A Distributed Data-Driven and Machine Learning Method for High-Level Causal Analysis in Sustainable IoT Systems

Wangyang Yu, Jing Zhang, Lu Liu, Yuan Liu, Xiaojun Zhai, and Ruhul Kabir Howlader

***Abstract*—A causal relationship forms when one event triggers another's change or occurrence. Causality helps to understand connections among events, explain phenomena, and facilitate better decision-making. In IoT systems, massive consumption of energy may lead to specific types of air pollution. There are causal relationships among air pollutants. Analyzing their interactions allows for targeted adjustments in energy use, like shifting to cleaner energy and cutting high-emission sources. This reduces air pollution and boosts energy sustainability, aiding sustainable development. This paper introduces a distributed data-driven machine learning method for high-level causal analysis (DMHC), which extracts general and high-level Complex Event Processing (CEP) rules from unlabeled data. CEP rules can capture the interactions among events and represent the causal relationships among them. DMHC deploys a two-layer LSTM attention mechanism model and decision tree algorithm to filter and label data, extracting general CEP rules. Afterward, it proceeds to generate event logs based on general rules with heuristic mining (HM), extracting high-level CEP rules that pertain to causal relationships. These high-level rules complement the extracted general rules and reflect the causal relationships among the general rules. The proposed high-level methodology is validated using a real air quality dataset.**

***Index Terms*—Energy management, IoT systems, Machine learning, Causal analysis, Petri nets, CEP.**

I. Introduction

The widespread implementation of IoT technology has triggered a substantial increase in energy demand [1]. However, energy consumption must be managed effectively, otherwise, it could lead to escalated air pollution, and hinder sustainable development. We can mitigate adverse environmental impacts of energy consumption by analyzing and optimizing the energy utilization of IoT systems. Energy efficiency is a significant concern, and researchers in this field have invested notable efforts [2], [3]. Energy consumption within IoT is prevalent across diverse aspects. Both the operation of IoT systems and energy usage of data centers consume substantial energy [4]. Improper energy consumption and utilization can lead to air pollution. Air pollution data can effectively reflect the extent of energy consumption. Interactions between air pollutants involve complex causal relationships [5]. Analyzing these causal relationships will assist in better managing and optimizing energy usage, thereby ensuring the regular operation of IoT systems while mitigating energy consumption and improving energy efficiency in line with sustainability goals.

For causal relationship analysis, many studies have been conducted by predecessors. Causal relationship analysis aids in quantifying the strength of causal relationships between variables and identifying key driving factors. Based on these analytical results, we can extract results and required rules, specifically rules where one event variable directly leads to the occurrence of another event variable. These rules visually represent the causal relationships among variables, facilitating a better understanding and analysis. In this paper, we utilize CEP rules to represent causal relationships. CEP rules are employed to identify and describe complex events occurring within data streams. These rules can help us identify the interrelationship among events, thereby revealing the causal relationships among events. CEP rules typically consist of event patterns, conditions, and actions. Event patterns describe sequences and temporal relationships of events, conditions specify the conditions under which event patterns occur, and actions define the operations to be performed while requirements are met [6]. By utilizing CEP rules, complex event patterns can be captured, anomalies detected [7], real-time decision support provided [8], and appropriate actions triggered when needed, thus enhancing system efficiency and responsiveness.

By extracting relevant CEP rules, correlations among events based on these CEP rules can be identified. Through the exploration of these event correlations, we can derive rules that represent causal relationships. Previously, experts often needed to intervene manually to select CEP rules. However, with the emergence and advancement of technologies such as machine learning and data mining, automated methods to extract CEP rules have been developed not long ago. These techniques have been seen in diverse applications in various domains, such as healthcare [9] and anomaly detection [10]. Similarly, these technologies can also automatically extract CEP rules from

This work was in part supported by the Open Research Fund of Anhui Province Engineering Laboratory for Big Data Analysis and Early Warning Technology of Coal Mine Safety, China, under Grant No. CSBD2022-ZD05, in part by the Fundamental Research Funds for the Central Universities under Grant No. GK202205039, and in part by the Open Research Fund of Key Laboratory of Embedded System and Service Computing (Tongji University), Ministry of Education, China under Grant ESSCKF2023-02. (*Corresponding authors: Jing Zhang; Lu Liu.*)

Wangyang Yu, Jing Zhang and Yuan Liu are with the Key Laboratory of Modern Teaching Technology, Ministry of Education, and the School of Computer Science, Shaanxi Normal University, Xi'an 710119, China (e-mail: ywy191@snnu.edu.cn; zj25@snnu.edu.cn; liu_yuan@snnu.edu.cn).

Lu Liu and Ruhul Kabir Howlader are with the School of Computing and Mathematical Sciences, University of Leicester, LE1 7RH Leicester, U.K. (e-mail: l.liu@leicester.ac.uk; smrkh1@leicester.ac.uk).

Xiaojun Zhai is with the School of Computer Science and Electronic Engineering, University of Essex, Colchester, CO4 3SQ, U.K. (e-mail: xzhai@essex.ac.uk).

extensive datasets [11], [12]. This automated rule extraction process significantly enhances efficiency and diminishes the risk of human errors.

Based on general CEP rules, we strive to extract high-level CEP rules that reveal the fundamental causal relationships within the general CEP rules. However, the extraction of high-level CEP rules is currently not ideal, with very limited related research. A universally applicable method for extracting high-level CEP rules has not yet been achieved. Furthermore, there is a lack of research attempting to extract high-level CEP rules from unlabeled data. In addition, a thorough analysis of high-level causal relationships among general CEP rules is crucial for a deep understanding of complex data connections. Currently, no research has attempted to address practical issues from this perspective, which severely limits our ability to conduct comprehensive and in-depth analysis of data and complex system behaviors.

Our proposed framework DMHC in this paper integrates machine learning methods, HM, and CEP techniques to address these issues. The first step involves utilizing a two-layer LSTM attention mechanism model and decision tree algorithm to label abnormal data. This step can extract general CEP rules from the processed data. The second step builds upon the first step, where event data logs are generated based on the extracted general rules. High-level rules are then extracted by utilizing HM. These rules represent the causal relationships existing among events. Our methodology contributes as follows:

1) Our methodology effectively extracts high-level causal relationships from unlabeled underlying observational data, which are innovatively represented using CEP rules, distinguishing it from existing research. It analyzes causal chains among datasets and refines high-level CEP rules representing complex causal relationships. These reveal deep data connections and the causal logic within general CEP rules.

2) Technically, our innovation lies in the integration of machine learning, HM, and Petri net technologies, enabling the extraction of more complex high-level CEP rules from general CEP rules. Our general and high-level CEP rules are formed based on in-depth data analysis. This provides a novel methodology for extracting and analyzing high-level CEP rules.

3) We have applied this methodology to the monitoring of air quality data and energy management, constructing event patterns through a CEP engine based on the extracted general and high-level CEP rules. These patterns can trigger alerts or early warnings for air quality data that meet specific criteria. This allows timely adjustment of energy strategies and control of pollutant emissions within acceptable limits.

By extracting general CEP rules and high-level rules, our methodology aids in a better understanding of causal relationships within the data. In an IoT system, if it is possible to determine excessive energy consumption based on the causal relationships among air pollution events data, the system can adjust energy allocation according to real-time data fluctuations. It ensures the regular operation of devices by minimizing energy wastage to the greatest extent possible. This methodology enables more efficient energy management and optimization, thereby driving the sustainable development of IoT systems.

II. Related Work

Diverse strategies have been employed so far for causal relationship analysis and modeling. Regression analysis and decision trees are widely used in causal relationship analysis [13], [14]. Additionally, Bayesian networks are extensively popular in this domain [15]. With the comprehensive understanding of the wide range of methods for causal relationship analysis and modeling are broadly categorized into two main classes: knowledge-driven methods and data-driven methods [16].

Knowledge-driven methods often require considerable domain expertise and a deep understanding of the system model. Among those knowledge-driven methods, a notable approach involves expressing causal relationships by constructing Fault Trees (FT). [17] through experiments and evaluation of cement material parts printed by the printer, identified the main fault groups and used the Fault Tree Analysis (FTA) method to find the causes, consequences, and affected components of system failures. Additionally, some researchers combined Fault Tree Analysis and Bayesian Networks to analyze drone-related risks, constructing fault trees from reports and literature, identifying initial risk factors, converting these into Bayesian Networks, and validating the model with real cases [18].

