International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7595
***Article Received**: 29 May 2023 **Revised**: 27 July 2023 **Accepted**: 11 August 2023*

CEP-DTHP : A Complex Event Processing using the Dual-Tier Hybrid Paradigm Over the Stream Mining Process

Mayur Panpaliya[1], Nihar Ranjan[2], Arun Algude[3]
[1]Department of Computer Engineering,
JSPM's Rajarshi Shahu College of Engineering,
Pune, Maharashtra, India
mjpanpaliya@gmail.com
[2]Professor, Department of Information Technology,
JSPM's Rajarshi Shahu College of Engineering,
Pune, Maharashtra, India
nihar.pune@gmail.com
[3]Department of Computer Engineering,
JSPM's Rajarshi Shahu College of Engineering,
Pune, Maharashtra, India
arunalgude@gmail.com

Abstract—CEP is a widely used technique for the reliability and recognition of arbitrarily complex patterns in enormous data streams with great performance in real time. Real-time detection of crucial events and rapid response to them are the key goals of sophisticated event processing. The performance of event processing systems can be improved by parallelizing CEP evaluation procedures. Utilizing CEP in parallel while deploying a multi-core or distributed environment is one of the most popular and widely recognized tackles to accomplish the goal. This paper demonstrates the ability to use an unusual parallelization strategy to effectively process complicated events over streams of data. This method depends on a dual-tier hybrid paradigm that combines several parallelism levels. Thread-level or task-level parallelism (TLP) and Data-level parallelism (DLP) were combined in this research. Many threads or instruction sequences from a comparable application can run concurrently under the TLP paradigm. In the DLP paradigm, instruc-tions from a single stream operate on several data streams at the same time. In our suggested model, there are four major stages: data mining, pre-processing, load shedding, and optimization. The first phase is online data mining, following which the data is materialized into a publicly available solution that combines a CEP engine with a library. Next, data pre-processing encompasses the efficient adaptation of the content or format of raw data from many, perhaps diverse sources. Finally, parallelization approaches have been created to reduce CEP processing time. By providing this two-type parallelism, our proposed solution combines the benefits of DLP and TLP while addressing their constraints. The JAVA tool will be used to assess the suggested technique. The performance of the suggested technique is compared to that of other current ways for determining the efficacy and efficiency of the proposed algorithm.

Keywords- Complex event processing (CEP), Hybrid parallelization, DLP, TLP.

I. INTRODUCTION

CEP is a prominent method for detecting arbitrarily complex patterns in enormous streams of data in real-time. It is extensively utilized in many fields where data is continually created in a streaming fashion and must be analyzed quickly and efficiently on handheld devices. CEP is a real-time event-processing solution that analyses and processes streams of events to generate relevant insights and stimulate appropriate actions [1, 2]. It permits organizations to make sense of massive amounts of event data and respond quickly to significant circumstances, opportunities, or threats. Numerous systems develop a constant stream of events in the modern digital world, including sensor readings, log entries, social media updates, financial transactions, user interactions, and more. These occurrences include significant information that, when examined collectively, can indicate patterns, trends, or anomalies that would be overlooked if individual events were examined in isolation [3, 4].

CEP tackles this issue by offering a framework for real-time event processing, correlation, and aggregation. Companies can recognize notable circumstances, discover patterns of interest, and activate actions based on observed occurrences by creating event patterns, rules, and temporal linkages. CEP systems excel in dealing with high-velocity, high-volume event streams, allowing for real-time decision-making and automation [5, 6].

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7595
***Article Received**: 29 May 2023 **Revised**: 27 July 2023 **Accepted**: 11 August 2023*

CEP has a wide range of applications in a variety of sectors. CEP, for instance, can be used in banking for fraud detection, algorithmic trading, and real-time risk assessment. It may monitor sensor data in manufacturing to detect equipment faults and optimize production operations. CEP can collect and analyze real-time data in transportation and logistics to manage supply chains, route trucks effectively, and identify delays or disturbances [7, 8].

A CEP system's key components generally comprise an event processing engine, an event stream ingestion approach, an event pattern language or query syntax, and a real-time response mechanism [9, 10]. The event processing engine examines the incoming event streams, compares them to specified event patterns or rules, and then takes suitable actions or generates warnings depending on the observed patterns [11, 10]. CEP systems also include temporal and contextual processing, allowing for the consideration of time frames, event sequences, and contextual links between events. Organizations may obtain deeper insights into trends, anomalies, and predictive analytics by including historical context and analyzing event patterns across time [13, 14].

Thus, CEP enables organizations to exploit the potential of real-time event data, permitting proactive decision-making, situational awareness, and intelligent automation [15, 16]. Organizations may gain a competitive advantage, increase operational efficiency, and enhance customer experiences by fast processing and analyzing events, recognizing trends, and triggering appropriate responses [17, 18]. The combination of different parallel processing algorithms in a computer system to improve performance and scalability can be described as hybrid parallelization. It makes use of the capabilities of various parallelization techniques to optimize resource utilization and overall system efficiency [19, 20].

A. Problem Statement

The challenge at hand is to create a scalable CEP model that uses hybrid parallelization approaches to successfully process complicated event streams in real-time. The model handles the following troubles: The CEP model should be able to handle large volumes of event streams from many sources. The system should be scalable to support increased data quantities without sacrificing performance or introducing a delay. The model should be able to evaluate events in real-time to deliver timely insights and enable quick decision-making. The system should minimize processing delays and guarantee that events are analyzed and connected within reasonable timescales. The CEP model ought to be fault-tolerant and resilient to failures, offering high availability and data integrity. It should smoothly tolerate system failures, network interruptions, and individual component failures without interfering with event stream processing and analysis. Addressing these issues will result in a scalable and efficient CEP model that can handle large-scale event streams, process complex event patterns, and provide real-time insights for a variety of applications such as fraud detection, anomaly detection, predictive maintenance, and situational awareness.

B. Contribution of the work

CEP offers a robust query and rule-based language for creating intricate event-processing logic. It enables users to communicate complex event queries and rules that define the circumstances and connections between events. The capacity to design and recognize intricate event patterns, sequences, or conditions that represent significant circumstances or conditions is provided. The contributions of our paper are,

- A distinct hybrid-parallel architecture for evenly distributing CEP tasks over several execution units.
- By introducing practical modifications to the underlying hybrid-parallel technique, we may improve pattern recognition performance and resource utilization.
- To boost event processing application performance, a mix of TLP and DLP is used.
- The theoretical performance of the suggested method is provided.
- An experiment is conducted to compare the proposed approach with existing methods.

The remainder of the paper is organized as follows. The second section summaries the existing paper's literature review. Section 3 encompasses the basic ideas of stream data mining, data stream pre-processing, and CEP that will be utilized throughout the paper, and Section 4 reviews and experimentally evaluates the related work. Section 5 concludes and offers future research prospects.

