Multi-Scale Bushfire Detection From Multi-Modal Streams of Remote Sensing Data

THANH CONG PHAN1, THANH TAM NGUYEN2, QUOC VIEU HUNG NGUYEN3, AND JUN JO3

1Faculty of Information Technology, Ho Chi Minh City University of Technology (HUTECH), Ho Chi Minh City 700000, Vietnam
2Hanoi University of Science and Technology, Hanoi 100000, Vietnam
3Griffith University, Gold Coast Campus, Southport, QLD 4215, Australia

Corresponding author: Thanh Tam Nguyen (nt.tam88@hutech.edu.vn)

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

ABSTRACT Bushfire is a destructive force that would change the course of a country and even the Earth. They are causing casualties and affect the quality of life of millions of people. Governments are calling for remote sensing methods to monitor and detect active bushfires around-the-clock. To fulfill this call, we develop a remote sensing framework on top of imagery satellite streams to monitor and detect bushfire timely before beyond control. However, detecting bushfires from satellite images needs to take into account several aspects including spatial pattern of fire-positive pixels, temporal dependencies, spectral correlation between channels, and adversarial effects. In this article, we propose a multi-scale deep neural network model that combines both satellite images and weather data for detecting and locating bushfires at both image and pixel level. We illustrate that the weather information with careful spatio-temporal alignment can be utilised to augment imagery data. Experiments on real-world datasets show that the proposed model is better than the baselines with 93.4% accuracy and detects bushfires 1.2 times faster. It is also robust to the effects of cloud and night-time.

INDEX TERMS Bushfire detection, multi-modal learning, remote sensing, spatio-temporal-spectral data, novel deep neural networks.

I. INTRODUCTION

Bushfire is a destructive force that would change the course of a country and even the Earth. It is estimated to cost about $2B to mitigate its damage every year [1]. Bushfires cause casualties and affect the quality of life of millions of people. Governments are calling for remote sensing methods to monitor and detect active bushfires around-the-clock, which is far cheaper than damage repair cost [1]. Bushfire detection is challenging as it requires ecology knowledge as well as algorithmic models [1].

Detecting bushfires from surveillance data and social media data is an active research direction [2]–[6]. However, there is a gap between the availability of these data and the timeliness of detection. Bushfires are often detected when already happened since these data are generated from post-observation feedback rather than environmental changes. Our work goes beyond these limitations by using around-the-clock streams of satellite imagery and weather measurements. Such remote sensing data cover a wide spatial region in near real-time [7]. They capture images periodically in multiple spectral channels, containing complementary information (e.g. infrared (IR) channels) of the observed locations as well as temporal information; and thus can effectively detect bushfires at the pixel level.

Bushfire mitigation specialists are having difficulties finding effective detection methods with high accuracy [8]. Traditional image-based techniques endures the curse of dimensionality when trying to use multiple spectral channels at the same time [9]. They often resort to use only one or few channels and sometimes ignore the temporal dimension [10], hindering the detection power to fires that already happen for a while (larger radiative power) and clear conditions (small measurement error of smoke emission) [11]. Another issue of existing monitoring systems is the false alarms, due to the lack of supporting information such as measurement error, fire radiative power, and the indistinguishability between a cloud and a smoke emission.
To overcome the above limitations, we propose a deep learning model on top of multi-modal data of satellite images and weather measurements. In particular, we develop a multi-level neural network on top of convolutional neural network (CNN) and long short-term memory (LSTM) architectures [12] to capture spectral-spatio-temporal dimensions of satellite data. To further avoid the measurement errors of satellite data, we follow a multi-modal approach that augments satellite images with weather data. The underlying intuition is that bushfires are more likely to happen during hot and dry days. Moreover, nearby weather meters, if close enough, provide additional signature of fire, such as temperature increasing and humidity decreasing [13].

Developing such a multi-modal and autonomous bushfire detection framework faces several challenges. First, the sampling rate of satellite data and weather data is different, making it difficult to align them in the same time frame for an accurate detection. Second, the spectral features are sensitive to different channels, different cloud and illumination conditions. Third, noisy data (missing values, measurement errors) is inevitable due to the nature of remote sensing.

Our contributions are summarised below.

- We develop a seamless processing engine for multi-modal data streams, including missing value imputation, and regions-of-interest query towards a bushfire monitoring system.
- We propose a spatio-temporal data integration algorithm to align weather data with satellite images for more accurate and stable bushfire detections.
- We design a bushfire prediction neural network model with several properties: (i) multi-scale – detects bushfires pixel by pixel and by images, (ii) spectral – unifies multiple spectral channels with 3D convolutional layers, (iii) temporal-aware – handles temporal dependencies via LSTM layers, and (iv) multi-modal – utilises satellite data and weather data simultaneously with fusion layers.

For the remainder of the paper, Section III summarises the preliminaries and our bushfire detection framework. We then discuss the details of our contributions, including data stream processing (Section IV), spatio-temporal data integration (Section V), and multi-modal bushfire prediction (Section VI). Next, Section VII presents empirical evaluations. Finally, Section II reviews related work and Section VIII concludes the paper.

II. RELATED WORK

Our research is conducted in the context of environment science and remote sensing technology that has a long and rich history of research [14], [15]. With the availability of environmental satellites in the recent decades, remote bushfire detection becomes possible with advanced prediction models.

A. BUSHFIRE DETECTION

Existing bushfire detection frameworks use raw spatio-temporal images from satellite sensors, then conduct a background temperature estimation, and finally perform the main fire detection step using some hotspot algorithms [10]. For example, the threshold-based algorithm of Moderate Resolution Imaging Spectroradiometer (MODIS) [7] follows several steps, including land masking, cloud masking, background characterization and threshold tests. Another threshold-based method is FIMMA (Fire Identification, Mapping and Monitoring) of the Advanced Very-High-Resolution Radiometer (AVHRR) system [16] for night-time detection problem. However, the algorithm is only accurate over forested regions. The Visible Infrared Imaging Radiometer Suite (VIIRS) system also has an algorithm, which is built on top of the well-established MODIS Fire and Thermal Anomalies product [17]. Recent methods include broad area algorithm for the AHI (Advanced Himawari Imager) system, which extracts temporal cycle during a day [10]. However, it has low accuracy and can only detect when the fires emit a large amount of smoke [10].

Different from traditional detection methods, our model captures spatial, temporal, and spectral information and dependencies at the same time. Moreover, we use weather data to augment satellite images in order to obtain higher confidence level.

B. REMOTE SENSING

1) MINING SPECTRAL IMAGES

Spectral images in general have been analysed by a wide range of statistical learning models for object detection and image classification [18]. For instance, [19] proposed a deep belief network (DBN) to classify ground covers from airborne spectral images. DBN is a stack of restricted Boltzmann machines, which are trained by a greedy layer-wise unsupervised learning method. However, DBN suffers from expensive computation due to the involvement of too many fully-connected layers. Moreover, it takes only the first three PCA components of spectral information; and thus would neglect complex dependencies between spectral channels. New approaches such as convolutional neural networks have been designed to capture spatial patterns [20]–[22]. However, they consider only global dependencies by applying a biased weighted combination of all spectral channels at the same time; and thus, could miss partial dependencies between them. Going beyond the state-of-the-art, we design a deep neural network architecture that exploits local spatial dependencies across spectral channels. Moreover, our model captures temporal patterns of satellite images captured at different timestamps for the same region of interest, enabling the possibility of forecasting applications to predict potential bushfires in near future.

2) MINING SPECTRAL SATELLITE IMAGES

Spectral images produced by satellite sensors poses unique challenges for spectral image classification. For example, due to adversarial conditions including rotations of the satellite sensors, data samples of the same class could show different
characteristics across sensor directions and capturing locations. As a result, training data of a particular class has great variance in the feature space, which is difficult for algorithmic discriminators [23]. Recent methods have been proposed to overcome these problems, such as a hierarchical convolutional neural network model that combines PCA and logistic regression in between convolutional layers to extract more invariant features. However, critical applications such as bushfire detection requires further examinations of adversarial conditions to avoid false alarms. For example, atmospheric scattering, cloud coverage, and illumination complex, can lead to very different feature maps across spectral channels. To satisfy these requirements, our model incorporates weather data to support the prediction on satellite images.