Though knowledge-driven methods are applied in multiple domains, they have some limitations [19]. In large-scale systems, causal variations of variables become challenging to understand entirely, thus making such models difficult to construct [20]. Therefore, data-driven methods are required. Methods that rely on data-driven techniques are effective in managing complex data [21]. One common data-driven technique for establishing causal relationships is Granger causality, which is widely used by researchers [22]. [23] enhanced the accuracy and noise resistance of bearing fault diagnosis by integrating Granger Causality Test with Graph Neural Network, utilizing feature transformation and causal analysis. Kiran et al. analyzed and ranked the performance of different sectors using Dempster-Shafer evidence theory, then determined inter-sector dependencies through Granger causality tests to invest in independent strong sectors, thereby improving investment efficiency [24].

Previous studies often focus on the direct connections among data, but frequently overlooking the indirect causal links that need to be elucidated when analyzing complex data combinations. Our methodology demonstrates clear innovation by using CEP rules to interpret causal relationships. It combines data-driven machine learning methods and process mining techniques to extract general and high-level CEP rules from unlabeled data. This aspect distinguishes it from existing research. These high-level CEP rules not only elucidate the connections among general CEP rules, establishing a causal framework based on these general CEP rules, but also depict the causal links among complex data combinations in low-level observational data. This aspect is crucial for a thorough understanding and analysis of system behavior or state changes. Our methodology introduces an innovative technique for extracting high-level CEP rules, providing a unique perspective and framework for analyzing the causal relationships within complex data combinations, effectively addressing the shortcomings in existing research.

III. Methodology

In this section, we initially present the overall framework of the Distributed Data-driven Machine Learning Method for High-level Causal Analysis (DMHC). Subsequently, we provide detailed explanations of the two main steps encompassed by this methodology.

A. Framework of DMHC

The overall framework of DMHC is presented in Figure 1. The collected data is stored in a historical database to be processed and analyzed by DMHC. The proposed methodology comprises two main steps.

At first, we employ machine learning algorithms to label abnormal data. Subsequently, we extract general CEP rules from the processed data. Building upon the first step, the second step involves generating event data logs based on the extracted general CEP rules. To derive high-level CEP rules, we utilize HM. These rules offer a broader perspective on causal relationships. The extracted CEP rules are input into the CEP engine, within which real-time risk alerts are applied to the incoming data. These rules enable the CEP engine to provide timely risk warnings based on established causal relationships. By analyzing data and extracting CEP rules, DMHC can assist researchers in understanding the causal relationships among pollution events, formulating more effective environmental policies and strategies, thereby reducing the impact on the environment and promoting sustainable development.

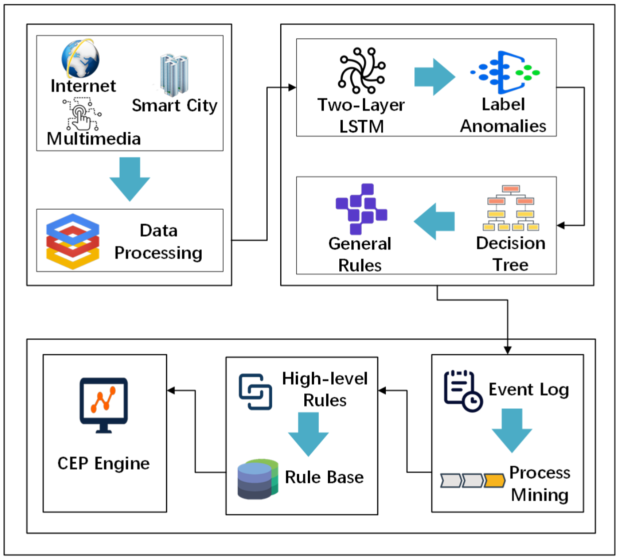


Fig. 1: The overall framework of DMHC

B. The Initial Step

In the beginning, a two-layer LSTM attention mechanism model will be described. Subsequently, the explanation of how this model is utilized to label abnormal data will follow. Additionally, we will discuss how the decision tree method is employed to extract general CEP rules.

1) Two-Layer LSTM Attention Mechanism Model

As discussed earlier, we used a two-layer LSTM attention mechanism model [25] here. The two-layer LSTM attention mechanism model is an extension of the traditional LSTM, designed to improve the performance of sequence classification tasks. Since our goal is to label anomalous data, we applied this model to this method. The model normalizes the raw data and splits the standardized air quality data into training and testing sets. LSTM model trained using the training set to achieve the optimal model. Then, the testing set is input into the model to predict regression and calculate the error between predicted and actual data. A threshold σ is set using three times the standard deviation of the reconstruction errors. When the reconstruction error of a data point exceeds this threshold σ, it is classified as an anomaly. Following these steps, a series of labeled data can be obtained.

In our framework, the process of identifying and marking abnormal data is crucial, as it lays the foundation for more accurate identification and analysis of air pollution events later on. By analyzing air quality data and marking those data points that deviate from the normal range, we can distinguish between actual pollution events and temporary data fluctuations. This not only enhances the accuracy of our pollution events detection but also allows us to identify potential pollution sources early by analyzing the anomalies in these air quality indicators. Therefore, this process provides strong data support for taking timely preventive measures, reducing the impact of pollution, and developing long-term environmental protection measures.

Analyzing the time complexity of the two-layer LSTM attention mechanism model requires considering the computational demands of each component within the model. The two-layer LSTM structure means that the input data are processed sequentially through two LSTM layers, with each layer executing a complete pass over the data. The attention mechanism adds additional computational steps, as it assesses the importance of each element in the input sequence and adjusts the model's response to these elements accordingly. The time complexity is primarily influenced by the sequence length (N), the number of features (d), and the number of hidden units (h). Specifically, the computational cost of each LSTM layer is generally proportional to the product of the sequence length and the number of hidden units, while the computational cost of the attention mechanism is proportional to the square of the sequence length and the number of hidden units. Therefore, the overall time complexity of the model is approximately $O(N \times (d \times h + 2h^2))$.

2) Extraction of General Rules

The decision tree is used to extract general CEP rules after labeling the data from the previous step. This decision tree method is usually employed to extract useful information from large datasets. The construction of a decision tree involves the recursive selection of optimal features and the division of the training dataset based on these features, ensuring the best classification for each subset. This process not only corresponds to the partitioning of the feature space but also to the construction of the decision tree.

When constructing a decision tree, the process begins by creating the root node and placing all training data samples within it. Subsequently, an optimal feature is selected, and the training dataset is divided into subsets based on this feature. This division ensures the best possible classification for each subset. If these subsets can be reasonably classified, leaf nodes are created, and subsets are assigned to their respective leaf nodes. However, if there are still subsets that cannot be adequately classified, the process of selecting the optimal feature is repeated, leading to further division and node construction. This recursive process continues until all training data subsets are correctly classified or no suitable features are available. Eventually, each subset is assigned to a leaf node, providing a clear classification result.

We can convert decision trees into general CEP rules. These general CEP rules reveal the interactions among pollutants and their impact on air quality, aiding in the precise monitoring and identification of key pollution sources. Additionally, by analyzing the relationship between pollutants and energy consumption, we can optimize energy use and reduce emissions. Utilizing these rules allows us to adjust production and energy patterns, decrease emissions of key pollutants, and contribute to environmental protection.

In the decision tree algorithm, general rules extraction is attained by transforming tree structure and paths into logical rules. Start from the root node, this process involves traversing the branches along a path until reaching a leaf node of the tree. Each node corresponds to a feature and a specific splitting condition. At the time of traversal, visited nodes and their corresponding splitting condition are recorded in this method. Upon reaching a leaf node, a general rule is generated by combining the features and splitting conditions encountered along the path, forming a logical expression. This process repeated for each path to formulate a general rule for each path in the tree. Consequently, every path constructs an independent general rule.

The average time complexity analysis of the decision tree algorithm considers that the tree construction process does not always perfectly bisect the dataset at each division. In the average case, it is assumed that the data is relatively evenly split. Under these conditions, the time complexity of constructing a decision tree can be approximated as $O(n \times m \times \log m)$, where n is the number of features, and m is the number of samples. This is because each node in the decision tree does not necessarily split the dataset into two equally sized subsets every time, but the division generally results in a progressively smaller amount of data, so the entire dataset's splitting process can be viewed as multiple iterations of the data, with each iteration corresponding to a layer of the tree. When selecting the optimal feature at each split, the algorithm needs to traverse m samples and evaluate n features to determine which feature best splits the dataset into subsets with distinct category labels.

C. The Subsequent Step

This section first provides a detailed introduction to HM. It then explains how we used HM to extract high-level rules and how Petri nets [26] is utilized, which encompass high-level rules among events. Additionally, It will describe how to derive high-level rules from the obtained Petri nets here.

