2021 15th International Conference on Advanced Computing and Applications (ACOMP) | 978-1-6654-0639-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/ACOMP53746.2021.00007

Realtime Bushfire Detection with Spatial-based Complex Event Processing

Thanh Tam Nguyen
Faculty of Information Technology
University of Technology (HUTECH)
Ho Chi Minh City, Viet Nam
nt.tam88@hutech.edu.vn

Thanh Toan Nguyen
Faculty of Information Technology
University of Technology (HUTECH)
Ho Chi Minh City, Viet Nam
nt.toan@hutech.edu.vn

Thanh Cong Phan
Faculty of Information Technology
University of Technology (HUTECH)
Ho Chi Minh City, Viet Nam
ptcongbk@gmail.com

Quang Duc Nguyen
Faculty of Computer Science and Engineering
HCMC University of Technology (VNU-HCM)
Ho Chi Minh City, Viet Nam
nguyenquangduc2000@gmail.com

Quoc Viet Hung Nguyen
School of ICT
Griffith University
Queensland, Australia
henry.nguyen@griffith.edu.au

***Abstract*—Bushfire is the primary destructive force that may cause damage to a large region, a country, or even the Earth. However, as bushfires spread too fast, they are often identified when they cannot control and cause significant damage. The reason is that existing works on remote sensing focus on low-level information processing, and thus, face the challenge of processing a massive amount of data in real-time. In this work, we employ complex event processing (CEP) to extract higher-level information to facilitate real-time bushfire detection. In particular, we propose a *"spatial extension"* to the ready-powerful CEP techniques to enable bushfire monitoring from the combinations of multiple spatial events. We further demonstrate the proposed spatial-based CEP on a real-time bushfire detection problem. Experimental results illustrate that our approach scales well while achieving the competitive detection performance.**

***Index Terms*—bushfire detection, complex event processing, spatial query.**

I. Introduction

Bushfires have been continuing to wreak havoc on people, economies and nature [1], and they are getting more devastating due to the significantly changing of the global climate [2]. The cost of bushfires to the economy is billions of dollars per year. Take Australia as an example, the damage cost of the summer of 2020 only was estimated at $2 billion, and this cost is still climbing [3]. Besides the immediate physical impacts, bushfires also cause long-term consequences to biodiversity, agriculture, and public health. Moreover, bushfires are the primary factor that contributes to global warming and, in the opposite direction, global warming makes the bushfires more intensively. This reciprocal relationship is known as the *"climate feedback loops"*. Therefore, the real-time mechanism to actively monitor the bushfire would significantly reduce unexpected damage to both long term and short term consequences [4] and contribute to the stability of the global environment.

With the emergence of new remote sensing technologies, bushfire detection has been intensively studied using satellite images which is available around-the-clock [5]. However, due to the immense volume of sensing data, real-time detection for a bushfire at a global scale is a notoriously hard task [6]. Many *thresh-hold based* methods have been introduced to solve the problem directly, such as FIMMA [5], GOES-AFP [7], MODIS [6], and VIIRS-AFP [8]. Although these approaches can deal with a rational low-latency and provide a (quasi) real-time monitoring system, they yield less competitive performance due to the simplicity of their approaches. Recently, *deep learning* methods [9], [10] improve the accuracy significantly by capturing information from multiple data sources. These studies focus on raw data input (i.e. low-level information), and thus, they encounter the difficulty of examining a massive volume of sensed data in real-time

Complex Event Processing (CEP) technologies enable the extraction of higher-level information from low-level and high velocity data streams. These data come in the form of *temporal-events*, and CEP correlates large amounts of such events for detecting the situations of interest [11]. As some situations of interest are critical in many domains of application, CEP technologies are widely applied in finance [12], supply chain [13], smart grid [14], traffic management [15]. While current CEP is ideal for monitoring such kind of temporal events, CEP technologies do not support native operators for *spatial-events* such as correlating the distance between two partial points, estimating the intersection between the boundary of two regions. These spatial events would facilitate a wide range of spatial-based applications.

We argue that the support for *spatial-events* in CEP systems is as equally crucial as *temporal-events* in different domains in environmental management. We consider bushfire monitoring in Fig.1 as a typical example. A system considers temperature signal only to estimate a region as a potential fire could lead to false-positive. However, the conclusion for a fire would be more accurate on a sub-region which is the intersection of high temperature and high CO_2 (i.e., smoke) areas. In addition, we

978-1-6654-0639-0/21/$31.00 ©2021 IEEE
DOI 10.1109/ACOMP53746.2021.00007

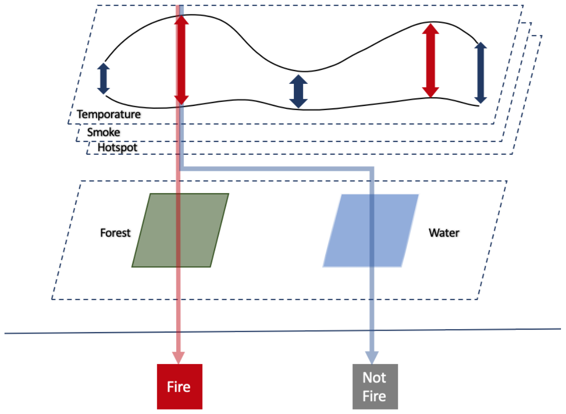


Fig. 1: Spatial events for bushfire detection.

are entirely confident to filter out a fire alert as the false alarm if the combination of these signals arises in water surface rather than a grassland or forest areas.