C. MULTI-MODAL LEARNING
There are several methods to combine data from different modalities [24]. Late fusion constructs one classifier for each data modal. The predicted class is then the combination of classification results via majority voting, maximum probability or average probability. However, this approach neglects the dependencies between data modals. Moreover, the improvement could be insignificant due to the arbitrary combination heuristics without a domain-specific goal. Early fusion eases the task of constructing multiple classifiers by performing fusion in the feature space via vector operations (component-wise max, average, sum). However it assumes that the feature data of given modalities already align well within the same semantic domain (which is often not true). Moreover the data from different modalities might not be available at the same time due to different input rates. Joint fusion: the joint fusion model changes the way the feature vectors are fused by introducing multiplicative loss functions so that the weaker modalities are down-weighted during the fusion. Different from these works, our paper follows a common-space fusion approach, where the feature vectors of image and weather data are projected into the same domain via a pairing constraint.

III. APPROACH OVERVIEW
A. PRELIMINARIES
1) SATELLITE SENSORS
A plethora of bushfire sensing satellites have been put in the space. The first generation includes mostly polar-orbiting systems such as MODIS [7], AVHRR [16], and VIIRS [17]. However, these satellites, despite having high spatial resolution, fly around the poles and thus are limited to daily imaging frequency only (no any spot on the Earth’s surface can be monitored continuously during a day) [11]. The second generation is geostationary systems such as AHI and ABI (Advanced Baseline Imager) that are popularly used, even
Table 1 presents the specifications and detection characteristics of bushfire sensors.

| Satellite system | Spatial resolution | Input Rate | Fire Size | Fire Radiative Power | False Alarms |
|------------------|--------------------|------------|-----------|----------------------|--------------|
| MODIS (1997) | 1 km | 6 hours | ~900 m$^2$ | ~15 MW | 5%–15% |
| AVHRR (1997) | 1.1 km | 3 hours | ≥1 km$^2$ | ~1 MW | 5%–10% |
| VIIRS (2011) | 375 m | 2 hours | ≥5 m$^2$ | ≥15 MW | 0%–4% |
| AHI (2014) | 1 km | 15 minutes | ≥2.2 km$^2$ | ≥30 MW | 3%–23% |

1) Geolocation of each pixel; 2) Low Earth orbiting; 3) Polar orbiting; 4) Geostationary.

**Table 2.** Spectral channels of a GOES-16 image.

| Channel | Central (μm) | Nickname | Type |
|---------|--------------|----------|------|
| 1       | 0.47         | Blue     | Visible |
| 2       | 0.64         | Red      | Visible |
| 3       | 0.865        | Veggie   | Visible |
| 4       | 1.378        | Curra    | Near-IR |
| 5       | 1.61         | Snow/Ice | Near-IR |
| 6       | 2.25         | Cloud Particle Size | Near-IR |
| 7       | 3.90         | Shortwave Window | Infrared |
| 8       | 6.9          | Upper-level Tropospheric Water Vapor | Infrared |
| 9       | 6.95         | Mid-level Tropospheric Water Vapor | Infrared |
| 10      | 7.34         | Lower-level Tropospheric Water Vapor | Infrared |
| 11      | 8.5          | Cloud Top Phase | Infrared |
| 12      | 9.61         | Ozone    | Infrared |
| 13      | 10.35        | Clean Infrared Longwave Window | Infrared |
| 14      | 11.2         | Infrared Longwave Window | Infrared |
| 15      | 12.3         | Dusty Infrared Longwave Window | Infrared |
| 16      | 13.3         | CO2 Longwave Infrared | Infrared |

**B. Wildfire Detection Framework**

Our framework is illustrated in Figure 1. The input stream is data from satellite sources and weather sources. Raw satellite images and weather measurements are cleaned and having smaller resolution, due to their frequent monitoring capability.

Table 1 presents the specifications and detection of typical satellite systems in the field.

- **MODIS**: MODIS is a flagship sensor of the Earth Observing System (EOS) project of National Aeronautics and Space Administration [25]. Together with EOS-Terra, its Aqua MODIS instruments are component of the low Earth orbiting system launched in 1999. Terra MODIS and Aqua MODIS capture the entire Earth’s surface every 1 to 2 days, generating images in 36 spectral channels. It can detect fires with area of 900 m$^2$ and 15 MW in average [7].

- **AVHRR**: The AVHRR sensor is first used in 1998 and currently onboard several polar-orbiting National Oceanic and Atmospheric Administration (NOAA) satellite platforms. It has a global standard of spatial coverage and spectral resolution to monitor Earth surface every day [26]. Each AVHRR instrument provides multiple thermal observations of the Earth in the mid to high latitudes on a daily basis [27], [28].

- **VIIRS**: The VIIRS sensor is aboard the joint NASA/NOAA satellite in 2011. Theoretically, it can detect fires with area as small as 5 m$^2$. Empirically, it can detect 1 million fires in night time [17].

- **AHI**: The AHI is a multispectral imager used by geostationary satellite platforms to capture images of the Asia-Pacific region. The resolution of AHI images is up to 500m resolution and can capture full-disk Earth’s view every 10 minutes [29]–[33].

- **ABI**: The ABI sensor is used by the GOES-16 (Geostationary Operational Environmental Satellite) platform in 2016. In the default “flex” mode of operation, the ABI produces images of the Earth every 15 minutes, with a spatial resolution of 0.5–2 km [34].

In this work, we select the GOES-16 platform as a data source since it operates upon US continent where the bushfire datasets are available (see Section VII). GOES-16 is the state-of-the-art geostationary system. It uses multispectral sensors, which are generally more preferred than hyperspectral ones since the latter has a high level of complexity (200 continuous channels compared to 16 representative channels per image) and channel redundancy (the difference between consecutive channels are too narrow) [35]. Other advantages of GOES-16 is the high spatial resolution and the high temporal resolution (i.e. input rate), suitable for fast-spreading events like bushfires. GOES-16 also have the advantage of geostationary, i.e. monitoring incidents without significant geometric correction of images, leading to robust spatial analyses [11].

2) **RAW SATELLITE DATA**

Imagery information is obtained from the satellite data streams available on Amazon Web Services (AWS) [36]. The data samples are generated by the ABI of GOES-16 satellite, which captures Earth’s radiance in 16 spectral channels (Table 2) via a variety of radiance detectors. Basically, they are digital maps of outgoing radiance values at the top of Earth’s atmosphere at visible, infrared, and near-infrared wavelengths. Then, the samples are compressed, packetized, and sent to the ground station, in which they are converted to geo-located and calibrated pixels [36], covering the whole America continent. The raw image pixels are kept in Network Common Data Form (netCDF) format, which is descriptive, flexible, standardized among large research projects [37]. Each channel of an image sample is kept in a separate netCDF file for each 15-minute interval.

3) **WEATHER MEASUREMENTS**

In our study, the three following weather measurements are taken into consideration: temperature, humidity and air pressure. Since bushfires are more likely to happen during hot and dry days, weather information provides an additional degree of confidence for remote detection using satellite images. Moreover, nearby weather stations, if close enough, might provide additional evidences of fire, such as temperature increasing, barometric pressure, and humidity decreasing, as recorded by weather meters within the burning zones [13]. For example, an average surface fire on the forest floor has flames reaching 1 metre in height and reaching temperatures of 800 celsius degree [38]. The readings of nearby stations are likely to be affected since the normal air temperature is often less than 50 celsius degree.
transformed into time-series and vector formats. More precisely, the GOES-16 data is hosted by AWS and fed into our system via Web API. However, the raw satellite images cover the whole America continent, which is too big for monitoring purpose. This component reads streaming raw satellite images and produce resample data to a particular region of interest in the shape of 16 spectral channels of brightness value. Missing values due to streaming errors are also imputed for continuity purposes. The details are described in Section IV.

