Fusion Assessment of Safety and Security for Intelligent Industrial Unmanned Systems

Rongyao Cai*
Institute of Cyber-Systems and Control
Zhejiang University
Hangzhou, China

Xiao Xv*
Institute of Industrial Technology Research
Zhejiang University
Hangzhou, China

Zhengming Lu
Institute of Cyber-Systems and Control
Zhejiang University
Hangzhou, China

Kexin Zhang
Institute of Cyber-Systems and Control
Zhejiang University
Hangzhou, China

Yong Liu†
Institute of Cyber-Systems and Control
Zhejiang University
Hangzhou, China

2024 7th International Symposium on Autonomous Systems (ISAS) | 979-8-3503-6317-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/ISAS61044.2024.10552597

***Abstract*—Fault tree analysis is the most commonly used methodology in industrial safety analysis to predict the probability or frequency of system failure. Although fault tree analysis has been proposed for more than six decades, the assumptions used in most commercial fault tree analysis codes have not changed significantly, which limits the ability of the method to represent design, operation, and maintenance characteristics in the context of the increasing complexity and specialization of modern industrial systems. The basic setup of traditional fault trees is unable to include dependencies between events, time-varying failures, and repair rate realities to explain complex maintenance strategies. To address the above shortcomings, we propose a fusion tree model combining fault tree and attack tree, and simplify the causal structure of the fusion tree by modularization, and utilize the dynamic Markov model to represent the complex coupling relationship between components or nodes. Finally, we demonstrate the calculation process of fusion tree in pressure vessel systems with temporal control.**

***Index Terms*—Fault tree analysis, Attack tree, Binary decision diagrams, Dynamic markov models**

I. Introduction

With the complexity of industrial process systems, the number of operating devices and network control nodes in the system has increased dramatically, and at the same time, there are complex topological causal structures and information conduction paths between the devices and nodes, which leads to the challenge of assessing and tracing the risk of failure of the system as a whole. When the system fails due to equipment failure or network control node attack, it is difficult to ensure the normal operation of the system and the economic benefits of the enterprise through equipment maintenance inventory or network traffic analysis to trace the risk after the fact. Based on this, online risk assessment and risk traceability has become an extremely important technical issue.

Traditional fault tree analysis as the major method that structured analysis the architecture of systems and fault conduction path has been greatly developed in recenter years [1], [2]. Traditional fault tree analysis is consisted with two stages. The first stage provides the minimum cut set, the list of necessary and sufficient basic events that give rise to the top events. The second stage quantifies system failure modes, top events, probabilities, or frequencies. Calculations of importance metrics can also be performed to determine the contribution of each component or minimum cut set to system failure modes, identifying weaknesses that can be addressed to improve system performance.

The fault tree focuses on the conduction path of the fault through the system. For specific sub-causes that trigger a fault, the attack tree [3], [4] is needed to specifically analyze the operating rules within the macro-unit. The attack tree decomposes the realization process of the attack into multiple necessary sub-conditions or steps for the identified attack target, and then constructs a complete attack tree model and performs security analysis based on the sub-conditions or steps.

Existing attack tree models focus more on the inter-device risk conduction process and the static risk evolution process in terms of system safety. At the same time, the fault tree model only analyzes the probability of failure of the device itself, but does not consider the possible controller network attacks that trigger the failure. The above problems lead to the traditional fault tree can not effectively deal with the modern complex industrial systems commonly exist in the parameters of time-varying, strong coupling of equipment and other issues. Moreover, with the complexity of industry, constructing a fault tree for the whole large system has the problems of chaotic system topology and causal structure, time-consuming risk calculation, difficult risk traceability, and difficult to effectively monitor the safety of the system.

In this paper, in order to solve the problems of poor time-variation, poor coupling relationship between describing device nodes, and lack of information security monitoring in the traditional risk assessment and traceability methods of industrial systems, an integrated risk assessment method of

This work was supported in part by the National Key R&D Program of China under Grant 2021YFB2012300.

* Equal contribution

†Corresponding author: Yong Liu (yongliu@iipc.zju.edu.cn).