With these observations, in this work, we explore the feasibility of extending *spatial-events* to the ready-powerful CEP technologies to facilitate a real-time bushfire detection system. To this end, we propose a spatial extension that can integrate with existing CEPs to facilitate the monitoring environment seamlessly. Following a discussion of related works (§II), the contributions of this work are summarised as follows.

- *Reminder on CEP background:* §III provides a brief background on CEP such as the event model, the query language, the meta model, and the native operators.
- *Spatial extension for CEP:* §IV proposes the spatial extension for CEP that supports for spatial and temporal queries simultaneously.
- *Real-time bushfire detection with spatial-based CEP:* §V further employs the proposed spatial-based CEP and implements a prototype system for real-time bushfire detection.

We evaluate our proposed spatial-based CEP with applications on the problem of real-time bushfire detection on four real-world datasets (§VI). The results show that our approach outperforms other baselines both in term of accuracy and timeliness of detection. In addition, we achieve competitive results on other fine-grained metrics. We conclude our paper in §VII.

II. Related Works

In this section, we present background information on the two main topics of this work: *bushfire detection* and *complex event processing*.

A. Bushfire detection

Current works on bushfire detection use raw satellite images, and then estimate the background temperature, and eventually employ some hotspot algorithms [16] to perform the fire detection. For instance, MODIS [6], a threshold algorithm of Moderate Resolution Imaging Spectroradiometer, conducts a variety of processing steps such as cloud masking, land masking, background characterisation before discovering a fire signal using the threshold tests. FIMMA (Fire Identification, Mapping and Monitoring) [5] is another threshold technique from the Advanced Very-High-Resolution Radiometer (AVHRR) system [17] which supports wildfire detecting at night. However, the algorithm performs well in the forest regions only. The VIIRS (Visible Infrared Imaging Radiometer Suite) also propose a technique that is able to work with the MODIS Fire and Thermal Anomalies product [8]. Recent approaches extract the temporal cycle of a day to identify the fire signal using the AHI (Advanced Himawari Imager) system [16]. However, they either yield low accuracy or only can detect when the fires had spread over a large area [16].

Recently, deep learning methods [9], [10] improve the accuracy significantly by capturing information from multiple and heterogeneous data sources. However, these studies focus on raw data input (i.e., low-level information), and thus, they confront the difficulty of investigating a massive amount of data in real-time. Therefore, unlike existing methods, we employed Complex Event Processing (CEP) to extract high-level information from low-level streaming of data input to facilitate the bushfire detection in a real-time manner.

B. Complex Event Processing

Complex Event Processing (CEP) technologies allow processing, analysing and correlating large amounts of temporal events from multiple data sources. CEP techniques could be classified into three main categories [18]: (1) *Logic-based* approaches [19]–[21] reply on logic inference mechanisms to derive the situation of interest from a stream of events that are defined by a formal and declarative language. (2) *Tree-based* approaches [22], [23] employ a non-declarative language to define the event, and the implementation is based on a cost model which is similar to the traditional relational database. (3) *Automata-based* approaches [11], [24], [25] derive the output by iterating through intermediate states of an automata model.

Although existing CEP techniques differ from each other in their implementation models and query language schema, they share a common trait. The common trait is that they do not support spatial operators that can correlate spatial events, e.g. events associated with spatial information. In this work, we go beyond the state-of-the-art CEP techniques and propose a spatial extension for CEP to facilitate environmental monitoring. Particularly in this work, we apply for bushfire detection.

III. Background on Complex Event Processing

Complex event processing (CEP) is a set of methods that support obtaining and monitoring a stream of data to discover the situation of interest in real-time. The situation of interest is user-defined via the form of rules or patterns. In this work, these rules are constructed using the SASE query language [26], but this is not a restriction, and our framework can be applied to any other CEP query languages. Below, we present some definitions which are used in our paper.

A. Preliminaries

Event. We define an event as an activity of interest that happens or contemplates as occurring in a system. Examples of events could be a failure of the IT network, an opening of the door, or an increase of temperature in a particular spatial location.

Event attribute. An event attribute is an element in the event that represents the property or characteristic of an event. Each attribute includes a name and a specific type (Integer, Float, Boolean, or String).

Event type. An event type is a collection of event objects that belong to the same class. Each event type is defined as a structure that comprises a set of attributes, and all event types contain a common timestamp. An example schema for door opening can be described as *(eventId, timestamp, doorId, value)*.

Primitive event. A primitive event is an event that cannot be decomposed to other events. A primitive event can also be referred to as an *atomic* event.

Complex Event. A complex event is a combination of *primitive* events that describe the situation of interest. A complex event is also called a *pattern*, and the system will actively report whenever the pattern is detected.

B. Event Model

In this model, an infinite sequence of events or an *event stream* is fed as the input of the event processing system. Each event is formulated as a *time annotation* and a *payload*. The former indicates the chronological order between events, while the latter reveals additional information about the timing of their appearance. Similar to the concept of type and instance in a programming language, the event model includes the event types representing different classes of events. The complex event processing requires processing events stream from multiple types, which is referred to as *heterogeneous* sources.