Satellite data and weather data are feed into Spatio-temporal data integration component, which align them along temporal and spatial dimensions. Next, the multi-modal data are streamlined to the Multi-modal bushfire prediction component to locate bushfires in real time and real location.

1) SPATIO-TEMPORAL DATA INTEGRATION
To improve detection accuracy, combining information from satellite and weather data simultaneously is an advantage of our framework. However, due to different sampling rates, weather data might not be available for a particular time of the satellite stream. Moreover, since a weather meter is put on the ground, it might not cover all geolocations of a satellite image. In this component, we develop a spatio-temporal integration algorithm to (i) infer weather information for all pixels of a satellite image from neighbouring weather stations and (i) interpolate missing weather data for all time steps of the satellite stream. Our experiments confirm that this leads to more accurate and continuous bushfire detections. The algorithm is detailed in Section V.

2) MULTI-MODAL BUSHFIRE PREDICTION
Developing this component needs to consider several aspects: (i) the feature maps of satellite images are unstable due to adversarial conditions such as rain and cloud, (ii) bushfires are spread in some spatial patterns, (iii) the speed of bushfires might follow a temporal pattern, (iv) brightness patterns of bushfires depends on different spectral channels and different regions of interest. The detail is discussed in Section VI.

IV. DATA STREAM PROCESSING

A. PROCESSING SATELLITE STREAMS

1) REGION OF INTEREST (ROI) QUERY
A ROI query [39] is \( R = (x_1, y_1, x_2, y_2, r) \), where \( x \) is latitude and \( y \) is longitude, \( r \) is the geographical size of each pixel (0.5–2km). We choose 1km resolution due to our pilot studies. To mimic real geographical distance, we use Lambert projection [40]. A ROI query is implemented via SatPy, a Python library using Lambert projection and netCDF format of GOES-16 as well as store meta-data (timestamp, size, etc.) efficiently.

2) MISSING DATA QUERY
Due to high sampling rate of the satellite stream and processing errors of the ground station, there could be some missing values of certain spectral channels at certain time. The missing values could be non-uniform, i.e. the missing-value event of each spectral channel happens at different timestamps. We solve the missing data imputation problem as follows. If there is only one missing value, we use the temporally-preceding value of the same spectral channel, since the input interval is quite small (15 minutes) (i.e. the chance that an error due to incorrect missing value imputation is small). However, if there are two or more missing values, the capture of that moment is skipped to avoid false prediction due to noisy observations. Other data imputation techniques [41] have been tried as well but did not perform better.

3) DATA STORAGE
Each channel of the output images is stored in PNG format, while their radiance values are kept in CSV format. In this work, we focus on the main problem of using spatial information, temporal information, and spectral information simultaneously to improve bushfire detection. Future works are expected to extend our model by employing more pre-processing techniques [42], [43].

B. PROCESSING WEATHER STREAMS

1) DATA QUERIES
The streaming source for weather information is Weather Underground (WU). WU contains data for 250,000+ personal weather stations as well as data from Cooperative Observer Program (National Weather Service), weather airports and weather balloons. For each station, automated measurements are taken at least once an hour. Moreover, observations are also manually obtained at least daily for high confidence [44].

Since weather measurements such as temperature and humidity should be kept free from direct solar radiation, only stations with accurate observations ranked by WU are used. Due to the fact that we cannot get the exact weather information for any geo-location, we will collect data from as many stations as we can. Approximate measurements for a target ROI are inferred from nearby stations.

2) DATA STORAGE
The weather information is stored in Mongo database along with station meta-data (code, name, longitude, latitude), time [date, time, captured date] and measurement (temperature, humidity, pressure). This storage facilitates the later integration of weather measurements and satellite images along both spatial and temporal dimensions as a multi-modal input of the prediction model.

C. PILOT STATISTICAL DATA ANALYTICS
We perform a pilot statistical data analysis of the satellite imagery data on real bushfires to motivate the design our prediction model.
FIGURE 2. Spectral distribution of fire pixels in a cloud day.

FIGURE 3. Spectral distribution of fire pixels in a clear day.

1) SPECTRAL DISTRIBUTION

Figure 2 presents the distribution of reflection and radiance values in 16 spectral channels of a real bushfire, which happens during a cloud day. Figure 3 shows the same distribution of another bushfire in a clear day. From these distributions, we can already observe some patterns of spectral information including the role of each spectral channel. In other words, the spectral values of fire pixels can be very different and thus it is imprudent to identify them by just thresholding. This confirms previous studies on spatio-spectral patterns of satellite images [18].

2) SPECTRAL CORRELATION

To understand the relationship between any two spectral channels as well as their own distributions, we use pair-plots [45]. In that, we also compare the correlations and distributions between fire and non-fire imagery data. For brevity sake, a sample comparison of 4 representative spectral channels is presented in Figure 4, in which yellow and blue data points represent the fire and non-fire pixels respectively. Full spectral correlation analysis can be found in the appendix (Figure 17).

TABLE 3. Satellite data vs. Weather Data.

It can be observed that each spectral has its own contribution in predicting fire pixels. For example, the channel 7 seems to be very important and sensitive with hot-spots. Its values are often very high for fire points. However, these values decrease quickly if it is a cloudy day. Such observations motivated us to use all 16 spectral channels for our prediction model instead of a single or a few spectral channels as in existing works [18].

V. SPATIO-TEMPORAL DATA INTEGRATION

Satellite data and weather data have different temporal sampling rates and different geographic locations. Table 3 shows the spatio-temporal characteristics of the satellite and weather data. As a result, weather data might not be available at the location and at the time point of a given satellite image. In this section, we will align the time series of weather data into the time series of satellite by means of interpolation.

A. PROBLEM STATEMENT

Since there is a limited number of weather stations, it is challenging to find the exact weather information for a geo-point in the map. Moreover, the sampling rate between weather stations are not the same and also different from that of the satellite.

The problem of finding the weather measurements at a particular location and at given time can be formulated as follows. There are $N$ climate stations at locations $\{(x_1, y_1), \ldots, (x_N, y_N)\}$. Considering a weather domain $D$...
(e.g. temperature, humidity, air pressure), each station $i$ records weather measurements over a series of time instants $\{h_D(x_i, y_i, t)\}_{m=0}^{M_i}$. Given a point of interest $p = (x, y)$ at time $t$, we want to compute its weather value as an interpolation of observations:

$$h_D(x, y, t) = f \left( \bigcup_{i=1}^{N} (h_D(x_i, y_i, t))_{m=0}^{M_i} \right)$$  \hspace{1cm} (1)

We argue that designing the interpolation function $f(\cdot)$ needs to satisfy the following requirements:

(R1) **Spatio-temporal alignment**: Measurements from a closer station might be less worthy than another if their timestamps are far away from the captured time of satellite image. We need to consider both spatial and temporal dimensions simultaneously.

(R2) **Time decay**: The interpolation effect of a measurement should be decreased with the temporal difference against the timestamp of satellite images.

To satisfy these requirements, our solution exploits spatio-temporal correlation of weather measurements via a physical diffusion model to infer missing measurements at a desired time and location. Figure 5 illustrates the overview of our solution.

**B. WEATHER DIFFUSION MODEL**

Diffusion is a physical phenomenon, which is common among different substances such as temperature, pressure, and humidity ($H_2O$). For example, heat always flows from a position with high temperature to a position with low temperature in a medium [46]. Such a phenomenon can be captured by a functional model, in which the measurements of weather stations are just sample data points [47]. The data samples can then be used to reconstruct the diffusion process, which capture the spatio-temporal evolution of e.g. temperature in the atmosphere [48].

