

Earth and Space Science



RESEARCH ARTICLE

10.1029/2024EA003997

Special Collection:

Analyzing Big Data for Understanding Climate Variability, Natural Phenomena and Rapid Environmental Changes

Key Points:

- This research addresses a gap in the literature by exploring the advantages of unsupervised learning in wildfire prediction
- This study focuses on Australia using a data set comprising historical weather and normalized difference vegetation index
- Our findings demonstrate the effectiveness of unsupervised learning using deep autoencoders for anomaly detection with an F1-score of 0.74

Correspondence to:

M. Arana-Catania,
miguel.aranacatania@cranfield.ac.uk

Citation:

Üstek, İ., Arana-Catania, M., Farr, A., & Petrunin, I. (2024). Deep autoencoders for unsupervised anomaly detection in wildfire prediction. *Earth and Space Science*, 11, e2024EA003997. <https://doi.org/10.1029/2024EA003997>

Received 27 SEP 2024

Accepted 28 OCT 2024

Author Contributions:

Conceptualization: İrem Üstek, Miguel Arana-Catania, Alexander Farr, Ivan Petrunin
Formal analysis: İrem Üstek
Methodology: İrem Üstek, Miguel Arana-Catania, Alexander Farr, Ivan Petrunin
Resources: Alexander Farr
Software: İrem Üstek
Supervision: Miguel Arana-Catania, Alexander Farr, Ivan Petrunin
Visualization: İrem Üstek
Writing – original draft: İrem Üstek, Miguel Arana-Catania, Ivan Petrunin

© 2024. The Author(s).

This is an open access article under the terms of the [Creative Commons Attribution License](#), which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

Deep Autoencoders for Unsupervised Anomaly Detection in Wildfire Prediction

İrem Üstek¹, Miguel Arana-Catania¹ , Alexander Farr², and Ivan Petrunin¹

¹School of Aerospace, Transport and Manufacturing, Cranfield University, Wharley End, UK, ²Spire Global, Inc., Boulder, CO, USA

Abstract Wildfires pose a significantly increasing hazard to global ecosystems due to the climate crisis. Due to its complex nature, there is an urgent need for innovative approaches to wildfire prediction, such as machine learning. This research took a unique approach, differentiating from classical supervised learning, and addressed the gap in unsupervised wildfire prediction using autoencoders and clustering techniques for anomaly detection. Historical weather and normalized difference vegetation index data sets of Australia for 2005–2021 were utilized. Two main unsupervised approaches were analyzed. The first used a deep autoencoder to obtain latent features, which were then fed into clustering models, isolation forest, local outlier factor and one-class support vector machines for anomaly detection. The second approach used a deep autoencoder to reconstruct the input data and use reconstruction errors to identify anomalies. Long Short-Term Memory autoencoders and fully connected (FC) autoencoders were employed in this part, both in an unsupervised way learning only from nominal data. The FC autoencoder outperformed its counterparts, achieving an accuracy of 0.71, an F1-score of 0.74, and an MCC of 0.42. These findings highlight the practicality of this method, as it effectively predicts wildfires in the absence of ground truth, utilizing an unsupervised learning technique.

1. Introduction

Forests are essential to maintaining the planet's ecological equilibrium, but they face threats from fires caused by both natural phenomena and human activities (Qiang et al., 2011). These wildfires are catastrophic events that negatively impact the environment, the economy, and the natural resources (Meddour-Sahar, 2015; Sayad et al., 2019). On the human front, they can lead to loss of life, respiratory ailments, and other health challenges. At the same time, economically, they can devastate communities, affecting livelihoods and leading to billions in damages. The impact on wildlife is also severe, threatening a diverse range of species with effects such as respiratory distress, altered behavior, and even the threat of extinction. Simultaneously, the natural environment suffers degradation through soil erosion, loss of biodiversity, and the destruction of essential ecosystems such as forests (Sanderfoot et al., 2022).

In recent years, the frequency of wildfires has surged, becoming a global concern that has drawn attention from various fields of study (Abid, 2021). A very recent example would be the wildfire episodes encompassing 25 million acres in Canada in 2023 (Canada, 2023). The growing threat of wildfire disasters is primarily attributed to climate change. These changes, characterized by higher temperatures and extended dry periods, have led to unprecedented bushfire activities, particularly within Australia (Vardoulakis et al., 2020).

In 2019, Australia, the focus area of this research, encountered its hottest year on record, with the annual national mean temperature surpassing the average by 1.52°C. It also experienced its driest year on record, with notable heatwaves occurring in January and December (Yu et al., 2020). Between 2019 and 2020, Australia witnessed one of the largest wildfires in modern record. A minimum area of 46 million acres of land was burnt (Sulova & Jokar Arsanjani, 2020). The 2019–2020 wildfire is noted for having the strongest measures on record in terms of both magnitude and intensity (Zhang et al., 2021). Regarding their impacts on climate change and the environment, these fires directly claimed 33 human lives and over one billion indigenous animals (Norman et al., 2021). Furthermore, 417 individuals succumbed to smoke inhalation, and it is believed that 80% of Australia's population was directly or indirectly affected by the fires (Borchers Arriagada et al., 2020; Norman et al., 2021; Ogie et al., 2022). Moreover, projections indicate that such wildfires will continue to occur due to multiple factors. Based on scientific estimates by the United Nations, extreme wildfires could increase by as much as 50% by the century's end, spurred by the climate crisis and poor land management (Qiang et al., 2011).

Writing – review & editing: İrem Üstek,
Miguel Arana-Catania, Alexander Farr,
Ivan Petrunin

Simulating wildfires is challenging due to the complex interactions of weather conditions, fuel setups, and unpredictable fire movements. The wide geographical and time scales involved make this difficulty even greater, making it a problem that requires significant computational resources (Sayad et al., 2019). The difficulty in predicting wildfires emphasizes the urgent need for innovative approaches, such as machine learning (ML) techniques. Indeed, numerous studies have been conducted to predict or detect wildfires in various scenarios using supervised learning techniques, such as support vector machines (SVM), random forests (RF), artificial neural networks (ANN), decision trees (DT), and more (Abid, 2021; Pang et al., 2022; Pérez-Porras et al., 2021; Shmuel & Heifetz, 2022; Sulova & Jokar Arsanjani, 2020). However, unsupervised learning in wildfire detection has not received much attention from researchers in the prediction phase, and only has been utilized in exploratory data analysis (Sayad et al., 2019). This research aims to bridge this gap by leveraging the advantages of unsupervised learning, such as accessibility to more data without the need for specific wildfire data sets, and the capability for broader and more flexible analysis without requiring labeled data. Advancements in unsupervised learning, exemplified by generative adversarial networks, a type of unsupervised autoencoder, have been shown to have potential for rapid, real-time applications in contexts such as wildfire detection (Zenati et al., 2018). This underscores the broader applicability and benefits of unsupervised learning techniques in this field. The benefits of using unsupervised learning methods are also shown in multiple and varied domains, requiring less computational power, less time consumption and use of less amount of data while yielding relatively good quality results in near-real time (Nhangumbe et al., 2023), reducing the data processing effort by up to 97% while being reliable enough for the task requirements (Lehr et al., 2021), showing strong scalability and interpretability (Chang et al., 2020), achieving the task requirements while reducing the training cost (Tuo et al., 2023), and sometimes producing results on par with supervised techniques (Haar et al., 2019; Pradhan et al., 2021).

The study will specifically narrow its focus to Australia. This area is particularly prone to wildfires and has a significant history in this domain, making it a valuable subject for examination. The research aims to predict future wildfires based on anomaly detection before they happen by employing unsupervised learning techniques, using deep autoencoders and clustering. The data set employed for this project is the combination of two separate data sets: historical weather and Normalized Difference Vegetation Index (NDVI) (IBM, 2020). It employs unsupervised learning techniques, specifically deep autoencoders and clustering, to predict wildfires through anomaly detection, utilizing a unique data set comprising historical weather and normalized difference vegetation index data. The techniques employed are some of the most common unsupervised methods for anomaly detection (Z. Cheng et al., 2021; Li et al., 2021). This research addresses a gap in the literature by exploring the potential advantages of unsupervised learning in wildfire prediction, such as its reliance on readily available data and reduced computational demands (Lehr et al., 2021; Nhangumbe et al., 2023; Tuo et al., 2023). In addition to primary contributions, this study also addresses challenges in wildfire prediction and recommends ways to enhance predictive models in this domain.

The remaining sections of this paper are organized as follows: Section 2 offers a literature review; Section 3 provides details about the study area, data set, and methodology; Section 4 presents results and findings; Section 5 discusses the results; and Section 6 provides conclusions.

2. Related Work

In this section we review relevant research to our project, beginning with a general study on wildfire prediction using ML, followed by specific reviews of the techniques employed in our work: anomaly detection, autoencoders, and the combination of both.

2.1. Wildfire Prediction Using ML Models

Various ML techniques and features have been applied and researched regarding forest fire prediction and detection. The most common and popular ML models used are the ANN, SVM, RF, XGBoost, and DT, as highlighted by Jain et al. (2020) in their review of 300 different papers.

A common focus in the works exploring ML techniques has been the joint use or comparison between SVM and ANN models. For instance, Sayad et al. (2019) combined in a single pipeline SVM and ANN to forecast wildfires using a data set based on the NDVI, land surface temperature (LST), and thermal anomalies from the Moderate Resolution Imaging Spectroradiometer (MODIS), and three characteristics relevant to the status of crops. The findings demonstrated high fire occurrence prediction accuracy (ANNs: 98.32%; SVM: 97.48%), defining it as

the ratio of correctly predicted instances to the total instances. The study integrated NDVI, LST, and thermal anomalies with ground truth labeled data. However, the researchers recommended integrating weather data due to its strong relationship with wildfire incidence, development, spread, and extinction. They especially highlighted utilizing certain variables which drastically affect wildfire incidences and likelihood: air temperature, wind, and soil moisture Sayad et al. (2019).

Another study included weather data, as suggested by Sayad et al. (2019). Sakr et al. (2011) compared SVM and ANN for predicting the likelihood of a fire using cumulative precipitation and relative humidity for several months of the year. SVM performed better in binary fire/no fire classification with the highest accuracy of 94.21%, while ANN obtained a maximum of 91.2%.

Gholamnia et al. (2020) took a distinct approach and explored ML models for mapping wildfire susceptibility instead of predicting them. Using wildfire inventory data from global positioning systems and MODIS fire occurrences, they considered 16 conditioning factors, including meteorological parameters and a vegetation index. Among the 11 ML techniques compared, RF showed the highest accuracy of 88%, followed by SVM at 79%. The study is noteworthy for implementing the unsupervised self-organizing maps (SOM) technique, which was however applied to a two-class data and has been adapted for use for classification purposes. The least accurate models were RBF, logistic regression, and SOM. Although not directly comparable to the subject of this paper, it serves as a valuable example of ML's potential in the wildfire susceptibility mapping domain.

Turning attention to different geographical contexts, Sulova and Jokar Arsanjani (2020) conducted research similar to this study in terms of the study area, focusing on predicting wildfire risk in Australia over 6 months. They used topography, vegetation, infrastructure, meteorology, and socioeconomic data. The study employed Naïve Bayes (NB), Classification and Regression Tree, and RF models, with RF achieving the highest accuracy of 96% and NB having the lowest accuracy of 64%.

Kondylatos et al. (2022) aimed to accurately forecast wildfire danger using deep learning models such as Long Short-Term Memory (LSTM) and ConvLSTM and traditional models like RF and XGBoost. The research covered Greece, parts of the Balkan peninsula, and western Turkey. They utilized various data sets, which were including daily weather data from ERA-5 Land and satellite variables from MODIS, such as NDVI and LST. Results from the study showed that LSTM and ConvLSTM achieved F1-scores greater than 0.8, outperforming RF and XGBoost in all metrics. ConvLSTM exhibited fewer false positives and higher precision, while LSTM had the lowest number of false negatives and highest recall.

Arpaci et al. (2014) examined fire occurrence data in Austria's Tyrol area from 1993 to 2011, incorporating terrain, vegetation, climate, and socioeconomic factors. Their research intended to locate fire-prone locations and predict the geographical spread of fires. They used the Maximum Entropy (MaxEnt) and RF methods to identify the primary variables influencing the spread of fires. Climate change and population density have become essential factors in the susceptibility of a given area to wildfires. RF model was able to forecast the distribution of fire ignition with a classification accuracy of between 75% and 78%. Both MaxEnt and RF algorithms were usually equivalent when categorizing fire susceptibility levels, except for the extremely high susceptibility class, where MaxEnt identified 28 places compared to 2 by RF (Arif et al., 2021).

Malik et al. (2021) conducted fire risk prediction in San Diego, California, using weather and fire history data, including temperature, soil moisture, relative humidity, and wind speed. SVM, XGBoost, and RF achieved accuracies of 98.18%, 98.23%, and 97.54%, respectively, while the F1-scores were 97%, 98%, and 91%, respectively.