1) Heuristic Miner

Process Mining (PM) [27] is a technology that automatically discovers, analyzes, and improves business processes from log data in an actual event. It combines techniques from data mining, business process management, and workflow technologies. It aims to reveal the sequence of activities, dependencies, and execution patterns within a business process to provide a deep understanding of organizational processes and optimization recommendations. In the context of environmental protection and sustainable development, this technology can be utilized to deeply understand the implicit relationships among pollutants, thereby identifying possible intervention points and energy-saving optimization schemes.

In the field of PM, various techniques have been implemented. Among them, HM is one well-known process mining algorithm. This algorithm is designed to handle noisy data and can effectively operate short and long loops, addressing the limitations of other algorithms. HM consists of the following four parts [28]:

a) Construct Dependency/Frequency Table (D/F table): Construct a dependency/frequency table based on actual event log data, recording the dependencies and frequencies between activities. The Alpha algorithm defines four basic log-based relations: 'directly follows,' 'causal,' 'parallel,' and 'independent.' [29] The 'directly follows' emphasizes that in the event log, activity A occurs immediately after activity B. 'causal' implies a direct cause-and-effect connection between two activities, meaning if activity A occurs, activity B will necessarily happen. 'parallel' indicates that there is no clear sequential order between two activities, and they can occur simultaneously without affecting each other. 'independent' means there is no direct link between two activities, and their occurrence does not affect each other. The 'directly follows' relationship is used here to determine the dependencies between activities. Its specific definition is as follows:

Activity B directly follows Activity A when there is a temporal relationship among them. It means that in a significant number of cases in the event log, Activity A is immediately followed by Activity B without any other activities in between. That implies a strong sequential dependency between Activity A and Activity B. Therefore, Activity B occurs right after Activity A in the process flow in a typical situation. For example, given an event log: H = [$< m, z >^6$, $< m, n, x, z >^9$, $< m, x, n, z >^9$, $< m, n, z >^1$, $< m, x, z >^1$, $< m, y, z >^9$, $< m, y, y, z >^2$, $< m, y, y, y, z >^1$] (The text inside the $<>$ represents the activity sequence or trajectory, and the numbers following it represent the frequency of the trajectory.)

Thus, the collection of direct follow relations in the event log H is represented as $> H$ = $\{(m, z)$, (m, n), (n, x), (x, z), (m, x), (x, n), (n, z), (m, y), (y, z), $(y, y)\}$. Then, based on the corresponding frequencies in the collection of direct follow relations, a dependency/frequency table is established as shown in TABLE I. ($|X >_H Y|$ represents the number of times Y directly follows X in H.)

TABLE I: The frequency of direct follow relations in event log H

$> H$	m	n	x	y	z
m	0	10	10	12	6
n	0	0	9	0	10
x	0	9	0	0	10
y	0	0	0	4	12
z	0	0	0	0	0

b) Establish Dependency Metric Table:

Using the dependency/frequency table, calculate the dependency metrics between activities to measure the strength of their relationships.

H is the previous event log on ζ. $X, Y \in \zeta$. $|X >_H Y|$ represents the number of times Y directly follows X in H. $|X \Longrightarrow_H Y|$ represents the dependency relationship value between X and Y. Therefore, the formula 1 [30] holds:

$$|X \Longrightarrow_H Y| = \begin{cases} \frac{|X \Longrightarrow_H Y| - |Y \Longrightarrow_H X|}{|X \Longrightarrow_H Y| + |Y \Longrightarrow_H X| + 1}, if X \neq Y \\ \\ \frac{|X \Longrightarrow_H X|}{|X \Longrightarrow_H X| + 1}, if X = Y \end{cases} \quad (1)$$

$|X \Longrightarrow_H Y|$ generates a value between -1 and 1. If $|X \Longrightarrow_H Y|$ is close to 1, then there is a strong positive dependency between X and Y, meaning that X is often the cause of Y. This value approaches 1 only when X is frequently followed directly by Y and Y is rarely followed directly by X. If $|X \Longrightarrow_H Y|$ is close to -1, then there is a strong negative dependency between X and Y, meaning that Y is often the cause of X. There is a special case when $|X \Longrightarrow_H X|$, which indicates the presence of a loop and a strong reflexive relationship if X is frequently followed by X. TABLE II presents the dependency measure of event log H.

c) Build Dependency Graph:

Create a dependency graph based on the dependency/frequency table, visualizing the relationships between activities. The resulting Figure 2 is as shown below. A dependency graph with a threshold of 5 for $| >_H |$ and a threshold of 0.9 for $| \Longrightarrow_H |$ is established. Due to $|y >_H y| = 4 < 5$ and$|y \Longrightarrow_H y| = 0.80 < 0.9$, there is no self-loop at y. Additionally, since $|m \Longrightarrow_H z| = 0.86 < 0.9$, there is also no connection between m and z.

d) Convert Dependency Graph to Petri net:

Transform the dependency graph into a Petri net, which represents the process model in a formal workflow notation.

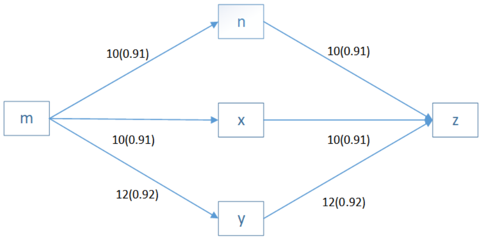


Fig. 2: A dependency graph

2) High-level rules extraction process

a) Generating Event Logs

As mentioned earlier, process mining involves extracting information about processes from event logs. Transforming the original dataset into an event log facilitates our subsequent mining activities. We assume the following way of recording events [28]:

Activity: Each event designates a step or activity defined in the process. These activities can be tasks, operations, or decisions in the business process.

Case: Each event represents a process instance or case. A case refers to a specific execution process in the business, such as a customer order, service request, or project execution.

Executor/Initiator: Each event can have an executor or initiator, referring to the person or role responsible for performing or initiating the activity. The executor can be an individual, a team, or a department.

Timestamp: Each event is associated with a timestamp that records the exact time when the event occurred. These timestamps can represent either the creation time of the event record or the actual time of the activity.

The event records are totally in order. It implies that events are sorted based on their timestamps, reflecting the actual sequence of activities. This ordered form of event records provides structured data for process mining. TABLE III presents an example log involving 10 events, 4 activities, and 5 executors.

In this research, firstly, filtered data is labeled with states in Navicat [31]. This labeling allows the data to correspond to general rules. After that, relevant algorithms are used to add segment nodes, traces, and timestamps to the labeled data. Through these procedures, the original dataset is transformed into an event log where each event has a timestamp and a Case ID. In this event log, Case ID represents a record for the events that contain.

b) From Event Log to Petri net

The constructed event log is imported into ProM 6.9 software [32]. In the field of process mining, ProM 6.9 is one of the most renowned open-source tools. It offers various process mining techniques such as process discovery, conformance checking, and process enhancement. With its diverse range of plugins, ProM caters to different user requirements. Through ProM, it is likely to acquire in-depth insights into business processes and carry out optimization and improvement. By utilizing ProM 6.9 software and its extensive plugin library, we are able to delve into a thorough analysis and understanding of various events and activities within the process, revealing

TABLE II: Based on the dependency measure of the 5 activities in event log H

$\vert \Longrightarrow_H \vert$	m	n	x	y	z
m	$\frac{0}{0+1} = 0$	$\frac{10-0}{10+0+1} = 0.91$	$\frac{10-0}{10+0+1} = 0.91$	$\frac{12-0}{12+0+1} = 0.91$	$\frac{6-0}{6+0+1} = 0.91$
n	$\frac{0-10}{0+10+1} = -0.91$	$\frac{0}{0+1} = 0$	$\frac{9-9}{9+9+1} = 0$	$\frac{0-0}{0+0+1} = 0$	$\frac{10-0}{10+0+1} = 0.91$
x	$\frac{0-10}{0+10+1} = -0.91$	$\frac{9-9}{9+9+1} = 0$	$\frac{0}{0+1} = 0$	$\frac{0-0}{0+0+1} = 0$	$\frac{10-0}{10+0+1} = 0.91$
y	$\frac{0-12}{0+12+1} = -0.92$	$\frac{0-0}{0+0+1} = 0$	$\frac{0-0}{0+0+1} = 0$	$\frac{4}{4+1} = 0.80$	$\frac{12-0}{12+0+1} = 0.92$
z	$\frac{0-6}{0+6+1} = -0.86$	$\frac{0-10}{0+10+1} = -0.91$	$\frac{0-10}{0+10+1} = -0.91$	$\frac{0-12}{0+12+1} = -0.92$	$\frac{0}{0+1} = 0$