C. CEP Query language

The SASE language is a SQL-like structure that supports *correlation, filtering, and transformation* of events. We use SASE language [26] to define how to filter specific events, how to correlate multiple events via value and time constraints, and how to construct the answer via correlated events. The overall formation of the language is defined as follows:

```
[FROM   <streaming events>]
 SEQ  <pattern>
[WHERE  <conditions>]
[WITHIN <window>]
[RETURN <pattern>]
```

The SASE language is a declarative language with semantics associated with each clause. The `FROM` clause specifies the input stream name. It can be omitted, and the system shall select the default input stream. The `SEQ`, `WHERE`, and `WITHIN` clauses are the main block of the query. The situation of interest is defined under the `SEQ` clause. The `WHERE` clause, if it exists, gives value constraints over the input events to satisfy the pattern. A sliding window for time constraint over the event pattern is further specified in the `WITHIN` clause. Finally, the `RETURN` clause yields a complex event for the final result of the query.

D. CEP Meta Model

Given the complex event language, we present the processing model for the language in this section. The query plan for this language comprises of the basic operators including: *sequence scan and construction*, *selection*, *negation*, *window*, and *transformation*.

- *Sequence scan and construction*. Sequence scan and sequence construction handle the predicates specified under `SEQ` clause. This operator aims to convert a stream of events into a stream of sequence, in which each sequence is a unique matching for the pattern of interest.
- *Selection*. Similar to relational database querying, the selection operator filters each sequence by assessing all the predicates specified by `WHERE` clause. If all evaluations are a success, the sequence is transferred to the output.
- *Window*. The window operator examines all conditions under the `WITHIN` clause. It considers each sequence and evaluates whether the temporal correlation between the relevant events is equal or lower than a certain time window specified in advance.
- *Negation*. The negation operator is expressed by '!' and defines in the `SEQ` clause. This operator is beneficial when the non-occurrence of a specific event plays a crucial role in the situation of interest.
- *Transformation*. Finally, the transformation operator transforms the resulting sequence back to a complex event by incorporating all the events in this sequence.

Due to brevity, we omit the detailed implementation of each operator. However, exciting readers may refer to that detailed implementation in the tutorial reported in [27].

IV. Spatial extension for CEP

Complex Event Processing technologies prove their efficiency in rules monitoring over data stream [28]. However, they lack the native support for spatial information of the events, which is vital for the environment and disaster monitoring. To resolve this shortcoming, we propose an extension that takes the standard meta model of CEP as a reference and endows it with the spatial extension. The proposed component enables extended operators that support spatial operations such as: `Union`, `Intersection`, `Distance`, among others.

A. Extended Events and Operators

Besides the standard events (in §III-A) for a CEP system, the extension required for defining additional events as follows.

Spatial Event. We define a spatial event as a primitive event associated with spatial information where the event is reported to occur. The spatial information can be a *location (lat,long)* or a *boundary area*.

Spatial Complex Event. A complex event is a combination of *spatial* events that describe the situation of interest. A complex event is also called a *pattern*, and the system will actively report whenever the pattern is detected.

B. Prototype applications

The extended spatial operators (see Figure 2) for the spatial query can be categorised into two main groups as follows:

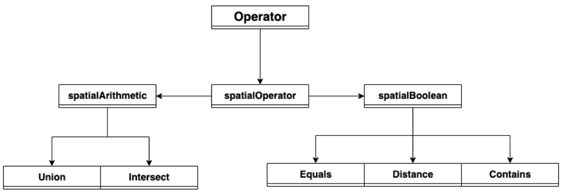


Fig. 2: Extended operators for spatial-based CEP.

- spatialArithmetic: They are those that receive n-operands of the geometry type and return another geometry. These are for example: `Union`, `Intersect`.
- spatialBoolean: They are those operators which receive n-operands and return a `true` or `false` value. These are the classic operators: `Equals`, `Distance`, `Contains`.

The former group of operators aims to return the boundary or area where the pattern occurs. The latter group, in particular, allows indicating whether an event occurs at a given location or area (or not). The proposed model makes it possible to represent spatial events and spatial operators of any type which is independent to any specific CEP platform.

C. Implementation Prototype for Spatial-based CEP

The implementation prototype is extended from the work [29], in which the standard operations for CEP are ready to use. In this section, we present an implementation for the spatial extension. The extensions consist of two main components: the *Query Analyser* and the *Pattern Matcher*.

Query Analyser. CEP system filters the situation of interest defined by the end-user from a continuous stream of events. The SASE language helps to define the situation of interest.

We use Flex [30] to implement the query analyser for the SASE language. Flex performs the query analysing via a deterministic finite automaton (DFA), a well-known theory in computer science that accepts regular languages. In particular, the two components are implemented: the *lexical analysis*, and the *syntax analysis*. The former component parses the query input into the meaningful tokens, and the latter component correlates these tokens to find the relationships between them. Besides, the standard tokens for standard SASE language such as `Pattern, SEQ, Where, Within, Return`, we reserve the additional tokens for our proposed operations such as `Union, Intersect, Equals, Distance, Contains`.