The diffusion process follows a Gaussian distribution as described by the 2D heat kernel [47]:

$$D(x, y, t) = \frac{1}{4\pi\alpha t} e^{-\frac{x^2+y^2}{4\alpha t}}$$  \hspace{1cm} (2)

where $\alpha$ is the diffusion constant depending on the weather domain [49]. Mathematically, it can be treated as $\alpha = 1$ without the loss of generality (since model parameters will be estimated from sample measurements accordingly). This model considers both spatial and temporal information at the same time (R1). Assuming at time zero, the weather function $h(\cdot)$ has an initial distribution $h(x, y, t_0) = h_0(x, y)$. Formally, we can model the weather distribution as follows. The temperature at time $t$ can be computed as a convolution of the initial heat at time $t_0$ with the diffusion kernel:

$$h(x, y, t) = [h_0 * D](x, y)$$  \hspace{1cm} (3)

The initial weather distribution can be modeled by multivariate Gaussian $h_0(x, y) = \frac{1}{\pi\sigma_1\sigma_2} e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2} - \frac{(y-\mu_2)^2}{2\sigma_2^2}}$, where $\{\mu_1, \mu_2, \sigma_1, \sigma_2\}$ are the model parameters.

**C. TIME-DECAY SMOOTHING**

For temporal interpolation, more recently measurements should be treated more importantly than old measurements to capture the trend (R2). To model such degradation, a monotonic decreasing function $g(t)$ is used:

$$g_m(t) = \exp^{-\lambda(t-t_m)}$$  \hspace{1cm} (4)

where $m = 1 \ldots M$ is the $m$-th measurement, $t_m$ is the recording time of measurement $m$, $\lambda$ is the decay rate and is set to 0.5 (maximal entropy principle), and $t$ is the time point we want to interpolate. In other words, the more preceding the measurement is ($t_m < t$), the less importance it has on the interpolation, and vice-versa.

Plugging the time-decay smoothing function into the interpolation algorithm, we have:

$$h(x, y, t) = f \left( \bigcup_{i=1}^{N} \bigcup_{m=0}^{M_i} (h(x_i, y_i, t_m) \times g_m(t)) \right)$$  \hspace{1cm} (5)

**D. INTERPOLATION ALGORITHM**

The time series of weather measurements by $N$ climate stations can be considered as the samples of the distribution function. Therefore, solution to Equation 1 can be computed by the following algorithm:

**Input**: Observed weather measurements as data samples.

1) Apply for time-decay smoothing for each data sample by Equation 5.

2) Infer the model parameters of the heat distribution function (Equation 3) from the set of samples $\bigcup_{i=1}^{N} (h(x_i, y_i, t))_{m=0}^{M_i}$ using the Random transform method [47].

3) Compute the weather value $h(x, y, t)$ for the given point $p = (x, y)$ at time $t$ from the inferred model parameters.

**Output**: The weather value at a given geo-location and at a given time.

The interpolation algorithm is then used to produce the weather “heat map” (Figure 5) for each time step of the satellite imagery data, guaranteeing a perfect spatio-temporal alignment between two different data modal.
VI. MULTI-MODAL BUSHFIRE PREDICTION

In this section, we use the integrated data of satellite images and weather information for bushfire detection. However, extracting robust features from spectral images is much more challenging than normal images due to the non-stationary characteristics of spectral channels such as illumination, sensor rotations, and air scattering. Such complex features can be learned from data via deep learning [50], [51]. Our approach builds a novel deep learning architecture that goes beyond the state-of-the-art deep neural networks by augmenting spectral images with weather data.

A. PROBLEM STATEMENT

A multispectral image has \( S = \{1, \ldots, N\} \) pixels of a multi-spectral image. Each pixel can be classified into 2 classes \( L = \{1, 2\} \): fire (or fire-positive) and no-fire (or fire-negative). Each pixel can be represented by a \( M \)-dimensional feature vector \( x_n = \{x_{n1}, \ldots, x_{nM}\} \), where \( M = 16 \) is the number of spectral channels. And each image is associated with \( N \) feature vectors \( X = \{x_1, \ldots, x_N\} \).

Problem 1 (Multi-Scale Prediction): Multi-scale prediction is a dual realisation of a many-to-many function \( f_1 : \mathbb{R}^{N \times M} \rightarrow \mathbb{R}^{N \times 2} \) and a many-to-one function \( f_2 : \mathbb{R}^{N \times M} \rightarrow \mathbb{R}^{2} \) from an image \( X \) to pixel-wise labels \( Y \) and an image-wise label \( y \):

\[
Y = f_1(X); \quad y = f_2(X)
\]

where the label \( Y = \{y_1, \ldots, y_N\} \) corresponding to \( N \) pixels. The labels \( y = \{y^1, y^2\} \) and \( y_n = \{y_n^1, y_n^2\} \in Y \) are 2-dimensional label vectors, where the elements \( y_n^1 \) and \( y_n^2 \) represent the probability of being \( l \in L \). The final label \( y^* \) for the whole image \( X \) and \( y^*_n \) for each pixel \( x_n \) will be computed using maximal probability principle:

\[
y^* = \arg\max_{l \in L} y^l \quad \text{and} \quad y^*_n = \arg\max_{l \in L} y^l_n.
\]

Due to the complex characteristics of GOES-16 images, we argue that the solution to this problem should satisfy:

(R1) Spatial context: captures the neighbourhood locality of a given pixel.

(R2) Temporal context: capture the trend of data over time as bushfire is a temporal phenomenon.

(R3) Cross-channel spectral dependencies: captures spectral correlation as observed in our pilot statistical analysis Section IV-C. Spectral channels are sensitive to illumination, atmospheric scattering, and sensor rotation. As a result, pixels of the same class (fire-positive or fire-negative) could show different characteristics in different times and locations.

(R4) Multi-modal learning: Discover relationships across modalities including weather measurements and spectral images for a better prediction.

B. MODEL STRUCTURE

Figure 6 illustrates our proposed deep neural network [14], [23], [50], [52] that captures these above requirements. It is compiled from several layers: (i) input layer – consumes the imagery and weather streams, (ii) convolutional layers – handles spatio-spectral contexts, (iii) fusion layer – integrates imagery data and weather information, (iv) LSTM layer – handles temporal patterns, (v) output layer – outputs the final result. It is noteworthy that the image-wise part and the pixel-wise part of our model are not separated. They complement each other to achieve better performance for each output. On the one hand, the latent features extracted by the CNN are connected to the image-wise part via the input of LSTM. On the other hand, the latent features extracted by LSTM are also back-propagated to the pixel-wise part.

1) INPUT LAYER

The input is, for each spectral channel, a set of \( m \times n \) matrices of pixels, where \( m \times n \) is the ROI size. The same procedures are applied for the pixel matrices of weather measurements.

a: PATCH NORMALISATION

Using images as training data for pixel-wise prediction would lead to overfit due to low number of samples. Augment training data via a patch normalisation layer. First, we divide each image into patches of size \( k_p \times k_p \) (\( k_p = 12 \) since \( 144 \text{ km}^2 \) is the minimum burnt region in bushfire datasets and each pixel corresponds to an \( 1 \text{ km}^2 \) area) with 50% overlap. Next, a normalisation operator is applied for each channel to ensure the same value domain by mean-shift [53].

b: PIXEL-WISE HOOK

For each patch, the coordinates of its center pixel in the respective image are fed as input along the output of convolutional layers to a fully connected layer/network (FCL/FCN) [54]. This mechanism allows spatial relationships between patches of the same image to be preserved and the output of our model to be pixel-wise.

Another way to enforce pixel-wise output is upsampling segmentation but satellite images do not contain natural objects to be segmented [55]. Besides, an upsampling layer would drop the spectral context between different channels.

2) CONVOLUTIONAL LAYERS

The convolution layers [50] model spatial context (R1) via convolutional and flatten operators.

a: 3D CONVOLUTION

There are three convolutional layers to process the input sequentially, where consecutive layers are connected via receptive fields [52]. A next layer is smaller than the previous one to capture more abstract features, focusing on locality of pixels regardless of their actual locations in the original image.