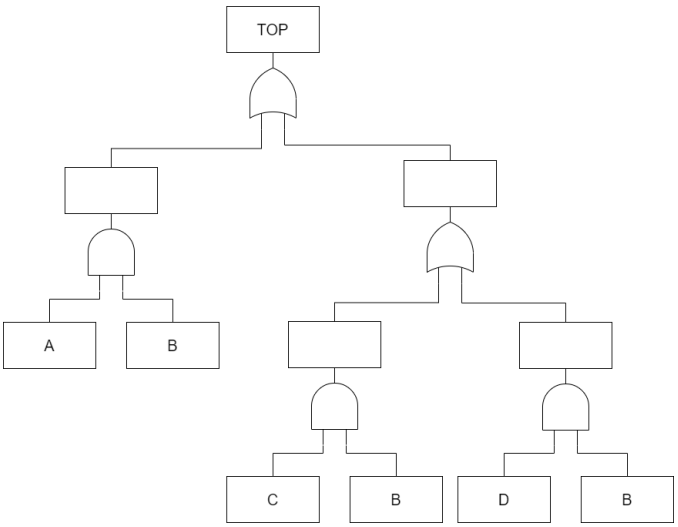

Fig. 1. Fault Tree

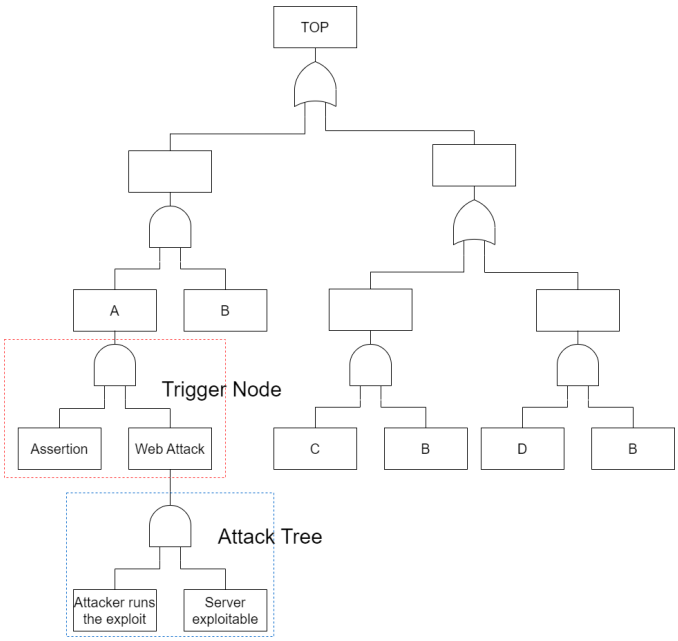

Fig. 2. Fusion Tree

functional safety and information security based on the fusion tree model is proposed. The major contributions are as follows:

- A novel fusion tree model is proposed by combining fault tree and attack tree to consider functional safety and information security together.
- Modularization are employed to merge fusion tree events, simplifying the topology and enhancing the computational efficiency of top event of fusion tree.
- Dynamic Markov models are introduced into fusion tree models to represent complex coupling relationships between nodes, giving the model dynamic properties.

II. Methodologies

A. Fault Tree Analysis

Fault trees are structured to represent the causality of system failure modes through combinations of AND and OR logic gates for component failure types, human errors, etc. [5], as illustrated in Fig. 1. Kinetic Tree Theory (KTT) [6] performs fault tree analysis in two phases. The qualitative phase forms a Boolean equation that represents the cause of the top event TOP in terms of basic events. This is then processed into an analytic paradigm (generating a minimum sum of forms):

$$TOP = C_1 + C_2 + \cdots + C_{N_c} \tag{1}$$

where C_i is the minimal sets, $i = 1, \ldots, N_c$ which can be extracted [7], TOP is the top event. C_i term is the sum of the basic event variables:

$$C_i = X_1 \cdot X_2 \cdot X_3 \ldots X_{N_{C_i}} \tag{2}$$

where X_j is the event variable, $j = 1, \ldots, N_{C_i}$, operator $\cdot$ means disjunction (OR) or conjunction (AND) relationship between X_j.