2.2. Anomaly Detection in Wildfire Prediction

Anomaly detection aims to identify anomalous patterns, often known as outliers or anomalies, that significantly deviate from the rest of the data (Thudumu et al., 2020). Such a task can be performed using supervised or unsupervised learning techniques. In this study, an unsupervised learning version will be implemented. In extreme event scenarios, it is difficult to find appropriately labeled data sets because detecting anomalous observations requires effort and expertise. Therefore, supervised learning based on labeled data may not be a viable and suitable solution in all anomaly detection circumstances (Esmaeili et al., 2023). This underscores the value and contribution of unsupervised anomaly detection.

To the best of our knowledge, there is no study implementing anomaly detection directly in wildfire prediction using unsupervised learning techniques. However, there exists a closely related study working in wildfire risk prediction using anomaly detection method, even though it is not an exact match with this research's topic. In their research, Salehi et al. (2016) set out to use weather data to better predict the risk of wildfires. They devised a new model named Context-Based Fire Risk (CBFR) that considers the changing patterns of weather over time using a context-based anomaly detection method. They used ECMWF ERA-5 weather data from the Blue Mountains in Australia. To assess the performance of their model, the authors compare it to the commonly used McArthur FFDI method. They looked at AUC and the straight-up accuracy rate. Their CBFR model performed better on both counts. Specifically, CBFR had an AUC of 0.96 and an accuracy of 0.94, while the FFDI method scored 0.89 for AUC and 0.87 for accuracy.

2.3. Autoencoders and Dimensionality Reduction

The first autoencoder neural network was developed to reduce dimensionality (Masci et al., 2011), showing specific advantages over other types of traditional dimensionality reduction methods. For instance, since they are non-linear, autoencoders can generally outperform linear traditional methods in the context of dimensionality reduction (Bank et al., 2020).

Although there have been no previous examples of such use of autoencoders in the wildfire prediction domain, there is one study focusing on post-fire assessment (Coca & Datcu, 2021). The authors introduced a framework for burned area estimation using multispectral images. Relying on ESA's Sentinel-2 (S2) sun synchronous satellite imagery, they focused on a wildfire-impacted area in Portugal from 2017. The authors employed an autoencoder to extract deep features from the data. They used the autoencoder's capability to not only reduce dimensionality, but also extract deep features from input data. After attaining these features, one-class Support Vector Machine (Shin et al., 2005) is utilized for anomaly detection. The experiment processed both normal and abnormal patches from the data set, achieving an accuracy of 0.71 at 7.9% abnormality.

2.4. Autoencoders and Anomaly Detection

2.4.1. Anomaly Detection Based on Reconstruction Error

Using autoencoders to detect anomalies using the reconstruction error is a technique well applied in several areas. The idea is as follows: a trained autoencoder would learn the latent subspace based on only normal (non-anomalous) instances. Once trained, the input data is reconstructed, and reconstruction errors, that is, the difference between the original and reconstructed data, are calculated. As a result, while normal instances have low reconstruction errors, anomalies have high reconstruction errors (Bank et al., 2020).

To the best of our knowledge, no existing literature has explored anomaly detection using only deep autoencoders and reconstruction error in the context of wildfire prediction. The forthcoming referenced studies, while they are from varying domains, share a methodology closely aligned with this research.

D. Cheng et al. (2022) utilized a ResNet autoencoder to identify anomalies in radar data. Their model incorporated both convolutional and LSTM layers. Convolutional layers captured features, while LSTM layers assessed temporal dependencies in radar data. The model achieved an accuracy potential of up to 85%. Esmaeili et al. (2023) trained deep autoencoders on normal data to spot anomalies in a similar domain. Using autoencoder-based prediction models on electrochemical aptasensor recordings with various signal lengths, they used reconstruction errors to set up a threshold for assigning anomalies and used kernel density estimation. They implemented vanilla, ULSTM, and Bi-LSTM autoencoder. The integrated ULSTM-vanilla model achieved about 80% accuracy for longer signal data sets, 65% and 40% for shorter ones.

Zhao et al. (2017) introduced a unique model called Spatio-Temporal Autoencoder (STAE) in a different domain that learns video representation and detects anomalies in the video by extracting features from spatial and temporal dimensions using deep autoencoders. The utilization of reconstruction error shares common points with the approach employed by Hasan et al. (2016). Once their model was trained, both studies produced a “regularity score” based on the reconstruction error, highlighting that video sequences with regular events had low errors. In contrast, anomalous sequences had high errors, allowing them to spot anomalies.

Table 1
Details of Used Data Sets (IBM, 2020)

Data set	Pre-processing details
Wildfires (MODIS/Aqua+Terra Thermal Anomalies/Fire locations 1km FIRMS V0061 NRT, 2020)	Spatial averaging into seven regions in Australia Aggregation by day from 2005 to 2021 Only high-confidence fires (>75%) considered
Historical Weather - ERA5-Land hourly data (Muñoz Sabater, 2019)	Raw ERA5 data was processed using IBM PAIRS Geoscope Temporal aggregation, unit conversion, and transformations are applied to each parameter. Spatial aggregation using physical area weights of each pixel
NDVI - MOD13Q1 MODIS/Terra Vegetation Indices 16-Day L3 Global 250 m SIN Grid V006 (Didan, 2015)	Spatial averaging into seven regions in Australia from 2005 to 2021

2.4.2. Anomaly Detection Based on Clustering Through Latent Features

Clustering is organizing unlabeled data into similar groupings known as clusters. A cluster is a group of data items that are similar to one another but differ from other clusters (Serra & Tagliaferri, 2019). When labeled data is not available, clustering offers a valuable alternative to supervised learning techniques.

Nevertheless, most traditional clustering algorithms are sensitive to the dimension of the input data, making them suffer from “the curse of dimensionality” (Saxena et al., 2017), which makes them less effective in high-dimensional spaces than in low-dimensional data (Aggarwal & Reddy, 2013). In order to solve this problem, dimensionality reduction-based clustering algorithms have been developed. These techniques translate data to lower dimensions utilizing techniques like PCA, kernel-based PCA, and autoencoders (Ma, 2022). For instance, Tian et al. (2014) described a simple method that involved employing stacked autoencoders to acquire the non-linear embedding of the features (latent features) and then feeding that information directly to k-means clustering. Similarly, Ma (2022) examined several models for various clustering techniques on several data sets, including MNIST, HHAR, and REUTERS-10K. It has been noted that the Gaussian Mixture Model (GMM) alone only achieved 47.8% accuracy for the MNIST data set. However, the accuracy rose to 79.34% when autoencoders and GMM were combined (AE + GMM), which might be a promising improvement. Like this, when AE and latent characteristics were included, normalized mutual information findings rose from 38.24% to 81.41%.

3. Methodology

The code within this research has been developed using Python 3.10.10. The procedures related to autoencoder modeling, including training, compilation, fine-tuning, and adaptation, were developed using TensorFlow 2.10.0 (Abadi et al., 2016). In the context of clustering, the Scikit-learn 1.2.2 library was employed (Pedregosa et al., 2018).

3.1. Data

This paper focuses on the region of Australia. It includes seven states: New South Wales (NSW), Northern Territory (NT), Queensland (QL), South Australia (SA), Victoria (VI), Western Australia (WA), and Tasmania (TA). The data set used is a combination of 2 data sets. The first data set, released by IBM, is the historical weather and vegetation index. It was released for the Call for Code Spot Challenge for Wildfires competition, and is publicly available (IBM, 2020). It includes historical weather and vegetation index information from 2005 to 2021 as daily aggregated values. The second data set, the Wildfires data set is a component of the main data set (IBM, 2020) and it is only used to evaluate the performances of the obtained model. Since we are following an unsupervised approach, the “wildfire” ground truth information obtained from the Wildfires data set is not used during any training phase. Necessary information regarding the data sets is summarized in Table 1.

The data presented is aggregated daily and encompasses estimations by NASA, having undergone certain processing stages (IBM, 2020).

Figure 1 provides an overview of fire area size across all Australian regions from 2005 to 2021. A significant surge in fire incidents stands out in 2006 and 2007. Remarkably, there was another notable increase in active fire occurrences during 2011 and 2012, followed by a subsequent rise in wildfires during 2019 and 2020.

Fire Area Size

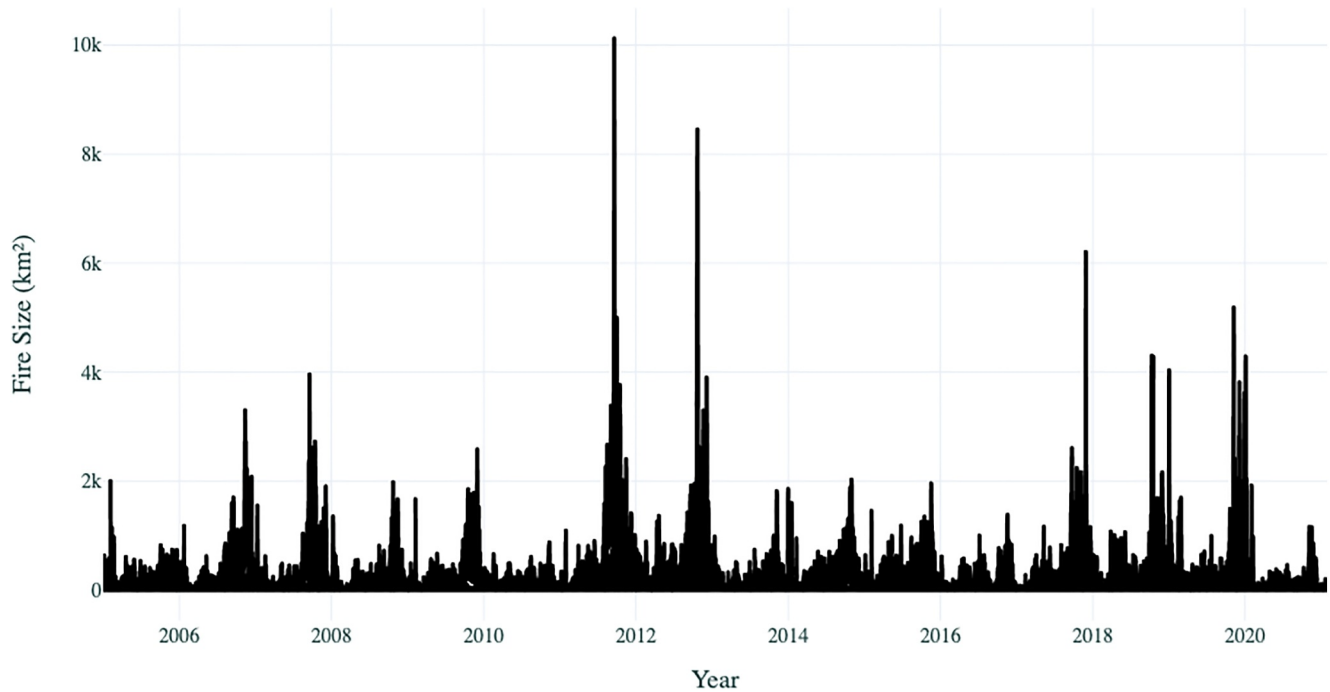


Figure 1. Fire area size over the years for all regions.

The two initial data sets (Weather and NDVI) were merged to be used as input data. Concerning the target variable, given that this is an unsupervised learning work, the actual wildfire ground truth is not utilized during the training and modeling phases. Nonetheless, to evaluate model performance and outcomes, a binary target variable labeled as “Fire” is generated from the Estimated Fire Area Size variable using the subsequent formula:

Table 2
Details of Features

Data set	Features	Range and unit
Wildfires	Date	2005-01-01 - 2021-01-23
	Region	[NSW, NT, QL, SA, VI, WA, TA]
	Estimated Fire Area Size	$0 \leq \text{Estimated Fire Area Size (km}^2\text{)} < 10120.94$
	(Estimated Fire Area Size is converted to a binary variable) Fire	[0, 1]
Historical Weather	Date	2005-01-01 - 2021-01-23
	Region	[NSW, NT, QL, SA, VI, WA, TA]
	Minimum, maximum, mean, and variance of each of these parameters: Precipitation	$0 \leq \text{Precipitation (mm/day)} \leq 509.83$
	Relative humidity	$0 \leq \text{Relative humidity (mm/day)} \leq 509.83$
	Soil water content	$0 \leq \text{Soil water content (m}^3 \text{ m}^{-3}\text{)} \leq 0.52$
	Solar radiation	$0.41 \leq \text{Solar radiation (MJ/day)} \leq 35.69$
	Temperature	$-5.05 \leq \text{Temperature (}^\circ\text{C)} \leq 41.73$
	Wind speed	$0.25 \leq \text{Wind speed (m/s)} \leq 24.27$
NDVI	Date	2005-01-01 - 2021-01-23
	Region	[NSW, NT, QL, SA, VI, WA, TA]
	Minimum, maximum, mean, and variance of NDVI	$0 \leq \text{NDVI (index)} \leq 1$