However, satellite data has spectral dimension beside 2D information (R3). In other words, different combinations of spectral channels could have correlation in terms of bushfire detection. As a result, we need 1 more dimension in convolutional layers to model such combinations. This calls for...
a 3D version of CNN, where each receptive field is a cube \( n_c \times n_c \times n_c \) instead of a matrix, where \( n_c \) is a hyperparameter.

Formally, the forward computation of CNN is captured by the weight-sharing function [50]:

\[
v_i = \varphi_c(b_i + w_i \ast x) \tag{7}
\]

where \( v_i \) is the output of \( i \)-th filter, \( \varphi \) is RELU activation function, \( \ast \) is the 3D convolution operation, \( x \) is the receptive field, \( b_i \) is bias factor of filter \( i \), and \( w_i \) is a shared weight vector. The output of each convolutional layer is \( n_{filters} \) feature map [50] corresponding to \( n_{filters} \) different filters. Each filter captures a partial dependency in both spatial and spectral dimensions.

In the first convolutional layer, we use receptive field of \( n_{c,1} = 7 \) with \( n_{filters,1} = 128 \) filters. In the second layer, the kernel size is \( n_{c,2} = 5 \) with \( n_{filters,2} = 64 \) filters. In the third layer, the receptive field reduced to \( n_{c,3} = 3 \) with \( n_{filters,3} = 32 \) filters. This configuration is motivated by the actual size of bushfires (at least \( 1 \text{ km}^2 \)) and works well in the experiments.

\( b \): FLATTEN

We flatten the last convolutional output by connecting them to a FCL to transform feature maps into prediction scores. More precisely, each feature map is connected with \( n_p \) neurons of the flatten layer, resulting in a vector \( v_p \in \mathbb{R}^{n_p} \) where each \( k \)-th element is:

\[
v^k_p = \varphi_p(w^k_p v + b^k_p) \tag{8}
\]

here, \( w^k_p \) and \( b^k_p \) are both parameters to learn and \( v \) is the last convolutional value.

\( c \): CONVOLUTION FOR WEATHER DATA

We also use a separate convolution layer to process of weather data due to capture the spatial dependency between locations. The difference is that the traditional 2D convolutional layers is used, since we consider the dependency between all weather measurements as a whole. In other words, similar to 2D CNN for RGB-images (which have 3 channels red, green, blue), using 2D convolutional layer is sufficient for weather data (which also has 3 “channels” temperature, humidity, air pressure) since the output of 2D convolution operation is already a weighted combination of the three measurements. The output is also flattened in the same manner by a fully-connected layer with the same output size for the fusion purpose next. It is noteworthy that this FCN shares the same weights with the FCN of satellite data to achieve the same output vector space.

\( C \): FUSION LAYER

The fusion layer combines the image feature vector \( v^I \) and weather feature vector \( v^W \) from the convolutional layer via an aggregation operation to produce a single vector:

\[
u = \text{avg}(v^I, v^W) \tag{9}
\]

where \( \text{avg} \) is the element-wise average function of two vectors. Other aggregation functions such as \( \text{max} \) have been tried but were not better. This indicates that bushfire detection should use two types of information simultaneously.

However, since image data and weather data does not share the same semantic space, we enforce a pairing constraint on a pair of image and weather data sharing the same class (fire or non-fire). Formally, the goal of the pairing constraint is to make a pair of image and weather vectors of the same sample, \( (v^I(s), v^W(s)) \), to be similar and of different samples \( (v^I(s), v^W(s')) \) to be different. We measure the similarity by the image Euclidean distance metric [56] \( d(\cdot, \cdot) \), which is proven to be robust to small perturbation in 2D data and efficient to compute.
The loss function for the pairing constraint is defined as the soft-max ratio of the given sample over a portion of other samples [57]:

$$L(s) = -\sum_{s' \in \Omega} \log \left( \frac{e^{-d(v'(s), v'(s'))}}{e^{-d(v'(s), v'(s'))} + e^{-d(v'(s), v(s'))}} \right)$$

(10)

where $\Omega$ is a sample of training data. The more samples we used, the more expensive of computation required for the training. Therefore, we perform a random sampling of 20% training data to generate $\Omega$ for each trained sample $s$.

1) PIXEL-WISE OUTPUT LAYER
The output is computed via a projection that transforms the fusion vector into scores, which is then normalised by a softmax layer. First, a linear mapping is applied:

$$y' = wu + b$$

(11)

where $y' = [y'_1, \ldots, y'_C]$ is the scores. The softmax layer helps to normalise for smoother transition:

$$y_c = \frac{\exp y'_c}{\sum_{c' \in C} \exp(y'_{c'})}$$

(12)

Note that during training, the final score $y_c$ is compared to the true label $\hat{y}_c$ of each pixel in the cross-entropy loss function to perform back propagation.

2) LSTM LAYER
In practice, it is essential to decide on the image-level whether there is a bushfire for automatic alarms. Traditional methods often follow a threshold-based approach to leverage pixel-wise information to classify the whole image [10]. However, such approaches are sensitive to threshold configuration, which is domain-specific and requires expert knowledge. On the other hand, treating the whole image as a separate classification would lose information at the pixel-level. To overcome these issues, after training the pixel-wise classification, we take the output of fusion layer of each data patch, connect all of them to a fully connected layer $z$ to represent the whole image.

Then each image will be connected across time by the LSTM layer, which consists of LSTM block, which allows to “remember” past information across multiple time steps (i.e. long range dependencies) by using a sequential structure of memory cells [52], satisfying (R2).

The core principle of LSTM is a continuously updated memory $c_t$, which is a combination of a part of existing memory and a new memory content $\tilde{c}_t$:

$$c_t = f(z_t, h_{t-1}) c_{t-1} + a(z_t, h_{t-1}) \tilde{c}_t$$

(13)

where $z_t$ is the input sequence at timepoint $t$, and $h_{t-1}$ is the output vector from the LSTM at the previous timepoint. The forget function $f(.)$ and adding function $a(.)$ are the sigmoid regression and thus always return a value in $[0, 1]$ to control how many percentages of each value should pass through.

The new memory content is given by:

$$\tilde{c}_t = \tanh(h_c + w_1 z_t + w_2 h_{t-1})$$

(14)

where the $\tanh(.)$ is used to smooth the memory value into $[-1, 1]$ (i.e. remember both “bad” and “good” memories). In short, the current output of the LSTM depends on three factors: the current input, the previous output, and the current memory:

$$h_t = o(z_t, h_{t-1}) \tanh(c_t)$$

(15)

where $o(.)$ is also the sigmoid regression, but with its own parameters.

Training LSTM layer requires batching the satellite images across time. Since the input rate of GOES-16 data is 15min, we group four temporally consecutive images into a batch, which is suitable for capturing the temporal pattern of bushfires.

3) IMAGE-WISE OUTPUT LAYER
The output from the LSTM layer is fed an image-wise output layer with similar structure as pixel-wise output layer with the projection layer, whose output is $\hat{y}' = wh + b$, and the softmax layer, whose output is $\hat{y}$. During training time, the cross-entropy function will be used to compute the loss. During testing time, the final label of the whole image is decided by $y^* = \arg\max(\hat{y}_1, \hat{y}_2)$, where $y^*$ is the predicted class and $\hat{y}_1$ and $\hat{y}_2$ are the classification scores of fire and non-fire classes respectively.

D. MODEL TRAINING

1) DESIGN CHOICES
We tried different model designs before the above architecture. For instance, using 2D convolutional layers only degraded the prediction results, which indicates correlations between spectral channels. And the CNN has a depth of 3 to justify to trade-off between prediction accuracy and computation time.

2) AVOID OVERFITTING
Neural networks prone to overfitting when the amount of training samples is small. This could be alleviated by different ways:

- **Regularisation**: We put a max-pool layer between two CNN layers and use batch normalisation for any fully-connected layer. We also tried drop-out layers (even with different dropout probabilities through the network) but they did not improve the prediction results.

