Enhancing Fault Diagnosis in Cyber-Physical Systems: A Model-Based Approach with Bayesian Networks

MUSTAFA HAYRI BILGIN, University of Twente, The Netherlands

The increasing complexity of cyber-physical systems has made it challenging to maintain their serviceability and diagnose faults. Model based diagnosis and Bayesian networks have been researched to diagnose faults in these systems. This study focuses on the use of a hybrid approach combining consistency-based diagnosis and Bayesian Networks to improve fault detection and diagnosis for complex systems. A relatively simple model of a cyber-physical system will be built and subjected to different health status node configurations to determine the best diagnosis performance. The proposed approach is expected to enhance fault diagnosis and maintain serviceability of cyber-physical systems.

Additional Key Words and Phrases: Model-based diagnosis, Probabilistic inference, Bayesian networks, Cyber-physical systems, Consistency-based diagnosis

1 INTRODUCTION

Cyber-physical systems have become an essential part of our lives, ranging from critical infrastructure such as power grids and transportation systems to everyday devices like smartphones and home appliances. Ensuring the uptime and smooth operation of these systems is crucial to prevent disruptions. However, as these systems have become more complex, diagnosing faults and maintaining their serviceability has become increasingly challenging. With this complexity comes a higher probability of component faults, which can lead to a cascade of misbehaving parts [2] and make it difficult to identify the root cause of the issue.

Research in the field of fault diagnosis and system serviceability for cyber-physical systems has been ongoing for several years, with numerous studies addressing various aspects of the topic. This research will be mainly focusing on the area of model based diagnosis (MBD) and Bayesian network (BN). MBD involves using a model of a system or device's structure and behavior to identify the cause of malfunctioning [8]. One popular approach to MBD is consistency-based diagnosis (CBD), which utilizes knowledge of normal device behavior to diagnose faults. This method is particularly useful for troubleshooting novel systems where there is limited or no prior experience with faults. However, incorporating uncertainty into CBD can be challenging, even though uncertainty is often desirable in fault diagnosis because it can help identify the most probable cause of a problem. The use of BN would be ideal in this case as they are known for providing a framework for dealing with uncertainty in knowledge-based systems [5].

This research will examine a diagnostic methodology that relies on model-based techniques and probabilistic reasoning to diagnose faults in a cyber-physical system. By leveraging design knowledge and probabilistic reasoning, this approach aims to improve fault detection and diagnosis for complex systems. To achieve this, a theoretical model of a water pipes cyber-physical system is built using hybrid model-based approach that combines consistency-based diagnosis with Bayesian networks. The model will be subjected to different configurations, and the health status node configuration that yields the best diagnosis performance will be determined.

This paper consists of six sections. The second section states the problem and research question. In section three, background and relevant literature is analyzed. The fourth section describes the methods and steps taken to answer the research question. The fifth section provides the results obtained from these experiments. Lastly, the sixth section presents the conclusions drawn from the study and highlights potential directions for future research.

2 PROBLEM STATEMENT

Although in the past there has been quite some research on the theoretical underpinning of the field of Bayesian MBD [4, 8, 9] there is not much empirical research that clearly indicates which particular approach is best. This holds in particular for the various possible graph configurations of the health nodes in these models. Previous research by Barbini et al. [1] made particular assumptions about the way in which health nodes are positioned in the graph, although, without proper justification. In our research the impact of various health node configurations on the accuracy of diagnosing faults is being investigated for the first time.

Research Question

The problem statement will lead to the following research question:

- How should health status variables be represented in a Model Based Diagnosis Bayesian Network for monitoring and diagnosing faults in a cyber-physical system?

3 BACKGROUND

Fault diagnosis can be achieved through several methods. One approach involves identifying the failure modes of faulty components and tracing the system based on a given causal theory which is similar to the idea of abductive diagnosis (for fault diagnosis) [7]. Another approach is MBD, where only the normal behavior of the system's components and relations between them are modeled which is similar to CBD. This research will focus on the latter method as it is often not possible to know the failure modes of a novel system and it is quite difficult to know all the possible failure modes to diagnose to a single component in relatively large systems.