TABLE III: An event log

$caseid$	$activityid$	$originator$	$timestamp$
$case1$	$activityA$	Amy	$6-3-2023:12.03$
$case2$	$activityC$	$Mike$	$6-3-2023:12.13$
$case2$	$activityC$	$Peter$	$6-3-2023:12.23$
$case3$	$activityB$	$Jane$	$6-3-2023:12.33$
$case5$	$activityD$	$Alice$	$6-3-2023:12.43$
$case1$	$activityA$	Amy	$6-3-2023:13.03$
$case4$	$activityA$	Amy	$6-3-2023:13.13$
$case2$	$activityD$	$Jane$	$6-3-2023:13.24$
$case5$	$activityB$	$Peter$	$6-3-2023:13.25$
$case3$	$activityD$	$Mike$	$6-3-2023:13.35$

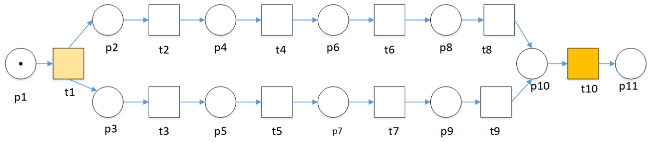


Fig. 3: The Petri net model

their interrelationships and dependencies. In the field of energy conservation, this process is significant. For example, this facilitates our in-depth analysis of the relationships between pollution events, thereby enhancing our understanding and management of energy consumption and allocation. By examining the interconnections and sequence of pollution events, we can anticipate future incidents and proactively adjust inappropriate energy usage, ensuring that energy is utilized more efficiently and environmentally friendly, thus advancing the goals of sustainable development.

We run the HM and obtain the Petri net model. Figure 3 illustrates an example of a Petri net model.

c) Extraction of High-level Rules

After obtaining the Petri net model, it is essential to analyze it and extract the desired CEP rules. In this part, we utilized a reachability graph to assist in the analysis of causal relationships. We conducted a reachability analysis on the Petri net, resulting in the reachability graph shown in Figure 4.

The nodes of the reachability graph represent the extracted general CEP rules, while the directed edges represent the relationships among these general CEP rules. Then, we analyze the relationships among nodes in the reachability graph and map these relationships into high-level CEP rules.

In our framework, by analyzing the Petri net model to extract high-level CEP rules, we delve into the causality and dependency relationships of pollution events. Through reachability graph analysis, we not only identify the interactions among pollution events but also uncover potential influencing factors, which are crucial for developing targeted pollution control strategies. For instance, identifying certain events that may trigger other pollution incidents helps us take preventive measures at critical points to prevent pollution or reduce its spread. This enhances our ability to formulate effective and sustainable environmental protection strategies.

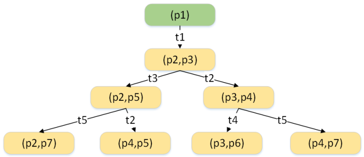


Fig. 4: The part of a reachability graph

The time complexity of the HM is primarily influenced by the total number of events in the event log and the number of different activity types. The time complexity is estimated to be $O(N+M^2)$, where N represents the total number of events in the event log, and M denotes the number of different activities. The algorithm's complexity mainly includes two aspects: First, the algorithm analyzes the event log to construct a heuristic relationship matrix. This matrix records the relationships between activities, including directly follows, causal, and parallel relationships. Constructing this matrix requires traversing the entire event log, which has a linear complexity of $O(N)$. Second, after building the heuristic relationship matrix, the algorithm needs to analyze the potential relationships between activities. This involves comparing the relationships between different activities and determining which relationships are significant. Since each activity needs to be compared with all other activities, this part of the process has a quadratic complexity of $O(M^2)$.

Our methodology primarily involves three major algorithms mentioned above. By integrating multiple algorithms, our methodology increases overall computational complexity but allows us to analyze and process data from multiple dimensions and perspectives, providing a more comprehensive and in-depth approach to solving complex problems.

IV. Experiments and Results

In this section, we will demonstrate the overall performance of the proposed DMHC using data. To better explain the application of the proposed framework, we review the DMHC workflow as illustrated in Algorithm 1. First, the raw data is normalized, then anomalies are detected and labeled using the two-layer LSTM attention mechanism model. Next, general CEP rules are extracted using the decision tree model. After mapping the raw data to the general CEP rules and generating event logs, HM is used to extract Petri net model representing high-level CEP rules. The Petri net model is then converted into reachability graphs for analysis, ultimately yielding high-level CEP rules. By integrating these general and high-level CEP rules into the CEP engine, we can monitor the data and generate alerts. To evaluate the overall predictive capability of our methodology, we selected air quality data from a smart city scenario.

A. Data Set

In our experiments, we utilize urban air pollution data collected by the Pulse project from *the City EU FP7 program* [33]. The dataset consists of $17,568$ samples, each containing eight features: particulate matter, sulfur dioxide, nitrogen dioxide, carbon monoxide, longitude, latitude, ozone, and a timestamp. The concentrations of various pollutants have similar maximum and minimum values, respectively at 215 and 15. For particulate matter, the average concentration is 124.90 with a standard deviation of 54.04. The average concentration of nitrogen dioxide is 107.10, with a standard deviation close to that of particulate matter, at 54.09. The average concentration of sulfur dioxide is 116.59, with a standard deviation of 54.61. Carbon monoxide has an average concentration of 98.13 and a standard deviation of 49.70, indicating a relatively more concentrated distribution of carbon monoxide concentrations. Lastly, ozone has an average concentration of 111.04, with the largest standard deviation of 55.04.

Algorithm 1: DMHC Workflow

```
Input: raw_data, threshold, lstm_model,
       decision_tree_model
Output: general_cep_rules, high_level_cep_rules,
        alerts
Step 1: Data Collection and Preprocessing
    normalized_data =
      min_max_normalize(raw_data);
Step 2: Anomaly Detection
    train_set, test_set =
      split_data(normalized_data);
    lstm_model.train(train_set);
    predicted = lstm_model.predict(test_set);
    reconstruction_error =
      calculate_error(predicted, test_set);
    anomalies =
      label_anomalies(reconstruction_error,
      threshold);
Step 3: Extraction of General CEP Rules
    decision_tree_model.train(anomalies);
    classification_results = ex-
      tract_classification_results(decision_tree_model);
    general_cep_rules = [];
    foreach result in classification_results do
        general_cep_rule = create_cep_rule(result);
        general_cep_rules.append(general_cep_rule);
Step 4: Generation of Event Logs
    labeled_data = label_data(anomalies,
      general_cep_rules);
    event_logs = generate_event_logs(labeled_data);
Step 5: Extraction of High-level CEP Rules
    petri_net_model =
      apply_heuristic_miner(event_logs);
    fitness = check_fitness(petri_net_model);
    reachability_graph =
      convert_to_reachability_graph(petri_net_model);
    edge_probabilities = calcu-
      late_edge_probabilities(reachability_graph);
    high_probability_edges =
      filter_edges(edge_probabilities, threshold);
    high_level_cep_rules = ex-
      tract_high_level_cep_rules(high_probability_edges);
Step 6: CEP Engine Analysis
    cep_engine =
      integrate_cep_engine(general_cep_rules,
      high_level_cep_rules);
    alerts =
      cep_engine.monitor_and_alert(raw_data);
    accuracy = evaluate_alerts(alerts);
return general_cep_rules, high_level_cep_rules,
  alerts;
```

This type of air pollution monitoring is also a typical application of IoT systems [34]. Monitoring and optimizing pollution levels contribute to achieving energy savings and emissions reduction, formulating energy-saving decisions to enhance energy utilization efficiency, and promoting green computing practices and sustainable development.

B. Experimental Environment

In the first step, we utilize the TensorFlow and Keras deep learning frameworks on the Python platform, which can train the model we need. Additionally, we employ the scikit-learn machine learning library to extract general CEP rules from its toolbox. In the second step, we utilize the ProM tool. It is based on Eclipse [35] and JDK 1.8. We extract Petri nets from the general CEP rules.

