the troubleshooting decision tree recommends actions that are impossible when operating in a remote distributed maritime environment. For instance, in the Operator's Manual, Light Armored Vehicle LAV-2A2 (Legacy) Automotive/Hull (TM 08594C-10/2-LG), the phrase "notify organizational maintenance" appears 116 times. While this option exists when training in CONUS or fighting as part of a light armored reconnaissance (LAR) battalion, the LAR platoon attached to a MEU or LCT does not have this luxury.

Common solutions to residual readiness problems at the organizational level include costly, cumbersome, risky, and generally unsavory options such as:

1. Committing more time and resources to training and equipping maintenance personnel
2. Restricting battalions to centralized operations due to vehicle and equipment maintenance and recovery support limitations

3. Civilian contractors having to deploy to dangerous remote outposts at significant cost to the federal government and taxpayer

1.2.4 A Look into the Future

If the Marine Corps plans on utilizing the LAV as an organic precision fires (OPF) platform, it must ensure the LAV is resilient enough to withstand mechanical and combat-related breakdowns and continue to operate as a critical asset to the EABO mission. When operating remotely while conducting distributed maritime operations (DMO), vehicle recovery is not an option. A commander has the option of either fixing or abandoning degraded or inoperable equipment. Resilient design and robust platforms do not eliminate the requirement for human intervention in restoring the functionality of degraded and dead-lined equipment. In an information age pervasive with nearly limitless pocket-sized computation and data storage power, we can continue to provide our first echelon maintainers with doubtfully useful 300+ page TMs, or we can use an automated program to streamline troubleshooting by quickly and accurately diagnosing system faults and failures. The need for such solutions grows as the complexity of the systems and technology put in the hands of the warfighter increases.

1.2.5 An Automated Approach

Machines can store data, compute statistics, and use algorithms to inform decisions. To fully harness the power of computation, machines can be used to detect patterns in data and provide inference where humans would not think to look. Visionaries like David Ferrucci (of IBM "Watson" fame) imagine machines "that can combine deductive and inductive processes to develop, apply, refine, and explain" (Lohr 2013).

Kasparov (2017) describes the decades when many of the brightest minds in computer science asserted that developing a chess playing program that could defeat a grandmaster was the holy grail of computer programming. He goes on to describe how ultimately, Moore's law won out, and simple "brute force" algorithms proved sufficient to beat world champion human players. Using these algorithms, a programmer can teach a chess machine the rules of chess. They can implement logic so it knows which pieces can move where and how checkmate is achieved. They can tell it how valuable a queen is relative to a pawn. "Knowledge" is anything that goes beyond these basic rules and mechanics. If you program it with the knowledge that a pawn is worth more than a queen, it will go into battle willing to lose the queen with no hesitation and likely throw the game away with it. Now, what if those values are not explicitly built into the program? What if you tell it the rules and let it figure out the rest? Let the machine figure out that rooks are more valuable than bishops, that doubled pawns can be weak, that open files can be useful. This opens up the possibility of creating a strong chess machine, and also a machine that allows humans to learn something new from what the machine discovers and how it discovers it (Kasparov 2017).

We can similarly ponder what happens if we omit our qualitative bias about the physical structure of a system and instead let the machine figure out that leaking coolant hoses cause radiator cores to malfunction, which in turn causes an engine to overheat. It may detect, without prompting, that a loss in steering function can be caused by either a broken hydraulic fluid reservoir or steering fluid pump then assign a probability distribution that it is one or the other given circumstantial evidence and information. Also, it may tell us that the condition of fuel injector nozzles only depends on injector pump timing if black smoke is being emitted from the exhaust. We often use computation to answer questions, but harnessing the true power of computation points us to questions humans would not think to ask (Kasparov 2017).

1.3 Thesis Objective

The author's approach to answering key issues and objectives involves implementation of a Causal Bayesian Network causal Bayesian network (CBN) to demonstrate how an automated program can better support the diagnosis of system faults and failures. A secondary purpose is to demonstrate how data can be used with network structure and model parameter learning methods to learn system components and the strength of causal dependencies among system components. Based on the author's familiarity with the LAV, the LAV hull and automotive component system was selected as the subject of the model. Three directed acyclic graphical models are built to demonstrate various methods for determining causal dependencies between LAV automotive system components and evidence of malfunction for the LAV automotive system. With these networks and accompanying data, we determined model parameters and utilize established algorithms implemented in software to query these various CBNs for inference. The first model, our "expert" network, uses the TM to determine dependencies between components and evidence. The second model, our "learned" network, uses causal dependencies detected in simulated data to learn network structure. The third model is our "hybrid" network, where each arc was reviewed to ensure directional dependencies identified by the software's algorithm were reasonable. The same data used to estimate network structure for our learned model was used to estimate network parameters for all three models.

1.3.1 Functional Diagnostic Models

For the sake of this research, we define evidence as an observed or measured system or system component status presented to the user. Some examples of evidence in our LAV example could include an engine (system component) that doesn't run (status) or a tire that's flat. In some cases, evidence could be an external factor like air temperature. Inputting evidence in the model updates the parameters, or conditional probability tables, associated with relevant system components or evidence nodes. These updated conditional probability tables inform the relative likelihood of the root cause of failure, the basis for model output. The outputs of these models can inform first echelon maintainers about the most likely source of mechanical failure, focusing their troubleshooting effort. System operators can combine this information with their experience and knowledge of the cost of testing and replacing various parts to quickly and accurately diagnose a mechanical failure and expeditiously take corrective action. Additionally, system analysts can use learned networks estimated from data to detect otherwise unknown causal dependencies between system components.

1.4 Thesis Organization

Chapter II of this thesis addresses the concept of Causal Bayesian Networks and provides a literature review of contributions and breakthroughs in the field of probabilistic graphical modeling. Chapter III will detail the methods used to build the three CBNs representing causal dependencies between LAV automotive components and evidence and the associated conditional probability table parameters. Chapter IV provides graphical and quantitative analysis of the models and addresses querying of the model for inference. Chapter V provides conclusions, recommendations on larger scale implementation, and recommendations on future studies.

CHAPTER 2: Background and Related Work

This chapter will provide a general overview of probability theory and CBNs, explore breakthroughs in the field of probabilistic graphical modeling and automated reasoning, and discuss the development and implementation of CBNs as a diagnostic support tool.

2.1 General Overview

2.1.1 Introduction

Aebischer et al. (2017) describes the CBN as a modeling methodology that meets all criteria to represent the combination of system expert knowledge and beliefs with empirical evidence and case data. With the help of open-source software, the resulting model can be easily encoded and rendered in an intuitive graphical user interface. The processes and mathematics supporting a CBN can provide exact closed form solutions or accurate approximate inference. They are capable of managing an extensive range of network breadth and complexity, but flexible enough to incorporate various types of data. Also, they are powerful enough to account for direct and indirect probabilistic relationships between several variables. Lastly, the body of research behind the CBN is mature enough to utilize machine learning techniques for analyzing and updating models. (Aebischer et al. 2017).

2.1.2 Network Structure

The CBN is a directed acyclic graphical representation of a system where each node corresponds to a component or attribute of the system and arcs indicate a causal dependency between a parent node and a child node. Therefore, this parent-child node relationship and the respective directed arc incident to both nodes comprise the foundation of a CBN (Aebischer et al. 2017). The most basic CBN is a two-node network where the parent node is considered a potential cause and the child node the resulting effect. We consider a mechanical example where the condition of an engine's exhaust valve may determine whether exhaust is present in the engine compartment, and the presence of exhaust depends only on the condition of the exhaust valve. We can think of the node as a discrete random variable, where the component status equates to variable levels. We can assign a probability that an observation of these random variables will take on a respective level. Figure 2.1 depicts this network and the respective probabilities that the representative nodes will be in a given state prior to attaining any knowledge or "evidence" about the status of these components. A model without inputted evidence is akin to selecting a random vehicle and having no prior knowledge of its mechanical status.