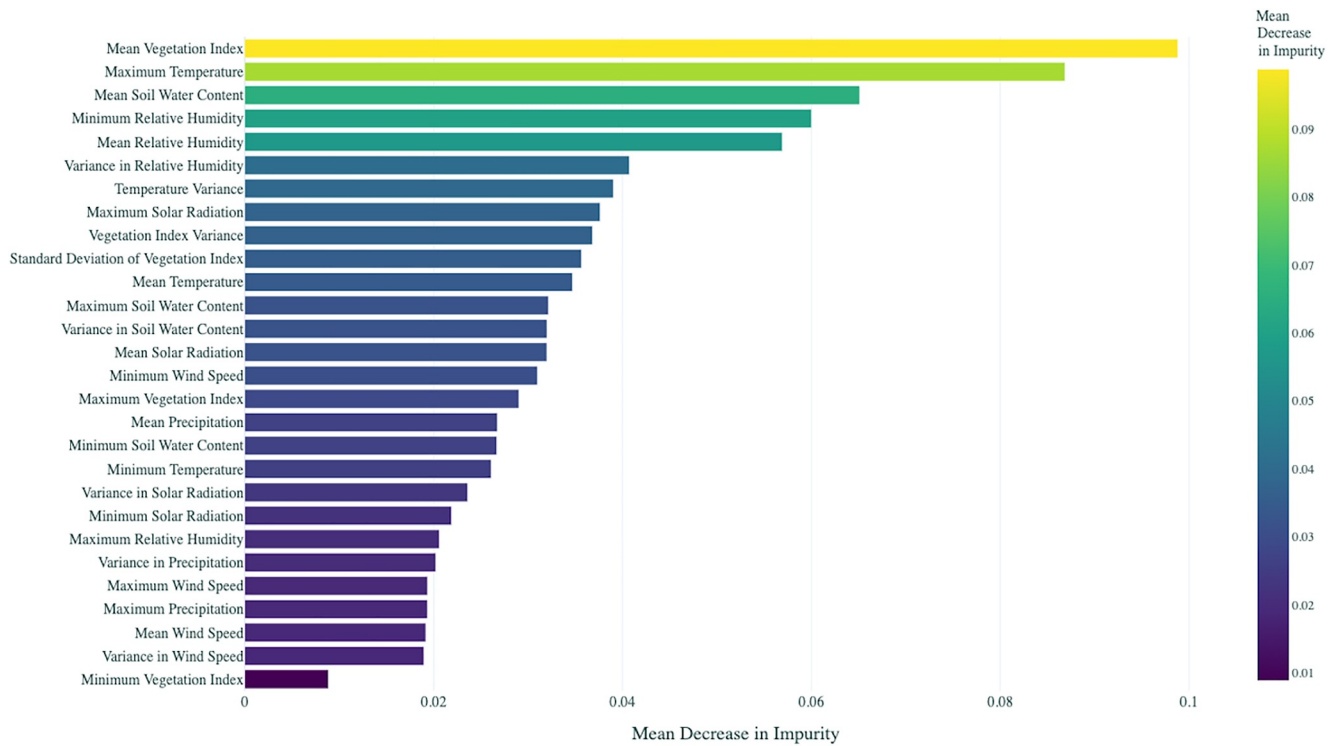


Figure 2. Feature importance values generated by random forests.

$$\text{Fire} = \begin{cases} 1, \text{Estimated Fire Area} \neq 0 \\ 0, \text{Estimated Fire Area} = 0 \end{cases} \quad (1)$$

Table 2 provides an overall list of all features and their range and units. In the final version of the data set, there are 26,677 instances of wildfires and 14,299 instances of non-wildfires. This imbalance arises because the data set is aggregated daily. If even a single wildfire occurs on a given day, that day is labeled as a “wildfire” day.

3.1.1. Feature Importance

Given that certain variables might be interrelated, and some may not provide significant value, RF models inherently select a subset of significant variables for classification, instead of using all variables, and thus enhancing the classification performance of high-dimensional data. This feature selection can be visualized through “Gini importance”, also known as mean decrease in impurity (MDI), which ranks the relevance of features (Sutera, 2021). RF was employed to get a feature importance plot with MDI values, as seen in Figure 2. Parameters used for the RF fit can be found in Table 3.

Figure 2 provides evidence that several variables, such as the Vegetation Index Min, Wind Speed Mean or Wind Speed Variance, contribute minimally to the model. This observation is particularly pronounced in the RF, highlighting numerous less impactful features.

Table 3
Parameters Used in the Random Forests Fit

Parameter	Value
Number of estimators	100
Criterion	“Gini”
Min samples split	2
Min samples leaf	2

In the deep autoencoder modeling part, two data sets will be analyzed:

1. Data set 1 includes all 28 variables without any exclusions.
2. Data set 2 is composed of 17 features. Based on a close look at Figure 2, it is seen that there is a significant decrease in MDI after the Vegetation Index Max value. Yet, since many studies emphasize the significance of the Temperature variable in this field, Temperature Min is retained as well. Consequently, any variable with an MDI value below Temperature Min was manually removed for experimental purposes.

Table 4
Features for Two Versions of the Data Set

Data set 1	Data set 2
Precipitation_Max	Vegetation_index_Mean
Precipitation_Mean	Temperature_Max
Precipitation_Variance	SoilWaterContent_Mean
RelativeHumidity_Max	RelativeHumidity_Min
RelativeHumidity_Mean	RelativeHumidity_Mean
RelativeHumidity_Min	RelativeHumidity_Variance
RelativeHumidity_Variance	Temperature_Variance
SoilWaterContent_Max	SolarRadiation_Max
SoilWaterContent_Mean	Vegetation_index_Std
SoilWaterContent_Min	Vegetation_index_Variance
SoilWaterContent_Variance	Temperature_Mean
SolarRadiation_Max	SoilWaterContent_Max
SolarRadiation_Mean	SoilWaterContent_Variance
SolarRadiation_Min	SolarRadiation_Mean
SolarRadiation_Variance	WindSpeed_Min
Temperature_Max	Vegetation_index_max
Temperature_Mean	Temperature_Min
Temperature_Min	
Temperature_Variance	
WindSpeed_Max	
WindSpeed_Mean	
WindSpeed_Min	
WindSpeed_Variance	
Vegetation_index_Mean	
Vegetation_index_Max	
Vegetation_index_Min	
Vegetation_index_Std	
Vegetation_index_Variance	

Comparison of these two data set results will allow us to assess the models' performance concerning variable contribution and dimensional variance. The complete set of features for both data versions can be found in Table 4. "Region" and "Date" features are omitted as they do not offer any insights into the model.

3.1.2. Data Split

The data is split into three parts: training, validation, and testing. For training, data with only normal (non-wildfire) cases are used. A mix of data from both classes is needed for the validation and test parts. The original data set has 14,299 non-wildfire (NW) cases (35%) and 26,677 wildfire (W) cases (65%). This imbalance within the classes arises due to the daily aggregation of data. If even a single minor wildfire is reported on a given day, that day is labeled as a "wildfire", increasing the wildfire count.

The autoencoder is trained solely on non-wildfire data. Therefore, a substantial portion of NW from the original data set (12,869 cases) is allocated to the train data, ensuring maximum use of the non-wildfire information present. The remaining NW cases are evenly divided between the test and validation data sets, with each receiving 715 NW cases. For test and validation data sets, wildfire cases are initially split into 13,338 and 13,339, respectively. However, given that the train data consisted of a total of 12,869 instances, these numbers seemed disproportionate. To address this, 1,000 wildfire cases are randomly dropped for both the test and validation data sets. Thus, each test and validation set consists of 1,000 wildfire cases with 715 non-wildfire cases, resulting in data sets with ($W = 1,000$, $NW = 715$).

This process ensures that the test and validation data set sizes are not excessively larger than the train data due to the class imbalance. Consequently, the distribution for training, testing, and validation sets is roughly 80%, 10%, and 10%. Still, it is worth noting that, since some instances had to be dropped due to the imbalance in the original data set, some information in the original data was not fully utilized during modeling, which could be listed as a limitation. This whole procedure is applied separately for both Data set 1 and Data set 2. Min-max scaling was employed to ensure that all features in the data set have a uniform scale, allowing them to contribute equally to the model's performance. Each data set - train, validation, and test, was scaled based on its own statistics. Therefore, it was ensured that data leakage did not

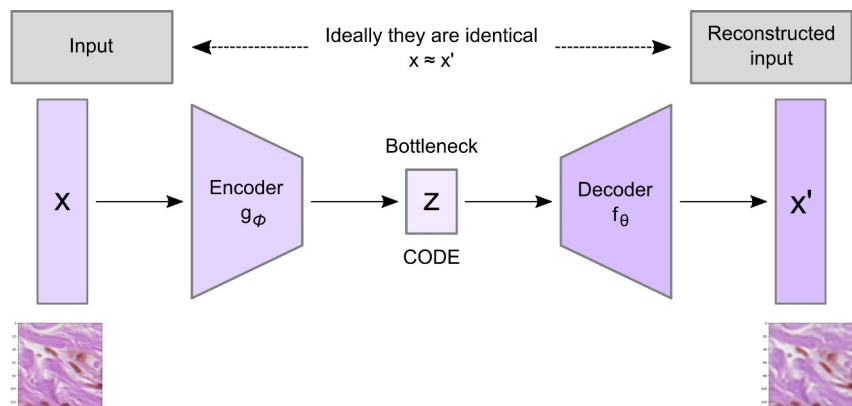


Figure 3. Architecture of a sample autoencoder (Kucharski et al., 2020).

Table 5
Shape of the Data Sets To Be Used in the Long Short-Term Memory Autoencoder

	Train (batch, timesteps, feature)	Test (batch, timesteps, feature)	Validation (batch, timesteps, feature)
Shape (Data set 1)	(1286, 10, 28)	(171, 10, 28)	(171, 10, 28)
Shape (Data set 2)	(1286, 10, 17)	(171, 10, 17)	(171, 10, 17)

occur, preserving the integrity of the evaluation. Through this scaling procedure, it is ensured that every feature within the data sets is within the range [0, 1].

3.2. Methods

In this study, we focus on two main unsupervised approaches: using a deep autoencoder to obtain latent features for clustering models and using a deep autoencoder to reconstruct input data and identify anomalies through reconstruction errors. These strategies have been discussed extensively in

the Related Work section, where their applications and effectiveness in different contexts were reviewed in detail. The main aim here is to fill the research gap in unsupervised learning for wildfire prediction using unlabeled data. Deep autoencoders are utilized to detect anomalies in data, classifying them as “wildfire” or “non-wildfire.” The model is trained on non-anomaly instances. Two strategies are explored:

1. Using the deep autoencoder for training, then reconstructing input data in lower dimensions and applying clustering methods to these latent features to differentiate wildfires.
2. Using the deep autoencoder to reconstruct input data, where high reconstruction errors signify anomalies. Here, both Fully Convolutional (FC) and Long-Short-Term-Memory (LSTM) autoencoders are employed, with the latter being adept at capturing temporal patterns in time-series data, a feature that FC architectures lack.

In the following sections, we start by explaining what a deep autoencoder is in Section 3.2.1, including its variants such as Fully Connected (FC) and LSTM Autoencoders. Then, in Sections 3.2.2.1 and 3.2.2.2, we detail the two main methods utilized in this paper that employ deep autoencoders for anomaly detection.

3.2.1. Deep Autoencoders

An autoencoder is a unique variant of feed-forward neural networks aiming to have the output closely mirror the input. Figure 3 shows that both the input and output layers have matching dimensions. An autoencoder consists of two main parts: an encoder and a decoder as provided in Figure 3.

The encoder section comprises one or multiple hidden layers, in this case, FC or LSTM layers, that compress the input into a condensed representation. Conversely, the decoder uses its hidden layers, which can also be FC or LSTM layers, to reconstruct this condensed representation back to the original input's dimensionality (Kucharski et al., 2020). The unsupervised training process optimizes these hidden layers to minimize the difference between the reconstructed output and the original input. This study will explore autoencoders with FC and LSTM hidden layers to comprehensively understand their performance and capabilities.

3.2.1.1. LSTM Autoencoder

As mentioned, a deep autoencoder can include hidden LSTM layers in its encoder and decoder components. Using LSTM layers provides the advantage of capturing temporal information, if any, in the time-series data. To utilize the LSTM autoencoder, the data set must be reshaped into a format compatible with what the LSTM requires. Thus, for the training, testing, and validation data sets, sequences of a 10-day length were prepared.