Several researchers have explored Bayesian Network as a method for fault diagnoses in cyber-physical systems [1, 6, 8]. Reiter [11] presented the first formal and precise description of consistency-based model-based diagnosis, which primarily discusses the logical structure of this diagnostic approach. A useful review and also input to this research was [1]. In that paper, Barbini et al. demonstrated

TScIT 39, July 7, 2023, Enschede, The Netherlands

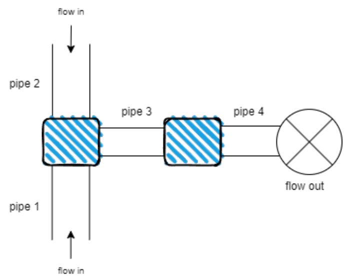


Fig. 1. A simple model of a pipe system

how to use model-based diagnosis to locate faults in complex cyber-physical systems. They achieved this by constructing diagnostic models utilizing Bayesian networks for diagnostic inference to deal with partial system observability.

However, despite the progress made in this field, none of this scientific research has investigated the impact of different health node configurations on Bayesian model-based diagnosis.

3.1 Model Based Diagnosis

Reiter [11] developed the notion of MBD to diagnose faults in a system by using first order logic. Essentially, a system consists of components ***CMP*** and a system description ***SD***. The former consists of a set of components of the system while the latter defines the normal behavior of the components in a set of first-order sentences. Each component within the system can have the following values: Normal, Abnormal. The Normal state refers to the case where a component is behaving as expected which can be found from the first-order logic in the ***SD***. The Abnormal state can be considered as the negation of the Normal behavior.

A diagnostic problem can be viewed as the combination of a system and a set of observations ***Obs***. When a system is behaving abnormally, a diagnosis ***D*** represents the minimal set of components that provides an explanation for the abnormal behavior and satisfies the following equation:

$$SD \cup Obs \cup \{Ab(c) \mid c \in D\} \cup \{\neg Ab(c) \mid c \in Comps - D\} \nvDash \bot \tag{1}$$

where $\bot$ represents falsum (or inconsistency) and $\nvDash$ represents the "does not prove" relation. Essentially, the expression is stating that the combination of sets on the left-hand side does not logically entail a contradiction.

3.2 Modeling

To visualize the idea of modeling in MBD, consider a relatively simple theoretical pipe system shown in Figure 1. This small system S consists of four pipes: pipe 1, pipe 2, pipe 3, and pipe 4. A flow meter is also attached to the end of the system which is the connection between pipe 4 indicating the end of the system.

Based on MBD, the system S consists of components CMP = {pipe 1, pipe 2, pipe 3, pipe 4} and a system description SD. Since each component is identical, their behavior is also expected to be similar. Each component has an input, output, and relations. The input and output are essentially variables and thus can have the following values: no flow, low flow, normal flow.

There exist relations between the components and these are represented using first-order logic. For instance, if the input is low flow, then the output can either be low flow or no flow logically. Besides the components of a system, we also have to consider the connections of components. These connections are the flow of the system in this case. Specifically, the output of a component is connected to the input of another component when two components are connected. For example, the output of pipe 1 and pipe 2 are connected to the input of pipe 3. These relations and connections make up the SD.

3.3 From MBD to BN

This paper implements MBD through the mapping of an SD of a system to a BN [10]. A BN is defined as a directed acyclic graph that represents the joint probability distribution of a group of random variables. Each node in the graph corresponds to a specific random variable and the connections between nodes depict causal dependencies among the variables. Additionally, a conditional probability table (CPT) is defined for each node in the BN.

There are multiple ways of translating an MBD to a BN problem and this paper will highlight three main approaches. The first approach discussed is the traditional method, providing a solid foundation for understanding the subsequent adaptations. The second approach represents an adaptation of the traditional method, introducing modifications to enhance its effectiveness. The third approach follows a similar structure to the second approach, further refining the representation of health nodes for components. It is important to note that each approach represents the configuration of health nodes of a component differently which allows to explore and find an answer to the research question of this paper.

Generally, a system can be translated to a BN using the following steps:

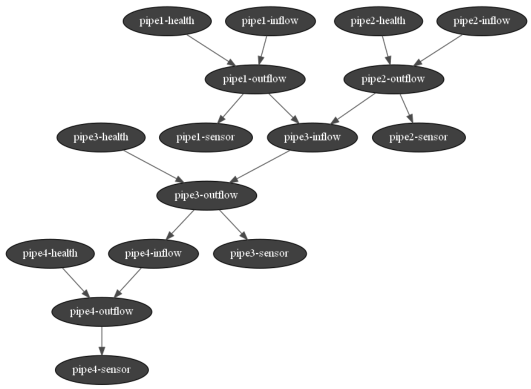
(1) Assign an individual input, output, and health variable for each component in the SD.
(2) Add a connection (edge) from the input to the output for each component.
(3) Create connections from the output of one component to the input of connected components within the system.
(4) The specific connections between the health node, input, and output variables vary depending on the chosen modeling approach.