Pattern Matcher. Pattern Matcher is the core processing unit in a CEP system. Pattern Matcher uses the Query Analyser to analyse the pre-defined situation of interest and transform that situation into an automata model. To enable spatial data handling, we also extend the Pattern Matcher to support spatial operations. Below, we demonstrate a sample implementation for the extended data structure of `Polygon` object and `Union` operator. The detailed implementation of other objects (i.e., `Polyline`, and `Point`) and other operators (i.e.,`Intersect, Equals, Distance, Contains`) are omitted due to space limitation. However, the interesting reader could easily re-implement them using the following code snippet.

```
namespace bg = boost::geometry;
typedef bg::model::polygon<bg::model::
d2::point_xy<double>> poly_t;
struct attr_e {
    enum {
        INT64_T,
        DOUBLE,
        POLYGON
    } tag;
    union {
        int64_t i;
        double d;
    }
    poly_t poly;
}
```

Next section, we present a detailed application on real-time bushfire detection. We demonstrate the working mechanism and the automata model behind the spatial-based CEP in a case study of bushfire detection. Although the bushfire detection is selected for demonstration in this work, our proposed spatial-based CEP could be applied to a wide range of environmental change monitoring.

V. Realtime Bushfire Detection with spatial-based CEP

A. Problem Statement

Given a data stream S, and it generates events belong to n types, $E = \{E_1, E_2, ..., E_n\}$. A CEP-based monitoring system then captures events of these types continuously. Each event type is defined as a structure that comprises a set of attributes, and all event types contain a common timestamp. The timestamp collects the time when the event happens. In the following section, we present the event types taken into account in this study; however, our approach is easily adapted to a wide range of other event types for environmental monitoring.

B. Event Type of Interest

Aerosol event. The amount of aerosol is used to precisely correct the atmospheric of remote sensing image, and measure the available solar resource. The aerosol source is constructed by solid convective cells, which are generated after a fire event occurs. This convection might be sufficiently strong to enter the tropopause and cause the injection of aerosol mass from the fire into the lower stratosphere. In addition, absorbing aerosol in the lower stratosphere and the upper troposphere can be uplifted to more significant altitudes from localized heating [31]. While many bushfires have occurred in the past, the effect of the stratospheric aerosol has just been observed in recent decades.

In general, the aerosol event (smoke) can be formulated as:
$Aerosol_{smoke} = c07.high\ AND\ (c07 - c14).high$

Hotspot. The ABI fire algorithm [32] is a thresholding and multi-spectral algorithm that mainly uses the channel 7 and 14 of satellite data to determine hotspot and retrieve their features.

Typically, there are brightness temperature differences between short-wave and long-wave infrared window observations of the clear sky. The reasons for that might be the water vapor attenuation, the reflected solar radiation, and surface emissivity differences. The differences would be more considerable when a portion of pixels (p) is significantly warmer than the remaining pixels (1-p). The hotter portion (p) shall contribute more radiance in the short-wave than the long-wave observations. Figure 3 presents a scan line that extends from a more extraordinary rain forest to a transition zone. When observing the brightness along the scan line, we see a general increase in both channels on different locations with possible fires.

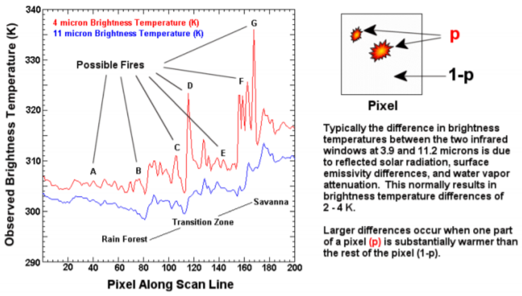


Fig. 3: Hot spot anomaly via water vapor [32].

As these peaks might be associated with a possible fire activity, we formulate the event for hotspot events as follows:
$hotspot = c07.high\ \ OR\ \ (c07 - c14).high$

CO2. Fires from biomass burning cause large quantities of CO2 emitted into the atmosphere. More precisely, it has been suggested that the amount of CO2 caused by biomass burning accounts for as much as 15–30% of global CO2 emissions [33]. The emission of CO2 during the active years of wildfire in the western U.S. accounts for up to 20% of annual emission [34]. Channel 16 in the satellite data is able to absorb CO2, and it has a particular wavelength of $13.3\mu M$. As it is closely associated with the amount of carbon dioxide emitted in the atmosphere, it is also called as CO2 channel, and the event for CO2 is defined as:
$CO2 = c16.high$

Fuel. The fuel loadings can cause more intense fire behavior. Therefore, integrating spatial information of fuel loadings is critical to improving the performance and trustworthiness of the fire detection system [35], [36]. For example, a fire alarm signal should be discarded to avoid a false alarm if it occurs in the water or wildland surface. This study will focus on categorization, which has high chances for bushfires, such as forests and grassland. All other types of land use will be ignored. Follow a proven idea in the work [37]; we classify the data from Landsat 8 using the SVM classifier, and each class represents a spatial event for fuel type.

C. Modeling Bushfire Patterns with Complex Spatial Events

We inherited domain knowledge from multiple disciplines to define the patterns of interest for bushfire [32]. As we have already had the spatial events of interest, the definition of a pattern of interest is straightforward. According to current results [9], because the fire patterns differentiate between day-time hay night-time, we define the pattern for day and night separately as follow.