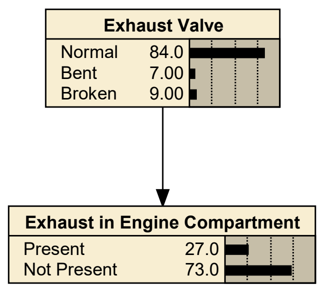


Figure 2.1. A two-node network

The trivial network in Figure 2.1 is a straightforward example. In order to understand confounding network structures, let's add more complexity to our network. We know that a defective valve spring can exert uneven or excessive pressure on the exhaust valve, causing it to break or bend at high engine temperatures. In probabilistic graphical terms, this is called a "serial" network, one of three basic confounding local network structures found in a CBN. Figure 2.2 depicts a serial network prior to the user attaining system knowledge or inputting that evidence into the network.

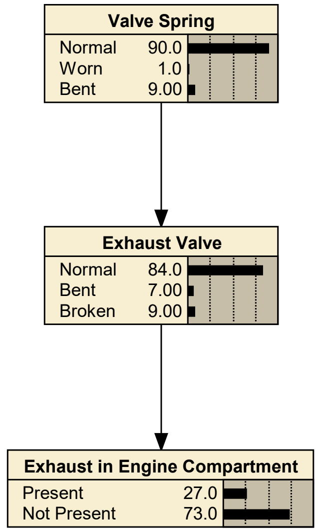


Figure 2.2. Serial network structure example

Slightly altering our example, consider a case where problems with an exhaust valve may potentially cause exhaust to appear in the engine compartment and/or a reduction in engine power due to compression loss. When a cause can have two or more potential effects, in graphical terms we call this a "divergent" network. Figure 2.3 depicts a divergent network and the respective probabilities the exhaust valve is in a particular state and the probabilities the user will observe exhaust in the engine compartment and/or a reduction in power. Again, no evidence has been input into this model, akin to approaching a vehicle blind to information about the exhaust valve, engine compartment, and engine power.

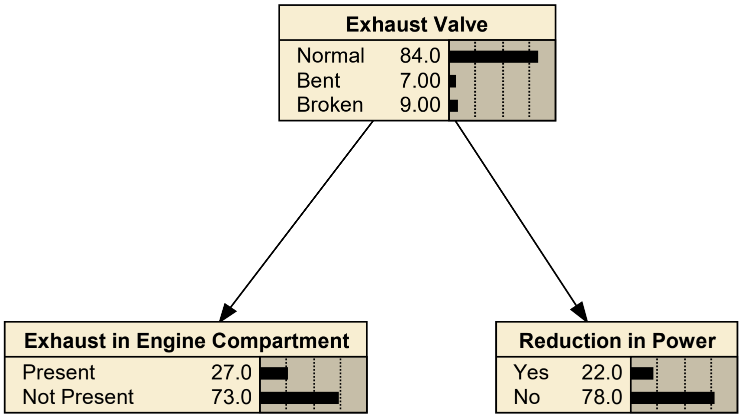


Figure 2.3. Divergent network structure example

The third and final basic network structure is the "convergent" network, which exists when one effect has two potential causes. For instance, exhaust in the engine compartment may be due to either a head gasket problem or an exhaust valve problem. This situation is graphically depicted in Figure 2.4. Again, we represent here a situation where no evidence has been input into the model, and only knowledge about the prior probabilities the components are in a respective state exists.

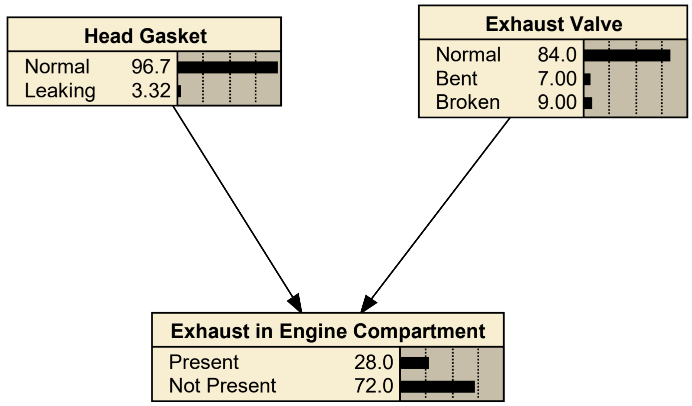


Figure 2.4. Convergent network structure example

These three basic network structures form the building blocks for constructing more complex probabilistic graphical networks. Figure 2.5 incorporates serial, convergent and divergent structures to form a simplified model of an engine system and its components. Attaining information about the status of a component renders its corresponding node an evidence node when that information is input into the model. Nodes with status information that is likely readily available for the user to input are highlighted in green in Figure 2.5.

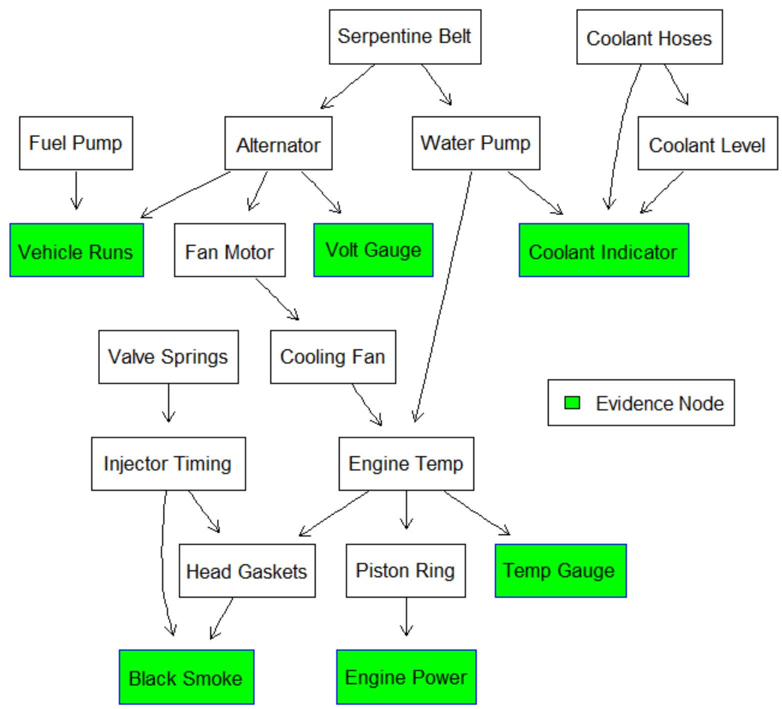


Figure 2.5. Simplified representation of causal dependencies between components and evidence of a diesel engine system

It is important to contrast this simplified model with what a visual of a real world graphical representation of a more complex system. Figure 2.6 depicts a small segment of a Bayesian Network with 448 nodes and 908 arcs built to support the diagnosis of medical patients. As the number of arcs and nodes increases, we lose visual tractability and gain an appreciation for the challenges in rendering a model by manually enumerating all nodes and arcs.

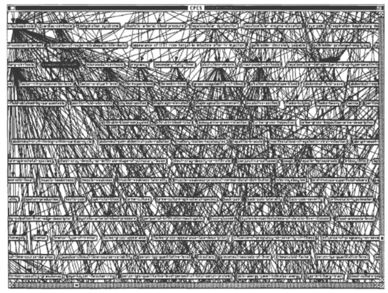


Figure 2.6. A small portion of a complex network including 908 arcs and 448 nodes. This network was built to support diagnosis of medical patients. Source: Pradhan et al. (1994).

2.2 Model Parameters: The Conditional Probability Table

Each node in a Bayesian network has an associated conditional probability table (CPT) that describes the probability a component is in a particular state given a set of known conditions. The values that comprise a node's conditional probability table are a function of its CPT and its parent node(s)' CPTs (Aebischer et al. 2017). Table 2.1 shows the CPT for the "Exhaust in Engine Compartment" node from Figure 2.4. The CPT in Table 2.1 describes the probabilities that we would see exhaust in the engine compartment given known evidence about the status of the exhaust valve and head gasket.