II. LITERATURE SURVEY

This section is the research of various available techniques for Complex event processing, which are used to protect data. Moreover, the primary benefits of existing approaches, together with their drawbacks, are outlined.

Chapnik et al. [21] developed a DALS algorithm for CEP Systems. They demonstrated a load-shedding system for sophisticated event processing in real time. Their method detects overload by collecting statistics. The method makes DALS judgments to remove the events which are less important to maintain a particular latency constraint while minimizing the loss in results quality. They investigated the issue of load shedding in CEP systems. Multiple optimization strategies, like parallelization and pattern rewriting, have been devised to reduce CEP processing time. However, if an unexpected rise in the input event stream exceeds the capacity of the system, these strategies may not be adequate or relevant.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7595
***Article Received**: 29 May 2023 **Revised**: 27 July 2023 **Accepted**: 11 August 2023*

Ramrez et al. [22] utilized convoluted event processing to build a rule-based pre-processing for data stream mining. Their strategy is to express pre-processing methods in event detection rules using an SQL language, providing domain specialists with a straightforward way of handling temporal data. This concept has been realized as a publicly available solution that combines a CEP engine with a library for online data mining. They offer three scenarios in which CEP rules pre-process data streams to add temporal information, alter features, and manage missing values to assess this technique. CEP rules are an effective language for expressing pre-processing activities in a modular and high-level fashion, with a low time and memory cost.

Zhao et al. [23] devised an LS for CEP. In which state and input-based techniques are employed. This approach assesses queries over event streams that might have unexpected input rates. During transient peak moments, exhaustive processing is no longer practicable, if not impossible, and systems have to turn to best-effort query assessment and try optimal result quality while adhering to a latency restriction. However, for CEP queries, this user may be highly dynamic. The impact of refusing a single event might vary significantly depending on the availability of insufficient matches. [23] Offered a state-based technique to input-based LS that discards weak matches.

Yankovitch et al. [24] developed a scalable CEP with an integrated Parallelization Approach. The method is based on a two-tier hybrid approach that combines different levels of parallelism. This multi-layered method dramatically boosts event detection throughput while minimizing latency and memory utilization by applying a fine-grained load-balancing paradigm. An exhaustive experimental examination of several real-world datasets demonstrates that their methodology regularly surpasses CEP parallelism processes are state-of-the-art by a factor of two or three orders of magnitude.

Lan et al. [25] created a Global CEP Mechanism Based on Edge Computing for Real-Time Monitoring of the Internet of Things. They provide a formalized hierarchical complex event model composed of raw, basic, and complex events, which reduces the complexity of event modeling. The paradigm gives sophisticated space and time semantics for specifying flexible complicated events in a programming approach. Based on this concept, they developed a CEP system architecture, with the system deployed at the network edge between sensing devices in the terminal and cloud applications. The intricate event description may be mapped to the CEP rule logic script to detect a likely abnormal occurrence promptly. The created CEP approach was general and may be used in any heterogeneous sensing device and CEP engine.

Flouris et al. [26] formulated a network-wide CEP system for geographically dispersed data sources. This investigation combines the network processing approach, which places processing components (i.e., CEP operators) closer to the sources of their input events, with a push-pull paradigm that limits the overall number of communicated events. They propose optimum detection approaches that attempt to minimize the highest possible bandwidth utilization considering input latency restrictions, in addition to efficient greedy and heuristic algorithmic modifications. At the system level, they indicate how current CEP engines may operate with minimum changes.

III. SYSTEM METHODLOGY

The present research presents a distinctive scalable complex event processing approach with hybrid parallelization that attempts to quickly identify critical events (like threats) in real-time. Our approach relies on statistics acquired to identify overload. The solution decides on load-shedding actions based on data to remove less significant events to maintain particular latency limitations while minimizing the reduction in result quality. Data stream mining, pre-processing, Event detection (CEP), load shedding, optimization, and result visualization are the five primary aspects of the suggested technique. The suggested model's architecture is depicted in Figure 1.

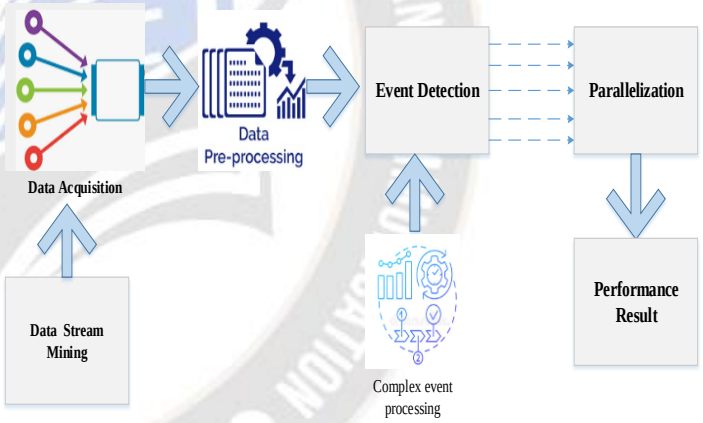

Figure 1. Proposed model.

A. Data stream mining

Data stream mining in CEP is the analysis and processing of ongoing, fast-moving data streams to quickly uncover significant correlations, patterns, and insights. By enabling the analysis of data as it is received, it complements the conventional batch processing and data mining methodologies and enables fast decision-making and proactive measures.

A data stream is an ordered succession of instances that may be unlimited and continually monitored over time. The collection of instances must be handled one at a time or in batches since, from the data perspective, they are not accessible in advance. To make the most efficient use of memory and storage space, each instance can only be visited a certain number of times before being destroyed. This suggests that the decision

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7595
***Article Received**: 29 May 2023 **Revised**: 27 July 2023 **Accepted**: 11 August 2023*

model is developed gradually, including new pertinent information while erasing out-of-date data. In addition to being able to address these problems, stream mining algorithms are anticipated to have low memory desires, real-time responsiveness, and prevent data queuing. Because of the severe constraints under which these algorithms may function, approximations are permissible. The intake of event streams from diverse sources is the first step in CEP data stream mining. In almost real-time, the CEP system collects and processes events. Window-based processing techniques are often employed in data stream mining to analyze a selection of recent occurrences within a sliding or tumbling window. The investigation of temporal patterns and correlations within the stream is possible with window-based processing. In CEP, data stream mining entails detecting complicated event patterns or sequences in the event stream. Identifying established patterns, such as temporal sequences, event hierarchies, or composite events formed out of numerous minor occurrences forms a component of architecture. When a connection is detected, the CEP system continually analyses incoming events against certain patterns and prompts actions or alarms. In data stream mining, stream-based analytics approaches are used to extract insights from the developing event stream [27]. As fresh data emerges, these approaches modify and update models progressively, enabling real-time predictions, anomaly detection, and trend analysis. CEP systems frequently use adaptive models that can respond to changes in the data stream on a whim. Machine learning techniques that adapt to concept drift, where the underlying data distribution varies over time, may be included in these models. Adaptive models guarantee that the CEP system continues to capture developing trends while retaining accuracy.