It is assumed that the same probabilities can be taken for the same type of modeled component, which in many cases will be a justifiable simplification.

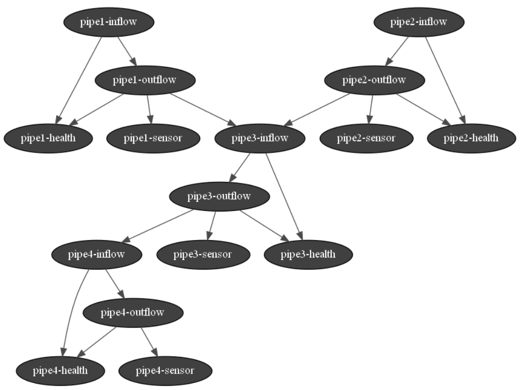
Note that it is also possible to add sensor nodes to a BN to represent the sensors in a system. For this research, it is assumed that each output node has a sensor attached to it in order to record the behaviour of the component.

The subsequent subsections present different approaches to configuring health nodes when translating MBD to BN, providing further details on each method.

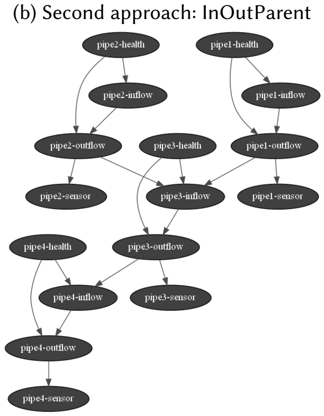
3.3.1 Traditional Approach. The traditional approach is based on the work of Srinivas [12]. Each component consists of an input,



(a) Traditional approach: HealthParentOut



(b) Second approach: InOutParent



(c) Third approach: HealthParentOut

Fig. 2. Different approaches of representing the pipe system in a BN.

output, and model. The input and output has the same description as explained in section 3.2. The model is essentially the health node and its value is determined by the behavior of the input and output.

In Figure 2a, the pipe system from Figure 1 has been translated into a BN following this method. As it can be seen, each pipe consists of inflow, outflow, and pipe health variable. The input (or inflow) has a connection to the output (or outflow) of a component. In addition, the health node has a connection to the outflow. For each connection of a pipe with another pipe, the output of the first pipe is linked to the input of the subsequent pipe. Based on inference logic, both the health and input node impact the output of a component. Moreover, the value of a component's input is influenced by the output of the connected pipes. In the later sections, this configuration will be referred to as "**HealthParentOut**".

3.3.2 Second Approach. The second approach shares several similarities with the traditional method. The overall structure remains the same, where each component possesses an input, output, and health variable. However, there is a distinction in the relationship between the health node and the input/output nodes. Figure 2b illustrates the translation of the pipe system from Figure 1 into a BN using this method.

In this approach, the health node can be considered dependent on the values of the input and output. Therefore, arrows connect from the input to the health node and from the output to the health node, respectively. This arrangement aligns logically since observations of the input and output implicitly provide information about the health of the component. For example, if the input exhibits expected behavior but an unexpected value is observed in the output, it may indicate a component fault. This type of relationship in the BN structure is well-suited for such inferences. Based on inference logic, the input and output variables serve as parents to the health variable, thereby influencing its assigned value. In the later sections, this configuration will be referred to as "**InOutParent**".

3.3.3 Third Approach. The third approach shares several similarities with the second approach, but with a key difference in the relationship between the health node and the input/output nodes. In this approach, the health node is considered the parent of the input and output nodes, indicating that the health node influences the values of both. Figure 2c illustrates the translation of the pipe system from Figure 1 into a BN using this method.

Logically, when a system has a problem or fault, it directly affects the values of the input and output variables. Therefore, the arrows in the BN structure go from the health node to the input and output nodes, reflecting this cause-and-effect relationship. This configuration aligns with the understanding that the health state of a component or system can impact the behavior and values observed in the input and output variables. Following the principles of inference logic, the health node's value influences the values of the input and output nodes. By considering the health node as the parent, any changes or issues in the health node propagate to affect the input and output variables accordingly. In the later sections, this configuration will be referred to as "**HealthParentOut**".

3.3.4 Forming a Diagnosis. After translating the MBD to a BN problem, the next crucial step is to define and find the diagnosis. This can be formed by using probabilistic inference and the calculation of posterior probabilities within the BN.