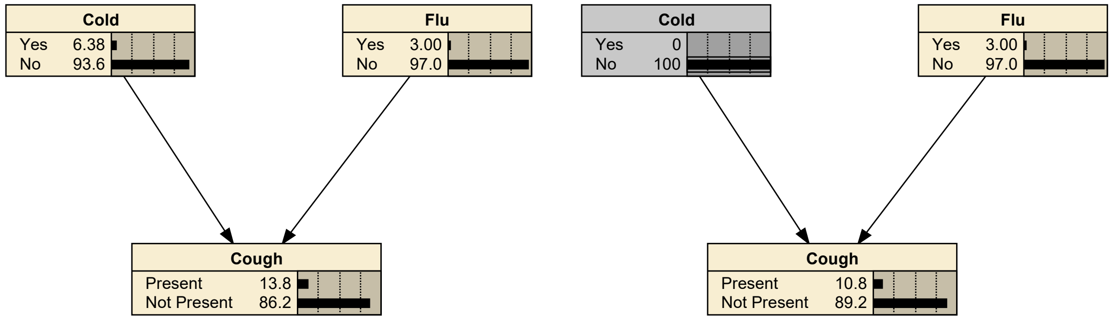
Table 2.1. Example conditional probability table for the network depicted in Figure 2.4

Node	**Exhaust Valve**	**Bent**		**Broken**		**Normal**	
	Head Gasket	**Normal**	**Leaking**	**Normal**	**Leaking**	**Normal**	**Leaking**
Exhaust in Engine Compartment	Yes	0.93	0.96	0.96	0.98	0.02	0.85
	No	0.07	0.04	0.04	0.02	0.98	0.15

For instance, if the exhaust valve is bent and the condition of the head gasket is normal, there is a 0.93 probability the user will observe exhaust in the engine compartment.

2.3 Automated Reasoning and Inference

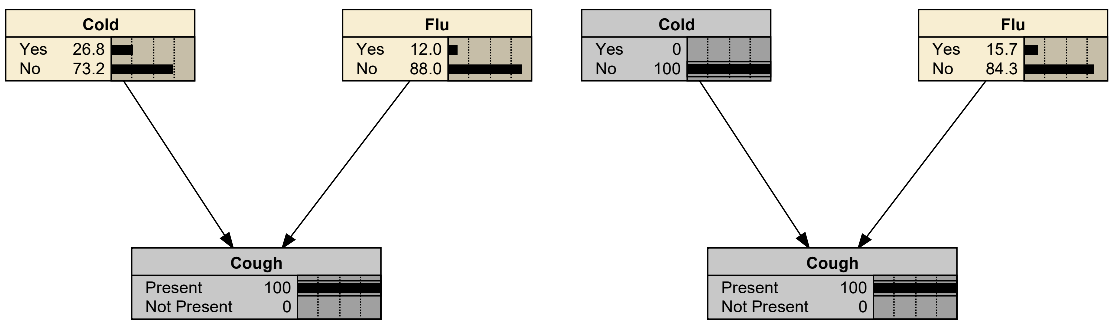
Reasoning algorithms, Bayesian probability theory, inference calculation techniques, and computation comprise the set of tools necessary to query a network for useful inference. When we input evidence information into our model, logic is applied via the directed separation (“d-separation”) algorithm to map dependencies between conditionally independent nodes. The resulting path contains the set of nodes and arcs that connect all direct and indirect causal dependency relationships between the evidence node and all other nodes in the network. Once the reasoning algorithm determines the dependency structure, exact or approximate inference techniques are applied to calculate updates to the relevant CPTs. The reasoning algorithm process ensures information input into the model will only influence adjacent nodes where marginal dependency exists. (Nodes that share an arc are considered adjacent nodes.) This is sometimes called network propagation or information flow. To exemplify how information inputs can alter flow, consider the converging networks in Figures 2.7 and 2.8 where the child node represents a medical patient potentially presenting a cough and the two parent nodes represent potential causes, a cold or the flu. Evidence input into the model is indicated by the gray boxes. With no information about the presence of a cough, there is no dependency between the presence of a cold and the presence of flu, and having confirmed the patient has a cold will not alter their probability of having the flu. Figure 2.7 demonstrates this temporary lack of dependency between the non-adjacent cold and flu nodes. Notice the probability of the patient having the flu remains at 3% after cold evidence is input into the model. However, confirming the presence of a cough “activates” a dependency between the two parent nodes. In this case, if you can confirm the patient does not have the cold, it is more likely the cough symptom is a result of the flu, and we would expect the probability of the flu to increase as we see with the same CBN depicted in Figure 2.8. The probability the patient has the flu has now increased from 12% to 15.7%, and there exists an “active trail” between cold and flu where there was once d-separation.



a. Cannot confirm or deny cough

b. Doctor confirms the patient does not have a cold

Figure 2.7. Lack of dependency between cold and flu node (note how inputting evidence about a cold does not impact the probability the patient has the flu)



a. Doctor confirms the patient has a cough

b. Additionally, Doctor confirms the patient does not have a cold

Figure 2.8. Activated dependency between Cold and Flu node (note how inputting evidence about a cold now impacts the probability the patient has the flu)

For small networks, exact inference can be calculated using Bayes' theorem and the chain rule of probability. For larger networks, an exact inference calculation is classified as NP hard, so a process that uses Monte Carlo simulation to randomly generate observations from the CBN can be used to perform approximate inference (Scutari and Dënis 2015).

2.4 Constructing Estimated Networks from Data

2.4.1 Benefits of a Data Driven Approach

A dataset of fully observed instances of network variables enables learning of network structure and network parameters. It is often more convenient, cost effective, and interesting for the analyst to use data to estimate network structure via learning algorithms. Koller and Fr̈iedman (2009) explain the pitfalls in relying purely on expert knowledge solicitation to construct CBNs. There are many examples of systems that are too large, too complex, and contain too many variable interactions to be effectively modeled by an expert with the requisite amount of knowledge in a reasonable amount of time. For some systems, such experts may not exist. As the methods for capturing data increase and the cost of data plummets, the information age may afford the analyst the ability to obtain readily available volumes of data more easily than human expertise, and the data may prove more informative than the expert. This data can be used to estimate network structure for diagnostic inference.

As a disclaimer, it is important to note that models learned from data will vary in their goodness-of-fit. As an approach limited by computation power and imperfect data samples, these models are limited if a near precise representation is required (Koller and Fr̈iedman 2009). In their seminal piece on network learning, Chow and Liu (1968) first determined how to approximate probabilistic causal network structure. They found it is important to consider empirical risk and overfitting when constructing a model from data. Conditional independence tests and network scoring methods exist to provide the analyst with relative quantifiable measures of model accuracy. When it is critical to assess the confidence in an approximate model, network construction via model averaging, conditional independence testing, and/or Markov chain Monte Carlo simulation may be employed (Koller and Fr̈iedman 2009). Cross-validation or network scoring (Scutari and Dënis 2015) can then be applied to learned CBNs to obtain unbiased estimates of goodness-of-fit (Scutari 2020).

A very different motivation for learning a model through data lies in our ability to explore a learned network to glean knowledge from the discovered arc set and various network paths. Because each individual arc represents a direct causal relationship while arc paths indicate indirect causal relationships, this distinction makes CBN structure learning a richer tool for detecting dependencies in data than other simpler statistical methods (Koller and Fr̈iedman 2009).

Not only does structure learning allow us to discover unidentified causal relationships between known variables, it also provides the opportunity to discover hidden causes. Heckerman (2020) describes how models can be scored with and without inclusion of “hidden variables” to determine if one exists. This, combined with search algorithms, provides one method for identifying possible hidden variables.