Pattern for day:

```
PATTERN bushfire-day SEQ (Satellite c07, Satellite c07minusC14, Satellite c16)
WHERE [boundary] AND c07.channelID = 1 AND c07.level = high
    AND c07.time >= 06h00 AND c07.time < 18h00
    AND c07minusC14.channelID = 714 AND c07minusC14.level = high
    AND c16.channelID = 16 AND c16.level = high
WITHIN 2h
RETURN (c16.time , INTERSECT(c07.boundary,c07minusC14.boundary,c16.boundary))
```

Pattern for night:

```
PATTERN bushfire-night SEQ (Satellite c06, Satellite c16)
WHERE [boundary] AND c06.channelID = 6 AND c06.level = high
    AND c06.time >= 18h00 AND c06.time < 06h00
    AND c16.channelID = 16 AND c16.level = high
WITHIN 2h
RETURN (c16.time, INTERSECT(c06.boundary,c16.boundary))
```

A logical prototype of our proposed spatial-based CEP system is presented in Figure 4. The system processes the incoming events in real-time and actively detect the patterns of interest according to the bushfire-day and bushfire-night patterns.

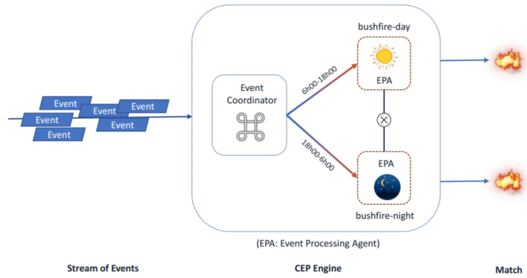
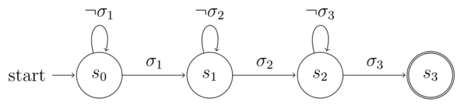


Fig. 4: A sketch solution of realtime bushfire detection.

From the defined pattern, the implemented *Pattern Matcher* analyse and transform the pattern into an automata execution schema for real-time filtering and matching. An example automata execution schema for the bushfire-day pattern is presented in Figure 5. The schema for bushfire-night exposed the same trend; therefore, discard it due to space limitation.



$\boldsymbol{\sigma_1}$: c07.channelID = 7 & c07.level = high
& c07.time >= 06h00 & c07.time < 06h00
$\boldsymbol{\sigma_2}$: c07.channelID = 7 & c07.level = high
& c07minusC14.boundary $\wedge$ c07.boundary
$\boldsymbol{\sigma_3}$: c16.channelID = 16 & c16.level = high
& c16.boundary $\wedge$ c07minusC14.boundary $\wedge$ c07.boundary

Fig. 5: Automata execution schema (bushfire-day).

The detected patterns are further cross-checked with the fuel type events to filter out all of the false-positives before alerting the final alarm.

VI. Experiments

In this section, we present the comprehensive evaluations of our proposed framework. We first demonstrate the experimental setting (§VI-A), and then evaluate various aspects of the approach as follows:

- The effect of real-time bushfire detection (§VI-B).
- The end-to-end performance of the framework (§VI-C).
- The real-time detection at pixel-wised level (§VI-D).
- The correctness of fire localisation (§VI-E).

A. Experimental setting

Dataset. We assess the proposed framework on different real-world benchmark datasets as follows.

- *County Fire:* The County Fire started on the Eastern side of the Berryessa Lake in Napa County and Yolo County in 2018. The total damage caused by this fire was approximately $365km^2$ of burned area [38].
- *Camp Fire:* The Camp Fire occurred in California in 2018. This fire was one of the most destructive fires in the state's history that covered an area of 620 km^2 [39].
- *Woolsey Fire:* The Woolsey Fire happened in Ventura Counties of California and Los Angeles in 2018. It was a significantly destructive wildfire that burned about 392 km^2 [40].
- *Kincade Fire:* The Kincade Fire started in Sonoma County, California in 2019. The fire covered above 314 km^2 [41] of the destroyed area.

The key characteristics of these datasets are summarised in (Table I).

TABLE I: Characteristics of the datasets

Dataset	Time	Size	Ratio [1]
County	14:12 30.06.2018	265 × 266	3,381,234 : 2,286
Woolsey	14:22 08.11.2018	110 × 145	761,854 : 2,018
Camp	06:33 08.11.2018	290 × 300	6,252,876 : 11,124
Kincade	21:24 23.10.2019	265 × 266	3,381,728 : 1,792

[1] The ratio between non-fire over fire pixels.

Baselines. We test our proposed framework against several competing baselines as follows:

- *GOES-AFP:* is the state-of-the-art technique [7] that uses the GOES system.
- *Threshold-based:* An ensemble of threshold-based methods [6], [8].
- *CNN-based:* is the deep learning technique for bushfire detection [9].

Metrics. We employ *Weighted F1-score* as the main metric, and we employed the class ratio as the weight factor [42]. Different from the *Accuracy* metric, Weighted F1-score captures the imbalanced class distribution of the employed datasets. We further evaluate our framework with more fine-grained metrics as follows.

- *Lag time:* the delay between the time of detection and the time when bushfire actually happens.
- *Distance-based:* To investigate the ability of fire localisation of detecting technique, we consider the distance between the predicted result in comparison with the real fire. Here, the *image Euclidean distance* metric [43], [44] is employed as it is proven to be efficient to small variants, and robust to assess the correctness of fire localisation of our method.

B. Real-time Bushfire Detection

To evaluate our model's effectiveness in the real-time detection of bushfire, we experiment on the streaming of events and the output is captured every minute. The results are shown in Figure 7.