1) Hybrid sliding and tumbling window

A combination of a sliding and tumbling window is a cutting-edge window-processing system that combines the advantages of sliding and tumbling windows. By involving both temporal continuity and discrete analysis at set intervals, it enables a more flexible and complete study of event streams. The hybrid technique might be effective in situations where real-time insights and periodic analyses are required. Here's how a hybrid sliding and tumbling window works [28,29].

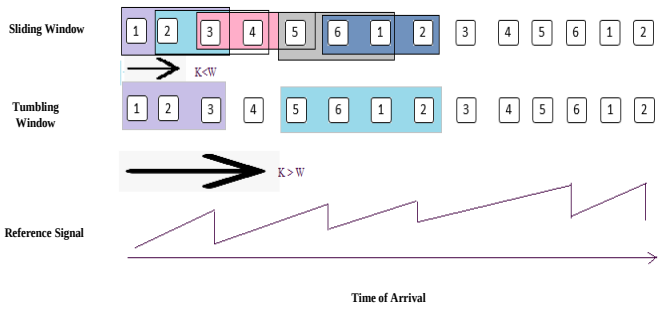

Figure 2. Sliding and tumbling window.

Where k indicates the Shift factor and w denotes the window size. Sliding and tumbling windows are shown in Figure 2. First, the following cases must be separated based on the size of the window and a shift factor k: If the shift factor k is less than the window size w, the result is a sliding window. A tumbling window will occur if the shift factor k is greater than the window size.

a) Configuration of Window

Compute the overall size of the hybrid window, which includes both the sliding and tumbling components. Then, within the combined window, specify the dimensions and durations of the sliding and tumbling windows. The sliding window advances with the arrival of new events, instead the tumbling window resets at preset intervals. Sliding Window Processing keeps a fixed-size buffer for the sliding window, recording the most recent events within its boundaries. Then, within the sliding window, carry out real-time analysis, such as event frequency or instantaneous pattern matching. As new events occur, the oldest events slide out of the sliding window, and the newest events enter. First, define the fixed intervals within the hybrid window for the tumbling window component in tumbling Window Processing. Perform a discrete analysis of the events within the tumbling window at the end of each period. To acquire a full perspective of the event stream, combine the findings of the sliding window and tumbling window analyses. Use the sliding window's real-time information to make quick decisions or take instant action. Use the tumbling window's periodic analysis to detect long-term trends and periodic patterns or execute batch processing on acquired data. The hybrid sliding and tumbling window technique combine the benefits of the sliding window's continuous real-time analysis and the tumbling window's periodic analysis capabilities. It enables the identification of changing patterns, instantaneous response to events, and periodic insights into long-term behavior.

B. Data pre-processing

In this approach, data pre-processing is crucial. It entails processing and converting incoming event data to assure its quality, consistency, and compliance with the CEP system. Data pre-processing involves methods that allow for the efficient adaptation of the content or format of raw data from several,

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7595
***Article Received**: 29 May 2023 **Revised**: 27 July 2023 **Accepted**: 11 August 2023*

sometimes diverse sources. Improving the correctness, completeness, and consistency of data not only simplifies its administration throughout the knowledge discovery process but also improves the efficacy of mining algorithms. Pre-processing approaches should be as light and automated as feasible when working with data streams. Data Cleaning and Data Integration are the two most important phases in pre-processing of data.

1) Data Filtering and Cleaning

Data cleaning is a crucial phase in data pre-processing that entails finding and dealing with missing values, outliers, and noise in the data. Its goal is to increase the quality and dependability of data for later analysis or processing. In the context of a novel scalable CEP model with hybrid parallelization, data cleaning can be performed by identifying and handling missing values in the data, this can happen for a multitude of causes including data collection errors or system failures. Determine a suitable technique for dealing with missing values, such as eliminating missing value records or characteristics if they are not critical to the analysis. A filtering rule seeks to identify events that fulfill certain criteria, which are stated as conditions. Events that failed to meet the constraints are simply discarded. A formal definition is provided by equation (1). In this situation, one instance is produced for each event that meets the rule's k criteria.

$$Z: E(x_1, \dots x_n) \rightarrow I(f_1, \dots f_n) \qquad (1)$$

As a result, the rule translates each event property to the instance feature directly, that is, and n=m. Logical and Selection operators are optimal choices for imposing limitations on event content. The filtering rule can be used to detect events that have no missing values or noise. Secondly, filter rules allow instance selection by simply specifying the criterion to be utilized to separate instances.

2) Data Intergration

Data integration is a vital step in the data pre-processing phase, in which data from numerous sources is merged and turned into a uniform representation suitable for analysis or processing. Data integration in the context of a unique scalable CEP model with hybrid parallelization involves the following major phases, Identify Data Sources: Identify the necessary data sources that must be incorporated into the CEP system and the data source types. Then, research each data source's structure and discover common properties or fields that may be utilized for integration. Map the properties from several data sources to a uniform schema to ensure data compatibility and consistency. Finally, data validation is carried out. To ensure appropriate event processing, examine the integrity and accuracy of the pre-processed data.

C. Complex Event Processing

CEP is the processing and analysis of incoming primitive events. It creates a limited number of output events from a large number of input events. A complicated event is a type of output event. A complex event is formed by combining the needed information from the basic events that contributed to its detection. As a result, the information is condensed. CEP frequently offers the context that converts raw data from a data stream into usable data. CEP is a type of information processing that uses events to recognize situations of interest. A simple event relates to low-level data in CEP terminology. To deduce the occurrence of complicated events, the expert must provide rules, which are specified using patterns that designate the events to pick and the actions that the rule must do. Traditional CEP systems use a publish-subscribe model, in which subscribers gather rule outputs to process further. Complex Event Processing (CEP) processes and analyses incoming primitive events. It creates a limited number of output events from a huge amount of input events. A complicated event is a type of output event. A complex event is formed by combining the needed information from the basic events that contributed to its detection. As a result, the data is condensed. CEP frequently offers the context that converts raw data from a data stream into usable data. Figure 3 depicts the CEP's architecture.