2.4.2 Causation vs. Correlation

Pearl (1988), developing extensions from the pioneering work of Chow and Liu (1968), explores whether it is possible to confidently ascribe direction to the arcs in a learned network. Recall arc direction represents the cause (emanating from parent node) and effect (to child node) dependency relationship between two nodes. If the direction of network arcs estimated from data indicate causal direction, how can we be sure this was not a detection of spurious correlation (Simon 1954)? To aid in determining if X causes Y or Y causes X, we can introduce variable Z. If Z is found to correlate with Y but not X, the tail of the arc projects from X to the arc head at Y. This does not mean X and Z are the ultimate causes of Y (Pearl 1988), and care should be given to review causal direction in an estimated network. Pearl (1988) concludes that “the construct of causality is merely a tentative, expedient device for encoding complex structures of dependencies in the closed world of a predefined set of variables.”

2.5 Learning the Parameters of Bayesian Networks

The required parameters in a discrete CBN model are the respective node probabilities conditioned on the various levels of its parent node (Scutari and Dënis 2015). These values form the substance of a CPT. Precise probabilistic numerical parameters are often more difficult to ascertain from experts than network structure, so parameter estimation is the more common technique (Koller and Fr̈iedman 2009). Two main approaches exist for estimating network parameters: 1) the maximum likelihood estimate (MLE), and 2) the Bayesian approach (Scutari and Dënis 2015).

2.6 Other Applications

There are many applications for CBNs beyond diesel engine maintenance diagnosis support. De Kleer and Williams (1986), Davis (1984), and Genesereth (1984) examined how belief networks can be used to diagnose multiple malfunctioning components in a digital circuit. Peng and Reggia (1986) used Bayesian networks to support detection of disease. Sutton and McCallum (2004) showed how the similar concept of conditional random fields can be used for text analysis and entity recognition, similar to the model in Figure 2.9. Temporal causal models can be constructed for target tracking and for tracking robot localization for automated movement (Fox et al. 1999). Causal models can be used for classification (Koller and Fr̈iedman 2009) and discovering user clusters (Breese et al. 1999). Bayesian networks can be used to conduct collaborative filtering for content delivery (social media feeds, search engine results, directed marketing, etc.), where graphical models of system user preferences can be used to determine preferences of users in general (Heckerman 2020). Deeper medical extensions include learning gene cell networks (Sachs et al. 2005) and prenatal diagnosis (Norman et al. 1998).

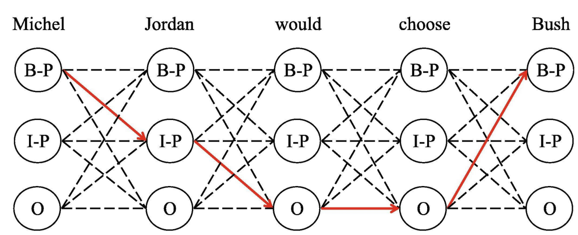


Figure 2.9. A graphical model developed for text analysis and recognition. Source: Chun Wei Lin et al. (2020).

2.7 The Value of Information

Pearl (1988) defines the value of any information source as "the difference between the utilities of two optimal strategies, one providing the freedom of choosing different actions for different source outcomes, and the other providing no such freedom." An objective application of this definition can assist in determining the usefulness of an information source and whether the cost of pursuing information from that source is worthy of an investment (Pearl 1988). Koller and Fr̈iedman (2009) explain how actions in mechanical troubleshooting involve running diagnostic tests to direct the user's attention to the problem, identifying and confirming the source of the problem, and supporting actions that remedy the issue. Both types of actions have a cost and an optimal strategy. Imagine a broken vehicle that requires either an engine replacement or an oil change. Pursuing either option will provide useful information. If the mechanic changes the oil and it fixes the problem, he receives information about the source of the problem. If he changes the oil and it doesn't fix the problem, he knows it requires an engine replacement. Prior to his decision, the user must consider whether the cost of pursuing either option (change the oil or replace the engine) is a worthy investment. A program using CBN logic tells you there's a high probability the engine is broken beyond repair and must be replaced, but there's also a low probability it only requires an oil change. Based on my intuitive cost-utility model, I will change the oil first regardless of the low likelihood it solves the problem. A maintenance support tool like the one built for this thesis can complement decision-theoretic utility techniques, or simple common sense, to help select a sequence of testing and repair actions. Heckerman, Breese and Rommelse (1995) take this a step further by augmenting probabilistic models with cost utility models to define an optimal series of repair and testing actions given a current state of known information, and the user can thereby compute the expected cost of repair at each step. Additionally, diagnostic CBNs can potentially inform a user about the probability they are witnessing a mere sensor or signal malfunction versus a potentially catastrophic malfunction, a potentially lifesaving distinction in an aircraft cockpit or elsewhere.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 3: Model Formulation

Programs driven by CBNs can support diagnosis and troubleshooting decisions based on input of multiple observed symptoms into a model that evaluates and returns a probability distribution of the most likely source of the defect. This chapter describes the composition of a CBN and the various methods used to construct a CBN model representation of a LAV. It explains the reasoning and processes behind model inference and presents a graphical user interface for easy model query.

3.1 Graphical Model of the LAV Automotive System

A graphical representation of the causal dependencies in a system can be constructed by connecting the multiple serial, diverging, and converging component sub-trees that comprise an entire network. When an expert possesses sufficient knowledge of these dependencies, it is possible to fully define the system network by manually enumerating the complete set of directed arcs corresponding to all parent (cause) and child (effect) relationships. Because the LAV TM 08594C-10/2-LG provides the basic causal relationships between LAV components and evidence necessary to diagnose a degraded LAV at the operator level, this same information can be used to determine CBN structure. For instance, the first step in the TM troubleshooting decision tree asks whether some evidence of a malfunction is present. This becomes a child node connected by one of the graph's directed arcs. If evidence exists, it directs you to the next step which may reference a malfunctioning component that is the cause of the evidence. This component represents the parent node, thus rendering one of the many directed arcs in our network. This step can be repeated multiple times until the decision tree directs you to the actual source of the problem, producing a new arc at every step in the process. For n steps in the troubleshooting path, there will be $n - 1$ directed arcs added to the overall set of arcs in the network. Once all possible routes in the troubleshooting decision trees are defined for all evidence of malfunction identified by the user and addressed in the TM, a directed acyclic graph (DAG) representation of the causal dependencies in our system is rendered. Figure 3.1 shows this resulting expert network derived from the LAV TM.

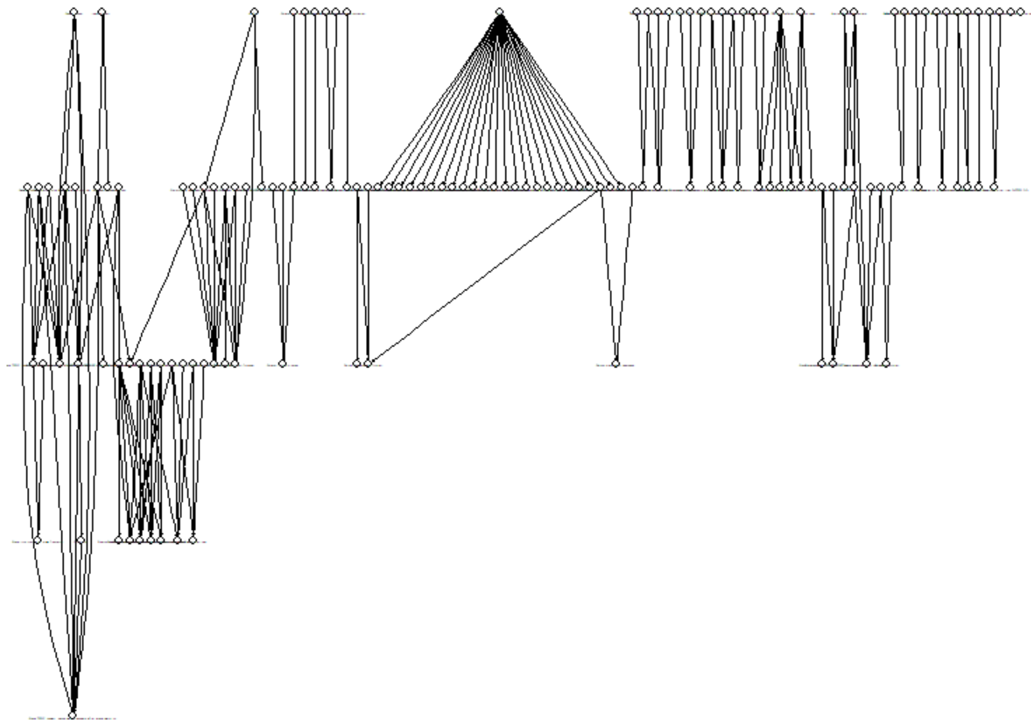


Figure 3.1. A graphical representation of causal dependency relationships between LAV automotive components and evidence nodes