To form a diagnosis using probabilistic inference and the calculation of posterior probabilities of a model-based BN, the following formula can be used:

$$D = \{k \mid \text{leak} \equiv \text{argmax}_v P(\text{Health}_k = v \mid E), k = 1, \ldots, N\} \quad (2)$$

for N different components of the cyber-physical system. By applying equation (2), the diagnosis process evaluates whether probability of a "leak" state in each kth "pipe-health" node (Health_k) and selects the pipe or pipes with the highest probability. If multiple pipes have the same highest leak probability, all of them are included in the diagnosis.

4 METHODOLOGY

In this paper the pipe system from Figure 1 is used as the experimental framework. In order to perform diagnosis and evaluate different BN models, it is essential to have both the system and relevant data. Since it is not feasible to collect data from the system as this requires gathering data from a real cyber-physical system, synthetic data will be generated based on a logical algorithm.

For the implementation of the experiments, the programming language Python was chosen. Specifically, the pyAgrum library [3] was used to construct and evaluate various BN models.

The following subsections provide a comprehensive description of the data generation process, the system itself, and the different evaluation methods employed.

4.1 System Structure

The system from Figure 1 has been translated to different BN configurations, as shown in Figure 2. These configurations represent different types of modeling approaches for the BN, while the components remain the same, with variations in the network relationships.

Once these configurations were constructed using the pyAgrum library, the next step involved assigning conditional probability tables (CPTs) for each node in the BN. There were two options for assigning the CPTs: manual assignment based on system knowledge or learning the CPTs from the synthetic data. For this research, a hybrid approach was chosen. Initially, the CPTs for each BN configuration were learned from the data. Subsequently, manual adjustments were made to the CPTs, except for the pipe-health nodes. Learning the CPTs for the health pipes from the data was preferred, as adding them manually would have been challenging without prior knowledge of the system's health status. Given that the primary research objective is to identify the most effective health node configuration, the manually adjusted CPTs were designed to be very similar across configurations. This ensured that the diagnosis results were primarily influenced by the health node configurations rather than the specific CPT values. Figure 3 contains an example CPT for the "HealthParentBoth" configuration.

It is worth noting that each pipe's outflow is connected to a sensor. This deliberate choice enables the use of inference without interfering with the inferential logic when providing evidence to identify the diagnosis. While additional sensors could be attached to other components, this research specifically adopted this approach.

4.2 Data Generation

A data generator algorithm was built to generate synthetic data for a theoretical pipe system. The functions built are modular to ensure that it adheres to any pipe system that follows a similar structure to the pipe system in Figure 1. By providing the names of the pipes and

		pipe3-outflow		
		lowFlow	noFlow	normalFlow
pipe3-health	pipe3-inflow			
leak	lowFlow	0.05	0.95	0.0
	noFlow	0.00	1.00	0.0
	normalFlow	0.98	0.02	0.0
normal	lowFlow	1.00	0.00	0.0
	noFlow	0.00	1.00	0.0
	normalFlow	0.00	0.00	1.0

Fig. 3. CPT of pipe3-outflow for the **HealthParentBoth** configuration

their corresponding relations (connections), the algorithm handles the rest of the data generation process.

Each pipe in the generated data consists of three main columns: *pipeX-inflow*, *pipeX-outflow*, and *pipeX-health*, where X denotes the pipe number. The inflow and outflow columns can have the values: *noFlow*, *lowFlow*, or *normalFlow*, while the *pipe-health* column can take either the value *normal* or *leak*.

Overall, the data generator makes use of logical probabilities to assign values for different states that a component can take. For example, assuming that 90% of the pipes are expected to function normally, the generated data reflects this probability distribution. Various factors influence the assignment of values to specific components based on the system's structure and behavior. The pseudo code for the data generator can be found in Algorithm 1. The logical flow for assigning values to each component is described below:

- *pipe-inflow*: The value of this component is determined by two factors: the pipe's own health status (*pipe-health*) and its connection to other pipes. This logical approach acknowledges that the health of the pipe can impact the inflow. Additionally, even if the pipe is in normal health, the inflow can be influenced by the outflow values from other connected pipes. For example, if pipe 1 and pipe 2 are connected, and pipe 1 has *noFlow*, then pipe 2's inflow will also be *noFlow*, regardless of the health of pipe 2.
- *pipe-outflow*: The value of this component is determined by two factors: the inflow of the current pipe (*pipe-inflow*) and the health status of the current pipe (*pipe-health*). Specifically, the value of the pipe's outflow can either match or be lower than its inflow value. Similarly to the *pipe-inflow* case, the health status of the pipe impacts the outflow value.
- *pipe-health*: This value is randomly determined based on a given probability. For the generated data in this case, it was assumed that 90% of the components would function properly.