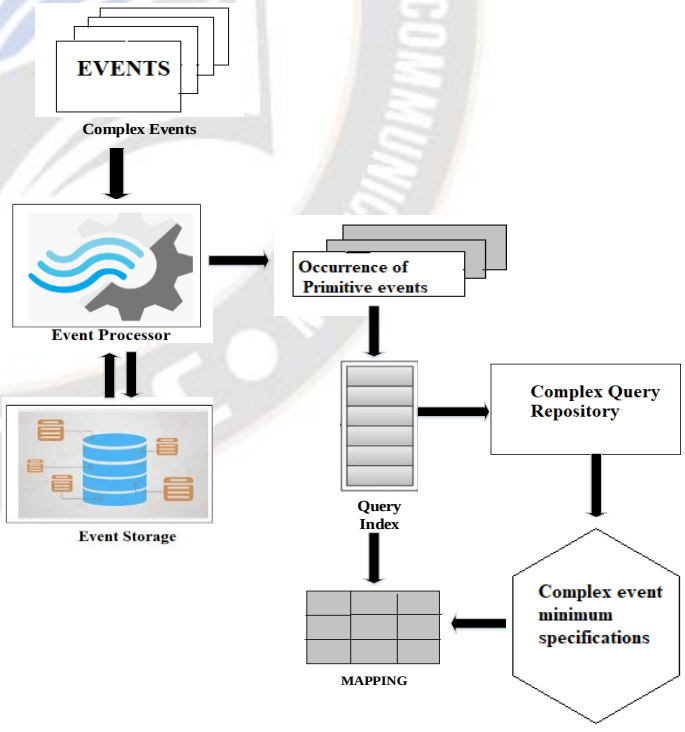

Figure 3. CEP Architecture.

The suggested CEP architecture is depicted in Figure 3. When an event is registered, it is given a new identifier and stored in the query index as a continuous query Qi. The proposed

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7595
***Article Received**: 29 May 2023 **Revised**: 27 July 2023 **Accepted**: 11 August 2023*

technique creates the query index to perform an incremental assessment of continuous inquiries whenever the status changes. The query index treats queries as data and data as questions to undertake an incremental assessment of ongoing inquiries in location-based services. The query index is used in complex event processing. To deal with a large number of complicated events, the recommended technique interprets complex occurrences as continuous inquiries. Then, in the difficult query repository. When a primitive event happens, the query index stores the occurrence history of primitive events in a bitmap. If an uphill struggle is connected to a primitive event in the query index, we examine the complex query repository to verify if the minimal conditions are satisfied. If the conditions for the difficult event are satisfied, we employ the event detection technique to determine it.

1) Complex Event Detection

Before processing, the complicated occurrences must be recorded in the gateway. A query index is generated in the logical grid when a sophisticated event is registered, and a bitmap is allocated to hold the history of the primitive events. One of the existing complex event processing algorithms runs through all possible combinations of complex events when basic events take place to find the complex events. There might be a significant overhead.

The proposed method carries out operations to identify complex occurrences after establishing their bare minimum. The most recent update timings in the nodes corresponding to the registered complicated events are updated when primitive events happen to ensure that the basic requirements are met. The basic needs-checking method is activated when events only take place in the trigger node. One of the basic requirements does not include the detection of complex events. The bitmap is then examined to take action to find the candidate set of difficult events.

D. Parallelization Approach

Optimization methods have been developed to reduce CEP processing time. As a consequence, changes in data arrival rates, system characteristics, and resource availability may cause the system to respond dynamically. One of the most obvious approaches to enhance event-processing application performance is by parallelizing CEP evaluation processes. Several methods for allocating the workload of a CEP system to a large number of execution units and managing their concurrent execution, including multi-core and completely distributed situations, have been developed. These solutions are essentially classified into two categories: data-level parallelism and task-level parallelism.

1) Hybrid TLP and DLP

Two alternative strategies for attaining scalability in complex event processing (CEP) systems are hybrid TLP and DLP. Combining these parallelization methodologies allows for the efficient processing of huge event workloads while maintaining high throughput and low latency. TLP entails breaking down processing jobs in a CEP system into smaller subtasks that may be processed concurrently. Concurrent execution is enabled by assigning each subtask to a different processing node or thread. TLP may be used at several phases of CEP processing, including event intake, pattern matching, and result creation. Incoming events in a TLP technique can be dispersed over numerous nodes or threads for parallel processing. Each processing node handles a subset of the events individually, lowering total processing time. TLP can be used during the pattern-matching phase when event patterns or rules are compared to the incoming event stream. Multiple processing nodes or threads can simultaneously examine distinct subsets of the event patterns, increasing processing speed. TLP can be used to generate the outcomes or complicated events that result from the pattern-matching step. Result event processing can be split across numerous nodes or threads, allowing for parallel execution and faster result development. DLP entails breaking down the data being processed into smaller parts and conducting parallel actions on these units at the same time. DLP can be applied to event streams or event characteristics in the context of CEP to facilitate parallel processing. DLP can be used to divide an incoming event stream into smaller groups that can be handled independently by various processing nodes or threads. Each node is responsible for a subset of events, allowing for parallel processing and lowering total processing time. When performing calculations or actions on specified properties, DLP can be applied to event attributes. For example, if many event characteristics must be aggregated or processed concurrently, they can be partitioned and handled concurrently, enhancing speed.

The execution units of a hybrid-parallel system are separated into two layers, and task allocation occurs in two stages. TLP initially assigns several execution units to each state based on its expected load. Second, inside each state, data-level parallelism (DLP) is used to divide the state's work among the many units.

To achieve scalability in CEP systems, hybrid parallelization blends TLP and DLP approaches. It is feasible to maximize resource utilization and efficiently manage greater event workloads by exploiting both task-level and data-level parallelism. TLP may be used to spread the event intake process over several nodes or threads in hybrid parallelization. DLP may be applied further at each node by separating incoming events into smaller groups for parallel processing. Using TLP, hybrid parallelization can allocate various subsets of event patterns to numerous processing nodes or threads during the pattern-matching phase. DLP may be used within each node to split the

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7595
***Article Received**: 29 May 2023 **Revised**: 27 July 2023 **Accepted**: 11 August 2023*

event stream and assess patterns or rules in parallel. TLP may be used in hybrid parallelization to divide the calculation of result events over many nodes or threads. DLP can be used inside each node to do parallel aggregation or processing on distinct subsets of event data.

The System Performance of the TLP is given as,

$$TLP = \left(\frac{1}{Execution\ Time}\right) * Number\ of\ Parallel\ Tasks \quad (2)$$

The System Performance of the DLP is given as,

$$DLP = \left(\frac{1}{Execution\ Time}\right) * Number\ of\ Parallel\ Data\ Units \quad (3)$$

To achieve scalability, hybrid TLP, and DLP in CEP combine these two parallelization approaches. TLP is concerned with separating processing jobs, whereas DLP is concerned with dividing data. By employing both strategies, the system may maximize resource utilization and effectively manage increasing event workloads.