In Figure 3.1, the multiple levels represent serial network structure and residual cause and effect relationships, similar to the smaller example from Figure 2.2.

3.2 From a DAG to a CBN

The second major step in constructing a CBN is defining the network parameters that comprise the values for each node's CPT. The dimensions of the associated CPTs are determined by the number of variables each node conditionally depends on, or the in-degree of each node. Looking at the CPT in Figure 2.1, we know that exhaust in the engine compartment depends on the status of the exhaust valve and the status of the head gasket. There are two ways to determine these conditional probabilities. The first method involves elicitation of a subject matter expert to extract conditional probabilities based upon their subjective degrees of belief (Koller and Frïedman 2009). These probabilities may then be directly applied to the CPTs. The second method involves estimating the parameters from data. Data in the form of a text file can be input into software that computes the classic frequentist and maximum likelihood estimates (MLEs) for parameters (Scutari and Dënis 2015). Table 3.1 provides an example of what a small portion of this dataset might look like, where each observation corresponds to a vehicle, each column header is a node in our network, and each value in the table represents one of the discrete levels or component statuses that node may take on. Open-source software packages can then apply MLE logic or Bayesian posterior distributions to translate these data into model parameters applied to an already pre-defined expert network to create a functional CBN model (Scutari and Dënis 2015).

Table 3.1. A subset of engine component data with one variable per column and one observation per line, similar to a text file format compatible with bnlearn R software (Scutari and Dënis 2015).

	Engine Temp	Cooling Fan	Oil Pressure	Water Separator	Coolant Pump
Vehicle 1	Normal	Normal	Normal	Full	Normal
Vehicle 2	Normal	Slow	Normal	Empty	Normal
Vehicle 3	Normal	Normal	Normal	Empty	Normal
Vehicle 4	Above 220°	Seized	Normal	Empty	Normal

3.3 Developing Data

The data-driven method for estimating network structure and parameters relies on the CBN developer's access to a dataset with the relevant information and a sufficient number of observations. The author developed a notional dataset for the sake of demonstration. The dataset structure is similar in form and content as the example in Figure 3.1, with one variable per column and one observation per line (Scutari and Dënis 2015). Construction of this dataset utilized random variate simulation to generate factor values by node and observation for the respective entries. The author was careful to build probabilistic dependencies into the values when it made reasonable sense based on a limited understanding of the system. For instance, an observation where the value for "brake shoes" was determined to be "worn" had a higher probability of "brake noise" taking on the value "squealing." Calculations confirmed the resulting dependency relationships satisfied the definition of probabilistic dependence. All data is categorical and derived from multinomial distributions with simulated probabilities determined by these built-in dependencies. This was replicated to populate values across 151 nodes identified in the TM for 400 observations to round out the dataset. This data was then applied to our network structure DAGs from Section 3.2 via the software's MLE logic for estimating parameters, completing our first of three CBNs.

3.4 Learned and Hybrid Networks

Multiple methods exist to estimate both parameters and network structure from data. To explore these methods, the author utilized the same open-source software and derived notional dataset to build a completely "learned" CBN. These methods have been extended to include processes for model selection to avoid over-fitting. The DAG representing this estimated structure is depicted in Figure 3.2. Each resulting arc can be explored to discover causal dependencies that were previously unknown to the system expert.

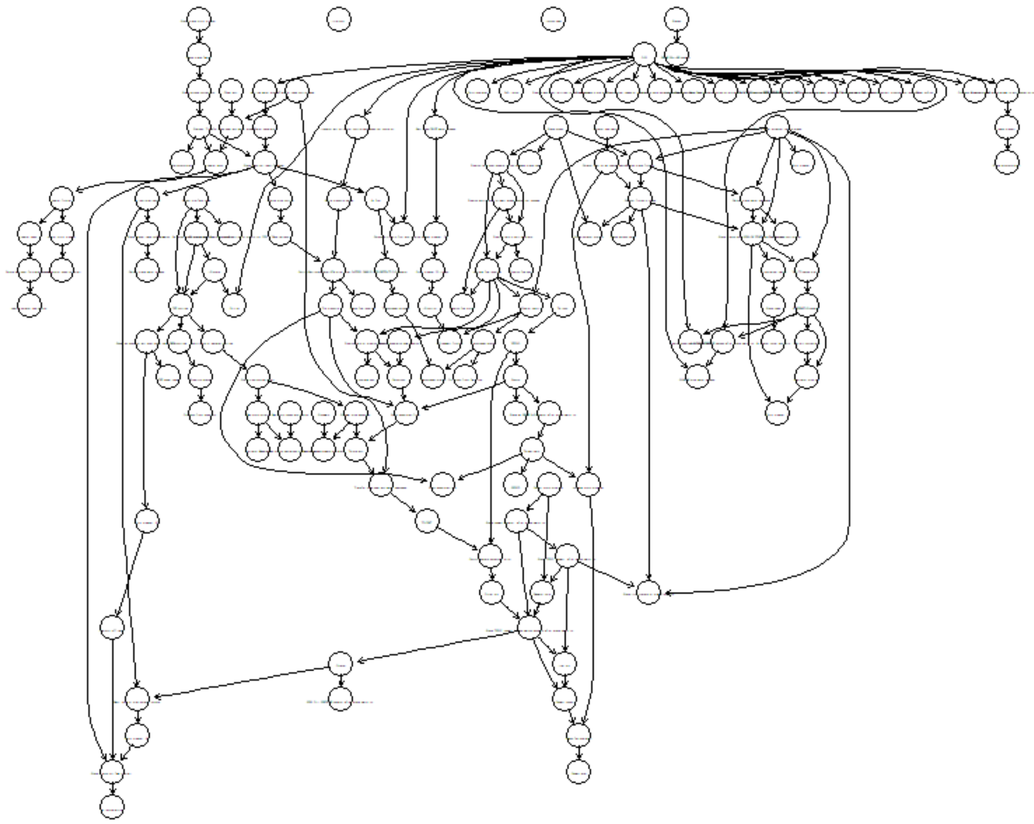


Figure 3.2. Learned network estimated from dataset

Given limits in computation, it is impossible for search algorithms to pick the best network from the set of all possible networks when every kth node has $k > 1$ parent nodes. This problem is classified as NP-hard and quickly becomes intractable as the number of nodes in a network increases (Heckerman 2020). The resulting estimated solution is an approximation. While clever extensions to search methods exist to mitigate this problem, when time and network size allows, the best solution may be supervised network quality control. For instance, when an estimated network has a manageable number of arcs, like the 185 in our learned network from Figure 3.2, it is possible for an expert to individually inspect each node and reverse the direction of the arc when dependency was detected but causation is attributed in the wrong direction. They may also decide to remove the arc completely if the relationship represented by the arc does not meet a subjective level of reason. This process was applied to our learned model, yielding a “hybrid” model structure depicted in Figure 3.3.