- **Semantic-preserving augmentation**: Since image-wise and pixel-wise prediction is rotation invariant and mirror invariant [53], i.e., a fire-positive pixel/image is still fire-positive after any rotation, we generate more samples as follows. We rotate the image 4 times with $k \times \pi/2$ rotations ($k = [0, 3]$), each of which is applied a vertical reflection, resulting in 8 more samples for each original sample.
Synthetic white-noises: For each class, we calculate the standard deviation of spectral values of training samples (doing separately for each spectral channel). Then, we use the calculated standard deviation as the parameter of a zero-mean multivariate normal distribution $N(0, \alpha^2 \Sigma)$, where $\alpha$ is a scale factor, and $\Sigma$ is a diagonal matrix containing $\alpha$ along the main diagonal entries. As a result, the augmented samples for a given sample of each class are generated by adding white noises, which are sampled from the distribution $N$, to the values of each pixel (doing separately for each spectral channel). The scaling factor $\alpha$ are tried with several values and fixed to 0.25 for the experiments [20] to balance between the learning efficiency and robustness (e.g. too big noises can make the model fail to converge).

3) AVOID UNDERFITTING
While training a complex model (i.e. with a large amount of parameters) is time-consuming, reducing model capacity (less parameters) might lead to underfitting. To overcome the trade-off between model capacity and computational efficiency, we incorporate pre-trained deep neural network models including VGG16 [58] and AlexNet [59], which turns out to be useful for domain-specific data since they are trained on the million-scale ImageNet dataset [60]. Formally, the trained VGG16 (or AlexNet) layers are put before the convolutional layer of our model.

4) PARAMETER OPTIMISATION
We trained the network using the state-of-the-art Adam optimiser, which theoretically and empirically outperform other optimisers such as Momentum and RMSProp [61].

VII. EMPIRICAL EVALUATION
In this section, we conduct experiments with real-world datasets. First, we discuss the experimental settings (Section VII-A). Then we report an end-to-end evaluation to show that our framework outperforms the baselines (Section VII-B). Next, we show efficiency test to show that our framework is fast enough for early detection of bushfires (Section VII-C). We also include ablation test to show the importance of each model component (Section VII-D). Our model is also robust to adversarial conditions (Section VII-F). Finally, qualitative showcases prove the interpretability of our model (Section VII-G).

A. EXPERIMENTAL SETUP
1) DATA
We evaluate the framework on the following real-world bushfire datasets (their key characteristics are summarised in Table 4).

- County Fire: The County Fire start from 2:12 pm on June 30, 2018 at the east of Lake Berryessa in Yolo County and Napa County. The fire burned approximately 365km$^2$ [62].

- Camp Fire: The Camp Fire started on November 8, 2018 in California. The fire covered an area of 620 km$^2$ [63] and is one of the most destructive fire in history of the state.

- Woolsey Fire: The Woolsey Fire is destructive wildfire that burned in Los Angeles and Ventura Counties of California. The fire began at 2:22pm PST on November 8, 2018 and burned about 392 km$^2$ [64].

- Kincade Fire: The Kincade Fire burned in Sonoma County, California from 9:24 p.m. PST on October 23, 2019 and subsequently burned above 314 km$^2$ [65].

2) BASELINES
The following baselines are used for comparison:

- GOES-AFP: is the state-of-the-art method [11] for the GOES system.
- AVHRR-FIMMA: is the algorithm developed for AVHRR imagery data [16], [28], [66], [67].
- MODIS-Terra: is the algorithm designed for MODIS satellite system [7].
- VIIRS-AFP: is the algorithm in VIIRS satellite system [17].

Note that we have reviewed several other baselines as well, such as BAT (a multi-temporal method to detect temperature change in AHI data) and ER [68] (a probabilistic reasoning on top of urban fire sensors). However, they are system-dependent (e.g. they utilised completely different channels of their own imagery sensors) and do not align with our settings and datasets.

3) METRICS
We use Weighted F1-score, where the weight factor is the class ratio [59]. Unlike Accuracy, it captures the imbalance class distribution as in our datasets. We also design a more fine-grained level of metrics:

- Pixel-level: Each pixel is considered as a data sample. This level concerns the ability of classification methods in locating the fires in the geographic area.
- Image-level: Each image is considered as a data sample to count the positive cases (containing fire) and negative cases (not containing fire). While the image level is less strict than the pixel level, it emphasises the the monitoring capability of classification methods in streaming data to quickly identify images with fires for timely alarms.

| Dataset   | Genesis     | #Pixels     | Class Distribution |
|-----------|-------------|-------------|--------------------|
| County    | 14:12 30.06.2018 | 265 × 266   | 3,381,234 : 2,286   |
| Camp      | 06:33 08.11.2018  | 290 × 300   | 6,252,876 : 11,124  |
| Woolsey   | 14:22 08.11.2018  | 110 × 145   | 761,854 : 2,018     |
| Kincade   | 21:24 23.10.2019  | 265 × 266   | 3,38,1728 : 1,792   |

1 Ratio of fire-negative pixels over fire-positive pixels
• **Distance-based:** Pinpointing the location of fires in a satellite image is important for timely reaction. In order to evaluate such localisation ability of classification methods, we consider both the detection result \(x\) and the ground-truth \(y\) as two binary images: \(x = (x_1, \ldots, x_{LW})\) and \(y = (y_1, \ldots, y_{LW})\), where \(x_i, y_i \in \{0, 1\}\) (pixel value 1 indicates fire), \(L\) and \(W\) are respectively the length and the width (in pixels) of the image. We employ the *image Euclidean distance* metric [56, 69], which is proven to be robust to small perturbation and efficient to compute:

\[
d(x, y) = \frac{1}{2\pi} \sum_{i,j=1}^{LW} \exp \left\{ \frac{-|P_i(x) - P_j(y)|^2}{2} \right\} (x_i - y_i)(x_j - y_j)
\]

where \(P_i(x) = (l, w)\) and \(P_j(y) = (l', w')\) denote the location of \(i\)-th pixel of \(x\) and \(j\)-th pixel of \(y\) respectively. \(|P_i(x) - P_j(y)| = \sqrt{(l - l')^2 + (w - w')^2}\) denotes the Euclidean distance between two pixels on the image lattice.

• **Lag time:** the detection delay counting from when the bushfire actually happens.

4) **TRAINING PROCEDURE**

We design different aspects of the training process:

• **Cross validation:** is used with \(k = 10\) to balance the amount of training data and the fairness of testing data.

• **Model tuning:** the training data is further split randomly 10 times into a training set and a validation set by 9/1 ratio like cross-validation. However, only the best model on validation set will be used to avoid over-fitting.

• **Early stopping:** To further avoid over-fitting as well as speed-up training time, we employ a best-practice stopping condition of training process by measuring the convergence of model performance on the tuning set instead of the learning set [50]. As such, the model can be prevented from over-fitting by not solely focusing on the training error and often converges faster.

5) **HYPERPARAMETER TUNING**

Hypeparameters of the model, including the regularization parameter \(\lambda\), the momentum coefficient \(\mu\), the learning rate \(\eta\), and the kernel sizes, are tuned by a random-walk process, which is more efficient than grid search [50].

6) **COMPUTING ENVIRONMENT**

We implement the model using Python 3 and Keras with 15,736 parameters including network weights and biases.

**B. END-TO-END EFFECTIVENESS EVALUATIONS**

1) **IMAGE-LEVEL ALARMS**

We evaluate the classification methods in detecting images having fires (fire-positive). In practice, monitoring systems allow user to configure a confidence threshold to cover fire alarms. For example, during hot days, user can decrease the confidence threshold to catch potential fires with the cost of having more false alarms. To this end, we put a cut-off threshold on the classification scores, in which only data sample with a score higher than the cut-off will be regarded as fire-positive.