Based on the minimum cut set and the probability and frequency of failure of the component, the top event probability Q_{SYS} can be calculated

X_j in actual system is usual the basic component which consisted with the whole systems. In most commercial fault tree tools, it is assumed that the component experiences a continuous rate of failures and repairs. Typical models used to assess the probability of component failure based on component maintenance are:

For non-repairable equipment:

$$Q(t) = 1 - e^{-\lambda t}, \tag{3}$$

For unscheduled maintenance of equipment:

$$Q(t) = \frac{\lambda}{\lambda + \nu}(1 - e^{-(\lambda+\nu)t}), \tag{4}$$

For schedule maintenance equipment for regular maintenance:

$$Q_{Av} = 1 - \frac{(1 - e^{-\lambda\theta})}{-\lambda\theta}. \tag{5}$$

where Q sit the fault probability of component, Q_{Av} is the average component fault probability in running, λ is the constant failure rate, ν is the constant repair rate, θ is the inspection interval and t is the time.

Logic gates (OR and AND) represent the causal relationship between root nodes and leaf nodes and determine the operation between X_j. The specific rules are as follow:

$$\mathrm{OR}(\mathrm{X_a}, \mathrm{X_b}) = \mathrm{P}(\mathrm{X_a}) + \mathrm{P}(\mathrm{X_b}) \tag{6}$$

$$\mathrm{AND}(\mathrm{X_a}, \mathrm{X_b}) = \mathrm{P}(\mathrm{X_a}) * \mathrm{P}(\mathrm{X_b}) \tag{7}$$

where $P(\cdot)$ represent the fault probability of component, X_a and X_b are specific components or events in fault tree.

B. Attack Tree

An attack tree [8], is a specialized topology used to delineate the steps involved in an attack process. Unlike fault trees, which primarily focus on system failures, attack trees are designed to describe and disseminate information about attack patterns within a system. As a result, they typically do not emphasize target scenarios to the same extent as fault trees. However, from a risk assessment perspective, it is advantageous to tailor attack trees to specific target scenarios to facilitate a more precise and comprehensive analysis. Each attack within an attack tree is characterized by its ultimate scope or motivation. For instance, the ultimate scope of a denial-of-service attack against a web server might be to obstruct a group of users from accessing the data hosted on that server. This ultimate scope serves as the target of the attack and can be defined simply as the objective of the attack.

An attack tree serves as an efficient depiction of the logical trajectory of a cyber attack directed at a particular target, outlining the complexity of each attack path. By integrating the attack tree into the fault tree, the fusion tree model gains the ability to concurrently assess the cumulative impact of conventional hardware failures (represented by the Fault Tree) and emerging cyber attacks (represented by the Attack Tree) on top events. Cyber attacks frequently disrupt the normal functioning of hardware devices by tampering with the input or output signals of system controllers. Hence, the attack tree needs to be linked to the fault tree event corresponding to its target attack to influence the failure probability of the respective event.

A novel operator called the trigger node has been introduced to regulate the flow of information from the attack tree to the corresponding fault tree event, as depicted in Figure 2. The trigger node operates only when the assertion is triggered and the attacker successfully executes the attack.

C. Modularization of the fault tree

Modularization involves redefining the original fault tree structure into a series of small independent modules, each of which can be efficiently solved, and the results can be recombined to produce a solution to the original problem. Two highly effective techniques for fault tree modularization involve creating gate dependencies by repeating basic events.

The first approach was developed by Riso and is utilized in their Faunet code [9], [10]. The second method is a linear time algorithm developed by Dutuit and Rauzy for identifying independent gates in fault tree structures [11]. While both methods excel at identifying small independent modules, they may not necessarily produce the smallest independent modules. Since they employ different methodologies, applying both methods sequentially may yield smaller modules than using either method alone.