Table 6
Hyperparameter Tuning Details for Long Short-Term Memory Autoencoder

Parameter & library	Range	Final choice
Number of layers (tf.keras.layers.LSTM)	[2, 3, 4, 5, 6, 7, 8]	5
Number of units (tf.keras.layers.LSTM)	[1024, 512, 256, 128, 64, 32, 16, 8]	[256, 128, 64, 32, 16]
Activation function (tf.keras.layers.LSTM)	[ReLU, Sigmoid, Tanh, Softmax]	Tanh
Optimizer (tf.keras.optimizers)	[Adam, SGD, RMSProp]	Adam
Learning rate scheduler (tfa.optimizers)	Exponential Decay, Polynomial Decay, Inverse Time Decay, Cosine Decay, Cyclical Learning Rate, None	None
Batch size	[16, 32, 64, 128, 256, 512]	32

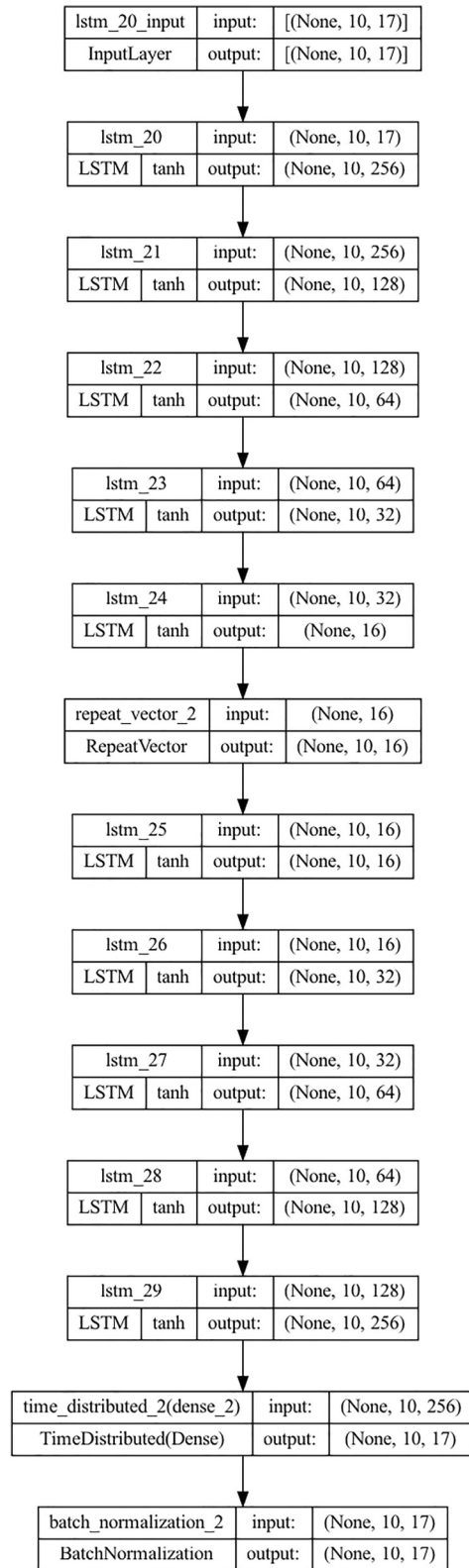


Figure 4. Final Long Short-Term Memory autoencoder architecture.

Table 5 showcases the precise shape details of the inputs prior to feeding them into the LSTM autoencoder. Table 6 provides detailed information on the final model's hyperparameter tuning details and the library used for this.

Figure 4 represents the LSTM autoencoder architecture. The design uses a stacked LSTM structure, where each layer feeds its output into the subsequent layer as its input. Since it is an autoencoder, a mirrored architecture could be observed in the layers.

3.2.1.2. Fully Connected (FC) Autoencoder

Like the LSTM autoencoder, the FC autoencoder incorporates hidden, fully connected layers in both its encoder and decoder parts. The hyperparameter tuning details for the FC autoencoder are detailed in Table 7, which shows the final model setup and the parameters explored. Like the LSTM autoencoder, the FC autoencoder also establishes mirrored architectures, formed with FC layers. Figure 5 represents the FC autoencoder architecture.

3.2.2. Anomaly Detection With Deep Autoencoders

This section delves into the anomaly detection methodology utilizing the deep autoencoder models discussed in the preceding sections.

The choice of the two detection strategies is informed by their proven efficacy in the domain of anomaly detection, as highlighted in Section 2. The first approach, using latent features obtained from a deep autoencoder for clustering models, leverages the ability of autoencoders to reduce dimensionality and extract significant features from the data. This method has shown promise in various applications, including environmental monitoring and anomaly detection, as discussed by authors such as Tian et al. (2014) and Ma (2022). The clustering models used—isolation forest, local outlier factor (LOF), and one-class SVM—are well-established techniques for identifying outliers in high-dimensional data (Breunig et al., 2000; Liu et al., 2012; Schölkopf et al., 1999).

The second approach, using reconstruction errors from a deep autoencoder to identify anomalies, exploits the inherent capability of autoencoders to learn the normal patterns in the data. When the autoencoder reconstructs data points, anomalies typically result in higher reconstruction errors, which can be flagged as outliers. This method has been successfully applied in various fields, including video surveillance and medical diagnostics, as demonstrated by researchers such as D. Cheng et al. (2022) and Esmaeili et al. (2023).

Implementation details of these two primary methods are explained in more detail in the following sections.

3.2.2.1. Anomaly Detection Based on Clustering Through Latent Features

In this case, the encoder segment of the FC autoencoder is employed to extract the latent features from the train and test data sets. These features are compressed from a 28-dimensional space down to an 8-dimensional one. An FC autoencoder is used, pre-trained on the train data (which consists of only non-wildfire cases). Specifically, to obtain latent features, the encoder component of the FC autoencoder is employed. The data dimension for both is reduced from 28 to 8. Then, latent train data is used to fit the cluster model. Three clustering techniques are adopted separately here: isolation forest, LOF, and

Table 7
Hyperparameter Tuning Details for FC Autoencoder

Parameter	Range	Final choice
Number of layers (tf.keras.layers.LSTM)	[2, 3, 4, 5, 6, 7, 8]	5
Number of units (tf.keras.layers.LSTM)	[1024, 512, 256, 128, 64, 32, 16, 8]	[512, 256, 128, 64, 32]
Activation function (tf.keras.layers.LSTM)	[ReLU, Sigmoid, Tanh, Softmax]	ReLU
Optimizer (tf.keras.optimizers)	[Adam, SGD, RMSProp]	Adam
Learning rate scheduler (tfa.optimizers)	Exponential Decay, Polynomial Decay, Inverse Time Decay, Cosine Decay, Cyclical Learning Rate, None	Cyclical Learning Rate
Batch size	[16, 32, 64, 128, 256, 512]	128

one-class SVM. Finally, the prediction of the latent test data is done using the clustering model, and anomalies are identified as a result. Each clustering model has multiple parameters that require tuning.

Each method has unique characteristics that make it suitable for specific aspects of anomaly detection in high-dimensional data. The isolation forest method, which isolates anomalies rather than profiling normal instances, operates by constructing binary trees through random selection of features and split values, creating shorter paths for anomalies due to their rarity in the data set, offering rapid anomaly detection efficiency (Liu et al., 2012). LOF measures the local density deviation of a data point relative to its neighbors, identifying anomalies in areas of varying density (Breunig et al., 2000). Lastly, one-class SVM is designed for scenarios where only the “normal” class is present during training; it defines a decision boundary around this class, with deviations from this boundary treated as anomalies, making it commonly suitable for the preliminary phases of environmental anomaly detection of the unsupervised learning scenarios (Schölkopf et al., 1999). The incorporation of isolation forest, LOF, and one-class SVM in this study is designed to facilitate a comprehensive comparative analysis. By evaluating each method's performance, the study aims to provide essential insights into their comparative strengths and weaknesses, thereby informing future research and practical applications in the field. To achieve this, all three methods are implemented concurrently, allowing for a direct and comprehensive comparison of their results.

Table 8 provides the parameters adjusted for each clustering model and the final choice based on the evaluation results. Only Data set 1 is utilized for this approach.

3.2.2.2. Anomaly Detection Based on Reconstruction Error

In this approach, both FC and LSTM autoencoders are employed independently, though the procedure for each remains the same. The hyperparameter tuning details for both FC and LSTM autoencoders are detailed in Table 9. The selected LSTM model is presented in Table 10. This model uses a batch size of 32, Tanh activation functions for both the encoder, decoder, and Adam optimizer with LSTM layers [256, 128, 64, 32, 16].

Hyperparameter tuning was exclusively conducted for Data set 1. Consequently, the same architecture of the chosen final model was also used when training Data set 2. For the FC case, two models emerge as strong candidates for the final selection. These top-tier models will be referred to as Model A and Model B and are presented in table 11. Having the highest F1 scores, they indicate a balance between precision and recall and a decent overall classifier performance. It is decided to assess both and compare their results, before finalizing the choice. Each model is trained separately on both Data set 1 and Data set 2 (on the test data).

3.2.2.3. Training Process

The training data set only consists of non-wildfire instances, while both the test and validation data sets contain a mix of cases. After obtaining the best combination of parameters for the model and finalizing the autoencoder model, training is performed. During the training, validation data is used to observe the validation loss and errors for each iteration of the training process.

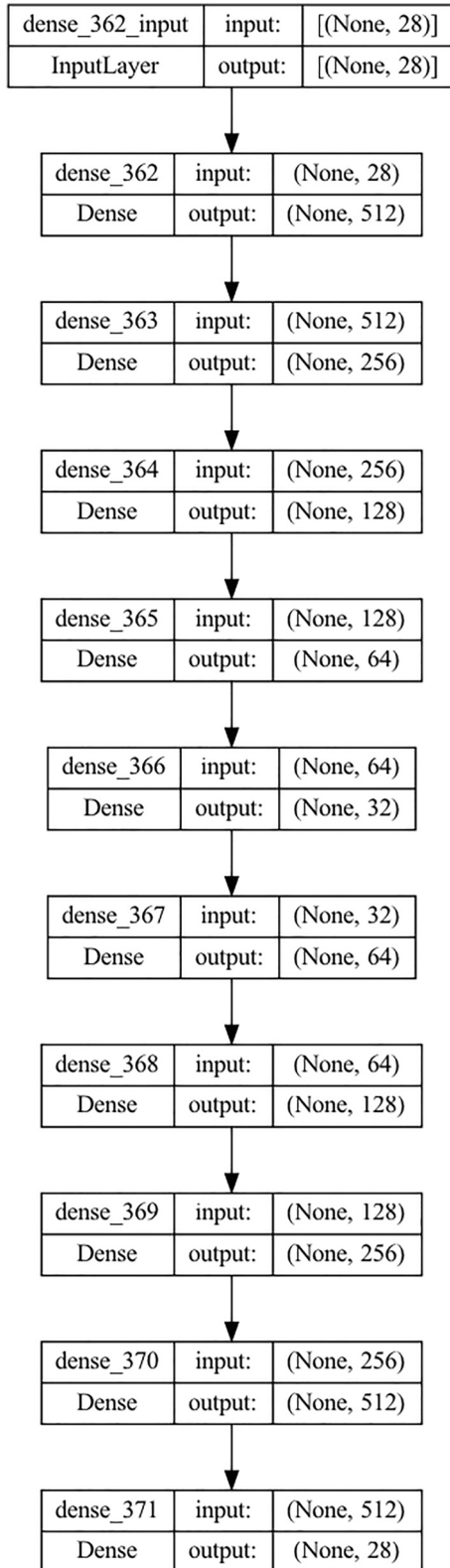


Figure 5. Final FC autoencoder architecture.

Initially, the train data is input into the FC/LSTM autoencoder, which has been previously trained on this data. The autoencoder then generates the reconstructed train data as its output. The reconstruction error, represented as the mean squared logarithmic error, quantifies the deviation between the original and the reconstructed train data. Therefore, subsequently, MSLETrain values are calculated to calculate the difference between the original train data and reconstructed train data. Using the MSLETrain values, a threshold is set up based on the formula provided below, here termed as ThresholdTrain.

$$\text{Threshold}_{\text{Train}} = \mu_{\text{MSLE}(\text{Train})} + 2\sigma_{\text{MSLE}(\text{Train})} \quad (2)$$

For the procedure for test data reconstruction and anomaly detection, similarly, the test data is input into the same FC/LSTM autoencoder. The autoencoder then generates the reconstructed test data as its output. Subsequently, MSLETest values are calculated to calculate the difference between the original test data and reconstructed test data. Finally, in the classification phase, MSLETest values are compared with ThresholdTrain to assign anomalies. Basically, if MSLETest is higher than ThresholdTrain, then the data point is categorized as a wildfire, denoted by “1.” If not, it is categorized as a non-wildfire, denoted by “0,” with the following formula:

$$\text{Fire}_{\text{Test}} = \begin{cases} 1, & \text{Threshold}_{\text{Train}} \leq \text{MSLE}_{\text{Test}} \\ 0, & \text{Threshold}_{\text{Train}} > \text{MSLE}_{\text{Test}} \end{cases} \quad (3)$$

3.2.2.4. Performance Metrics

For both approaches, after getting the anomaly detection/classification results based on test data, the performance of the model on unseen data needs to be evaluated. For this, the model's performance is evaluated against five metrics: Accuracy, precision, recall, F1-score, and Matthew's coefficient (MCC). The following equations present formulas for calculating these metrics, where TP = True Positive, TN = True Negative, FP = False Positive, and FN = False Negative:

$$\text{Accuracy} = \frac{TP + TN}{P + N} \quad (4)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (5)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (6)$$

$$\text{F1} = 2 \frac{(\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})} \quad (7)$$

$$\begin{aligned} &\text{Matthew's coefficient (MCC)} \\ &= \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \end{aligned} \quad (8)$$

A confusion matrix for each model (Demir, 2022), based on test data, was also produced. Lastly, a ROC curve plot and AUC are plotted and calculated. A ROC curve plots the TP rate in comparison to the FP rate. These values shift

Table 8
Hyperparameter Tuning Details for Clustering Models

	Parameter	Range	Final choice
Isolation forest	Number of estimators	[50, 100, 200]	200
	Max samples	[0.5, 0.7, 0.9]	0.9
	Contamination	["auto", 0.01, 0.02, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5]	0.5
LOF	Number of neighbors	[2, 3, 4, 5, 6, 7, 8, 10, 12, 15, 20]	8
	Contamination	["auto", 0.01, 0.02, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5]	0.5
	Distance metric	[Euclidean, Manhattan, Minkowski]	Manhattan
One class SVM	Kernel	[Linear, Polynomial, RBF, Sigmoid]	Linear
	Nu value	[0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6]	0.6

relative to one another. Essentially, when the TP rate peaks, the FP rate is at its minimum, and vice versa (Maxion & Roberts, 2004). AUC is a key metric for assessing a binary classifier's performance. It ranges from 0.5 (equivalent to random guessing) to 1.0 (a flawless classifier). AUC evaluates the effectiveness of score classifiers by considering all potential classification thresholds (Melo, 2013).