Figure 7 presents the result, in which we vary the confidence threshold from 50% to 80%. The X-axis represents different datasets, whereas the Y-axis measures the performance in Weighted F1. The key finding is that when the confidence threshold increases, precision increases while recall decreases. This is, indeed the trade-off between early detection and detection accuracy: investigating all suspicious outputs would incur more wasted time and cost in case of false alarms.

2) **PIXEL-LEVEL PATTERNS**

Since bushfires are temporal events that spread over geolocations, monitoring at the pixel-level would help to understand bushfire development behaviours. To this end, we run pixel-wise classification during during the life-time of a bushfire.

Figure 8 presents the result, in which the X-axis is the difference between current time and genesis time of a bushfire and the Y-axis is prediction performance in terms of Weighted F1. The interesting observation is that the accuracy increases...
significantly at the beginning and starts to converge in the middle of the fire lifetime. This could be explained by the neighbourhood effect: when a fire becomes larger, spatial dependency between adjacent pixels is clearer and enable more accurate prediction.

3) FIRE LOCALISATION
In practice, the prediction output could be offset by some margins, i.e. the fire-positive pixels are not exactly but near the ground-truth pixels. As a result, it is acceptable to consider offset pixels as true alarms, even their locations are approximate to the true event. To this end, we measure the distance between the detected pixels compared to the actual locations of fires.

Figure 9 presents the result, in which the X-axis is the timeline of the bushfires and the Y-axis is the image Euclidean distance metric (Equation 16). At the beginning, the distance is high due to the lack of confidence. Later it decreases and becomes stable, but never reach zero. This could be explained by the resolution and integration errors when querying streams of satellite data and weather data for the region of interest.

C. COMPREHENSIVE COMPARISON
We now compare our model with all baselines. The results are reported in Table 5 by averaging over all datasets. In general, our model has at least 1.8% higher predictive power (Weighted F1) and 1.2 times earlier detection.

D. ABLATION TEST OF MODEL COMPONENTS
In this experiment, we want to verify that whether all the model components contributes to the overall performance. To this end, we measure the accuracy, training time, testing time, lag time to detection when replacing model components by other designs.

Table 6 depicts the result when replacing the convolutional layer and LSTM layer of our model with other designs. It can be observed that the performance drops for all datasets when we replace the convolutional layer with a multilayer perceptron (MLP), which is a FCN with drop-out. The performance degradation could be explained by the fact that local spatial dependencies are not captured by MLP. However, using MLP is indeed faster in training and testing, but with increased lag time. On the other hand, using a FCN of convolutional layer does not lead to convergence due to its high complexity.

Another interesting finding is that, replacing LSTM layer with a recurrent neural network only reduces the performance slightly (93.41% compared to 93.17% F1-score) and has a similar lag time. This is because, bushfires spread quite fast over time and thus might not require capturing preceding information that is too old.

E. IMPORTANCE OF MULTI-MODAL DATA
1) EFFECTS OF WEATHER DATA
This experiment evaluates the effects of weather data on classification performance. To this end, we compare our model (with weather) with the case without weather by removing the weather data and fusion layer out of our model. Figure 10 illustrates the result. It can be seen that using weather data improves the performance for all fire datasets, hinting the correctness of our approach in exploiting multi-modal information.
2) EFFECTS OF TEMPORAL DATA

We also evaluate the effects of the temporal information on the prediction. To this end, we compare two cases: (i) temporal: our proposed model with the temporal layer, (ii) non-temporal: we remove the temporal layer out of our model and consider images in different time-points as individual data samples. Figure 11 depicts the result, in which the X-axis is dataset and Y-axis is the prediction performance. The interesting observation is that the model performs worse without temporal data, even though this would result in more training data for model learning.

F. ROBUSTNESS TO ADVERSARIAL CONDITIONS

This experiment shows that our model is robust to adversarial conditions.

1) EFFECTS OF NIGHT-TIME

We compare the prediction performance in daytime vs. nighttime in Figure 12. Our model is better than the best baseline so far. Even in night-time, our model can still achieve prediction power with $\geq 0.9$ F1-score.

2) EFFECTS OF CLOUD

The bushfire detection methods are compared separately on cloudy images vs. non-cloud images. Figure 13 summarises the result. Again, our model outperforms the best baseline (MODIS) according to previous experiments. Especially,
FIGURE 17. Correlation of 16 Spectral Channels in Satellite Images. Blue points: values of fire-negative pixels. Yellow points: values of fire-positive pixels.

MODIS (and even other baselines) cannot detect in cloudy setting. Whereas, our model still achieves $\geq 0.85$ in cloudy situations.

G. INTERPRETABILITY OF PREDICTION RESULTS
We show the following qualitative usecases to prove the interpretability of our model.

1) SHOWCASE ON FERGUSON FIRE DATASET
Figure 14 depicts the qualitative comparison of bushfire detection results between baselines. The online detection performance of the baselines are queried from the USDA Forest Service [70]. All of the images are captured at day 14 in July 2018 at 22:00 UTC since all satellite systems have full for this area at that moment.

Overall, our model and MODIS give highly accurate predictions. This is because MODIS is low-Earth orbiting system which provides very high spatial-resolution images. However, the disadvantage of orbiting systems such as MODIS and AVHRR is that not a single spot on the Earth’s surface can be sensed continuously due to their daily cycle, hindering the need of early detection need. Examining more specifically, there is a slight difference between MODIS and our model: MODIS shows 2 false alarms (fire-positive pixels)
corresponding to the smoke at the edge of the fires while our model produced a more accurate prediction by locating the center of the fires.

It is interesting to observe that VIIRS (which is a MODIS enhanced version) could not produce the right prediction. This is because its satellite position does not capture the fire at the right time. This observation emphasizes the weakness of polar-orbit systems, which can only visit the same point in the Earth at most two times per day. In contrast, imagery from the GOES-16 is continuous and frequent enough with new data every 15min.

2) SHOWCASE ON CAMP FIRE DATASET
In this usecase, we evaluate our method and MODIS-Terra (which is chosen for showcase due to its superior performance against other baselines above) on Camp Fire dataset. Figure 15 shows the qualitative result of the baseline on: (a) the true color image captured by MODIS on November, 8th 2018, (b) Terra prediction at 14:00 UTC, (c) Terra prediction at 20:00 UTC. The visualisation is only shown at these two moments since the Terra platform only acquires MODIS data twice per day at mid-latitudes [49]. Although its output is visible at 20:00 UTC, the fire, in fact, already started at approximately 14:00 UTC, which is 6-hour before-hand.

Figure 16 depicts the qualitative output of our model on the same event. In contrast with the baseline, our model outperforms w.r.t. lag time. The output is captured every 15m from 15:15 UTC to 16:00 UTC. Our model can detect the fire from 15:30 UTC, which is only 1.5 hour behind – a significant damage mitigation over the baseline. In addition, with a short interval (15m), users can monitor and raise the alarm in near real-time.

VIII. CONCLUSION
This article proposes a remote sensing bushfire detection framework by integrating multi-modal data. In this framework, we propose (i) a processing engine for raw data streams, (ii) a spatio-temporal data integration algorithm to align satellite data and weather data across space and time dimensions, and (iii) a multi-scale bushfire prediction model that simultaneously captures spatial, temporal, spectral, and multi-modal patterns. The real-world experiments show the superiority of our model with 93.4% accuracy and 1.2x smaller lag time.

The developed system has profound implications for government and other co-operations seeking to better prevent bushfire damages. Our work could be extended in several directions. First, one might need to further enhance the capacity of the developed neural network to incorporate illumination information, as daytime and nighttime produces different features map for spectral images. Second, one can incorporate further data streams such as drone [71], air quality [72], cloud and moisture imagery, and lightning mapper [36]. Third, further pre-processing techniques such as cloud removal [42] and missing data reconstruction [43] could be employed to improve the performance. Finally, the model could be enhanced to counter the landscape drift problem, in which the model trained on one ROI is not transferrable to another ROI due to the differences in their landscape characteristics.