The structure shown in Fig. 3 and Fig. 4, where duplicate events serve as inputs to all gates of the same type at one level, is reconstructed as depicted. This reconstruction is undertaken

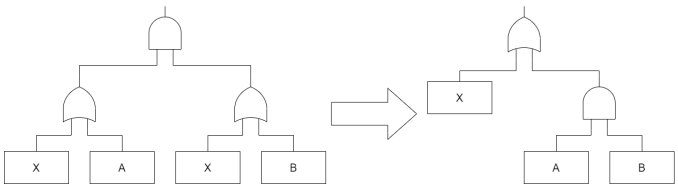

Fig. 3. AND Node

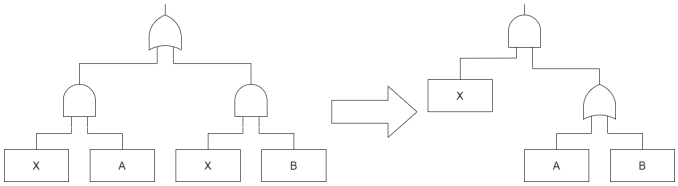

Fig. 4. OR Node

to further simplify the fault tree structure. The mathematical expression is shown in the following equations:

$$(X \text{ OR A})\text{AND}(\text{X OR B}) = \text{X OR (A AND B)}, \quad (8)$$

$$(X \text{ AND A})\text{OR}(\text{X AND B}) = \text{X AND (A OR B)}. \quad (9)$$

Based on the aforementioned rules, the fusion tree model illustrated in Fig. 2 can undergo event merging to produce the simplified fusion tree model depicted in Fig. 5.

D. Dynamic Markov Model

The Markov property stipulates that system failure and repair processes are uniform and memoryless, resulting in constant transmission rates. This implies that the immediate future state of the system depends solely on its current state. Models of continuous-time Markov processes enable the analysis of systems with dependencies, where failure and repair rates remain constant. An example Markov model is depicted in Fig. 6.

Markov models [12] consist of two components: states (nodes) and transitions (directed edges). Nodes represent the system's state based on the states of its components, while edges represent transitions between states, each with associated parameters such as transition rates. These models facilitate the formulation of state equations.

$$S(t)' = AS(t) \quad (10)$$

where $S(t)$ is the vector of state probabilities and A is the state transition matrix, $S(t)'$ is the transited state probabilities. The calculation process of Fig. 6 is as follow:

$$\begin{bmatrix} P(A)' \\ P(B)' \\ P(C)' \\ P(D)' \end{bmatrix} = \begin{bmatrix} 1 & P_5 & P_9 & P_4 \\ P_1 & 1 & P_6 & 0 \\ 0 & P_2 & 1 & P_7 \\ P_8 & 0 & P_3 & 1 \end{bmatrix} \begin{bmatrix} P(A) \\ P(B) \\ P(C) \\ P(D) \end{bmatrix} \quad (11)$$