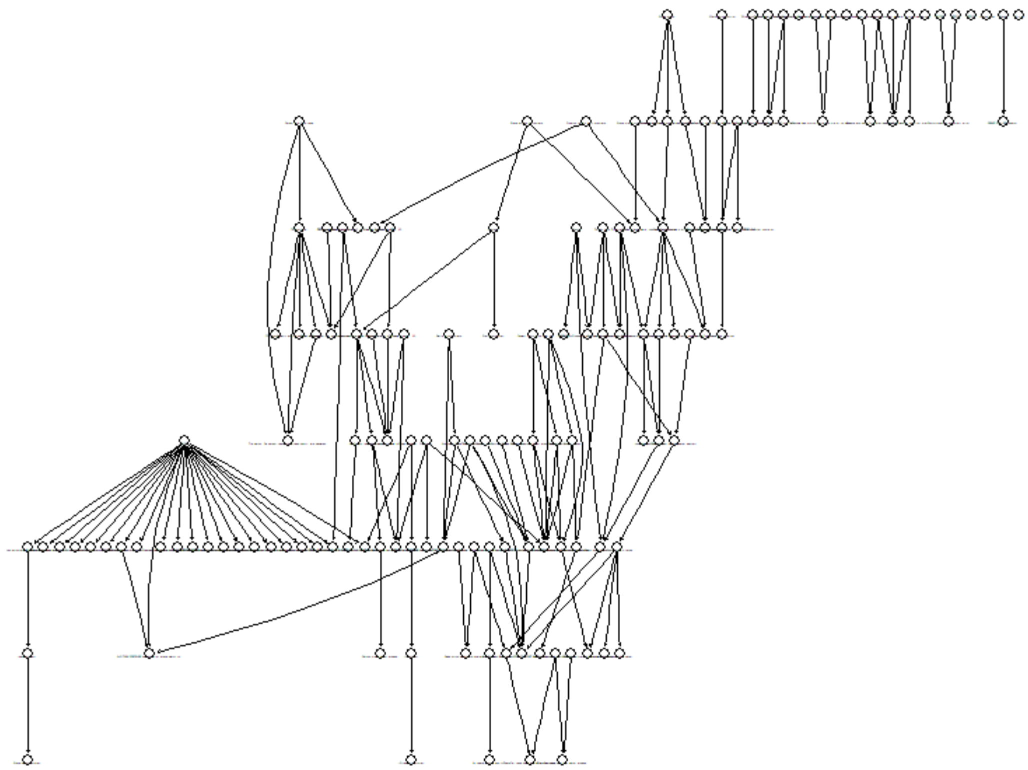


Figure 3.3. Hybrid network estimated from dataset and refined by human system expert

3.5 Model Inference

To query the network, the user may input known evidence into the system then analyze the updated CPTs to glean useful diagnostic decision support. Just as Figures 2.7 and 2.8 demonstrated conditional information flow in converging network structure, the directed separation algorithm determines the set of node CPTs that will be updated throughout the entire network given the input evidence. Approximate inference methods, like the Monte Carlo method implemented in the software used for this research, alleviate the NP-hard computational burden when querying the LAV model.

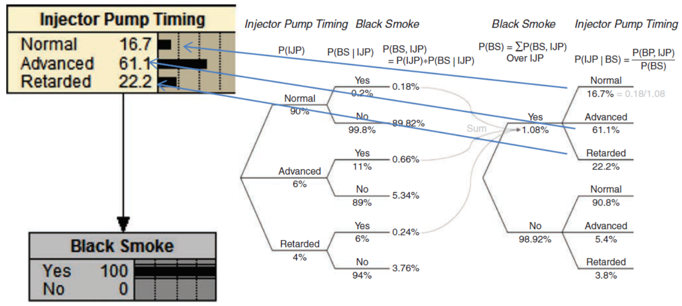


Figure 3.4. How Bayes' theorem and the chain rule of probability is used to determine exact inference in simple networks. Source: Aebischer et al. (2017).

For smaller models like the two-node network depicted in 3.4, it is possible to calculate exact inference using Bayes' theorem and the chain rule of probability.

3.6 Graphical User Interface

In addition to understanding and implementing model construction techniques and inference logic, the analyst must provide a medium for ease of providing model inputs, processing the information, and interpreting outputs. This will maximize model applicability and usability for the end user.

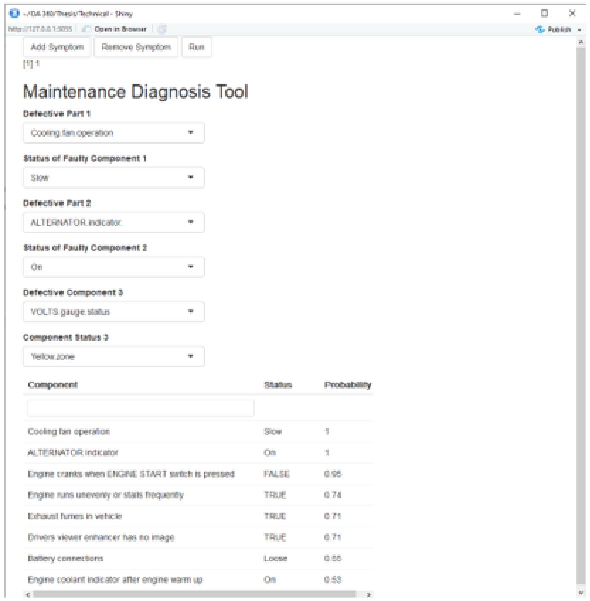


Figure 3.5. Diagnostic tool graphical user interface developed in Shiny

While model inputs and outputs can be expressed as code and programming language, we can demonstrate how easy-to-use graphical interfaces are utilized to support input and output visualization while software running in the background handles the logic and computation. Figure 3.5 is a screenshot of a graphical user interface (GUI) application implemented in the Shiny software package and embedded in a web browser that allows for ease in querying our expert model.

CHAPTER 4: Analysis

This chapter exhibits results of model analysis and presents techniques for model comparison and model selection. A user interface to support model query is also presented.

4.1 Model Comparison

Applying the various methods for determining network structure and model parameters renders a variety of resultant models. It is possible to determine the degree to which two models agree with one another by counting and classifying the specific arcs across both networks. We can select an incumbent network and compare it with a contending network. Specifically, we can count the number of true positive arcs, false positive arcs, and false negative arcs. True positives are arcs that appear in both networks. False positives appear in the contending network but not the incumbent. False negatives appear in the incumbent but not the contending (Scutari 2020). Figure 4.1 is a visual comparison of our expert and learned models. The red arcs represent false positives, and blue arcs represent false negatives. Of particular interest in this situation are the false positives, as they potentially indicate causal dependencies between nodes that are detected by the data but otherwise unknown to the expert. One particular false positive suggests a directional dependency relationship from "shocks" to "communications equipment error."