APPENDIX

STATISTICAL DATA ANALYTICS OF SPECTRAL CHANNELS
Figure 17 presents the full spectral correlation analysis by showing the pair-plots between every pair of spectral channels and their own distributions. It can be observed that the dependency between spectral channels are rather local, e.g. the hints of detecting fire-positive pixels can be found in the correlation between two channels.

REFERENCES
[1] P. E. Dennison, D. A. Roberts, and L. Kammer, “Wildfire detection for retrieving fire temperature from hyperspectral data,” J. Sci. Eng. Res., vol. 4, no. 7, pp. 126–133, 2017.
[2] Y. H. Habiboglu, O. Gunay, and A. E. Cetin, “Real-time wildfire detection using correlation descriptors,” in Proc. 19th Eur. Signal Process. Conf., Aug. 2011, pp. 894–898.
[3] O. Gunay, B. U. Toreyin, K. Kose, and A. E. Cetin, “Entropy-functional-based online adaptive decision fusion framework with application to wildfire detection in video,” IEEE Trans. Image Process., vol. 21, no. 5, pp. 2853–2865, May 2012.
[4] V. Slavkova, S. Verstoeck, S. Van Hoecke, and R. Van de Walle, “Review of wildfire detection using social media,” Fire Saf. J., vol. 68, pp. 109–118, Aug. 2014.
[5] I. R. Felix, L. A. Castro, L.-F. Rodriguez, and O. Banos, “Mobile sensing for behavioral research: A component-based approach for rapid deployment of sensing campaigns,” Int. J. Distrib. Sensor Netw., vol. 15, no. 9, Sep. 2019, Art. no. 155014719987418.
[6] H. Chen, B. Guo, Z. Yu, and L. Chen, “A location-constrained crowdsensing task allocation method for improving user satisfaction,” Int. J. Distrib. Sensor Netw., vol. 15, no. 10, Oct. 2019, Art. no. 155014771988598.
[7] L. Giglio, W. Schroeder, and C. O. Justice, “The collection 6 MODIS active fire detection algorithm and fire products,” Remote Sens. Environ., vol. 178, pp. 31–41, Jun. 2016.
[8] T. K. McGee, “Public engagement in neighbourhood level wildfire mitigation and preparedness: Case studies from Canada, the US and Australia,” J. Environ. Manage., vol. 92, no. 10, pp. 2524–2532, Oct. 2011.
[9] F. Bach, “Breaking the curse of dimensionality with convex neural networks,” J. Mach. Learn. Res., vol. 18, no. 19, pp. 629–681, 2017.
[10] 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,” Int. J. Digit. Earth, vol. 12, no. 9, pp. 1030–1045, Sep. 2019.
[11] W. Xu, M. J. 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,” Remote Sens. Environ., vol. 114, no. 9, pp. 1876–1895, Sep. 2010.
[12] T. Poggio, H. Mhaskar, L. Rosasco, B. Miranda, and Q. Liao, “Why and when can deep-but not shallow-networks avoid the curse of dimensionality: A review,” Int. J. Autom. Comput., vol. 14, no. 5, pp. 503–519, Oct. 2017
[13] D. M. Doolin and N. Sitara, “Wireless sensors for wildfire monitoring,” Proc. SPIE, vol. 7565, pp. 477–485, May 2005.
[14] E. Maggiori, Y. Tarabalka, G. Charpillet, and P. Alixie, “Convolutional neural networks for large-scale remote-sensing image classification,” IEEE Trans. Geosci. Remote Sens., vol. 55, no. 2, pp. 645–657, Feb. 2017.
[15] K. A. Nguyen, R. A. Stewart, and H. Zhang, “An autonomous and intelligent expert system for residential water end-use classification,” Expert Syst. Appl., vol. 41, no. 2, pp. 342–356, Feb. 2014.
[16] S. P. Flasse and P. Ceccato, “A contextual algorithm for A VHRR fire detection,” Int. J. Remote Sens., vol. 17, no. 2, pp. 419–424, Jan. 1996.
[17] W. Schroeder, P. Oliva, L. Giglio, and I. A. Csiszar, “The new VIIRS 375m active fire detection data product: Algorithm description and initial assessment,” Remote Sens. Environ., vol. 143, pp. 85–96, Mar. 2014.
[18] R. Xu, H. Chen, X. Liang, and H. Wang, “Priority-based constructive algorithms for scheduling agile Earth observation satellites with total priority maximization,” Expert Syst. Appl., vol. 51, pp. 195–206, Jun. 2016.
T. C. Phan et al.: Multi-Scale Bushfire Detection From Multi-Modal Streams of Remote Sensing Data
[67] 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,” in Global and Regional Vegetation Fire Monitoring From Space, Planning and Coordinated International Effort. The Hague, The Netherlands: SPB Academic Publishing, 2001, pp. 199–225.

[68] E. Zervas, A. Mpimpoudis, C. Anagnostopoulos, O. Sekkas, and S. Hadjieothymiades, “Multisensor data fusion for fire detection,” Inf. Fusion, vol. 12, no. 3, pp. 150–159, Jul. 2011.

[69] J. Gou, H. Ma, W. Ou, S. Zeng, Y. Rao, and H. Yang, “A generalized mean distance-based k-nearest neighbor classifier,” Expert Syst. Appl., vol. 115, pp. 356–372, Jan. 2019.

[70] USDA. (2019). Daily Georeferenced Modis Image Subsets. [Online]. Available: https://fsapps.nwcg.gov/afm/imagery.php

[71] Y.-H. Ho, Y.-T. Huang, H.-H. Chu, and L.-J. Chen, “Adaptive sensing scheme using naive bayes classification for environment monitoring with drone,” Int. J. Distrib. Sensor Netw., vol. 14, no. 1, Jan. 2018, Art. no. 155014771875603.

[72] C. Feng, Y. Tian, X. Gong, X. Que, and W. Wang, “MCS-RF: Mobile crowdsensing-based air quality estimation with random forest,” Int. J. Distrib. Sensor Netw., vol. 14, no. 10, Oct. 2018, Art. no. 155014771880470.

**THANH CONG PHAN** received the master’s degree in computer science from Griffith University, Australia. He is currently a Lecturer with HUTECH, Vietnam. He has publications in prestigious journals and conferences, such as TIST, Information Fusion, WIMS, and ADC. His research interests include artificial intelligence, data mining, recommender systems, and big data analytics.

**THANH TAM NGUYEN** received the Ph.D. degree in data science from the Ecole Polytechnique Federale de Lausanne (EPFL), Switzerland. He is currently a Lecturer with HUTECH, Vietnam. He has published several articles in top-tier venues, such as SIGMOD, VLDB, SIGIR, KDD, AAAI, ICDE, IJCAI, JVLDB, TKDE, TOIS, and TIST. His research interests include data integration, data quality, information retrieval, trust management, recommender systems, machine learning, and big data visualization.

**QUOC VIET HUNG NGUYEN** received the Ph.D. degree from EPFL, Switzerland. He is currently a Senior Lecturer with Griffith University, Australia. He has published several articles in top-tier venues, such as SIGMOD, VLDB, SIGIR, KDD, AAAI, ICDE, IJCAI, JVLDB, TKDE, TOIS, and TIST. His research interests include data integration, data quality, information retrieval, trust management, recommender systems, machine learning, and big data visualization.

**JUN JO** received the Ph.D. degree from The University of Sydney, in 1994. He has been working on various research projects, including computer vision and machine learning, and their applications in various areas, including robots, autonomous cars, drones, the IoTs, satellite image analysis, and medical image analysis. He has published more than 150 refereed publications. He is also the President of the Australian Robotics Association and the Chair of World Innovative Technology.

---

**THANH DAT HOANG** received the bachelor’s degree in computer science from the Hanoi University of Science and Technology, Vietnam, in 2020. He has publications in top-tier conferences and journals, such as ICDE, ESWA, and PRICAI. His research interests include deep learning and benchmarking technology.