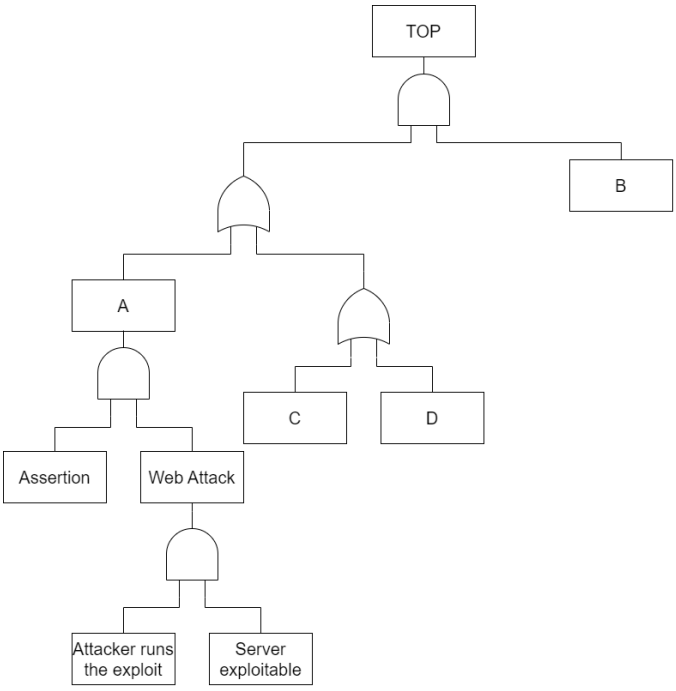

Fig. 5. Simplified Fusion Tree

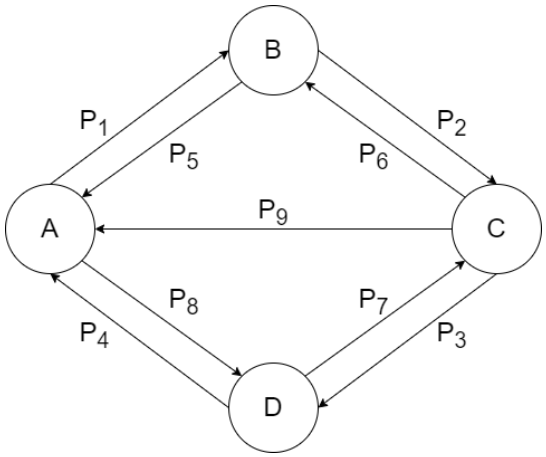

Fig. 6. Dynamic Markov Model

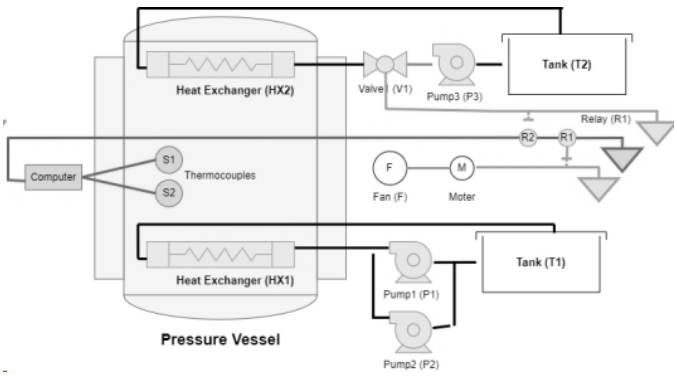

Fig. 7. Pressure vessel systems with temperature control

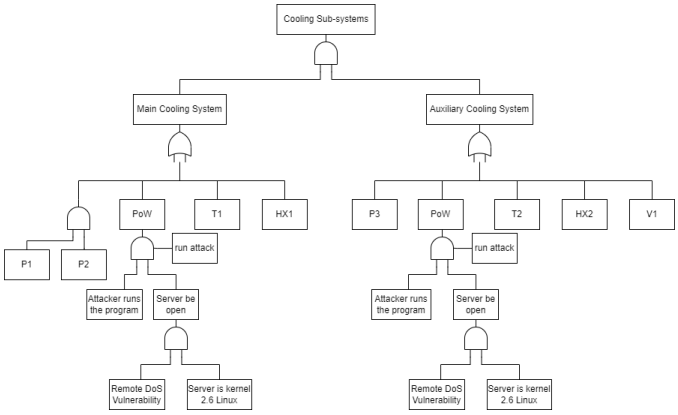

Fig. 8. Experiment Fusion Tree

III. EXPERIMENTS

In order to demonstrate the proposed fusion tree algorithm, we take the pressure vessel cooling system as an example to perform demonstration calculations, as illustrated in Fig. 7. The system described is a pressure vessel employed in an exothermic chemical reaction process that necessitates cooling. Cooling is achieved using the main cooling system, which involves water supply from tank T1. This water is conveyed to the heat exchanger (HX1) by two pumps (P1 and P2), both powered by a shared power supply, PoW. Failure of the main cooling system leads to an elevation in vessel temperature, which is monitored by the thermocouples S1 and S2. If either thermocouple detects a high vessel temperature, the computer will deactivate relays R1 and R2 and engage two backup cooling systems. The first backup system mirrors the primary setup and consists of water supply T2, heat exchanger HX2, and a single pump, P3. When relay R2 is deactivated, its contacts close, activating pump P3 and opening motorized valve V1. The second cooling mechanism involves a fan (F) powered by a motor (M). This system is activated when relay R1's contacts close upon relay deactivation. The fan, motor, pump P3, and valve V1 are also powered by the supply PoW. The Basic information of pressure vessel system is showed in TABLE III. Because the power supply in the system needs to be connected to the Internet to upload data in real time, it is more vulnerable to network attacks. Therefore, this example assumes that only the power supply is attacked.