C. Experimental Results

a) Label Anomalous Data

In our methodology, we employ the two-layer LSTM attention mechanism model to label anomalous data. Firstly, we normalize the raw data using the min-max normalization method. The specific formula for min-max normalization is as follows [36]:

$$normalized\_value = \frac{(x - min\_value)}{(max\_value - min\_value)} \tag{2}$$

Where 'x' is the original value, 'min_value' is the minimum value of the data, and 'max_value' is the maximum value of the data. Next, the normalized standard air quality data is randomly split into training and testing sets. The training set is used to train and fit the two-layer LSTM model, aiming to obtain an optimal model. Then, the testing set is input into the trained model for regression prediction, and the model outputs the predicted values. After receiving predicted values, we calculate the reconstruction error (RE) between the predicted and actual data. We select three times the standard deviation of the reconstruction errors as the threshold. If the reconstruction error of a data point exceeds the threshold, it is marked as an anomaly. By performing the above operations, we obtained a series of labeled data. In the labeled data, 0 represents normal data, while 1 represents abnormal data.

b) General CEP Rules Extraction and Application

Next, we used the decision tree algorithm to extract general CEP rules from the labeled data. To evaluate the performance of the decision tree algorithm, we selected the Support Vector Machine (SVM) and Random Forest for comparison because these algorithms are widely used in the field of machine learning for classification problems, and their performance often serves as a benchmark for evaluation and comparison. Common metrics for assessing machine learning algorithms include Precision, Recall, F1 Score, and Overall Accuracy. The performance metrics of the decision tree algorithm, SVM, and Random Forest are presented in Table IV. The results show that the decision tree model, even with default parameters, exhibited good performance, especially in terms of overall accuracy, reaching 90% and outperforming other models. This high accuracy highlights the effectiveness and robustness of the decision tree in processing this dataset. For the SVM model, despite parameter tuning (*C*=10, *gamma*=0.01), its performance in classifying class 1 data was still inferior to the decision tree, with an overall accuracy of only 86%. After parameter adjustment (*n_estimators*=10, *max_depth*=30), the Random Forest achieved an overall accuracy of 88%, but showed lower recall for class 1 data, indicating a deficiency in classifying positive classes. Considering the comprehensive advantages of the decision tree in terms of accuracy, balance across classes, simplicity, and interpretability of the model, we believe that the decision tree is the most suitable classification model.

TABLE IV: Performance Metrics

Metric / Classifier	Decision Tree	SVM	Random Forest
Precision (Class 0)	0.92	0.91	0.89
Precision (Class 1)	0.83	0.62	0.80
Recall (Class 0)	0.97	0.91	0.97
Recall (Class 1)	0.62	0.62	0.50
F1-Score (Class 0)	0.94	0.91	0.93
F1-Score (Class 1)	0.71	0.62	0.62
Overall Accuracy	0.90	0.86	0.88

We visualize the decision tree model for easier extraction of the desired rules from the tree model. The partial decision tree obtained is shown in Figure 5.

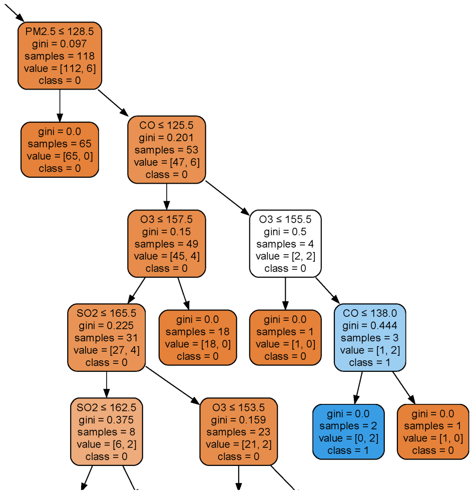


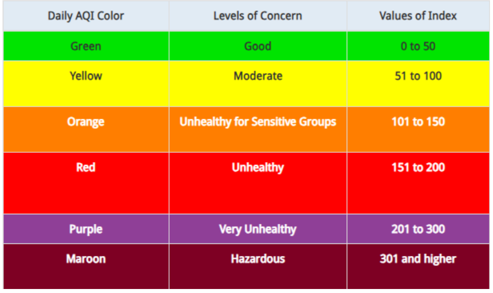
Fig. 5: Partial decision tree model

By using the relevant code, we can convert the obtained decision tree model into applicable general CEP rules, which help us monitor and understand air pollution patterns. Applying these general CEP rules allows us to track the occurrence of pollution events, enabling timely measures to mitigate pollution. For example, we can shift to using clean energy or adjust energy allocation and usage strategies to reduce the emissions of specific pollutants, thereby effectively alleviating environmental pollution and promoting the implementation of energy-saving and environmental protection measures. Two example rules are :

“$carbon\ monoxide > 98.0\ and\ sulfur\ dioxide > 144.0\ and\ particulate\ matter > 128.5\ and\ carbon\ monoxide <= 125.5\ and\ ozone <= 157.5\ and\ sulfur\ dioxide <= 165.5\ and\ sulfur\ dioxide <= 162.5\ and\ particulate\ matter > 150.0\ and\ particulate\ matter <= 152.0$”

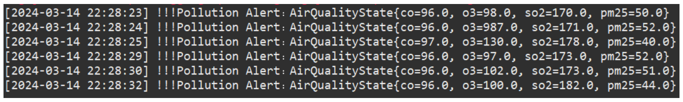
“$carbon\ monoxide > 98.0\ and\ sulfur\ dioxide > 144.0$ *and particulate matter* $>$ *128.5 and carbon monoxide* $<=$ *125.5 and ozone* $<=$ *157.5 and sulfur dioxide* $>$ *165.5 and ozone* $<=$ 153.5 *and sulfur dioxide* $>$ *203.5 and particulate matter* $<=$ *144.0 and ozone* $<=$ 146.5”

We apply the extracted general CEP rules to the Esper CEP engine and create CEP pattern events based on these rules. We utilize these events to monitor air quality data and issue early warnings. The CEP engine generates alerts for abnormal atmospheric data. For instance, in the rule $P2$: *carbon monoxide* $<=$ *98.0 and ozone* $>$ *95.5 and sulfur dioxide* $>$ *175.0 and carbon monoxide* $<=$ *82.0*. In the extracted rule $P2$, the AQI value of “$sulfur\ dioxide$” falls between 151 and 200, which is marked as unhealthy by the World Health Organization (WHO) [37] as illustrated in Figure 6. It can be harmful to human health. In this case, our CEP engine will issue an unhealthy alert as displayed in Figure 7. Relevant personnel can utilize this alert information to become aware that energy consumption or production activities may lead to excessive emissions of “$sulfur\ dioxide$”. Decision-makers may take measures such as optimizing production processes, reducing the use of high-sulfur coal, or transitioning to cleaner energy sources to reduce “$sulfur\ dioxide$” emissions. By taking these actions, general CEP rules not only help monitor air pollution but also prompt practical measures to optimize energy use and reduce emissions of various pollutants. This not only improves energy efficiency and effectiveness but also contributes to environmental protection.



Daily AQI Color	Levels of Concern	Values of Index
Green	Good	0 to 50
Yellow	Moderate	51 to 100
Orange	Unhealthy for Sensitive Groups	101 to 150
Red	Unhealthy	151 to 200
Purple	Very Unhealthy	201 to 300
Maroon	Hazardous	301 and higher

Fig. 6: AQI



```
[2024-03-14 22:28:23] !!!Pollution Alert: AirQualityState{co=96.0, o3=98.0, so2=170.0, pm25=50.0}
[2024-03-14 22:28:24] !!!Pollution Alert: AirQualityState{co=96.0, o3=987.0, so2=171.0, pm25=52.0}
[2024-03-14 22:28:25] !!!Pollution Alert: AirQualityState{co=97.0, o3=130.0, so2=178.0, pm25=40.0}
[2024-03-14 22:28:29] !!!Pollution Alert: AirQualityState{co=96.0, o3=97.0, so2=173.0, pm25=52.0}
[2024-03-14 22:28:30] !!!Pollution Alert: AirQualityState{co=96.0, o3=102.0, so2=173.0, pm25=51.0}
[2024-03-14 22:28:32] !!!Pollution Alert: AirQualityState{co=96.0, o3=100.0, so2=182.0, pm25=44.0}
```

Fig. 7: Part of the result of the unhealthy alert

We compared the warning results of general CEP rules with a series of actual labels (indicating whether the status falls within the unhealthy AQI range) to calculate the accuracy of warnings generated based on the general CEP rules. Through calculation, the accuracy based on general CEP rule alerts is 90%.

c) Convert Labeled Data into Event Logs

Next, we proceed to extract high-level CEP rules based on the extracted general CEP rules. High-level CEP rules reflected correlations behind the general air pollution events. Firstly, in Navicat, we label the processed data with their corresponding states. Then, we use relevant algorithms to add segmentation nodes, traces, and timestamps to the labeled data. Through these operations, the original dataset is transformed into an event log. Here is an example shown in TABLE V of a partial event log.