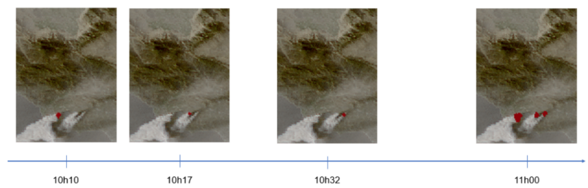


Fig. 7: Camp Fire prediction by our model.

We see that the model can localise the fire and raise the alarm with a short interval (i.e., several minutes). An interesting finding is that the model can successfully detect multiple fires, although these fires happened quickly. For example, the fire at 10h32 and the fire at 11h00 arose within a time frame of fewer than 30 minutes. However, the model successfully identified them as separate fires in different regions, thanks to the support for the spatial query of the spatial-based CEP model. The detection of multiple fires is crucial as it increases the reliability of the system.

C. End-to-end Comparison

To investigate the efficiency of our framework, we assess the performance of all competing methods in an end-to-end manner against two different metrics: (i) (i) *Weighted F1* and (ii) the *Lag time (h)*. Table II depicts the detailed comparison.

In general, our framework consistently outperforms other baselines over all considered metrics. GOES-AFP performs the worst as it adopts a greedy strategy in attracting all possible wildfire signals, which might raise false alarms [7].

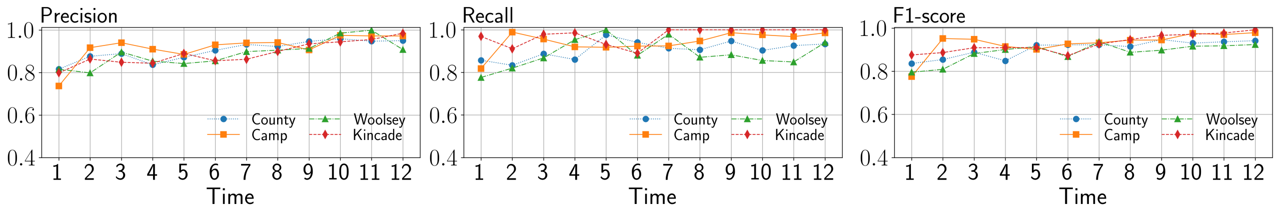


Fig. 6: Pixel-wise detection

TABLE II: Region-level alarm.

	Weighted F1	Lag time (h)
GOES-AFP	85.5%	4 - 8
Threshold	91.7%	3 - 14
CNN-based	93.89%	2.6
Ours	**94.39%**	**0 - 0.24**

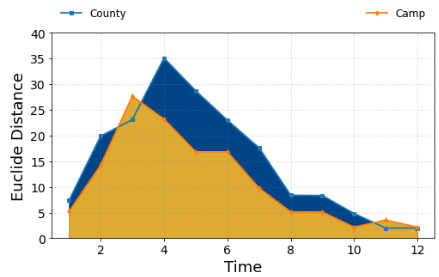


Fig. 8: Difference between actual pixels and detected pixels.

Although threshold and CNN-based methods yield a moderate performance and could be competitive with our performance, they suffer the high latency (i.e., the high lag time).

D. Real-time pixel-wise monitoring

Because a bushfire is a spatial event that spreads over time, looking at the pixel level of the bushfire would help us understand how it develops and increase the chance to mitigate it in the future. To achieve this goal, we conduct a set of pixel-wise predictions during the lifetime of a bushfire.

Figure 6 reveals the result. The difference between the starting time and current time of a bushfire is shown in the X-axis, and predicted results in terms of Precision, Recall, and Weighted-F1 is presented in the Y-axis. In general, the performance significantly fluctuates at starting and begins to be stable from the center of the lifetime of the fire. Such phenomenon can be explainable because when a fire grows into intensive, the spatial dependency between neighboring pixels is more visible and enables more precise predictions.

E. Fire localisation

In spatial analysis application, the prediction could be different from the actual ground-truth by a reasonable margin. The lower of the offset, the better of the performance. Therefore, in this experiment, we measure how difference between the actual locations of the fires compared to the detected pixels by our framework.

Figure 8 depicts the result, in which Y-axis is the *Euclidean distance* and the X-axis is the timeline of the bushfires. At the starting point, the offset is high as some necessary events are insufficient. The distance gap then reduces and matures at a stable level. However, it cannot reaches zero. This behavior could be interpreted by the integration errors and the resolution of the satellite data stream.

VII. Conclusion

In this paper, we proposed a spatial extension to existing Complex Event Processing (CEP) techniques in order to monitor the situations of interest related to the environment. The extension allows to define spatial events and correlate the events based on their spatial information. We demonstrate the effectiveness of the spatial-based CEP on real-time bushfire detection problem. The empirical results show that our detection approach outperforms other baselines on a variety of metrics. In future, we intend to develop a generic framework that supports environmental monitoring in different aspects such as water pollution, water surface and land-use change over time [45]–[53].

Acknowledgment

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.01-2019.323.