A very important note is the conservation of water in pipes. If the *pipe-inflow* of a pipe has the value *lowFlow*, then the outflow can only be *lowFlow* or *noFlow*, as the flow cannot increase in a pipe but can only stay the same or decrease.

By taking into account these logical principles, as well as additional conditions within the code, the synthetic data was generated.

Algorithm 1 Random Pipe Generator

```
function DATA_GENERATOR(pipes, relations)
    for pipe in pipes do
        pipe_health = randomly select 'normal' or 'leak'
        if pipe is the first pipe then
            pipe_inflow = randomly select from ['noFlow', 'lowFlow', 'normalFlow'] based on pipe_health
        else
            find connected_outflows based on pipe relations
            pipe_inflow = randomly select from ['noFlow', 'lowFlow', 'normalFlow'] based on connected_outflows and pipe_health
        end if
        pipe_outflow = randomly select from ['noFlow', 'lowFlow', 'normalFlow'] based on pipe_inflow and pipe_health
    end for
end function
```

4.3 Evaluation Methods

4.3.1 Cartesian Product of Evidences. In order to gain an understanding of all potential diagnoses that can be generated by a particular model, it is possible to generate and test every unique combination of evidences to formulate a diagnosis. The Cartesian Product can be used to obtain all possible combinations of evidence. This method involves considering all possible combinations of the source pipes (Sp), sensor nodes (Sn), and possible values (Pv). The formula used for this method is as follows:

$$\mathrm{E} = \{c \in (\mathrm{Sp} \times \mathcal{P}(\mathrm{Sn}) \times \mathcal{P}(\mathrm{Pv})) \mid \forall \mathrm{sp} \in \mathrm{Sp} : \mathrm{sp} = \text{'normalFlow'}\} \tag{3}$$

The above formula ensures that each combination in the set of evidences includes all the source pipes with the value 'normalFlow', while allowing the sensor nodes to take any value from the possible values set. Additionally, it accounts for scenarios where not all sensor nodes are present, as long as all the source pipes are included.

The adaptation of the Cartesian Product of Evidences and the creation of this formula were necessary to consider logical and appropriate diagnosis that can effectively represent real-life evidences. By incorporating the points below, the formula provides a realistic representation of the evidential scenarios:

- In all the combinations, all the source pipes must be present. Moreover, we are only interested in the cases where the flow is normal from the source pipes. This is because diagnosing faults of the source pipes can be achieved through a simple observation of the system.
- The combinations also include cases where not all sensor nodes are present, while ensuring that all source pipes remain present.

Note that the code which generates these combinations take into account additional factors. For instance, if a pipe is connected to another pipe, then it is not possible to have an evidence combination where the former pipe sensor detects no flow while the latter pipe sensor detects low flow or normal flow.

4.3.2 Accuracy. A BN model can form a diagnosis by following the steps from Section 3.3.4. Once the BN selects its diagnosis, which may consist of one or multiple pipes, it becomes essential to evaluate the accuracy of this diagnosis. This evaluation involves examining each data point from the synthetic dataset that adheres to the evidence criteria.

The accuracy of the diagnosis can be defined as follows:

$$\text{Accuracy} = \frac{\text{Number of Correct Diagnoses}}{\text{Number of Rows that Meet the Evidence Criteria}} \tag{4}$$

4.3.3 Incalculable Inference. The concept of incalculable inference in Bayesian networks refers to situations where the network fails to compute inference due to a mismatch between the provided evidence and the network's structure. This negatively impacts the evaluation of the BN model, as it suggests a potential mismatch between the model and the observed data.

4.3.4 Diagnosis Probability Log Likelihood. The Diagnosis Probability Log Likelihood (DPLL) provides an objective measure of the BN model's performance in capturing the true diagnosis across different combinations of evidence. Essentially, the DPLL measures the cumulative log-likelihood of the correct diagnosis given different combinations of evidence. The idea of this methodology is described as follows:

(1) Logical Generation of Evidences: All logically possible combinations of evidence are generated using the formula described in Section 4.3.1.
(2) Determining the Correct Diagnosis: For each combination of evidence, the correct diagnosis is determined by analyzing the synthetic data. The observed evidence is compared with the actual data to identify the diagnosis that aligns with the ground truth.
(3) Performing Inference: Inference is performed on the BN model using each combination of evidence. This step allows for the computation of probabilities associated with the correct diagnoses given the observed evidence.
(4) Logarithmic Transformation: Instead of directly summing the probabilities of the correct diagnosis across all combinations, the logarithm of the leak probability for each combination is taken. This transformation is motivated by the following reasons:
   (a) Numerical Stability: The probabilities of individual combinations can vary significantly, ranging from very small to very large values. Summing these probabilities directly may result in numerical instability or loss of precision.
   (b) Comparative Analysis: The logarithmic transformation allows for easier comparison of results across different combinations and datasets.
(5) Calculation of DPLL: The DPLL is calculated by summing the logarithms of the leak probabilities for each combination of evidence. The formula is as follows:

$$\text{Diagnosis Probability Log Likelihood} = \sum_{i=1}^{N} \log p_i \tag{5}$$

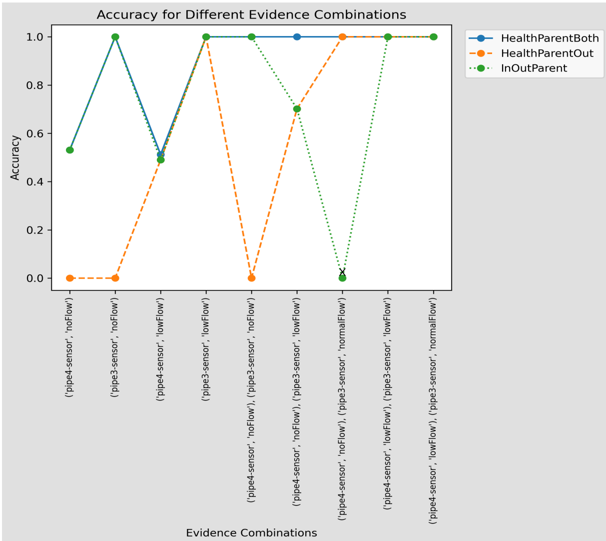


Fig. 4. Line-chart: Accuracy for different Evidence Combinations

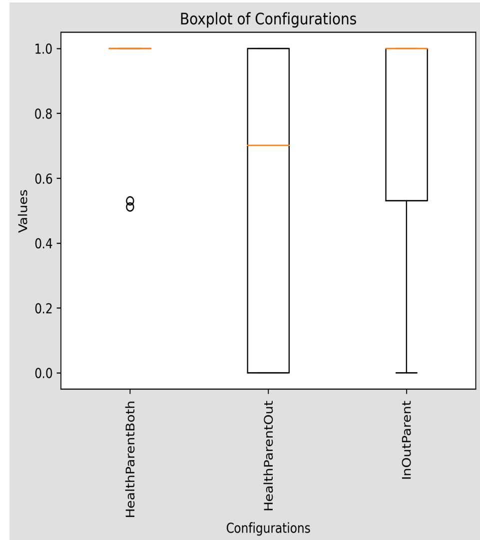


Fig. 5. Box-plot: Accuracy information

where p_i represents the leak probability of the correct diagnosis for each combination of evidence and N is the total number of combination of evidences.

Note that certain BN models cannot compute the inference of certain combinations of evidence, as explained in Section 4.3.3. The number of such incalculable inferences is considered alongside the DPLL, as these factors mutually influence each other's values.

4.4 Experiments

In order to address the research question, a series of experiments will be conducted to test the three BN configurations using various methodologies. The accuracy and number of incalculable inferences will be evaluated for each configuration. The accuracy assessment will be based on synthetic data consisting of 200,000 rows.

The experiments will involve applying the Cartesian Product of Evidences to each BN configuration. This approach will allow to compare the accuracy of the configurations across different evidences. By testing the configurations with all logical combination of evidences, the relative performance of each configuration will be analyzed, and any patterns or variations in accuracy will be identified.

Additionally, the number of incalculable inferences will be determined for each configuration. This metric provides insight into the robustness and reliability of the BN configurations in handling different types of evidences. Moreover, it indirectly measures how well the BN structure fits the data structure.

5 RESULTS

The results of the experiments are presented in this section, including a box plot, a line chart, and a table summarizing the performance of different BN configurations.

5.1 Box plot Analysis

The box plot from Figure 5 provides a visual summary of the distribution of accuracy values across different BN configurations. Analyzing key statistics such as the minimum accuracy, maximum accuracy, median accuracy, interquartile range (IQR), upper quartile, and lower quartile allows for insights into the performance of each BN configuration.

The "HealthParentBoth" configuration has the highest minimum accuracy value of 0.512, indicating that even its lowest accuracy values are relatively high compared to the other configurations. On the other hand, the "HealthParentOut" and "InOutParent" configurations both have minimum accuracy values of 0.0.