Based on the process flow in Fig. 7, we build the fusion tree model of the system in Fig. 8. The fusion tree model is then simplified as Fig. 5 to facilitate top event failure probability calculations.

After obtaining the simplified fusion tree model, we will use Eq. 3-5 to calculate the failure probability of leaf event according to the type of leaf events (Non-repairable, scheduled maintenance and unscheduled maintenance), and then use the logic gate formula to obtain the failure probability of the final top-level event. The fault probability of leaf node events at timestamp t are showed in TABLE II.

However, due to the process flow, the equipment in the sys-

TABLE I
BASIC INFORMATION OF PRESSURE VESSEL SYSTEM

Event code	Description	Type	Failure rate (/hour)	Mean time to repair (hour)	Inspect interval (hour)
P1,P2	Pumps	scheduled maintenance	1×10^{-4}	-	2190
P3	Pump	scheduled maintenance	3×10^{-4}	-	730
HX1	Heat Exchanger	unscheduled maintenance	4×10^{-5}	3×10^{-5}	-
HX2	Heat Exchanger	unscheduled maintenance	3.5×10^{-5}	2.5×10^{-5}	-
T1	Tank	scheduled maintenance	1×10^{-5}	-	2190
T2	Tank	scheduled maintenance	2×10^{-5}	-	21090
V1	Value	Non-repairable	5×10^{-5}	-	-
PoW	Power Supply	scheduled maintenance	1×10^{-4}	-	730

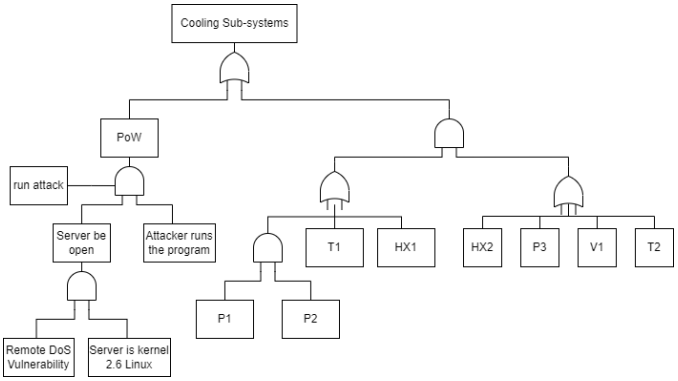

Fig. 9. Simplified Experiment Fusion Tree

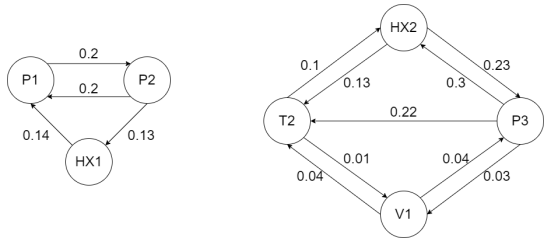

Fig. 10. Dynamic Markov Models of Pressure Vessel System

tem has complex equipment failure coupling situations. We use two dynamic Markov models (Fig. 10) to graphically represent the coupling relationship. The coupled fault probability of leaf node events at timestamp t are showed in TABLE III.

The fault probability of top event (Cooling Sub-systems) is calculated as Eq. 12 at timestamp t.