4. Results

4.1. Anomaly Detection Based on Clustering Through Latent Features

In this section, two different models and their results will be presented for each cluster model type (Isolation Forest, LOF, and one-class SVM). While the first one is the best model according to the hyperparameter tuning, the other one will have different parameters for the sake of the comparison of the parameters and performance.

4.1.1. Isolation Forest

Based on Table 12, Model (A) assumes half of the data is anomalous (0.5), while Model (B) only assumes 0.2 of the data is anomalous due to the contamination parameter. This discrepancy leads to a significant difference in their performance.

Figure 6 illustrates Model (A) has a better balance between precision and recall, resulting in a higher F1-score of 67%. In contrast, Model (B) has a higher precision of 71.7%, suggesting that among the anomalies it detected, a greater percentage were true anomalies. However, its recall is considerably low at 33.7%, which means it failed to capture a significant proportion (around 66%) of the true anomalies present in the data set.

4.1.2. LOF

In Table 13 and Figure 7, it is seen that Model (C) outperformed in recall (76.6% vs. 60%) and F1-score (71.9% vs. 65.5%), indicating its superiority in detecting real anomalies. In contrast, while Model (D) has a higher precision of 72%, it struggles to identify all real anomalies, as revealed by its low recall of 60%.

While drawing definitive conclusions on the influence of the number of neighbors from just two models is challenging, there is a trend: models with better accuracies consistently have a higher number of neighbors in

Table 9
FC and Long Short-Term Memory Autoencoder Final Hyperparameter Tuning Range

Parameter & library	Range
Activation function (tf.keras.layers.LSTM)	[ReLU, Sigmoid, Tanh, Softmax]
Optimizer (tf.keras.optimizers)	[Adam, SGD, RMSProp]
Learning rate scheduler (tfa.optimizers)	Exponential Decay, Polynomial Decay, Inverse Time Decay, Cosine Decay, Cyclical Learning Rate, None
Batch size	[16, 32, 64, 128, 256, 512]

Table 10
Long Short-Term Memory Autoencoder Model Selected After Hyperparameter Tuning on the Test Data

Batch size	Encoder activation function	Decoder activation function	Optimizer
32	Tanh	Tanh	Adam

LOF. Similarly with isolation forest, the contamination parameter significantly drops LOF's performance. It typically reflects the proportion of outliers in the data set and was highly influential for both isolation forest and LOF models.

4.1.3. One-Class SVM

Both models, (E) and (F), employ the RBF kernel, allowing us to compare the performance metrics due to the variations in the Nu parameter Table 14. The difference in the Nu values significantly impacts the models' performances. Model (E), with a Nu value of 0.6, exhibits better overall performance across all metrics than Model (F), which has a Nu value of 0.3. Specifically, Model (F)'s performance lags, especially in terms of recall (32.2%), which is considerably lower than Model (E)'s recall (66.2%) as in Figure 8.

4.1.4. Visualization of the Results

Figure 9 represents the anomaly detection results of all three techniques on 3D maps with PCA features. PCA features are created here just for visualization purposes. The actual clustering, fitting, and prediction processes utilize the original latent data generated through an autoencoder, detailed in Section 3.2.2.1. Latent test data, which is 8-dimensional, is reduced into 3-dimensional data using PCA so that it is possible to visualize the whole data in a 3D dimension plot instead of randomly choosing three variables.

Upon reviewing Figure 9, it is clear that anomaly and normal data points do not lie in the different spaces, which makes the clustering approach challenging. Furthermore, the significant reliance of these models on specific parameters like contamination and Nu is evident, as indicated in previous sections. While these approaches showed initial promise, they might not be the most suitable choice for imbalanced data sets like this one.

4.2. Anomaly Detection Based on Reconstruction Error

4.2.1. LSTM Autoencoder

The model definition used for the training was presented in Table 10. Both data sets training underwent training for 200 epochs, incorporating an early stopping mechanism to mitigate overfitting. Figure 10 displays the training and validation loss plots of the LSTM autoencoder for both Data set 1 and Data set 2.

For Data set 1, although the initial epochs show fluctuations, the model seems to stabilize and converge without any evident signs of overfitting or underfitting. Both the training and validation losses exhibit a consistent declining trend across the 200 epochs in Figure 10. Early stopping is not triggered, suggesting that extending the training might further reduce validation loss. For Data set 2, the loss plots appear less consistent. Around the 125th epoch, there is a sudden spike in loss followed by a rapid decline. Even as training continues to the 175th epoch, the model does not attain much additional learning. Table 15 presents the obtained performance metrics of models on the test data for Data set 1 and Data set 2.

From Table 15 and Figure 11, it is observed that the model trained on Data set 1 demonstrates better performance in Accuracy, Precision, Recall, and F1-score compared to the model trained on Data set 2, which retains only 17 variables. However, an important point for both models is that MCC is very low. While the first model's MCC is closer to zero, suggesting a near-random prediction, the second model's MCC is significantly lower, indicating a

Table 11
FC Autoencoder Models Selected After Hyperparameter Tuning on the Test Data

Model	Activation function	Batch size	Optimizer	Learning rate scheduler
Model A	ReLU	128	Adam	Cyclical Learning Rate
Model B	ReLU	32	RMSProp	None

Table 12
Parameter Information and Results for Two Models of Isolation Forest

Model	Number of estimators	Max samples	Contamination	Accuracy	Precision	Recall	F1-score	MCC
Model (A)	100	0.9	0.5	0.612	0.664	0.675	0.67	0.242
Model (B)	100	0.9	0.2	0.536	0.717	0.337	0.459	0.264

slightly worse prediction than random guessing. Analyzing the ROC curve plots in Figure 12, the second model exhibits a better AUC value of 0.61 than the first model with 0.51. However, one could say that this type of ROC curve plot is not desirable.

In conclusion, while the first model, trained on Data set 1, outperforms traditional performance metrics, the second model's ROC curve seems better. Nevertheless, the overall quality of both models is not very promising, given that an AUC close to 0.5 implies nearly random prediction. However, one can note that this reflects the inherent difficulty in predicting wildfires with limited historical data.

4.2.2. FC Autoencoder

The models A and B trained in this scenario were presented in Table 11. Each model's training process spanned 400 epochs, with an early stopping (patience criteria as 20) integrated to avoid potential overfitting.

Based on Figure 13, the FC Autoencoder Model A demonstrates steady convergence in its training and validation loss plots for both Data set 1 and Data set 2. This smooth progression suggests effective learning without erratic fluctuations, which is favorable. However, the early stopping could have been triggered earlier. This might indicate that the early stopping criteria, the patience parameter, which is 20, in this case, were set too early, which allowed the model to train longer than necessary without significant performance gains.

In Figure 14, the FC Autoencoder Model B shows a less consistent convergence than Model A across both data sets. Notable fluctuations in loss are observed, with early stopping triggered around the 120th epoch for Data set 1 and the 100th epoch for Data set 2. Additionally, a significant gap between training and validation losses in Data set 1 suggests potential overfitting, contrasting with the performance observed in Model B.

Table 16 represents the obtained performance metrics of Models A and B on the test data of Data set 1 and Data set 2. Upon analyzing the performance metrics from Table 16 and confusion matrices in Figures 15 and 16, several insights emerge regarding the efficacy of Models A and B across Data sets 1 and 2. For Data set 1, both models show relatively similar performance metrics. However, a key distinction arises in the recall, where Model B has a noticeable advantage with 71.4% while Model A's recall is 68%. On the contrary, the MCC value of Model A is slightly higher than Model B, which may be due to its use of the Cyclical Learning Rate scheduler, enabling more refined learning rate adjustments. For Data set 2, although both models showcase very close precision, Model B compensates again with a better recall, offering 60.7% against Model A's 55.5%. Model B also marginally outperforms Model A in accuracy and MCC.

Figures 17 and 18 display the ROC curve plots for Models A and B across both Data set 1 and Data set 2 (using test data). All these plots fall within the 77%–78% range, making it difficult to draw meaningful comparisons between the models and data sets. Nonetheless, the near-80% performance indicates that the models consistently show quite promising results across both data sets.

Analyzing the influence of data sets on model performance reveals some significant insights. Both models demonstrate strong performance metrics even when applied to Data set 2, which has been reduced to 17 variables. There is only a slight decrease in metrics when the models transition from Data set 1 to Data set 2. An interesting

Table 13
Parameter Information and Results for Two Models of Local Outlier Factor

Model	Number of neighbors	Contamination	Distance metric	Accuracy	Precision	Recall	F1-score	MCC
Model (C)	20	0.5	Manhattan	0.65	0.677	0.766	0.719	0.329
Model (D)	12	0.3	Manhattan	0.631	0.72	0.6	0.655	0.361

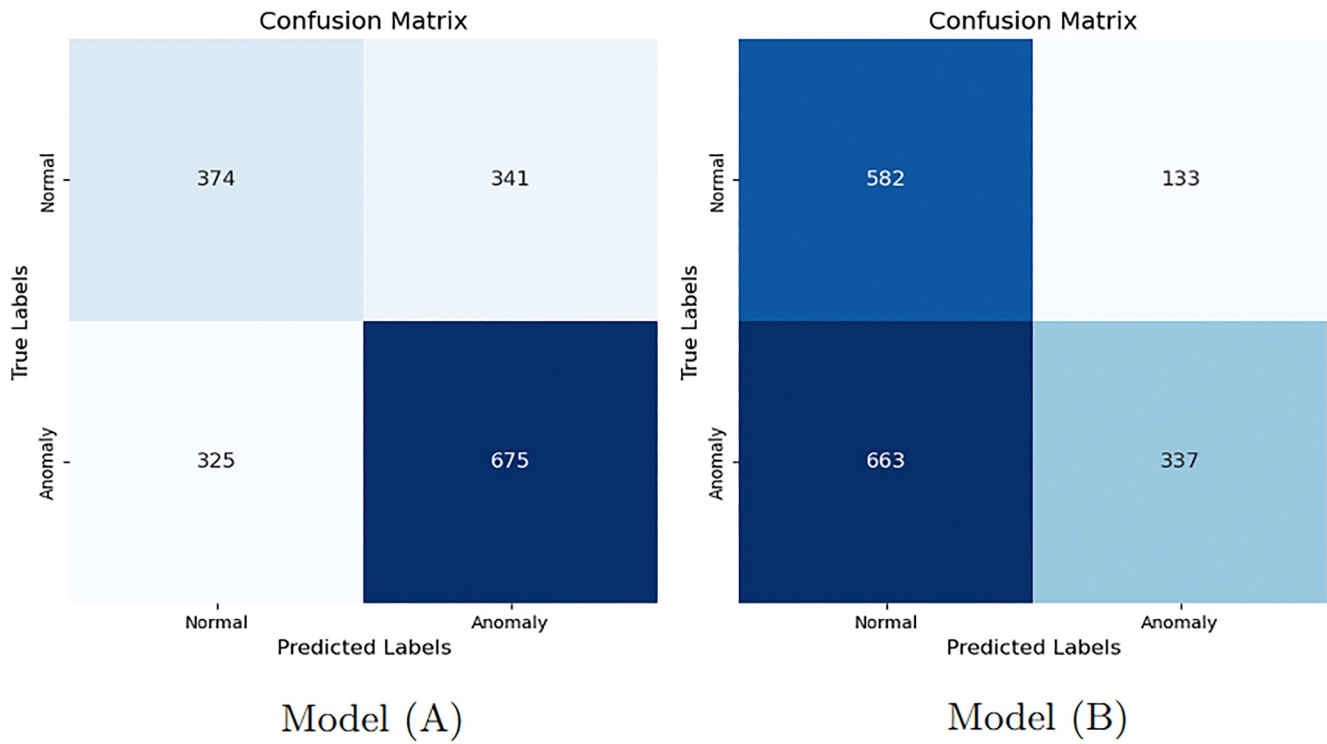


Figure 6. Confusion matrix comparison for Model (A) and (B) for isolation forest.