TABLE V: Partial event log

$DataTime$	$PM2.5$	O_3	CO	SO_2	$label$	$status$	$case$
9:00:00	-1	-1	-1	-1	-1	$P0$	$trace1$
9:00:01	65	174	119	169	0	$P3$	$trace1$
9:00:02	64	179	120	171	0	$P12$	$trace1$
9:00:40	91	186	124	168	1	$P15$	$trace1$
9:00:45	97	175	128	170	1	$P13$	$trace1$
9:00:46	97	176	125	170	0	$P14$	$trace1$
9:00:47	97	173	124	165	0	$P3$	$trace1$
9:00:49	100	175	129	165	0	$P13$	$trace1$
9:00:50	99	176	126	164	0	$P14$	$trace1$
9:00:51	104	174	122	166	0	$P3$	$trace1$
9:00:59	-1	-1	-1	-1	-1	end	$trace1$

d) Apply HM to Generate Petri net

The constructed event log was imported into ProM 6.9 software. We run the heuristic miner algorithm plugin, and then we can obtain a Petri net model. The part of the Petri net model is illustrated in the following Figure 8.

In a Petri net, each transition represents an event in the system. In our experiment, each transition corresponds to a general CEP rule. The trajectories of a Petri net refer to sequences of transitions occurring within the Petri net, describing changes in the system’s states. Taking our dataset as an example, the transitions represent general air pollution events. The trajectories reflect the correlation relationships among air pollution events. In the obtained Petri net, places represent the states of the trajectory. Transitions fall into two categories. Black transitions are silent transitions, which lack real-world significance and are included to ensure the net’s structural completeness. In contrast, white transitions have real-world significance and represent general CEP rules.

For PM, we need to perform conformance checking to assess the quality of the obtained model. The expected measurement used for this purpose is fitness, which indicates how

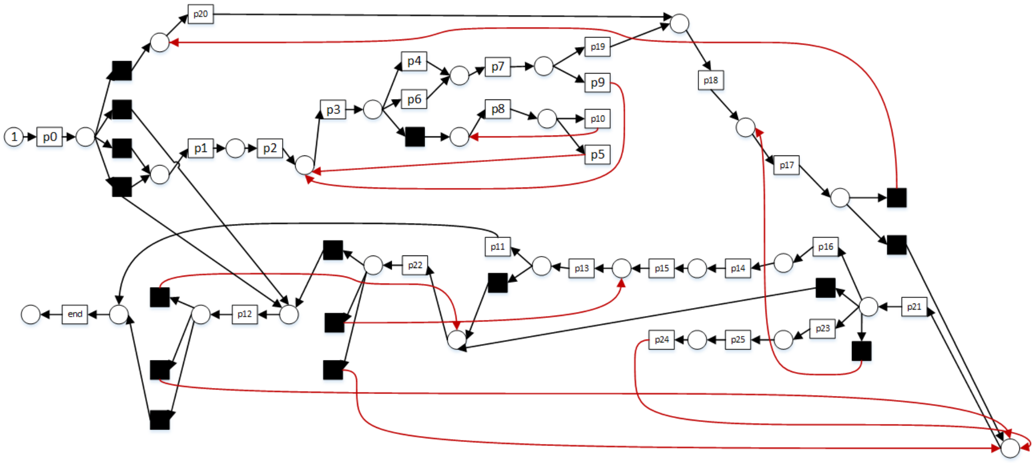


Fig. 8: The part of the Petri nets model

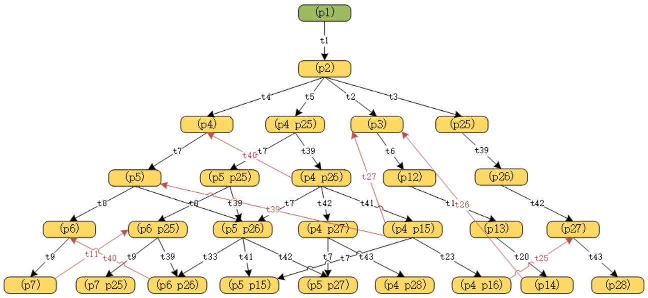


Fig. 9: A portion of the resulting reachability graph

well the model reproduces the majority of traces in the log. A higher fitness value indicates a better fit, implying that the model is ideal. We use a plugin to calculate the fitness of our obtained model, which is found to be 0.82, indicating a good fit.

e) High-level CEP Rules Extraction and Application

After obtaining the Petri net model, we need to analyze this to extract the desired high-level CEP rules. To facilitate rule extraction, we utilize relevant Petri nets processing tools. We convert the Petri net into a reachability graph using the processing tool. A portion of the resulting reachability graph is displayed in Figure 9.

Multiple structures in the above graph contain all the high-level CEP rules we can get. For example, based on the relationship among adjacent three-layer nodes, we can extract some high-level CEP rules. In this reachable graph, there are directed arcs across layers. Through this relationship, we can also get some high-level CEP rules. By extracting these high-level CEP rules, we can uncover the underlying causal relationships inherent in the general CEP rules. Analyzing these causal relationships assists in making better decisions. In this experiment, our analysis of high-level CEP rules aims to uncover the underlying high-level causal relationships behind air pollution events. Let's take an example of an adjacent three-layer node to illustrate how to extract high-level CEP rules.

Definition 1 [38]: Let $PN(S, T; F, M)$ be a bounded Petri net, where the triple $G = (R(M0), S, k)$ represents a reachability graph, where:

- $R(M0)$ is the set of reachable markings of PN, constituting the vertex set of G;
- S represents the set of arcs in G, defined as $S = \{(Mi, Mj) \mid Mi, Mj \in R(M0), \exists\ tk \in T : Mi[tk > Mj\}$;
- k: $Ar \to T$, $k(Mi, Mj) = b$ if and only if $Mi[b > Mj$, and when $k(Mi, Mj) = b$, b is referred to as the label of the $arc(Mi, Mj)$.

Definition 2: Given a reachability graph $G = (V, E)$, where V is the set of nodes, and E is the set of arcs (edges). $R = \{(ti, tj) \mid ti, tj \in E, pm, pn, pk \in V, \exists\ pm \to pn, pn \to pk\}$.

- $R = (ti, tj)$ stands for a high-level rule, where R signifies that event represented by ti can be derived from event represented by tj.
- $pi \to pj$ implies the existence of an arc from node pi to node pj.

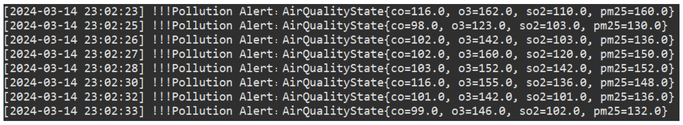
To make our extracted rules more convincing, we need to calculate the probability that each extracted high-level CEP rule is true. We use P to represent probability. We use the Bayesian formula [39] to calculate the probability P. When the value of P is greater than or equal to the median, we consider the corresponding rule to be extractable. By applying this filtering condition, we can retain the rules we require.

In a Petri net, the system state is represented by an array composed of 0 and 1. 0 and 1 typically represent the quantity of tokens within places. 0 indicates the absence of tokens in place, while 1 indicates the presence of a single token in place. When the token quantity in a place meets certain conditions, it triggers the corresponding transition. After a transition occurs, the system's state changes. For example, through the occurrence of transition $t39$, the state changes from (*0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0*) to (*0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0*). Similarly, through the occurrence of transition $t33$, the state changes from (*0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0*) to (*0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0*). Transition $t39$ corresponds to the label $P12$, while transition $t8$ corresponds to the label $P2$. This means there is a causal relationship between the two general rules:

General Rule P12: carbon monoxide > 98.0 and sulfur dioxide < 144.0 and particulate matter <= 128.5.

General Rule P2: carbon monoxide <= 98.0 and ozone > 95.5 and sulfur dioxide > 175.0 and carbon monoxide <= 82.0.