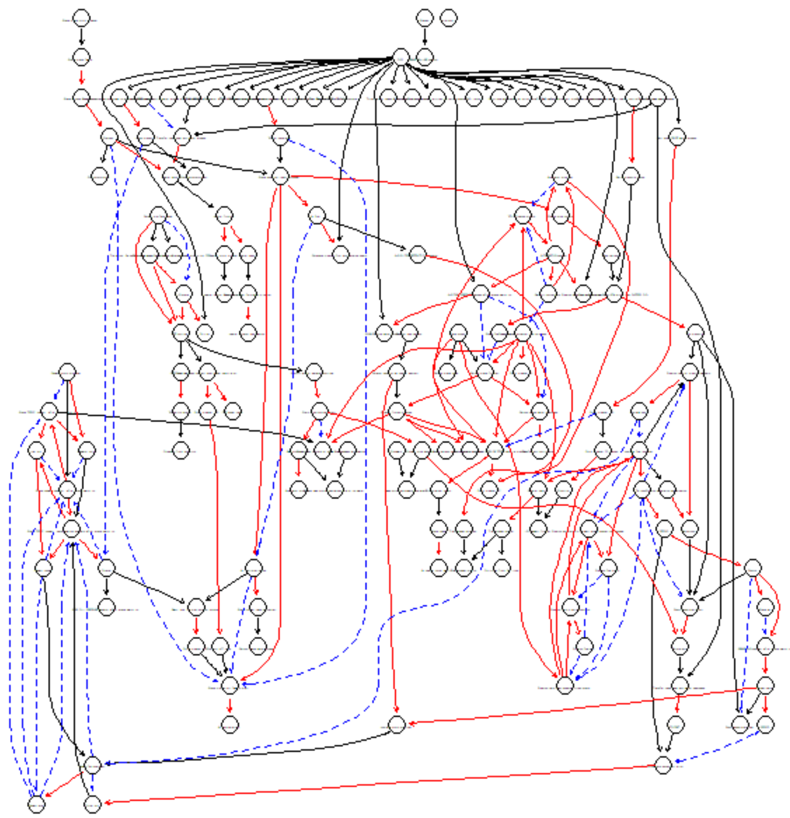


Figure 4.1. An overlay comparing our expert model with the learned model.

Table 4.1 conveys the similarities and differences in our models by the number of true positives, false positives and false negatives. We see more false positives than true positives when treating the expert model as the incumbent and comparing it to the the learned model. This tells us that there are a significant number of potential causal dependencies between components that are unknown to the subject matter expert or system technician. To investigate the arcs that fall in the false positive category, we look at the symmetric differences between the two models' sets of arcs. Inspecting these arcs, we see that there is a possible causal dependency between the turbo clamps being loose and the engine not starting, the status of the coil spring may affect the steering, and broken shocks can impact the communications equipment on the vehicle. For this last example, LAV operators generally understand their radios are more likely to cut out when riding over bumpy terrain, but this is neither intuitive nor explicitly addressed in the TM. Looking further down Table 4.1, we see our hybrid model agrees more with the initial expert design. Recall our hybrid model relies on human applied subjective reasoning, overriding instances where the machine found random or spurious correlation in the data.

Table 4.1. Comparing models by exploring the intersection of arc sets.

Incumbent	Contender	True Positives	False Positives	False Negatives
Expert	Learned	83	102	74
Learned	Hybrid	122	31	63
Expert	Hyrbid	102	51	55

4.2 Model Selection

4.2.1 Model Scoring

Given a model and a data set similar to the one in Table 3.1, we can compute a model score that is a goodness-of-fit indicator measuring how well our model reflects the dependence structure identified in our data (Scutari and Dënis 2015). This value can support model selection decisions when considering scaling our network. If a network score is calculated for both a sparse model and an augmented model, a quantifiable justification can be made for including or rejecting the augmenting nodes and arcs. Also, as potentially new data becomes available, a model can be scored against this new information to assess fit. Running a score for our three models by computing a Bayesian information criterion (BIC) value (where values closer to 0 indicate a better fit), we compute scores of -19,762.42, -18,132.95 and -19,098 for the expert, learned and hybrid models respectively. These results confirm the efficacy of the scoring method as the data used to compute the score was also used to create the learned model.

4.2.2 Cross Validation

Another way to evaluate model performance is through a cross validation holdout technique. To execute this technique, first split the data observations into two disjoint sets, a training and test set and estimate network structure using the training set. Next, use a loss function to measure its performance against the test set (Koller and Friedman 2009). Computing the expected loss for our three models using the log-likelihood loss function and the same input data used to construct our learned model, we compute values of 42.64, 42.04, and 43.99 for our expert, learned and hybrid models, respectively, where the lower value indicates a better fit. This suggests that although our hybrid model is loosely built from causal dependencies found in the data, cross validation using the log-likelihood loss function prefers selection of the expert model.

CHAPTER 5:
Summary and Conclusions

5.1 Summary of Results

The methods explored to design an automated tool for diagnostic support rendered limited models that demonstrate the utility of probabilistic graphical networks. The CBN is undoubtedly an apt representation of beliefs about system component causal dependencies that is useful in determining direct and indirect causes of malfunctions depending on presented symptoms. These beliefs can be determined by expert elicitation or data analysis, which form the two main approaches to CBN model construction. Coupled with node dependency algorithms and computation power, the resulting systems have the useful property of being easily queried for inference by inputting observed evidence. The outputs, updated CPTs for particular nodes of interest, can then direct the user to the likely source of the problem.

5.1.1 Reasoning in Expert Systems

CBNs may be constructed by eliciting expert knowledge. Three major components must be explicitly enumerated by the expert: the network nodes, network arcs, and respective node CPTs. The network nodes consist of the union of the set of system components and the set of all evidence that gives insight into the status of a component. The expert can then determine which nodes directly interact with one another and how they influence one another, yielding the set of network arcs and arc directions. The conditional probability table framework for each node is defined by the in-degree of that node and the associated levels of its parent nodes. The expert will assign probabilities that the node is in a given state for every combination of parent node states. The resulting CBN can then be queried for inference. This is a viable approach for constructing a Bayesian network for relatively simple networks where knowledge of the network is abundant and the expert's time is not too valuable to divert their attention. When these criteria are not met, a data driven solution becomes more appealing.

5.1.2 The Power of Data

It is not always practical or preferable to construct a CBN from expert elicitation. Given the powerful data driven methods and freely available software demonstrated in this research, the CBN programmer should consider if estimating a network from data is the preferred method. For instance, construction of a CBN representation of a fifth-generation fighter jet would require participation from numerous industry representatives and is likely too costly if not unrealistic. Also, the diminishing cost and abundance of accessible data means data driven CBN solutions may be the less cumbersome option. Our results tell us that models built from data provide feasible solutions for CBN programmers. For the sake of model accuracy, it is important to rely on a large enough quantity of data that is representative of system causal dependencies. Even when this data exists, we have seen how network estimation algorithms can often detect dependencies but fail to properly ascribe the cause and effect direction. Lastly, recall that networks learned from data allow the analyst the potential to discover direct and indirect causal dependencies that are otherwise unknown.

5.2 Recommendations

If the status quo for conducting higher echelon corrective maintenance does not translate to DMO and the future operational environment, then we must fully enable maintainers at the tactical unit level to independently diagnose and repair degraded systems. While much research has focused on automated supply chain operations in this future environment, a resupply that fails to deliver the correct Class IX supplies due to misdiagnosis costs time, resources, and most importantly, warfighting capability. The current approach outlined in the TM treats various pieces of fault evidence and information as independent from another, but when multiple symptoms are present, the whole of the information is likely more useful than the sum of the particular pieces. A rich CBN model is able to weigh input interaction effects on outputs, capitalizing on the synergy of dependency in information about system components. This research demonstrates how this logic can be implemented in software and uploaded onto a technician's handheld device. The Marine Corps should direct attention and resources to continue this research by partnering Marine Corps Systems Command (MCSC) as a topic sponsor with Naval Postgraduate School (NPS). Concurrently, investments can be made to modernize how we track system maintenance, which would have the additional benefit of digital data capture for the purpose of developing system specific CBNs.

5.2.1 Data Management

Upgrades in maintenance related data capture and management can provide an avenue for designing complex networks and continuously informing and improving the models with model comparison methods as new data becomes available. Currently, LAV maintenance records are transcribed by hand into maintenance logbooks. A digitized logbook would provide the perfect platform for capturing this data. A digital logbook would harness the power of computation by submitting to long-term memory all vehicle data and maintenance records. In addition to the typical function of a maintenance logbook, a digital maintenance logbook program installed on a tablet or toughbook could continuously update a CBN model as new data is constantly being input by multiple users across the service and allow the user to run queries for diagnostic support. Results can be input into a cost utility model that lays out optimal step-by-step troubleshooting steps. Photos and help tools could render along with the output, guiding the user through each performance step. Additionally, these enhanced logbooks would have the benefit of network connectivity where it can passively interact with Global Combat Support System (GCSS), streamlining the Class IX supply part replenishment process.