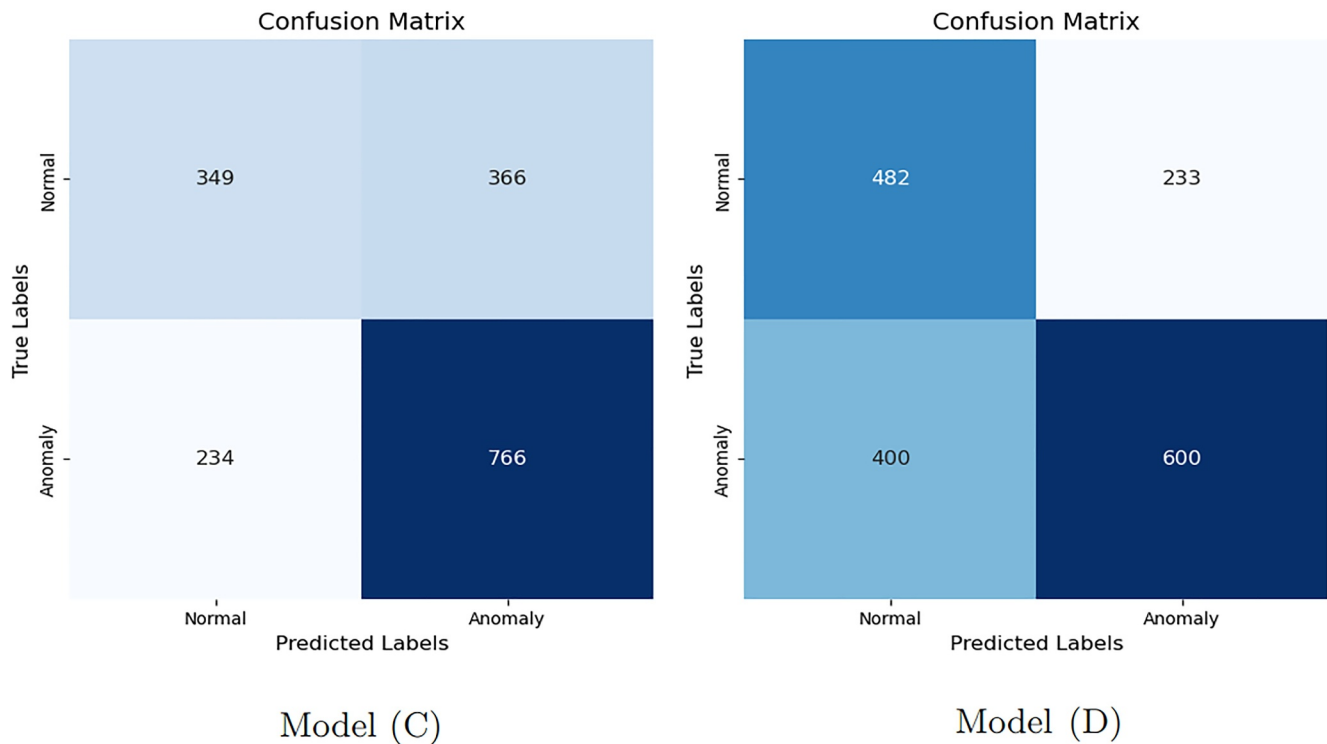


Figure 7. Confusion matrix comparison for Model (C) and (D) for local outlier factor.

Table 14
Parameter Information and Results for Two Models of One-Class Support Vector Machines

Model	Kernel	Nu value	Accuracy	Precision	Recall	F1-score	MCC
Model (E)	RBF	0.6	0.554	0.608	0.662	0.634	0.074
Model (F)	RBF	0.3	0.482	0.604	0.322	0.42	0.032

observation is that Model B's accuracy on Data set 2 drops slightly from 71.1% to 69.5%, yet its AUC remains consistent at 78%. This consistency is particularly impressive given that the models' hyperparameters were specifically tuned for Data set 1. Naturally, one might expect better results on Data set 1 because of this tuning. However, the minor decline in performance on Data set 2 suggests that this compact data set effectively integrates the most influential variables.

In a direct comparison of results, the standout models still come from Data set 1. Model A outperforms precision and MCC, whereas Model B is better in recall and F1-score. Given the importance of accurately detecting wildfires, a focus on recall becomes crucial. Hence, in such a scenario, Model B, with its superior recall, may be the more advantageous choice.

5. Discussion

In this research, two different unsupervised learning approaches to detect wildfires in Australia with deep autoencoders have been implemented successfully.

The first approach was anomaly detection based on clustering through latent features using deep autoencoders. Here, first an FC autoencoder is trained and applied to reduce the data dimension from 28 to 8. Then, this latent data is fed into three different clustering methods (isolation forest, LOF, and one-class SVM) to spot outliers. This approach presented several challenges. First, the performance metrics of all three methods were closely dependent

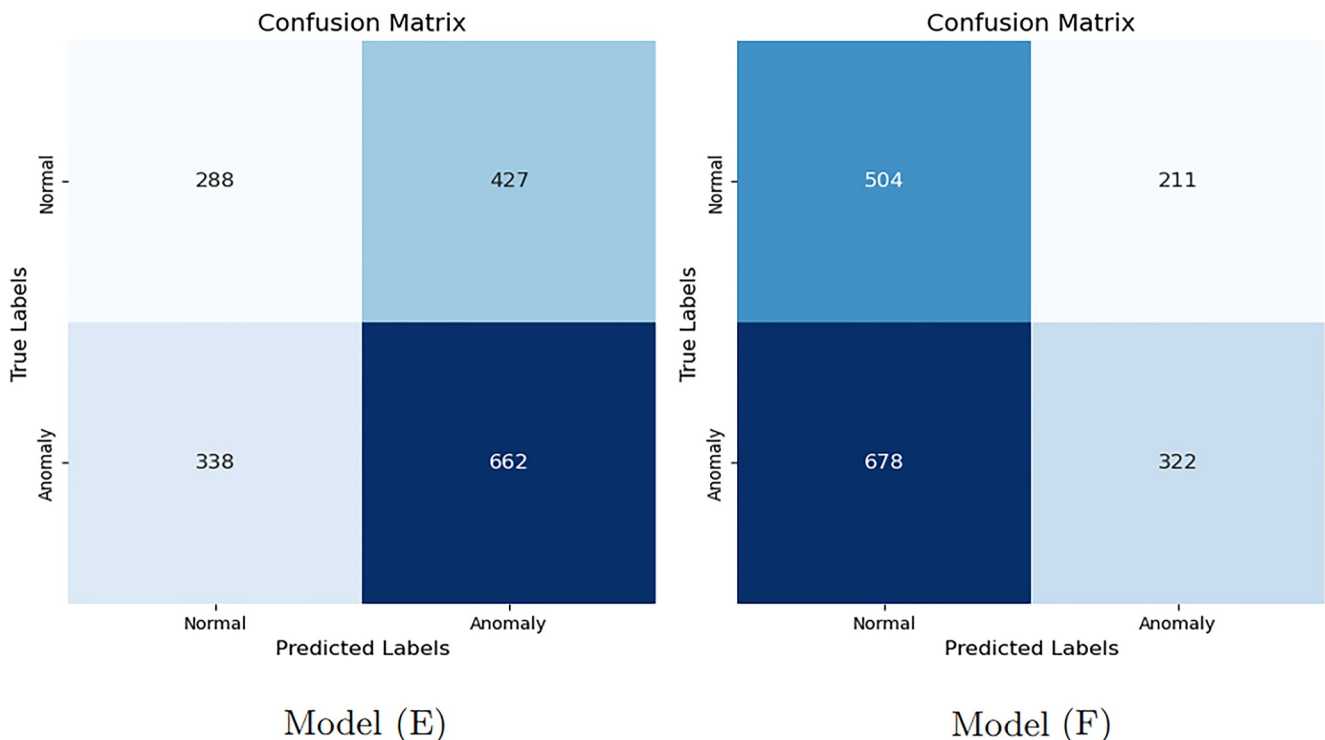


Figure 8. Confusion matrix comparison for Model (E) and (F) for one-class support vector machines.

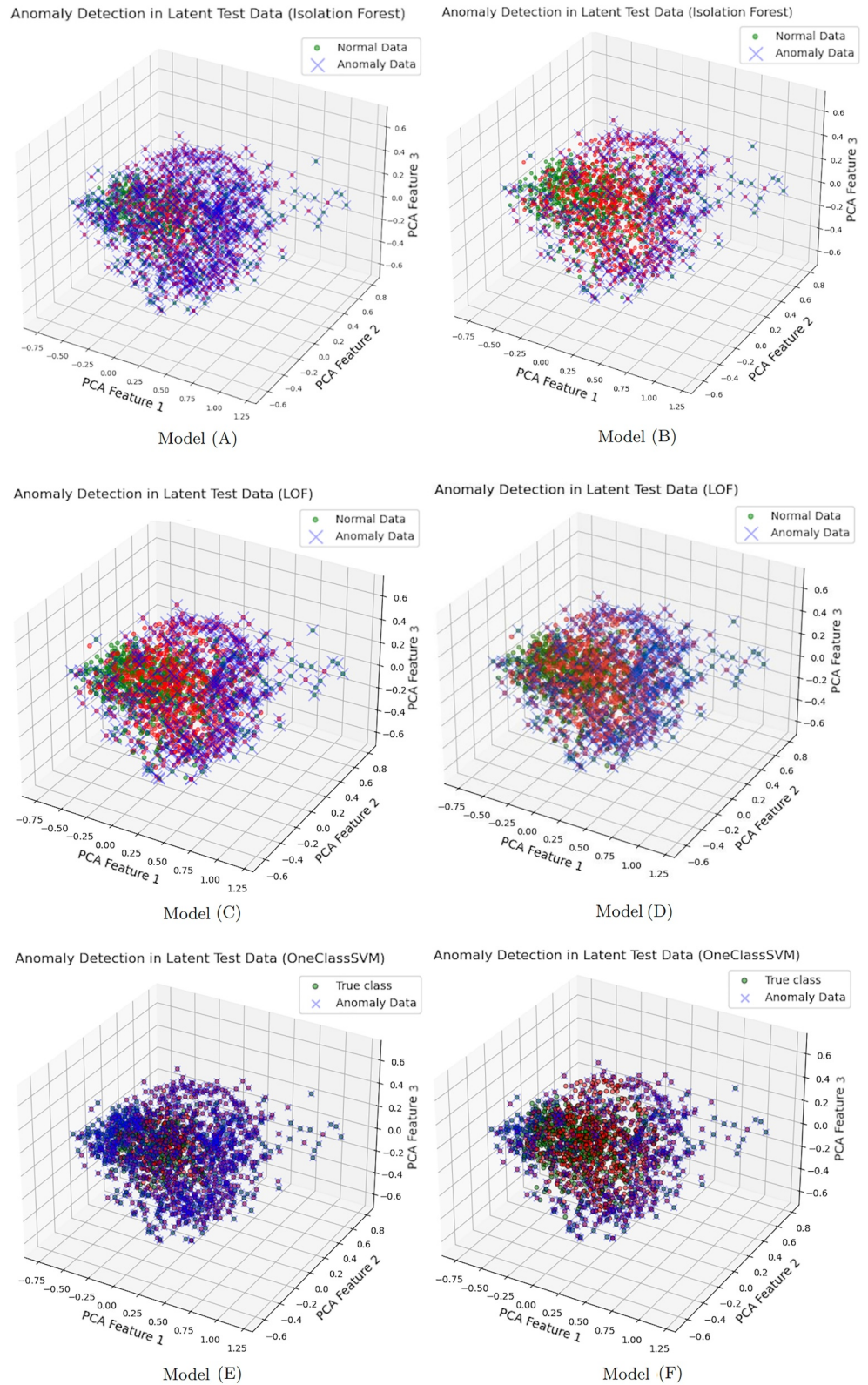


Figure 9. Anomaly detection results of isolation forest on 3D maps (PCA is used for only visualization purposes) for Model (A) and (B) (top row), (C) and (D) (middle row), and (E) and (F) (bottom row).

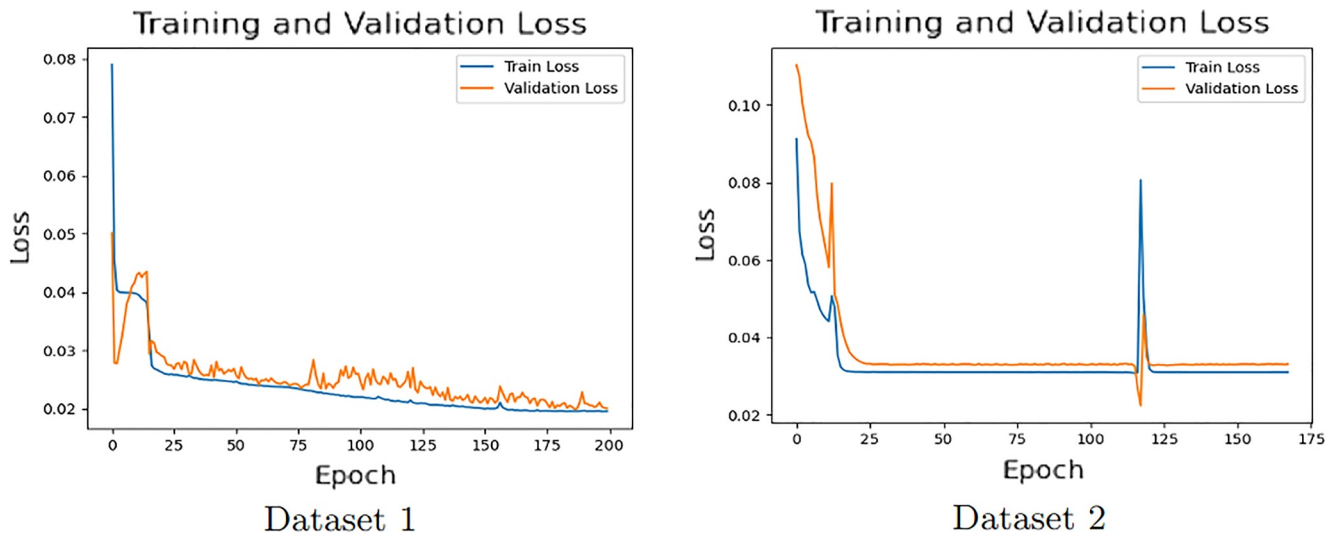


Figure 10. Long Short-Term Memory Autoencoder training and validation loss plots. Data set 1 on the left, Data set 2 on the right.

on specific parameters, especially the contamination and Nu values. For instance, the recall value for the isolation forest model dropped sharply from 67.5% to 33.7% when contamination was decreased from 0.5 to 0.2. Adjusting contamination and Nu values, while helpful, may only serve as a specific short-term solution. In addition, considering that these techniques are implemented in an “unsupervised learning” way, it is important to remember that a real-life scenario will not have ground truth labeled (wildfire or non-wildfire) data available for parameter adjustment. Another challenge related to how the anomaly data spread out. Thanks to 3D map visualizations done using PCA components, it was observed that anomaly data outliers did not cluster separately from the usual normal data points. Consequently, due to these challenges and results, clustering techniques alone were not sufficient, robust, or efficient in distinguishing anomalies in this wildfire data set. This realization triggered the exploration of alternative strategies, guiding us to the second approach: anomaly detection using deep autoencoders, primarily based on reconstruction error.