We can obtain a high-level CEP rule $R1$: the air quality condition corresponding to general CEP rule $P12$ leads to the air quality condition corresponding to general CEP rule $P2$. In the $P2$ rules extracted above, the AQI value of "sulfur dioxide" is between $151-200$. This AQI value is labeled as unhealthy by the WHO and will cause harm to the human body. According to $R1$, the values of each air quality index corresponding to the general CEP rule $P12$ meet the requirements of the health range, so the corresponding air quality data is healthy. However, it is followed by unhealthy air quality data, which is the air quality data corresponding to general CEP rule $P2$. Thus, our CEP engine will issue an alert against the general CEP rule $P12$. The early warning results are shown in Figure 10. In this scenario, decision-makers can take preemptive actions such as adjusting production schedules, reducing production activities during peak hours, or transitioning to cleaner and more efficient energy sources such as solar or wind energy. This helps to keep pollutant emissions levels within reasonable limits. Such adjustments not only help mitigate anticipated pollution events but also optimize energy use, reduce reliance on traditional high-polluting energy sources, thereby achieving energy conservation and emission reduction.



```
[2024-03-14 23:02:23] !!!Pollution Alert: AirQualityState{co=116.0, o3=162.0, so2=110.0, pm25=160.0}
[2024-03-14 23:02:25] !!!Pollution Alert: AirQualityState{co=98.0, o3=123.0, so2=103.0, pm25=130.0}
[2024-03-14 23:02:26] !!!Pollution Alert: AirQualityState{co=102.0, o3=142.0, so2=103.0, pm25=136.0}
[2024-03-14 23:02:27] !!!Pollution Alert: AirQualityState{co=102.0, o3=160.0, so2=120.0, pm25=150.0}
[2024-03-14 23:02:28] !!!Pollution Alert: AirQualityState{co=103.0, o3=152.0, so2=142.0, pm25=152.0}
[2024-03-14 23:02:30] !!!Pollution Alert: AirQualityState{co=116.0, o3=155.0, so2=136.0, pm25=148.0}
[2024-03-14 23:02:32] !!!Pollution Alert: AirQualityState{co=101.0, o3=142.0, so2=101.0, pm25=136.0}
[2024-03-14 23:02:33] !!!Pollution Alert: AirQualityState{co=99.0, o3=146.0, so2=102.0, pm25=132.0}
```

Fig. 10: Part of the result of the early unhealthy alert

Similarly, based on the accuracy calculation formula mentioned earlier, we calculate the accuracy of the warnings issued based on the general CEP rules and high-level CEP rules to be 98.6%. From this, we can see that by extracting high-level CEP rules to supplement the general CEP rules and issuing warnings about air quality data based on the combination of both, the accuracy of the warnings can be improved.

Therefore, general CEP rules extracted by our proposed method can be applied to identify abnormal values in air quality data. Detecting air anomalies is closely related to energy-saving decisions. These anomalies may indicate unnecessary energy waste or high energy consumption. The extracted high-level CEP rules can provide early warnings for the air state prior to the impending polluted conditions. By anticipating potential pollution events, it facilitates proactive adjustments and optimization of energy consumption in relevant sectors, allowing proactive measures to be taken before the situation worsens. Hence, analysis based on air data anomaly detection can provide strong support for making energy-saving decisions. It helps to optimize resource utilization, thus contributing to sustainable energy management goals. This mitigates the negative effects on the environment to some degree, fostering the development of a clean and aesthetically pleasing environment.

V. Discussions

We proposed a new framework named DMHC to extract high-level CEP rules in this research. Machine learning methods, HM, and CEP techniques are merged here in this methodology. Our methodology provides a generic framework for extracting high-level CEP rules, which has been overlooked in previous work. In our work, the extracted general and high-level CEP rules can be used for monitoring or warning of air pollution states. This methodology can uncover implicit causal relationships from unlabeled data. It also has a variety of applications. In IoT systems, our methodology can extract meaningful general and high-level CEP rules. High-level CEP rules reflect the causal relationships in air pollution events, as well as the relationships among general CEP rules. Based on the early warning results from these rules, we can intelligently allocate energy consumption within the IoT system to reduce or prevent pollution, thereby promoting sustainable development. Additionally, our work innovatively employs HM for extracting high-level CEP rules, and explores the causal relationships among general CEP rules to derive high-level CEP rules. That stands in contrast to prior research efforts.

VI. Conclusion and Future Work

Within the IoT, there is widespread energy consumption that impacts diverse aspects. Inadequate handling of energy resource allocation and utilization can hinder sustainable development. To some extent, air pollution data can reflect whether energy is being overconsumed or misused. There is a complex causal relationship among air pollution events. A deeper analysis of these relationships will aid in more effectively managing and optimizing energy utilization. Here, we propose DMHC based on machine learning methods, CEP techniques, and HM. Our proposed methodology extracts both general and high-level CEP rules from unlabeled data. These extracted rules reveal the comprehensive causal relationships among data. We apply this methodology to energy-saving within the IoT domain. Monitoring or warning changes in air status to understand energy consumption allows for intelligent energy-saving strategies based on real-time data variations. That contributes to improving energy utilization efficiency and promoting the sustainable development of IoT systems.

Nonetheless, our proposed methodology also has some limitations. Although our proposed system can extract high-level CEP rules, when encountered with complex events, the generated Petri nets might exhibit other intricate structures. In the future, we intend to investigate the extraction of essential rules within even more complex configurations.

References

[1] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, "ifogsim: A toolkit for modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments," *Software: Practice and Experience*, vol. 47, no. 9, pp. 1275–1296, 2017.

[2] J. Panneerselvam, L. Liu, N. Antonopoulos, and Y. Bo, "Workload analysis for the scope of user demand prediction model evaluations in cloud environments," in *2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing*. IEEE, 2014, pp. 883–889.

[3] U. U. Tariq, H. Ali, L. Liu, J. Panneerselvam, and X. Zhai, "Energy-efficient static task scheduling on vfi-based noc-hmpsocs for intelligent edge devices in cyber-physical systems," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 6, pp. 1–22, 2019.

[4] M. Koot and F. Wijnhoven, "Usage impact on data center electricity needs: A system dynamic forecasting model," *Applied Energy*, vol. 291, p. 116798, 2021.

[5] J. Y. Zhu, C. Zhang, H. Zhang, S. Zhi, V. O. Li, J. Han, and Y. Zheng, "pg-causality: Identifying spatiotemporal causal pathways for air pollutants with urban big data," *IEEE Transactions on Big Data*, vol. 4, no. 4, pp. 571–585, 2017.

[6] D. Luckham, "A brief overview of the concepts of cep," *Carbon*, vol. 45, p. 15, 2007.

[7] F. Terroso-Saenz, M. Valdes-Vela, and A. F. Skarmeta-Gomez, "A complex event processing approach to detect abnormal behaviours in the marine environment," *Information Systems Frontiers*, vol. 18, pp. 765–780, 2016.

[8] E. Brazález, H. Macià, G. Díaz, M. Baeza_Romero, E. Valero, and V. Valero, "Fume: An air quality decision support system for cities based on cep technology and fuzzy logic," *Applied Soft Computing*, vol. 129, p. 109536, 2022.

[9] W. Yu, X. Wang, X. Fang, and X. Zhai, "Modeling and analytics of multi-factor disease evolutionary process by fusing petri nets and machine learning methods," *Applied Soft Computing*, vol. 142, p. 110325, 2023.

[10] W. Yu, Y. Wang, L. Liu, Y. An, B. Yuan, and J. Panneerselvam, "A multiperspective fraud detection method for multiparticipant e-commerce transactions," *IEEE Transactions on Computational Social Systems*, 2023.

[11] R. Mousheimish, Y. Taher, and K. Zeitouni, "Automatic learning of predictive cep rules: bridging the gap between data mining and complex event processing," in *Proceedings of the 11th ACM international conference on distributed and event-based systems*, 2017, pp. 158–169.

[12] M. U. Simsek, F. Yildirim Okay, and S. Ozdemir, "A deep learning-based cep rule extraction framework for iot data," *The Journal of Supercomputing*, vol. 77, pp. 8563–8592, 2021.

[13] J. H. Yoon, D. J. Kim, and Y. Y. Koo, "Novel fuzzy correlation coefficient and variable selection method for fuzzy regression analysis based on distance approach," *International Journal of Fuzzy Systems*, vol. 25, no. 8, pp. 2969–2985, 2023.

[14] J. Yang, S. C. Han, and J. Poon, "A survey on extraction of causal relations from natural language text," *Knowledge and Information Systems*, vol. 64, no. 5, pp. 1161–1186, 2022.

[15] M. R. Waldmann and L. Martignon, "A bayesian network model of causal learning," in *Proceedings of the twentieth annual conference of the Cognitive Science Society*. Routledge, 2022, pp. 1102–1107.

[16] K. Nadim, A. Ragab, and M.-S. Ouali, "Data-driven dynamic causality analysis of industrial systems using interpretable machine learning and process mining," *Journal of Intelligent Manufacturing*, vol. 34, no. 1, pp. 57–83, 2023.

[17] G. Felfili, M. H. de Oliveira, and J. de Almeida Martinelli, "Performance evaluation of the 3d printing system through fault tree analysis method (ftam)," *Journal of Building Pathology and Rehabilitation*, vol. 8, no. 2, p. 68, 2023.