IV. EXPERIMENTAL RESULTS

The effectiveness of the model is evaluated by measuring the efficiency and scalability of CEP systems that use numerous parallelization techniques to handle and analyze complicated event streams. The system generates a graphical representation of the result with the highest possible accuracy, precision, recall, and root mean square error. Researchers and practitioners may obtain insights into the scalability, efficiency, and effectiveness of complex event processing utilizing hybrid parallelization approaches, allowing them to optimize CEP systems for handling large-scale event streams.

A. Comparative Analysis

In CEP, a comparison study analyses different CEP systems or methodologies based on several criteria. This study can assist you in understanding the strengths, flaws, and applicability of several solutions for your individual needs. The experimental assessment should compare the hybrid parallelization model's performance to that of alternative parallelization methodologies. In terms of throughput, latency, scalability, or resource utilization, the results may highlight the benefits of hybrid parallelization.

1) Throughput

Throughput in this framework refers to the rate at which the CEP system can process and manage incoming events or event patterns. It is a critical performance indicator that assesses the system's capacity to handle events effectively and in real-time. One of the primary benefits of hybrid parallelization is the possibility for enhanced throughput, which allows the CEP model to handle a greater volume of events per unit of time. The experimental results may greatly boost throughput when compared to a non-parallelized or typically parallelized CEP model. Throughput can be impacted by numerous parameters when using hybrid parallelization approaches in CEP, including Parallelization Efficiency, Data Partitioning, connection cost, and Scalability.

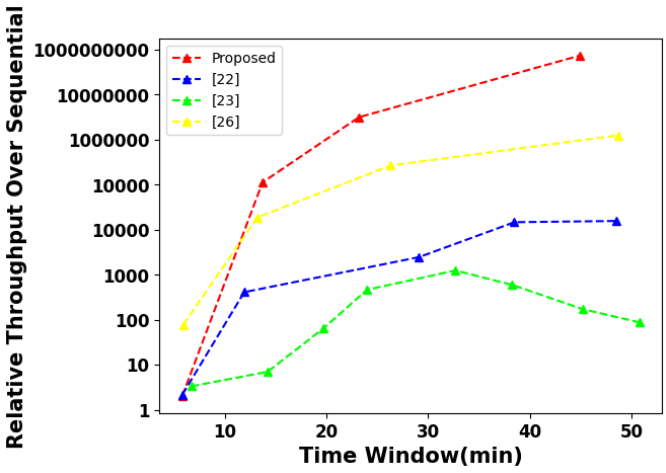

Figure 4a. Relative Throughput Vs Time Window for sliding window.

The performance of the system should be evaluated by altering the event arrival rates and monitoring the throughput attained under various parallelization settings. To determine optimal configurations that maximize throughput in the CEP system, performance analysis should include the influence of parallelization strategies on throughput, scalability, and resource utilization. Figure 4a depicts the relative throughput of a CEP system using a time window versus a sequential approach, comparing the throughput obtained by the CEP system with a time window configuration to the throughput obtained by a sequential CEP system without a time window.

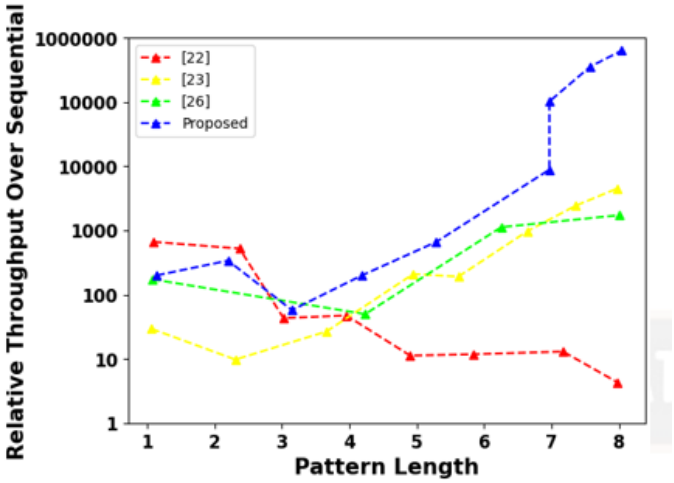

Figure 4b. Relative Throughput Vs Pattern Length for sliding window.

The relative throughput can be affected by how the time frame is configured, such as the length or event count. Different

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7595
***Article Received**: 29 May 2023 **Revised**: 27 July 2023 **Accepted**: 11 August 2023*

window widths may have an impact on the system's capacity to handle events within the timeframe specified. Evaluating different window layouts aids in understanding their impact on relative throughput. If the event arrival rate is high, the time window-based CEP system may have a greater relative throughput by processing events selectively inside the window. However, if the event arrival rate is low, the sequential system's throughput may be equivalent. Figure 4b depicts the sliding window's relative throughput vs Pattern length.

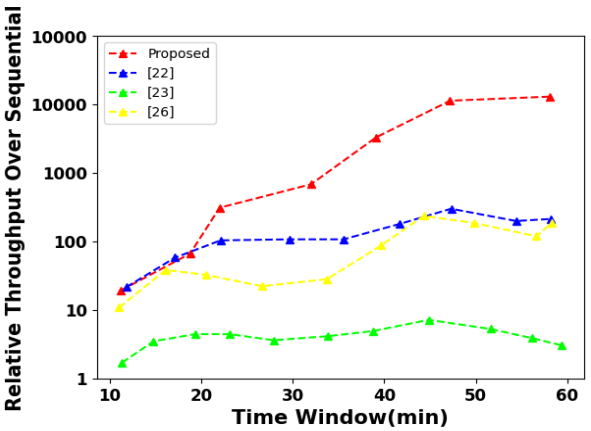

Figure 5a. Relative Throughput Vs Time Window for tumbling window.

Figure 5a depicts the tumbling window's Relative Throughput over the Sequential vs Time Window. The model's Relative Throughput gradually grows as the window time increases. The suggested model outperforms the other current techniques in terms of throughput. A time window in CEP is a method that specifies a time length or event count within which events are evaluated for processing. It enables the CEP system to concentrate on a subset of events within a certain temporal environment.

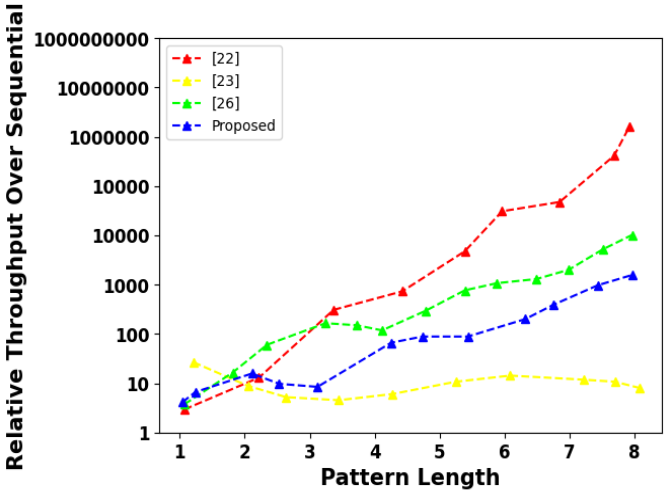

Figure 5b. Relative Throughput Vs Pattern length for tumbling window.