The second approach involved using deep autoencoders to detect anomalies by using the reconstruction error. The approach involved training the deep autoencoder on normal data points and then using this trained model on the reconstruction of test data. Anomalies have higher reconstruction errors while normal data points have low errors. Two different deep autoencoder type were implemented: LSTM and FC.

In the LSTM autoencoder case, a final model architecture was determined after hyperparameter tuning and was applied to both Data set 1 and Data set 2. The model trained on Data set 1 showcased better performance metrics with an accuracy of 56.3%, whereas the model trained on Data set 2 managed only 45.7%. A consistent issue with both models was their tendency to misclassify non-wildfire cases, often labeling them as wildfires. Ultimately, the performance of both models is not sufficient, as their AUC values close to 0.5 suggest nearly random predictions. Our initial assumption was that utilizing LSTM layers would be advantageous given the time-series nature of the data, hoping to capture temporal information. However, this assumption did not hold true in practice. Potential reasons include the short sequence span of 10 days, which might not provide enough temporal context. Additionally, the daily aggregation of the data could have lost valuable temporal patterns that LSTM layers might have

Table 15

Two Long Short-Term Memory Autoencoder Models Based on Data Set 1 and Data Set 2, on Test Data

Model	Accuracy	Precision	Recall	F1-score	MCC
LSTM Autoencoder (Data set 1)	0.563	0.581	0.897	0.705	−0.013
LSTM Autoencoder (Data set 2)	0.457	0.526	0.692	0.598	−0.20

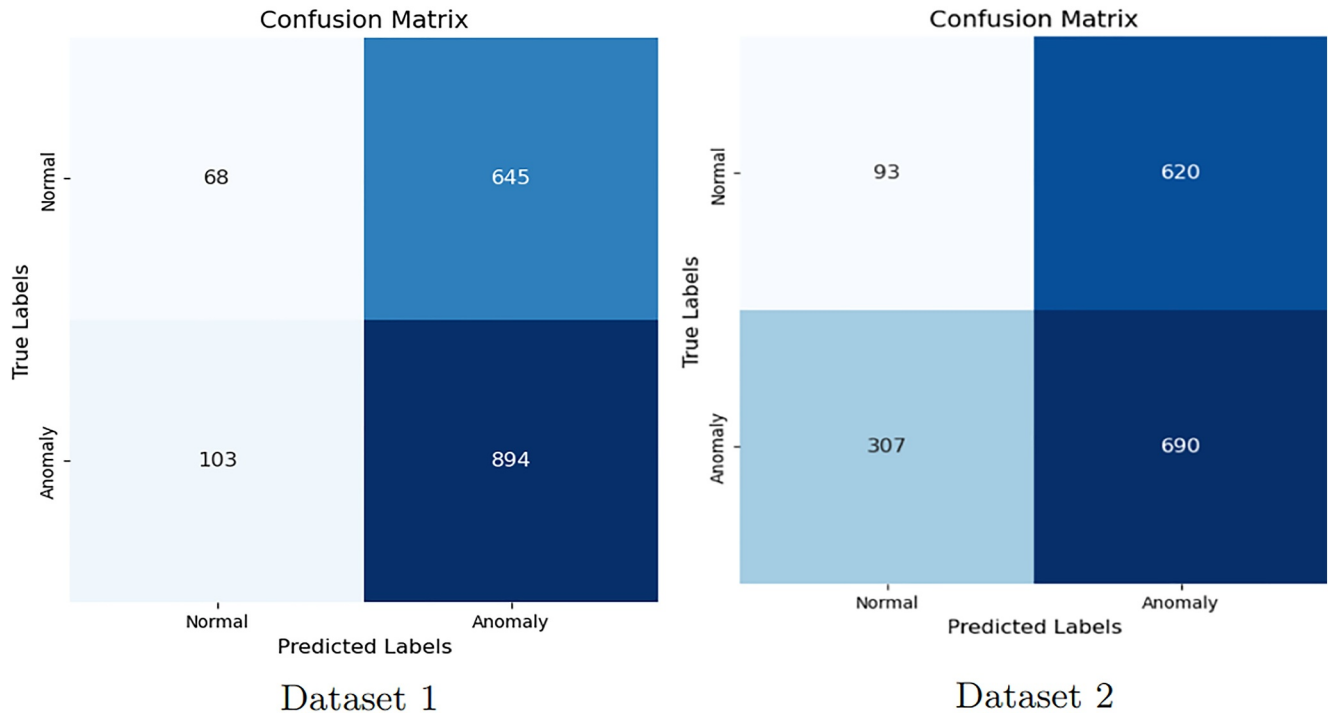


Figure 11. Confusion matrices of Long Short-Term Memory autoencoder for Data set 1 and Data set 2.

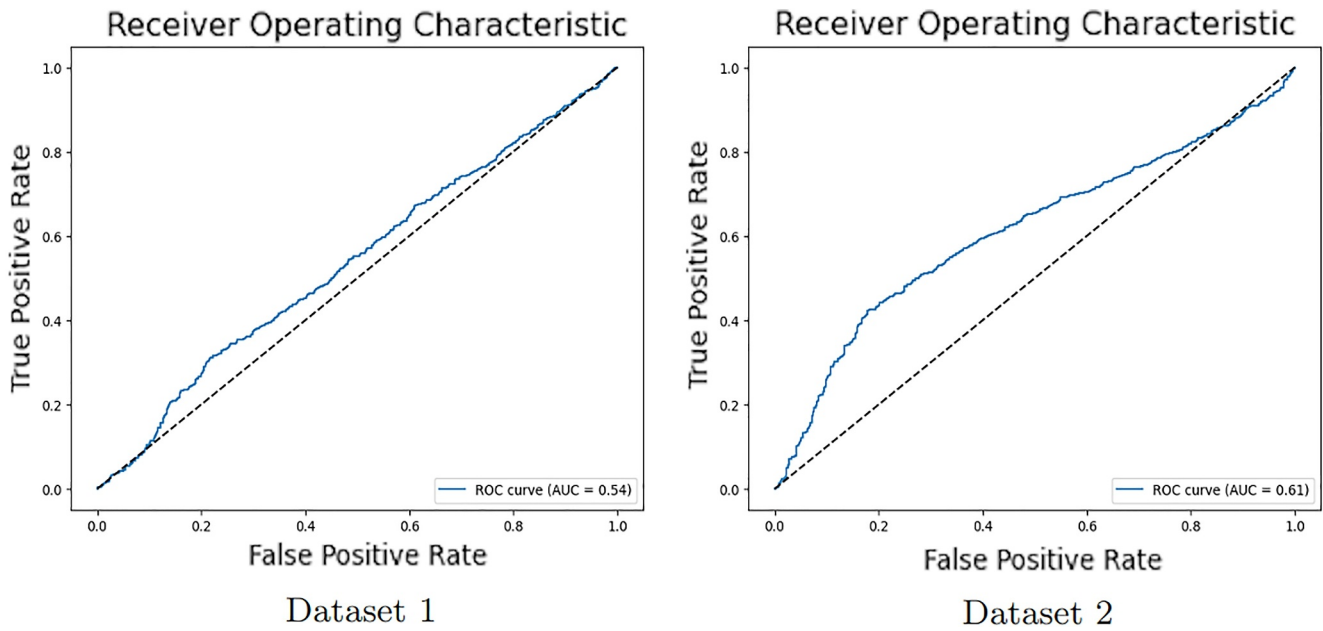


Figure 12. ROC curve plots of Long Short-Term Memory autoencoder Data set 1 and Data set 2.

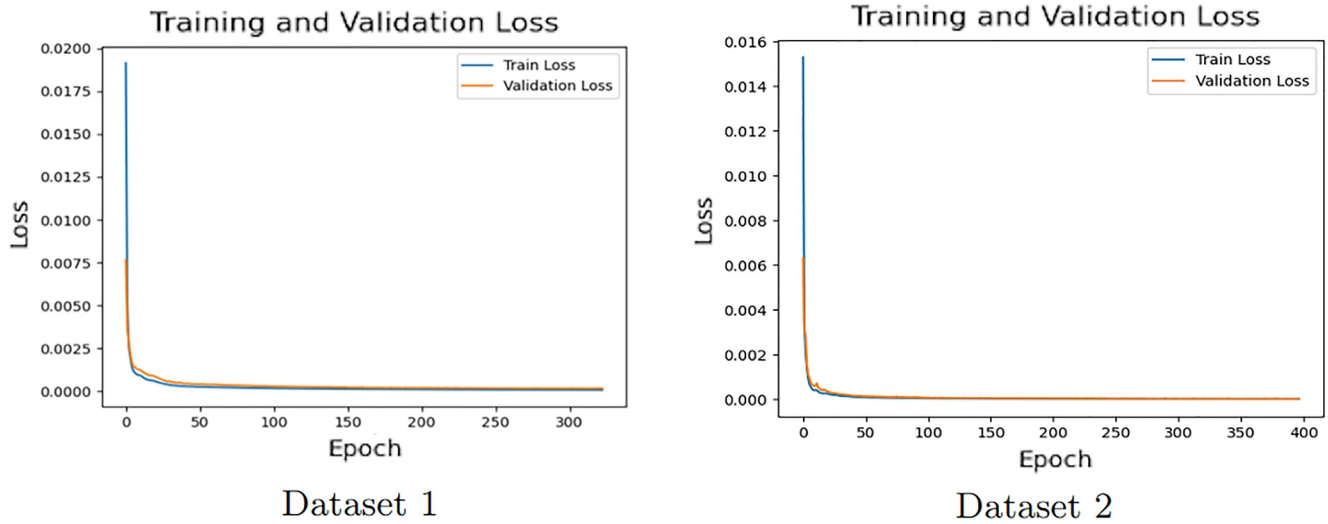


Figure 13. FC Autoencoder (Model A) training and validation loss plots. Data set 1 on the left, Data set 2 on the right.

captured. Lastly, one can note that low values of accuracy are expected given the challenging nature of the data and the lack of historical ground truth. Eventually, the goal of this study is not to filter out methods with poorer performance but to provide a comprehensive comparison.

The final modeling phase employed an FC autoencoder. Two promising models obtained from hyperparameter tuning were utilized on both Data set 1 and Data set 2. The first model leveraged the Adam optimizer with a learning rate scheduler, while the second relied on RMSProp without any scheduler. Despite the compact 17-feature dimension of Data set 2, the models maintained strong metrics on this data set as well. However, Data set 1's models did perform better eventually. Most impressively, the top model - using RMSProp without any scheduler - recorded an accuracy of 71.1%, an F1-score of 74.2%, and an MCC of 0.417. These figures are significantly higher than what is achieved with earlier methods, highlighting the effectiveness of this approach and the use of FC layers in deep autoencoders.