References

[1] S. Pascoe, “The 2009 victorian bushfires royal commission: Lessons for the conduct of inquiries in australia,” *AJPA*, vol. 69, no. 4, pp. 392–400, 2010.
[2] D. J. Wuebbles, D. W. Fahey, and K. A. Hibbard, “Climate science special report: fourth national climate assessment, volume i,” *US Global Change Research Program*, 2017.
[3] J. Irvine, “Economic cost of bushfires,” 2020.
[4] USGCRP, “Impacts, risks, and adaptation in the united states: Fourth national climate assessment, volume ii,” *US Global Change Research Program*, 2018.
[5] Z. Li, Y. J. Kaufman, C. Ichoku, R. Fraser, A. Trishchenko, L. Giglio, J. Jin, and X. Yu, “A review of avhrr-based active fire detection algorithms: Principles, limitations, and recommendations,” *Global and regional vegetation fire monitoring from space, planning and coordinated international effort*, pp. 199–225, 2001.
[6] L. Giglio, W. Schroeder, and C. O. Justice, “The collection 6 modis active fire detection algorithm and fire products,” *RSE*, vol. 178, pp. 31–41, 2016.
[7] W. Xu, M. Wooster, G. Roberts, and P. Freeborn, “New goes imager algorithms for cloud and active fire detection and fire radiative power assessment across north, south and central america,” *RSE*, vol. 114, no. 9, pp. 1876–1895, 2010.