Figure 5b depicts the tumbling window's relative throughput vs Pattern length. When evaluating the relative throughput of a CEP system utilizing a tumbling window to the pattern length, the throughput attained by the CEP system with different pattern lengths inside the tumbling window configuration is referred to. A tumbling window is a time-based window in CEP that divides events into non-overlapping fixed-sized intervals. For event processing, each interval is considered a separate window, and once the window time is achieved, the events within that interval are processed as a group. When the pattern length of the model rises, so does the related throughput value.

2) Latency

File size and latency are two critical elements in CEP that can impact system performance and efficiency. Furthermore, hybrid parallelization can help minimize event processing latency, allowing the CEP model to handle events more quickly. In dealing with huge event loads, experimental findings may show reduced latency values as compared to non-parallelized or conventionally parallelized models. Figure 6 depicts the delay of several models as the file size varies.

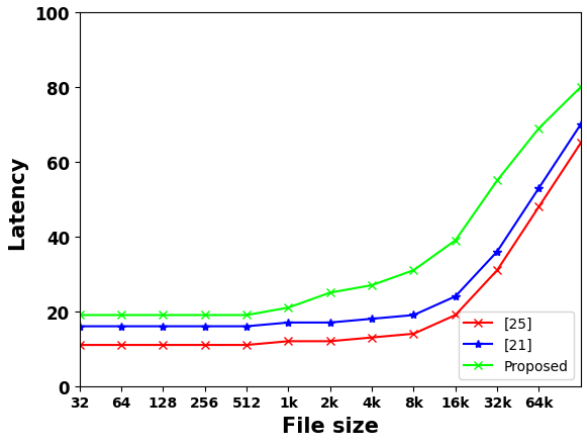

Figure 5. File Size vs Latency.

Figure 6 shows the model's delay curve concerning file size. This graph shows that as the size of the file rises, so does the latency of the corresponding method; the suggested model is compared to other current strategies. As a result, the suggested hybrid parallelization strategy has a greater latency level. Performance testing and benchmarking are recommended to examine the specific file size vs. latency relationship in your CEP system. By methodically adjusting the file size while monitoring the accompanying delay, you may collect empirical data that gives insights into the behavior of your individual CEP system under varied file size situations. This can help you understand the performance characteristics and make educated decisions for optimizing latency in your CEP setup.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7595
***Article Received**: 29 May 2023 **Revised**: 27 July 2023 **Accepted**: 11 August 2023*

3) Optimal Resouce Utilization

Optimal utilization of resources is essential in CEP to enable efficient and effective event processing while maximizing resource utilization. There are various ways for maximizing resource utilization in CEP. Figure 7 depicts a resource utilization vs event count comparison chart. Hybrid parallelization entails making use of both parallelisms and efficiently utilizing existing computational resources. The experimental results suggest that the hybrid parallelization model optimizes resource utilization by making the most use of available processors and memory.

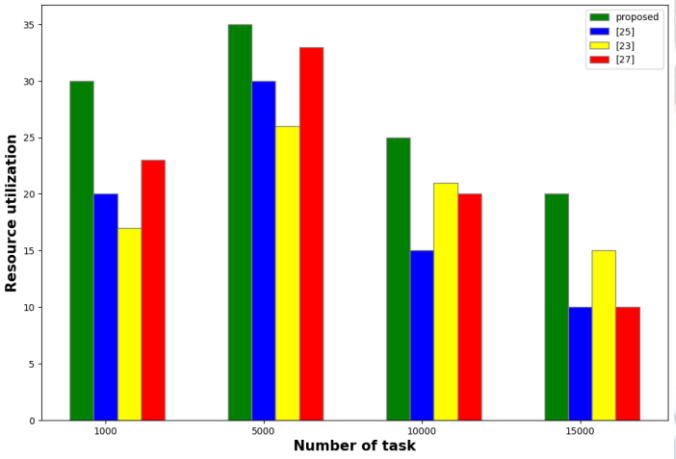

Figure 6. Comparison chart of the resource utilization with the number of events.

B. Trade-offs and Considerations

It's important to consider any trade-offs or limitations associated with hybrid parallelization. Experimental results may uncover factors such as increased overhead or complexity in managing the hybrid parallelization framework. These trade-offs can be evaluated and weighed against the performance benefits to provide a comprehensive assessment of the hybrid approach. Events are generated at a relatively constant rate over time. In this when the number of events processed by the CEP system will increase linearly with time, assuming the system can handle the incoming event load.

Figure 8 illustrates the graph of the number of events vs time in seconds. In CEP, the number of events processed over time can vary significantly depending on, event sources, event generation frequency, and the system's processing capabilities. The relationship between the number of events and time in CEP can be influenced by several factors.

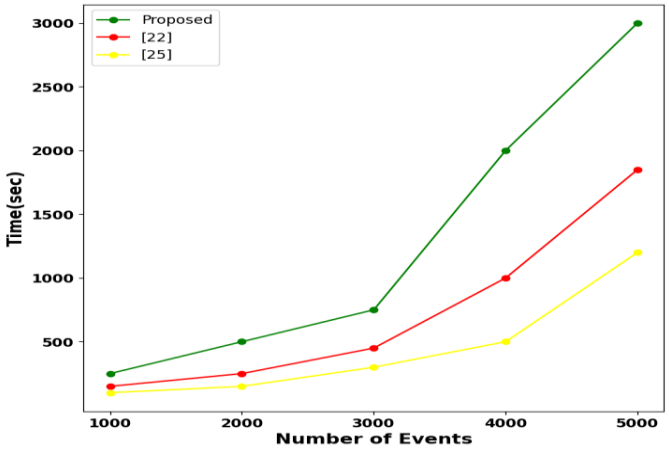

Figure 7. Number of events vs Time(s).

The above figure explains, when the number of events rises, the corresponding time duration of the event also increases. The relative throughput in CEP systems can be influenced by the sequential length (number of events in a sequence) and the pattern length (number of events required to match a pattern).