5.2.2 Program Improvement

Results from this research suggest that using CBNs to support equipment diagnostics has the potential to provide real benefits to material readiness and repair part budgets. The research presented satisfies much of the model verification burden, but much is yet to be gained by model validation. An enriched model approved by subject matter experts could be tested against current procedures to quantify the benefit of automated diagnostic tools. Once this head to head evaluation provides marginal measurements of performance improvement, these parameters can be used in a discrete event simulation (DES) similar to the one depicted in Figure 5.1. The probability of a misdiagnosis, $P(M)$, and the time to troubleshoot, t_T, can be captured and treated as systematically controlled independent variables, data can be collected over multiple iterations, and statistical analysis performed to determine measures of effectiveness such as mean number of operational vehicles and repair part expenditures over an operational cycle.

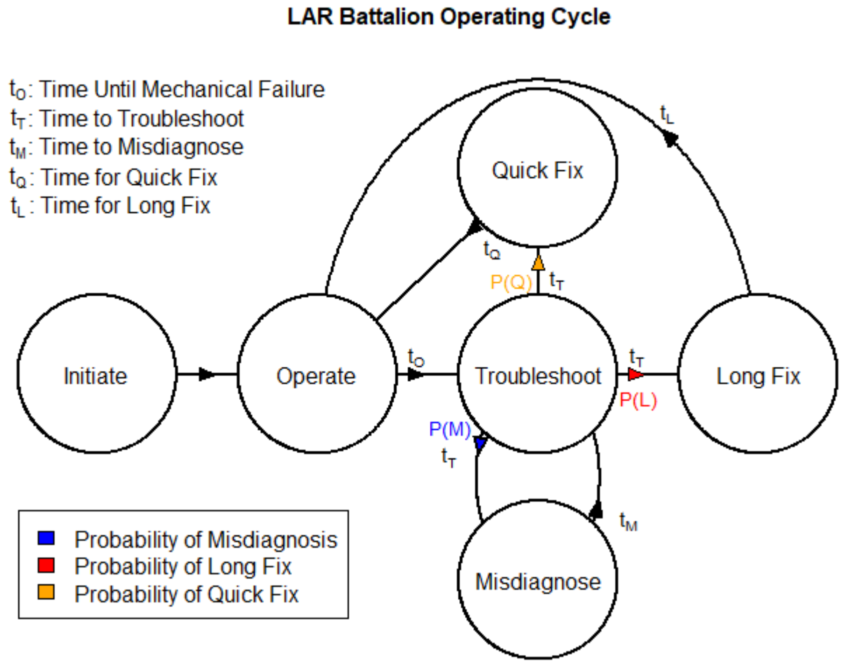


Figure 5.1. Discrete event graph representing the stages of an operational cycle for the LAV.

The results of this experiment would quantify the level of model effectiveness and potential return on investment, signaling the proper breadth and depth of investment in implementation across other systems maintained by the Marine Corps.

5.2.3 Program Implementation

For relatively simple, closed systems like an LAV, very minimal investments of time, resources and energy would be required to develop of a working model that could be fielded to LAV mechanics. The presented method of engineering a CBN from a TM can be used to build an initial working model to be improved upon after expert review and comparisons with data. The open-source software freely available to run these programs and develop a user interface are sufficient for this purpose. However, larger and more complex systems would likely require additional support from industry. The Marine Corps should designate specific systems similar in number of sub-components and complexity for initial implementation. After implementing and monitoring improvements in corrective maintenance turn-around, an incremental approach can be taken when designating additional platforms for method replication.

5.3 Areas for Future Research

Due to the author's limited access to real world data, a notional dataset was constructed that loosely represents real world causal dependencies between LAV system components. Researchers attempting to further this work should assess whether GCSS can provide input data for model construction. Otherwise, a concerted data collection effort could consist of inputting handwritten vehicle maintenance logs into a digital format. Also, having a way to use data to determine otherwise unknown conditional causal dependencies between components can inform research on using CBNs to determine where sensors should be placed on vehicles to detect potential symptoms of mechanical problems. Combining this with an understanding of which parent node component status is likely to trigger a malfunction presents an opportunity to develop of a scheme for preventing breakdowns and malfunctions before they occur. Additionally, undirected graphical representation of systems and Markov networks could potentially support a program for automated diagnostic support and should be explored further.

THIS PAGE INTENTIONALLY LEFT BLANK

List of References

Aebischer D, Vatterott Jr J, Grimes M, Jordan R, Reinso C, Baker B, Aldrich W, et al (2017) Bayesian networks for combat equipment diagnostics. *INFORMS Journal on Applied Analytics* 47(1):85–105.

Berger D (2019) The Commandant's planning guidance, 38th Commandant of the Marine Corps. Washington, DC, https://www.hqmc.marines.mil/Portals/142/Docs/

Breese J, Heckerman D, Kadie C (1999) Empirical analysis of predictive algorithms for collaborative filtering. *Proc. 14th Conference on Uncertainty in Artificial Intelligence (UAI)* 01(01):43–52.

Chow C, Liu C (1968) Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory* 14(3):462–467.

Davis R (1984) Diagnostic reasoning based on structure and behavior. *Artificial Intelligence* 24(01):347–410.

De Kleer J, Williams B (1986) Reasoning about multiple faults. *In AAAI* 01(01):132–139.

Fox D, Burgard W, Thrun S (1999) Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research* 11(01):391–427.

Genesereth M (1984) The use of design descriptions in automated diagnosis. *Artificial Intelligence* 24(01):411–436.

Heckerman D (2020) A tutorial on learning with Bayesian networks. *Innovations in Bayesian Networks* 01(01):1–40.

Kasparov G (2017) *Deep Thinking: Where Machine Intelligence Ends and Human Creativity Begins* (Public Affairs, New York).

Koller D, Fïiedman N (2009) *Probabilistic Graphical Models: Principles and Techniques* (The MIT Press, Cambridge, MA).

Lohr S (2013) David Ferrucci: Life after Watson. *The New York Times* (May 6).

Mattis J (2018) Summary of the national defense strategy of The United States of America. Https://dod.defense.gov/Portals/1/Documents/pubs/2018-National-Defense-Strategy-Summary.pdf.

Norman J, Shahar M, Kupperman M, Gold B (1998) Decision-theoretic analysis of prenatal testing strategies. Technical Report SMI-98-0711, Stanford University, Palo Alto, CA, section on Medical Informatics.

Pearl J (1988) *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference* (Morgan Kaufmann, San Mateo, CA).

Peng Y, Reggia J (1986) Plausibility of diagnostic hypotheses. *Proc. 2nd Conference on Artificial Intelligence (AAAI)* 01(01):140–145.

Sachs K, Perez O, Pe'er D, Lauffenburger D, Nolan G (2005) Causal protein signaling networks derived from multiparameter single-cell data. *Science* 308(5721):523–529.

Scutari M (2020) Bayesian networks and cross validation. Bnlearn. Accessed April 5, 2021, https://www.bnlearn.com/examples/xval/.

Scutari M, Dënis J (2015) *Bayesian Networks with Example in R* (CRC Press, Boca Raton, FL).

Simon H (1954) Spurious correlations: A causal interpretation. *Journal of American Statistical Association* 49(01):467–492.

Sutton C, McCallum A (2004) Selective segmentation and labeling of distant entities in information extraction. *ICML Workshop on Statistical Relational Learning and Its Connections to Other Fields* 01(01):01–01.

United States Marine Corps (2021) Tentative manual for expeditionary advanced base operations. Washington, DC, PCN 501 007704 00.

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California