When considering the maximum accuracy, all three configurations achieve a maximum accuracy of 1.0, suggesting that they can reach the highest level of accuracy for some combinations of evidence. However, it is important to note that this value alone does not provide a comprehensive comparison.

Analyzing the median accuracy, the "HealthParentBoth" and "InOutParent" configurations stand out with a median accuracy of 1.0, indicating a consistently high performance compared to the "HealthParentOut" configuration, which has a median value of 0.702.

When examining the IQR, it can clearly be seen that "HealthParentOut" has the highest IQR. A wider variability in accuracy values suggests that the performance is likely to be more dependent on the specific combinations of evidence than overall structure.

In terms of the upper quartile, all three configurations have the same accuracy value of 1.0, indicating a consistent performance for the top 25% of accuracy values across the configurations. However, for the lower quartile accuracy, the "HealthParentBoth" configuration has the highest value of 1.

In summary, the boxplot analysis reveals that the "HealthParentBoth" configuration generally exhibits higher accuracy based on the median, minimum, and maximum values. However, it is essential to

Table 1. General table

Configuration	Average Accuracy	Incalculable inference	Diagnosis Probability Log Likelihood Sum
HealthParentBoth	0.894	0	-1.315
HealthParentOut	0.577	0	-6.984
InOutParent	0.747	1	-9.579

consider the wider spread of accuracy values and the performance of the "InOutParent" configuration. The comparison and performance of these two configurations will be more evident in the Line Chart analysis below.

5.2 Line chart Analysis

The line chart from Figure 4 provides a visual representation of the accuracy values for different BN configurations across various combinations of evidence. Each line on the chart represents a different BN configuration, and the points on the line indicate the corresponding accuracy values. Analyzing the line chart allows for observation of performance trends for each BN configuration and identification of patterns based on the accuracy values.

The "HealthParentBoth" configuration consistently demonstrates high accuracy values across different combinations of evidence, with the majority of data points reaching a value of 1.0. This suggests a robust and reliable performance for this configuration, regardless of the specific evidence used.

The "HealthParentOut" configuration exhibits varying accuracy values across the combinations of evidence. This configuration demonstrates an intriguing pattern where the accuracy is either consistently high (1.0) or consistently low (0) for the majority of evidence combinations. This behavior indicates that the performance of this configuration is highly dependent on the specific combinations of evidence used.

For the "InOutParent" configuration, the accuracy values vary the most across different combinations of evidence. While most data points indicate high accuracy (1.0), there is one data point marked as '**X**' (incalculable inference), which is interpreted as 0. Similar to the "HealthParentOut" configuration, this configuration might be influenced by specific combinations of evidence, leading to lower accuracy in certain cases.

5.3 Table Analysis

The table 1 presents key metrics for each BN configuration, including average accuracy, the number of incalculable inferences, and the diagnosis probability log likelihood sum.

The "HealthParentBoth" configuration achieves an average accuracy of 0.894, indicating the highest overall performance. Notably, this configuration does not have any incalculable inferences, suggesting its robustness in handling various evidence combinations and its suitability with the structure of synthetic data. Additionally, this configuration has the lowest diagnosis probability log likelihood sum (-1.315), indicating the most favorable estimation of probabilities for all possible evidence combinations.

In comparison, the "HealthParentOut" configuration demonstrates a lower average accuracy of 0.577. Similar to the "HealthParentBoth" configuration, it does not have any incalculable inferences. However, the diagnosis probability log likelihood sum for this configuration is -6.984, indicating a less accurate estimation of probabilities compared to the "HealthParentBoth" configuration.

Lastly, the "InOutParent" configuration achieves an average accuracy of 0.747. It has one incalculable inference, suggesting some limitations in handling specific evidence combinations. Although it has a higher average accuracy than "HealthParentOut", the diagnosis probability log likelihood sum for this configuration is lower (-9.579), indicating a less accurate estimation of probabilities.

6 DISCUSSIONS AND CONCLUSIONS

In conclusion, this research investigated the representation of health status variables in a Model Based Diagnosis Bayesian Network for fault monitoring and diagnosis in a cyber-physical system. The findings demonstrated the impact of different health state configurations on fault diagnosis accuracy.

The "HealthParentBoth" configuration exhibited the highest overall performance, showcasing its robustness and suitability for handling various evidence combinations. The "HealthParentOut" configuration provided insights into the sensitivity of accuracy to specific evidence combinations, while the "InOutParent" configuration highlighted the trade-off between accuracy and limitations in handling certain evidence combinations. By understanding the implications of these findings, practitioners can make informed decisions when selecting health state configurations for fault diagnosis in cyber-physical systems, improving the accuracy and reliability of the diagnostic process.