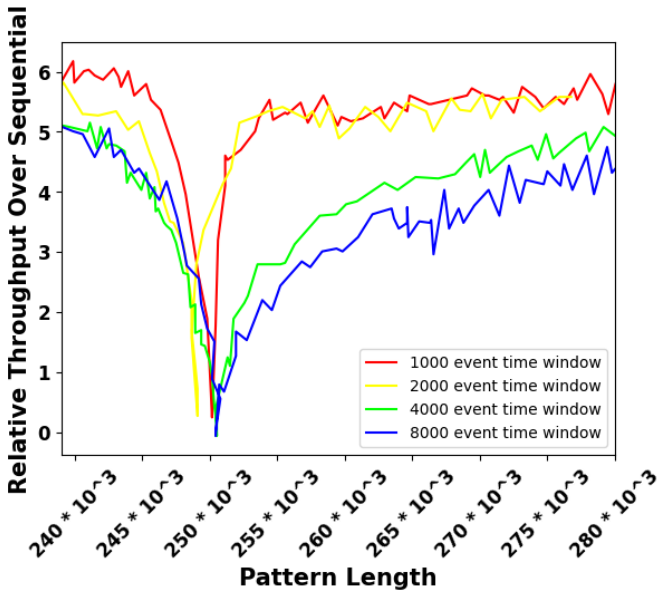

Figure 8. Relative Throughput over sequential vs pattern length.

The sequence and pattern lengths of the model are represented graphically in Figure 9. A CEP system's throughput may be affected by increasing the consecutive duration. Sequential durations that are longer suggest that more events must be gathered and processed before a pattern match can take place. Because the system must wait for a longer sequence to arrive and match the desired pattern, the throughput might therefore drop. The individual requirements and features of the application should be taken into consideration while deciding on the sequential duration and pattern length. Finding the right sequence and pattern durations can be aided by balancing the

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7595
***Article Received**: 29 May 2023 **Revised**: 27 July 2023 **Accepted**: 11 August 2023*

required pattern complexity, event correlation requirements, and allowable throughput.

C. Efficiency

The number of processors or parallel processing units used can have an impact on a CEP system's efficiency. It may be done in a CEP system by utilizing hybrid parallelism, allowing for the simultaneous processing of events and patterns. The system may split the processing effort over more units as the number of processors rises, potentially increasing total efficiency.

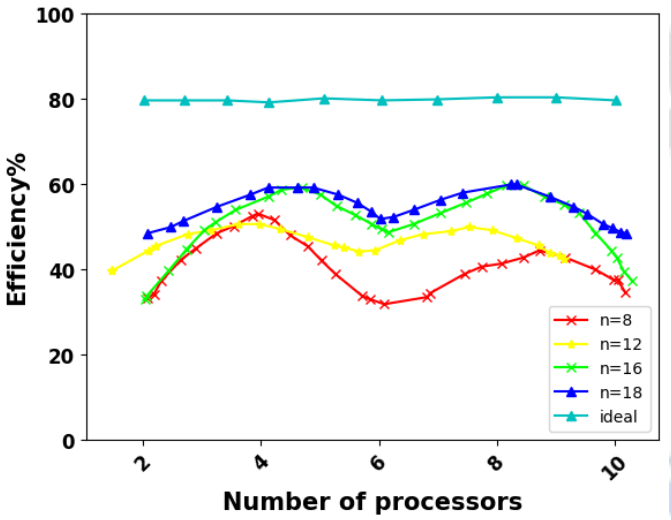

Figure 9. Efficiency of the model concerning the number of processors.

Figure 10 displays the model's efficiency as a percentage. It is essential to evaluate the parallelizability of a model, take into account the computational and communication needs, and choose an appropriate parallelization method to maximize a model's efficiency in the number of processors. Additionally, analyzing the system and keeping an eye on its performance might assist find bottlenecks and enhance efficiency by optimizing parallel execution.

TABLE I. COMPARISON OF EXECUTION TIME, F1 SCORE, AND ACCURACY WITH THE EXISTING TECHNIQUE.

Sr.No.	Model	Accuracy	F1-score	Execution Time
1	Spector	95.3	94.6	47.50 ± 9.56
2	Hypersonic	84.48	85.25	44.04 ± 22.38
3	Stretch	93.33	81.22	42.88 ± 12.22
4	Proposed Model	99.25	98.46	0.77 ± 0.16

Specific information regarding the existing approach is needed to give a full comparison of the accuracy, F1 score, and execution time between Complex Event Processing (CEP) and the present technique. How effectively a system accurately recognizes and categorizes events or patterns is measured by its accuracy. Useful assessment techniques, such as confusion matrices or precision-recall curves, should be used to compare the accuracy of CEP with the current methodology. The F1 score offers a thorough assessment of a system's performance by combining recall and accuracy. The execution times of both strategies must be measured and compared when contrasting CEP with an existing method. Our method is said to be more efficient since it performs event detection or pattern identification with a better degree of precision.

Better performance in terms of recall and precision is indicated by a higher F1 score. And when compared to the already used methodologies, the suggested model's execution time is minimal.

V. CONCLUSION

In our research, we created a novel method for hybrid parallelizing CEP programs. To the greatest extent of our expertise, this model is the first to combine the advantages of data-parallel and task-parallel approaches while removing their considerable disadvantages. Our experiments showed a significant throughput boost compared to cutting-edge methods while achieving lower latency and needing less memory. It permits limitless parallelism since any number of execution units may be allocated to an agent. Furthermore, because the agents interact via the match streams, no complicated synchronization mechanism is necessary. Scalable CEP systems can provide increased throughput, lower latency, and more effective resource utilization by integrating TLP and DLP methods in a hybrid manner. Specific hybrid parallelization implementation specifics and performance advantages may vary depending on the CEP system architecture, event characteristics, and parallelization frameworks or technologies used. When compared to the standard approach, the suggested model achieves the highest accuracy, precision, recall, and lowest root mean square error.

REFERENCES

[1] Xiao, F. (2021). CEQD: A complex mass function to predict interference effects. IEEE Transactions on Cybernetics, 52(8), 7402-7414.

[2] Roldán, J., Boubeta-Puig, J., Martínez, J. L., & Ortiz, G. (2020). Integrating complex event processing and machine learning: An intelligent architecture for detecting IoT security attacks. Expert Systems with Applications, 149, 113251.

[3] Sahal, R., Breslin, J. G., & Ali, M. I. (2020). Big data and stream processing platforms for Industry 4.0 requirements mapping for a predictive maintenance use case. Journal of manufacturing systems, 54, 138-151.

[4] Semlali, B. E. B., El Amrani, C., Ortiz, G., Boubeta-Puig, J., & Garcia-de-Prado, A. (2021). SAT-CEP-monitor: An air quality monitoring software architecture combining complex event processing with satellite remote sensing. Computers & Electrical Engineering, 93, 107257.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7595
***Article Received**: 29 May 2023 **Revised**: 27 July 2023 **Accepted**: 11 August 2023*

[5] Boubeta-Puig, J., Rosa-Bilbao, J., & Mendling, J. (2021). CEPchain: A graphical model-driven solution for integrating complex event processing and blockchain. Expert Systems with Applications, 184, 115578.