[18] Q. Xiao, Y. Li, F. Luo, and H. Liu, "Analysis and assessment of risks to public safety from unmanned aerial vehicles using fault tree analysis and bayesian network," *Technology in Society*, vol. 73, p. 102229, 2023.

[19] J. Nie, J. Jiang, Y. Li, H. Wang, S. Ercisli, and L. Lv, "Data and domain knowledge dual-driven artificial intelligence: Survey, applications, and challenges," *Expert Systems*, p. e13425, 2023.

[20] A. Ragab, M. El Koujok, H. Ghezzaz, M. Amazouz, M.-S. Ouali, and S. Yacout, "Deep understanding in industrial processes by complementing human expertise with interpretable patterns of machine learning," *Expert Systems with Applications*, vol. 122, pp. 388–405, 2019.

[21] U. Ali, S. Bano, M. H. Shamsi, D. Sood, C. Hoare, W. Zuo, N. Hewitt, and J. O'Donnell, "Urban building energy performance prediction and retrofit analysis using data-driven machine learning approach," *Energy and Buildings*, vol. 303, p. 113768, 2024.

[22] Y.-H. Xue, R. Chen, J.-G. Wang, W. Liu, Y. Yao, J.-L. Liu, and H.-L. Chen, "Granger-based root cause diagnosis with improved backward-in-time selection," in *2023 IEEE 12th Data Driven Control and Learning Systems Conference (DDCLS)*. IEEE, 2023, pp. 1853–1858.

[23] Z. Zhang and L. Wu, "Graph neural network-based bearing fault diagnosis using granger causality test," *Expert Systems with Applications*, vol. 242, p. 122827, 2024.

[24] K. Bisht and A. Kumar, "A portfolio construction model based on sector analysis using dempster-shafer evidence theory and granger causal network: An application to national stock exchange of india," *Expert Systems with Applications*, vol. 215, p. 119434, 2023.

[25] Y. Liu, W. Yu, C. Gao, and M. Chen, "An auto-extraction framework for cep rules based on the two-layer lstm attention mechanism: A case study on city air pollution forecasting," *Energies*, vol. 15, no. 16, p. 5892, 2022.

[26] W. Reisig, *Petri nets: an introduction*. Springer Science & Business Media, 2012, vol. 4.

[27] O. Loyola-González, "Process mining: software comparison, trends, and challenges," *International Journal of Data Science and Analytics*, vol. 15, no. 4, pp. 407–420, 2023.

[28] A. J. Weijters, W. M. van Der Aalst, and A. A. De Medeiros, "Process mining with the heuristicsminer algorithm," 2006.

[29] W. Van der Aalst, T. Weijters, and L. Maruster, "Workflow mining: Discovering process models from event logs," *IEEE transactions on knowledge and data engineering*, vol. 16, no. 9, pp. 1128–1142, 2004.

[30] A. Burattin and A. Burattin, "Heuristics miner for time interval," *Process mining techniques in business environments: theoretical aspects, algorithms, techniques and open challenges in process mining*, pp. 85–95, 2015.

[31] G. Ozar, *MySQL management and administration with Navicat*. Packt Publishing Ltd, 2012.

[32] B. F. Van Dongen, A. K. A. de Medeiros, H. Verbeek, A. Weijters, and W. M. van Der Aalst, "The prom framework: A new era in process mining tool support," in *Applications and Theory of Petri Nets 2005: 26th International Conference, ICATPN 2005, Miami, USA, June 20-25, 2005. Proceedings 26*. Springer, 2005, pp. 444–454.

[33] M. Giatsoglou, D. Chatzakou, V. Gkatziaki, A. Vakali, and L. Anthopoulos, "Citypulse: A platform prototype for smart city social data mining," *Journal of the Knowledge Economy*, vol. 7, pp. 344–372, 2016.

[34] X. Dai, W. Shang, J. Liu, M. Xue, and C. Wang, "Achieving better indoor air quality with iot systems for future buildings: Opportunities and challenges," *Science of The Total Environment*, p. 164858, 2023.

[35] N. Devillard, "The eclipse software." *The Messenger, vol. 87, p. 19-20*, vol. 87, pp. 19–20, 1997.

[36] A. Jain, K. Nandakumar, and A. Ross, "Score normalization in multimodal biometric systems," *Pattern recognition*, vol. 38, no. 12, pp. 2270–2285, 2005.

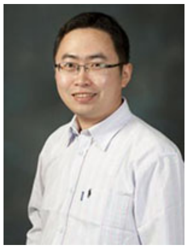
[37] A. T. Teologo, E. P. Dadios, R. G. Baldovino, R. Q. Neyra, and I. M. Javel, "Air quality index (aqi) classification using co and no 2 pollutants: a fuzzy-based approach," in *TENCON 2018-2018 IEEE Region 10 Conference*. IEEE, 2018, pp. 0194–0198.

[38] X. Ye, J. Zhou, and X. Song, "On reachability graphs of petri nets," *Computers & Electrical Engineering*, vol. 29, no. 2, pp. 263–272, 2003.

[39] D. Isa, L. H. Lee, V. Kallimani, and R. Rajkumar, "Text document preprocessing with the bayes formula for classification using the support vector machine," *IEEE Transactions on Knowledge and Data engineering*, vol. 20, no. 9, pp. 1264–1272, 2008.

Wangyang Yu received the M.S. degree in computer software and theory from Shandong University of Science and Technology, Qingdao, China, in 2009, and the Ph.D. degree in computer software and theory from Tongji University, Shanghai, China, in 2014. He is currently an Associate Professor with the College of Computer Science, Shaanxi Normal University, Xi'an, China. He was also a Visiting Scholar with the University of Derby, Derby, U.K., from 2016 to 2017. His research interests include the theory of Petri nets, formal methods in software engineering and trustworthy software.

Jing Zhang obtained her bachelor's degree in Computer Science and Technology from Henan Normal University, China, in 2022. She is currently pursuing a master's degree at the School of Computer Science, Shaanxi Normal University, Xi'an, China. Her research interests encompass the theory of Petri nets, Machine Learning, Process Mining, and CEP. Jing Zhang is dedicated to advancing her knowledge and expertise in these fields as she continues her academic journey.



Lu Liu (Member, IEEE) received the Ph.D. degree from the Surrey Space Centre, University of Surrey, Guildford, U.K. He is the Head of the School of Informatics, University of Leicester, Leicester, U.K., from 2019 to 2023. He had worked as a Research Fellow with the WRG e-Science Centre, University of Leeds, Leeds, U.K., on the EPSRC/BAE funded NECTISE Project and the CoLaB Project which was jointly funded by the EPSRC and the China-863 Program. He has over 120 scientific publications in reputable journals, such as IEEE TRANSACTIONS ON COMPUTERS, IEEE TRANSACTIONS ON SERVICE COMPUTING, and ACM Transactions on Embedded Computing Systems. He has secured many research projects which are supported by U.K. research councils, BIS, and leading U.K. industries. Prof. Liu serves as an editorial board member of six international journals and the guest editor for five international journals. He has chaired over 20 international conference workshops and presently or formerly serves as the program committee member for over 50 international conferences and workshops. He is a Fellow of the British Computer Society.

Yuan Liu obtained his bachelor's degree in Computer Science and Technology from Shaanxi Normal University, Xi'an, China, in 2021. He is currently pursuing the master's degree with the School of Computer Science, Shaanxi Normal University, Xi'an, China. His research interests include the theory of Petri nets, Process Mining, Machine Learning, and Complex Event Processing (CEP). Yuan Liu is committed to expanding his knowledge and expertise in these fields as he continues to pursue his academic journey.

Xiaojun Zhai received the PhD degree from University of Hertfordshire, UK, in 2013. He is currently a senior lecturer in the School of Computer Science and Electronic Engineering, University of Essex. He has authored/co-authored over 100 scientific papers in international journals and conference proceedings. His research interests mainly include the design and implementation of the digital image and signal processing algorithms, custom computing using FPGAs, embedded systems, and hardware/software co-design. He is a BCS, IEEE member, and HEA Fellow.

Ruhul Kabir Howlader is a PhD in Computer Science student at the University of Leicester. His main research area is Federated Learning for Healthcare Informatics. He has served for around four years as an Artificial Intelligence Engineer and Software Engineer at two renowned IT companies in Bangladesh. He has completed MSc in Computer Science from American International University - Bangladesh (AIUB) and BSc in Computer Science and Engineering from East West University (EWU), Bangladesh.