It is important to note that these results are specific to the system and data used in this study. Therefore, further experimentation with different types of cyber-physical systems is necessary to generalize this observation. Moreover, future research can build upon these findings by refining assumptions, exploring alternative representations, and investigating additional factors to enhance fault diagnosis accuracy. One interesting area of research for the future is to examine how the addition of more sensor nodes in the system, thereby expanding the available evidence, can impact the accuracy of fault diagnosis. Understanding the relationship between the number of sensor nodes and the diagnostic accuracy can provide valuable insights into the optimal deployment of sensors in cyber-physical systems. Furthermore, investigating the effects of different types of evidence and their relative importance in the diagnostic process can contribute to the development of more accurate and reliable fault diagnosis methods. By addressing these aspects, future research can further advance the field and pave the way for more effective fault diagnosis strategies in cyber-physical systems.

REFERENCES

[1] Leonardo Barbini, Carmen Bratosin, and Emile van Gerwen. 2020. Model based diagnosis in complex industrial systems: a methodology. *PHM Society European Conference* 5, 1 (July 2020), 8–8. https://doi.org/10.36001/phme.2020.v5i1.1174 Number: 1.

[2] Sergey V. Buldyrev, Roni Parshani, Gerald Paul, H. Eugene Stanley, and Shlomo Havlin. 2010. Catastrophic cascade of failures in interdependent networks. *Nature* 464, 7291 (April 2010), 1025–1028. https://doi.org/10.1038/nature08932 arXiv:0907.1182 [cond-mat].

[3] Gaspard Ducamp, Christophe Gonzales, and Pierre-Henri Wuillemin. 2020. aGrUM/pyAgrum : a toolbox to build models and algorithms for Probabilistic Graphical Models in Python. https://www.semanticscholar.org/paper/aGrUM-pyAgrum-%3A-a-toolbox-to-build-models-and-for-Ducamp-Gonzales/1b7b7b4d8abe5b7e1c122150fffb4220e5b015d9

[4] I. Flesch, P.J.F. Lucas, and Th.P. van der Weide. 2007. Conflict-based diagnosis: adding uncertainty to model-based diagnosis. In *Proceedings of IJCAI-2007*. Morgan-Kaufman, San Francisco, 380–388.

[5] David Heckerman, Abe Mamdani, and Michael P. Wellman. 1995. Real-world applications of Bayesian networks. *Commun. ACM* 38, 3 (March 1995), 24–26. https://doi.org/10.1145/203330.203334

[6] Zhen Huang, Cheng Wang, Milos Stojmenovic, and Amiya Nayak. 2015. Characterization of Cascading Failures in Interdependent Cyber-Physical Systems. *IEEE Trans. Comput.* 64, 8 (Aug. 2015), 2158–2168. https://doi.org/10.1109/TC.2014.2360537

[7] Peter J. Lucas. 1997. Symbolic diagnosis and its formalisation. *The Knowledge Engineering Review* 12 (June 1997). https://doi.org/10.1017/S0269888997002026

[8] Peter J. F. Lucas. 2001. Bayesian model-based diagnosis. *International Journal of Approximate Reasoning* 27, 2 (Aug. 2001), 99–119. https://doi.org/10.1016/S0888-613X(01)00036-6 Publisher: Elsevier.

[9] Judea Pearl. 1988. . Morgan Kaufmann, San Francisco (CA).

[10] Judea Pearl. 1988. Chapter 3 - MARKOV AND BAYESIAN NETWORKS: Two Graphical Representations of Probabilistic Knowledge. In *Probabilistic Reasoning in Intelligent Systems*, Judea Pearl (Ed.). Morgan Kaufmann, San Francisco (CA), 77–141. https://doi.org/10.1016/B978-0-08-051489-5.50009-6

[11] Raymond Reiter. 1987. A theory of diagnosis from first principles. *Artificial Intelligence* 32, 1 (April 1987), 57–95. https://doi.org/10.1016/0004-3702(87)90062-2

[12] Sampath Srinivas. 1994. A probabilistic approach to hierarchical model-based diagnosis. In *Uncertainty Proceedings 1994*, Ramon Lopez de Mantaras and David Poole (Eds.). Morgan Kaufmann, San Francisco (CA), 538–545. https://doi.org/10.1016/B978-1-55860-332-5.50073-0