[6] Boubeta-Puig, J., Rosa-Bilbao, J., & Mendling, J. (2021). CEPchain: A graphical model-driven solution for integrating complex event processing and blockchain. Expert Systems with Applications, 184, 115578.

[7] Sun, A. Y., Zhong, Z., Jeong, H., & Yang, Q. (2019). Building complex event processing capability for intelligent environmental monitoring. Environmental Modeling & software, 116, 1-6.

[8] Bezerra, E. D. C., Teles, A. S., Coutinho, L. R., & da Silva e Silva, F. J. (2021). Dempster–shafer theory for modeling and treating uncertainty in iot applications based on complex event processing. Sensors, 21(5), 1863.

[9] Moreno, N., Bertoa, M. F., Burgueño, L., & Vallecillo, A. (2019). Managing measurement and occurrence uncertainty in complex event processing systems. IEEE Access, 7, 88026-88048.

[10] Nawaz, F., Janjua, N. K., & Hussain, O. K. (2019). PERCEPTUS: Predictive complex event processing and reasoning for IoT-enabled supply chain. Knowledge-Based Systems, 180, 133-146.

[11] Naseri, M. M., Tabibian, S., & Homayounvala, E. (2022). Adaptive and personalized user behavior modeling in complex event processing platforms for remote health monitoring systems. Artificial Intelligence in Medicine, 134, 102421.

[12] Slo, A., Bhowmik, S., & Rothermel, K. (2020). State-aware load shedding from input event streams in complex event processing. IEEE Transactions on Big Data, 8(5), 1340-1357.

[13] Flouris, I., Giatrakos, N., Deligiannakis, A., & Garofalakis, M. (2020). Network-wide complex event processing over geographically distributed data sources. Information Systems, 88, 101442.

[14] White, M., Hall, K., López, A., Muñoz, S., & Flores, A. Predictive Maintenance in Manufacturing: A Machine Learning Perspective. Kuwait Journal of Machine Learning, 1(4). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/154

[15] Akintoye, S. B., Han, L., Zhang, X., Chen, H., & Zhang, D. (2022). A hybrid parallelization approach for distributed and scalable deep learning. IEEE Access, 10, 77950-77961.

[16] Kumar, S., & Mohbey, K. K. (2022). A review on big data based parallel and distributed approaches of pattern mining. Journal of King Saud University-Computer and Information Sciences, 34(5), 1639-1662.

[17] Lima, M., Lima, R., Lins, F., & Bonfim, M. (2022). Beholder–A CEP-based intrusion detection and prevention systems for IoT environments. Computers & Security, 120, 102824.

[18] Al-Mansoori, A., Abawajy, J., & Chowdhury, M. (2020, May). BDSP in the cloud: scheduling and load balancing utlizing SDN and CEP. In 2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID) (pp. 827-835). IEEE.

[19] Liu, Y., Yu, W., Gao, C., & Chen, M. (2022). An Auto-extraction framework for CEP rules based on the two-layer LSTM attention mechanism: A case study on city air pollution forecasting. Energies, 15(16), 5892.

[20] Ramírez, A., Moreno, N., & Vallecillo, A. (2021). Rule-based preprocessing for data stream mining using complex event processing. Expert Systems, 38(8), e12762.

[21] Gulisano, V., Najdataei, H., Nikolakopoulos, Y., Papadopoulos, A. V., Papatriantafilou, M., & Tsigas, P. (2022). STRETCH: Virtual shared-nothing parallelism for scalable and elastic stream processing. IEEE Transactions on Parallel and Distributed Systems, 33(12), 4221-4238.

[22] Chapnik, K., Kolchinsky, I., & Schuster, A. (2021). DARLING: data-aware load shedding in complex event processing systems. Proceedings of the VLDB Endowment, 15(3), 541-554.

[23] Ramírez, A., Moreno, N., & Vallecillo, A. (2021). Rule-based preprocessing for data stream mining using complex event processing. Expert Systems, 38(8), e12762.

[24] Rajendran, P. S. ., & Kartheeswari , K. R. . (2023). Feature-Based Machine Intelligent Mapping of Cancer Beating Molecules. International Journal of Intelligent Systems and Applications in Engineering, 11(4s), 266–277. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/2652

[25] Zhao, B., Hung, N. Q. V., & Weidlich, M. (2020, April). Load shedding for complex event processing: Input-based and state-based techniques. In 2020 IEEE 36th International Conference on Data Engineering (ICDE) (pp. 1093-1104). IEEE.

[26] Yankovitch, M., Kolchinsky, I., & Schuster, A. (2022, June). Hypersonic: A hybrid parallelization approach for scalable complex event processing. In Proceedings of the 2022 International Conference on Management of Data (pp. 1093-1107).

[27] Lan, L., Shi, R., Wang, B., Zhang, L., & Jiang, N. (2019). A universal complex event processing mechanism based on edge computing for the Internet of Things real-time monitoring. IEEE Access, 7, 101865-101878.

[28] Flouris, I., Giatrakos, N., Deligiannakis, A., & Garofalakis, M. (2020). Network-wide complex event processing over geographically distributed data sources. Information Systems, 88, 101442.

[29] Mayer, R., Slo, A., Tariq, M. A., Rothermel, K., Gräber, M., & Ramachandran, U. (2017, December). SPECTRE: Supporting consumption policies in window-based parallel complex event processing. In Proceedings of the 18th ACM/IFIP/USENIX Middleware Conference (pp. 161-173).

[30] Traub, J., Grulich, P. M., Cuellar, A. R., Breß, S., Katsifodimos, A., Rabl, T., & Markl, V. (2021). Scotty: General and efficient open-source window aggregation for stream processing systems. ACM Transactions on Database Systems (TODS), 46(1), 1-46.

[31] Shaikh, S. A., Matono, A., & Kim, K. S. (2020). A Distance-Window Approach for the Continuous Processing of Spatial Data Streams. International Journal of Multimedia Data Engineering and Management (IJMDEM), 11(2), 16-30.

[32] Nihar Ranjan, Midhun C. (2020). A Brief Survey of Machine Learning Algorithms for Text Document Classification on Incremental Database. TEST Engineering and Management, ISSN: 0193-4120, Volume 83, 25246 – 25251.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7595
***Article Received**: 29 May 2023 **Revised**: 27 July 2023 **Accepted**: 11 August 2023*

[33] Nihar Ranjan, Midhun C. (2021). Evolutionary and Incremental Text Document Classifier using Deep Learning" International Journal of Grid and Distributed Computing Vol. 14, No. 1, 587-595.

[34] Nihar Ranjan, Zubair Ghouse. (2017). A Multi-function Robot for Military Application. Imperial Journal of Interdisciplinary Research (IJIR) Vol-3, Issue-3, ISSN: 2454-1362, 1785-1788.