[8] W. Schroeder, P. Oliva, L. Giglio, and I. A. Csiszar, "The new viirs 375 m active fire detection data product: Algorithm description and initial assessment," *RSE*, vol. 143, pp. 85–96, 2014.
[9] N. T. Toan, P. T. Cong, N. Q. V. Hung, and J. Jo, "A deep learning approach for early wildfire detection from hyperspectral satellite images," in *RiTA*, pp. 38–45, IEEE, 2019.
[10] T. C. Phan, T. T. Nguyen, T. D. Hoang, Q. V. H. Nguyen, and J. Jo, "Multi-scale bushfire detection from multi-modal streams of remote sensing data," *IEEE Access*, vol. 8, pp. 228496–228513, 2020.
[11] N. P. Schultz-Møller, M. Migliavacca, and P. Pietzuch, "Distributed complex event processing with query rewriting," in *DEBS*, pp. 1–12, 2009.
[12] K. Teymourian, M. Rohde, and A. Paschke, "Knowledge-based processing of complex stock market events," in *EDBT*, pp. 594–597, 2012.
[13] T. Tran, C. Sutton, R. Cocci, Y. Nie, Y. Diao, and P. Shenoy, "Probabilistic inference over rfid streams in mobile environments," in *ICDE 2009*, pp. 1096–1107, 2009.
[14] R. C. Fernandez, M. Weidlich, P. Pietzuch, and A. Gal, "Scalable stateful stream processing for smart grids," in *DEBS*, pp. 276–281, 2014.
[15] A. Artikis, M. Weidlich, F. Schnitzler, I. Boutsis, T. Liebig, N. Piatkowski, C. Bockermann, K. Morik, V. Kalogeraki, J. Marecek, *et al.*, "Heterogeneous stream processing and crowdsourcing for urban traffic management.," in *EDBT*, vol. 14, pp. 712–723, 2014.
[16] B. Hally, L. Wallace, K. Reinke, S. Jones, and A. Skidmore, "Advances in active fire detection using a multi-temporal method for next-generation geostationary satellite data," *IJDE*, vol. 12, no. 9, pp. 1030–1045, 2019.
[17] S. Flasse and P. Ceccato, "A contextual algorithm for avhrr fire detection," *IJRS*, vol. 17, no. 2, pp. 419–424, 1996.
[18] A. Artikis, A. Margara, M. Ugarte, S. Vansummeren, and M. Weidlich, "Complex event recognition languages: Tutorial," in *DEBS*, pp. 7–10, 2017.
[19] D. Anicic, P. Fodor, S. Rudolph, R. Stühmer, N. Stojanovic, and R. Studer, "A rule-based language for complex event processing and reasoning," in *RR*, pp. 42–57, 2010.
[20] F. Chesani, P. Mello, M. Montali, and P. Torroni, "A logic-based, reactive calculus of events," *Fundamenta Informaticae*, vol. 105, no. 1-2, pp. 135–161, 2010.
[21] A. Artikis, M. Sergot, and G. Paliouras, "An event calculus for event recognition," *TKDE*, vol. 27, no. 4, pp. 895–908, 2014.
[22] M. Liu, E. Rundensteiner, K. Greenfield, C. Gupta, S. Wang, I. Ari, and A. Mehta, "E-cube: multi-dimensional event sequence analysis using hierarchical pattern query sharing," in *SIGMOD*, pp. 889–900, 2011.
[23] Y. Mei and S. Madden, "Zstream: a cost-based query processor for adaptively detecting composite events," in *SIGMOD*, pp. 193–206, 2009.
[24] J. Agrawal, Y. Diao, D. Gyllstrom, and N. Immerman, "Efficient pattern matching over event streams," in *SIGMOD*, pp. 147–160, 2008.
[25] G. Cugola and A. Margara, "Tesla: a formally defined event specification language," in *DEBS*, pp. 50–61, 2010.
[26] D. Gyllstrom, E. Wu, H.-J. Chae, Y. Diao, P. Stahlberg, and G. Anderson, "Sase: Complex event processing over streams," *arXiv preprint cs/0612128*, 2006.
[27] A. Artikis, A. Margara, M. Ugarte, S. Vansummeren, and M. Weidlich, "Tutorial: Complex event recognition languages," in *DEBS*, pp. 7–10, 2017.
[28] E. Wu, Y. Diao, and S. Rizvi, "High-performance complex event processing over streams," in *SIGMOD*, pp. 407–418, 2006.
[29] B. Zhao, N. Q. V. Hung, and M. Weidlich, "Load shedding for complex event processing: input-based and state-based techniques," in *ICDE*, pp. 1093–1104, 2020.
[30] J. Levine, *Flex & Bison: Text Processing Tools*. " O'Reilly Media, Inc.", 2009.
[31] A. De Laat, D. C. Stein Zweers, R. Boers, and O. N. Tuinder, "A solar escalator: Observational evidence of the self-lifting of smoke and aerosols by absorption of solar radiation in the february 2009 australian black saturday plume," *JGR: Atmospheres*, vol. 117, no. D4, 2012.
[32] "Goes algorithms," 2013.
[33] H. Lee, "Intergovernmental panel on climate change," 2007.
[34] S. Urbanski, W. Hao, and B. Nordgren, "The wildland fire emission inventory: western united states emission estimates and an evaluation of uncertainty," *ACP. 11: 12973-13000.*, pp. 12973–13000, 2011.
[35] E. Chuvieco, I. González, F. Verdú, I. Aguado, and M. Yebra, "Prediction of fire occurrence from live fuel moisture content measurements in a mediterranean ecosystem," *IJWF*, vol. 18, no. 4, pp. 430–441, 2009.
[36] R. D. Ottmar and E. Alvarado, "Linking vegetation patterns to potential smoke production and fire hazard," in *SNSS.*, vol. 193, pp. 93–96, 2004.
[37] K. Jia, X. Wei, X. Gu, Y. Yao, X. Xie, and B. Li, "Land cover classification using landsat 8 operational land imager data in beijing, china," *Geocarto International*, vol. 29, no. 8, pp. 941–951, 2014.
[38] Wikipedia, "County fire." https://en.wikipedia.org/wiki/County_Fire, 2018.
[39] C. Gov, "Camp fire." https://www.fire.ca.gov/incidents/2018/11/8/camp-fire/, 2019.
[40] Wikipedia, "Woolsey fire." https://en.wikipedia.org/wiki/Woolsey_Fire, 2018.
[41] Wikipedia, "Kincade fire." https://en.wikipedia.org/wiki/Kincade_Fire, 2019.
[42] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, pp. 1097–1105, 2012.
[43] L. Wang, Y. Zhang, and J. Feng, "On the euclidean distance of images," *TPAMI*, vol. 27, no. 8, pp. 1334–1339, 2005.
[44] J. Gou, H. Ma, W. Ou, S. Zeng, Y. Rao, and H. Yang, "A generalized mean distance-based k-nearest neighbor classifier," *ESWA*, vol. 115, pp. 356–372, 2019.
[45] T. T. Nguyen, M. T. Pham, T. T. Nguyen, T. T. Huynh, Q. V. H. Nguyen, T. T. Quan, *et al.*, "Structural representation learning for network alignment with self-supervised anchor links," *ESWA*, vol. 165, p. 113857, 2021.
[46] T. T. Nguyen, M. Weidlich, H. Yin, B. Zheng, Q. H. Nguyen, and Q. V. H. Nguyen, "Factcatch: Incremental pay-as-you-go fact checking with minimal user effort," in *SIGIR*, pp. 2165–2168, 2020.
[47] T. T. Nguyen, T. T. Nguyen, T. T. Nguyen, B. Vo, J. Jo, and Q. V. H. Nguyen, "Judo: Just-in-time rumour detection in streaming social platforms," *Information Sciences*, vol. 570, pp. 70–93, 2021.
[48] B. Zhao, H. van der Aa, T. T. Nguyen, Q. V. H. Nguyen, and M. Weidlich, "Eires: Efficient integration of remote data in event stream processing," in *SIGMOD*, pp. 2128–2141, 2021.
[49] N. T. Tam, H. T. Dat, P. M. Tam, V. T. Trinh, N. T. Hung, Q.-T. Huynh, and J. Jo, "Monitoring agriculture areas with satellite images and deep learning," *Applied Soft Computing*, p. 106565, 2020.
[50] C. T. Duong, T. T. Nguyen, H. Yin, M. Weidlich, S. Mai, K. Aberer, and Q. V. H. Nguyen, "Efficient and effective multi-modal queries through heterogeneous network embedding," *TKDE*, 2021.
[51] N. T. Tam, H. T. Trung, H. Yin, T. Van Vinh, D. Sakong, B. Zheng, and N. Q. V. Hung, "Entity alignment for knowledge graphs with multi-order convolutional networks," in *ICDE*, pp. 2323–2324, 2021.
[52] N. T. Tam, M. Weidlich, B. Zheng, H. Yin, N. Q. V. Hung, and B. Stantic, "From anomaly detection to rumour detection using data streams of social platforms," *PVLDB*, vol. 12, no. 9, pp. 1016–1029, 2019.
[53] T. T. Nguyen, M. Weidlich, H. Yin, B. Zheng, Q. V. H. Nguyen, and B. Stantic, "User guidance for efficient fact checking," *PVLDB*, vol. 12, no. 8, p. 850–863, 2019.