TABLE II
FAULT PROBABILITY OF LEAF NODE EVENTS

Event	Fault probability in t
P1,P2	0.1019
P3	0.1019
HX1	$0.5714(1 - exp(-7 \times 10^{-5}t))$
HX2	$0.5833(1 - exp(-6 \times 10^{-5}t))$
T1	0.0108
T2	0.0216
V1	$1 - exp(-5 \times 10^{-5}t)$
PoW	0.0356

TABLE III
COUPLED FAULT PROBABILITY OF LEAF NODE EVENTS BASED

Event	Fault probability in t
P1	$0.2023 - 0.0800exp(-7 \times 10^{-5}t)$
P2	0.1223
P3	$0.2401 - 0.1342exp(-6 \times 10^{-5}t) - 0.0400exp(-5 \times 10^{-5}t)$
HX1	$0.5846 - 0.5714exp(-7 \times 10^{-5}t)$
HX2	$0.6160 - 0.5833exp(-6 \times 10^{-5}t)$
T1	0.1018
T2	$0.1598 - 0.0758exp(-6 \times 10^{-5}t) - 0.0400exp(-5 \times 10^{-5}t)$
V1	$1.0033 - exp(-5 \times 10^{-5}t)$
PoW	0.0356

$$
\begin{aligned}
P(TOP) &= 1.2877 - 0.6697exp(-5 \times 10^{-5}t) \\
&- 0.4919exp(-6 \times 10^{-5}t) - 1.1736exp(-7 \times 10^{-5}t) \\
&+ 0.4611exp(-1.3 \times 10^{-4}t) + 0.6277exp(-1.2 \times 10^{-4}t)
\end{aligned} \tag{12}
$$

IV. CONCLUSION

The fusion tree model is proposed by integrating the fault tree and attack tree to address dependencies between events, time-varying failures, and issues with repair rate realities that challenge traditional fault tree models. Modularization and dynamic Markov modeling are introduced to simplify the fusion tree model and represent the complex coupling relationship between events in the fusion tree model. Finally, we illustrate the calculation process of the fusion model for pressure vessel systems with temporal control. The proposed fusion tree model offers improved evaluation of risks arising from both traditional functional safety and emerging information security threats to the system. Additionally, it demonstrates good interpretability.

REFERENCES

[1] J. Andrews and S. Tolo, "Dynamic and dependent tree theory (d2t2): A framework for the analysis of fault trees with dependent basic events," *Reliability Engineering & System Safety*, vol. 230, p. 108959, 2023.

[2] S. Byun, M. Papaelias, F. P. G. Márquez, and D. Lee, "Fault-tree-analysis-based health monitoring for autonomous underwater vehicle," *Journal of Marine Science and Engineering*, vol. 10, no. 12, p. 1855, 2022.

[3] J. P. McDermott, "Attack net penetration testing," in *Proceedings of the 2000 workshop on New security paradigms*, Feb 2001.

[4] J. Steffan and M. Schumacher, "Collaborative attack modeling," in *Proceedings of the 2002 ACM symposium on Applied computing*, Mar 2002.

[5] L. Meshkat, J. Dugan, and J. Andrews, "Dependability analysis of systems with on-demand and active failure modes, using dynamic fault trees," *IEEE Transactions on Reliability*, vol. 51, no. 2, pp. 240–251, 2002.
[6] W. Vesely, "A time-dependent methodology for fault tree evaluation," *Nuclear Engineering and Design*, vol. 13, p. 337–360, Aug 1970.
[7] A. Rauzy, "Toward an efficient implementation of the mocus algorithm," *IEEE Transactions on Reliability*, p. 175–180, Jun 2003.
[8] I. N. Fovino and M. Masera, *Through the Description of Attacks: A Multidimensional View*, p. 15–28. Jan 2006.
[9] O. Platz and J. Olsen, "Faunet: A program package for evalution of fault trees and networks," Jan 1976.
[10] K. A. Reay and J. D. Andrews, "A fault tree analysis strategy using binary decision diagrams," *Reliability Engineering & System Safety*, vol. 78, p. 45–56, Oct 2002.
[11] Y. Dutuit and A. Rauzy, "A linear-time algorithm to find modules of fault trees," *IEEE Transactions on Reliability*, vol. 45, p. 422–425, Jan 1996.
[12] J. Andrews and T. Moss, "Reliability and risk assessment," Apr 1994.