6. Conclusion

In environmental science and disaster prevention, detection, and prediction of wildfires in time, especially in vulnerable regions such as Australia, remains a crucial concern. This research successfully employed several

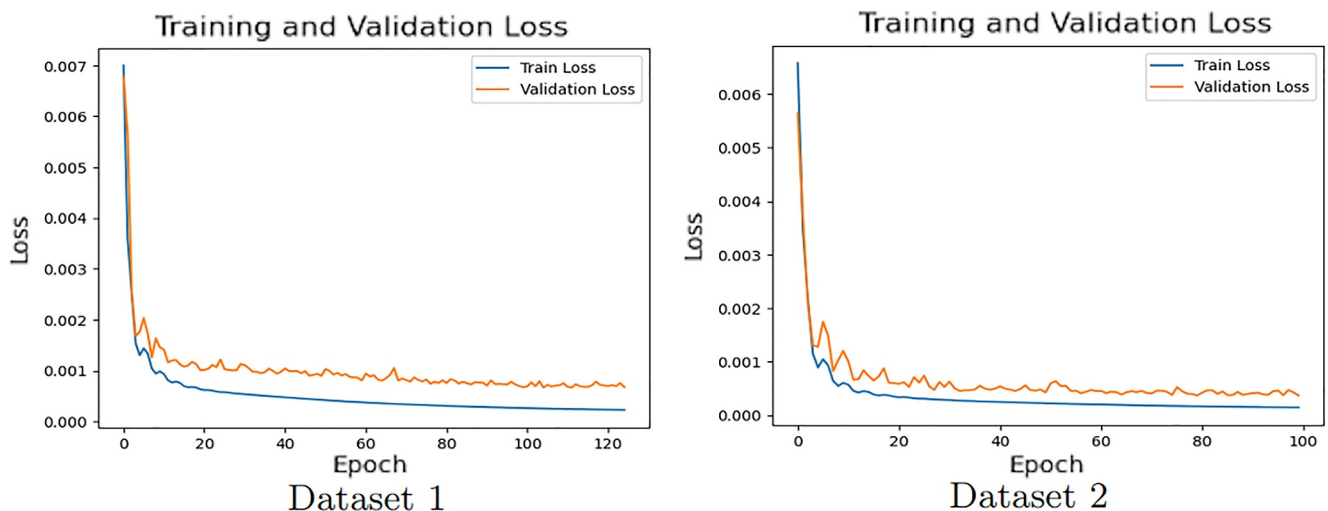


Figure 14. FC Autoencoder (Model B) training and validation loss plots. Data set 1 on the left, Data set 2 on the right.

Table 16
Model A and B Performance Metrics on Both Test Data of Data Set 1 and Data Set 2

Model	Accuracy	Precision	Recall	F1-score	MCC
Model A (Data set 1)	0.712	0.797	0.68	0.733	0.432
Model A (Data set 2)	0.678	0.838	0.555	0.667	0.41
Model B (Data set 1)	0.711	0.773	0.714	0.742	0.417
Model B (Data set 2)	0.695	0.823	0.607	0.698	0.423

unsupervised learning techniques, in the context of anomaly detection with deep autoencoders, to predict wildfires in Australia. This approach fills a gap in the field of wildfire prediction.

Our research explored two primary methods. The first method focused on anomaly detection using latent features extracted from a deep autoencoder and demonstrated initial promising results. Utilizing the FC autoencoder enabled significant dimensionality reduction, and when paired with clustering techniques like isolation forest, LOF, and one-class SVM, yielded some encouraging results. However, while these methods showed potential, they were somewhat sensitive to specific parameters, which could affect their

reliability in real-life scenarios without labels. Furthermore, the fact that anomaly data is not scattered as a different cluster, challenged these techniques even more.

The second method, using anomaly detection through reconstruction error, presented a clearer potential. The LSTM autoencoder, despite its theoretical applicability to time-series data, failed to yield the expected outcomes. Yet, the FC autoencoder showcased impressive results. The final FC autoencoder outperformed its counterparts, achieving an accuracy of 71.1%, an F1-score of 74.2%, and an MCC of 0.417 on Data set 1.

This work proposed wildfire prediction by employing unsupervised learning approaches without the need for ground truth data of historical wildfires. Our findings demonstrate the feasibility of these methodologies in complex real-life scenarios linked to wildfire detection and filling a significant gap in the literature. The use of deep autoencoders for wildfire prediction shows the effectiveness of the method. Given the critical role of forests in the global ecosystem and the pressing need for efficient wildfire detection, this study aims to present a meaningful and practical solution in wildfire prediction, reinforcing the critical bridge between nature's preservation and technological innovation using AI.

Several recommendations could be given for future work for researchers working in this area. To begin, while the LSTM autoencoder used in this research was limited to 10-day sequences due to computational limitations, longer sequences, such as weekly or monthly intervals, may improve its performance. Second, given the data set's temporal nature, Bi-LSTM autoencoders, which can capture bidirectional temporal information, might be

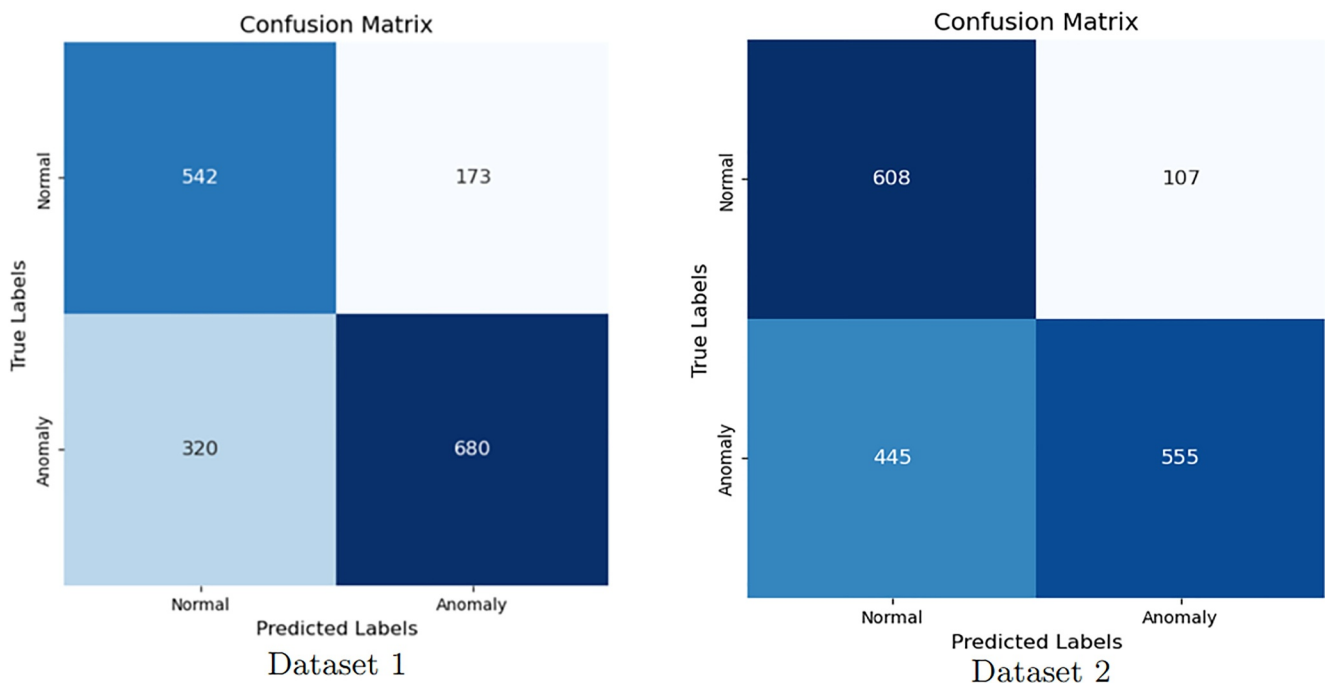


Figure 15. Confusion matrices of FC Autoencoder (Model A) on the test data. Data set 1 on the left, Data set 2 on the right.

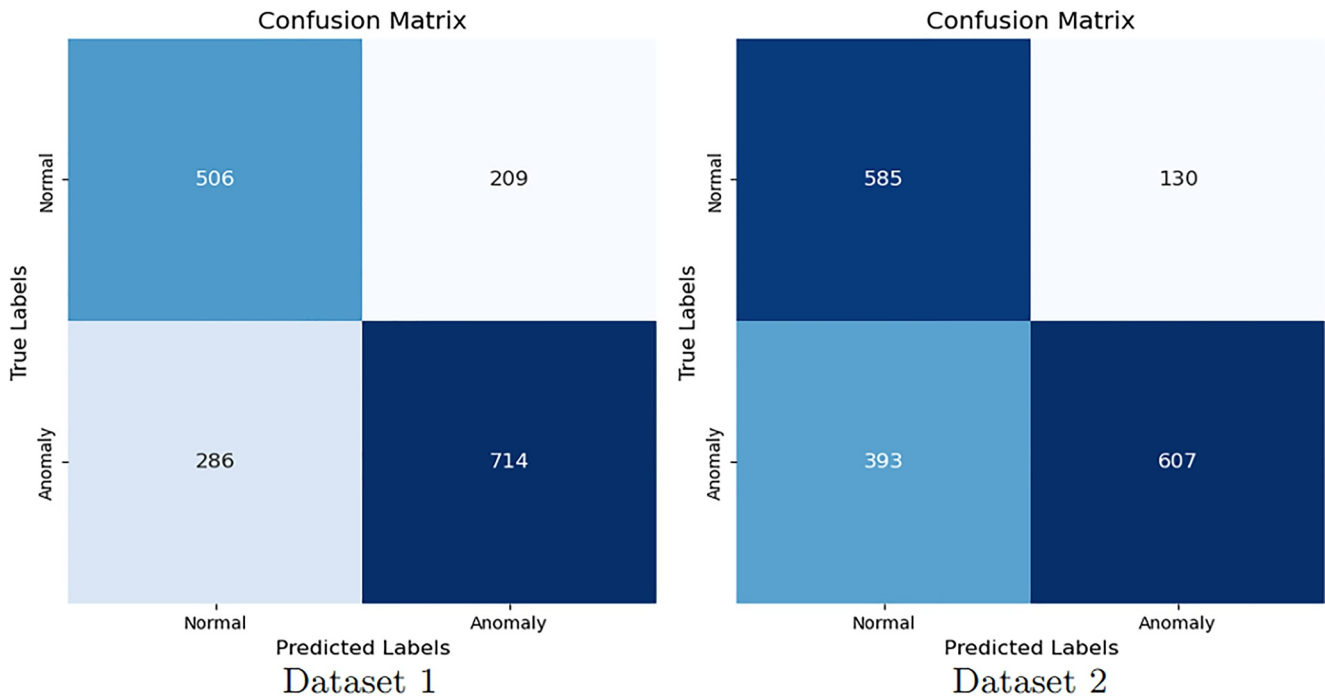


Figure 16. Confusion matrices of FC Autoencoder (Model B) on the test data. Data set 1 on the left, Data set 2 on the right.

explored as an alternative to the traditional LSTM. Furthermore, there is an opportunity for more extensive hyperparameter tuning to discover possible alternative optimum model configurations. Modification of the decision rule can provide a probability of wildfire instead of a detection outcome. Finally, given the encouraging improvements in the autoencoder area, experimenting with Variational Autoencoders (VAEs) might be a promising next step. These proposals can help to refine and improve the predictive ability of the models in the

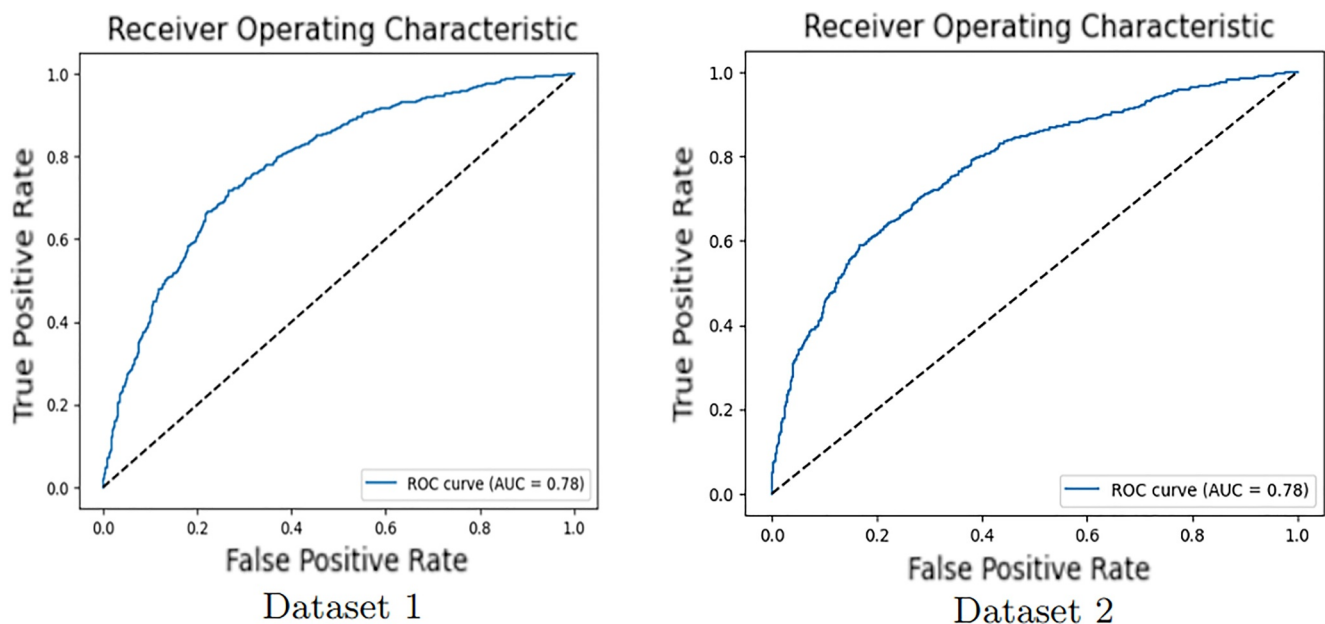


Figure 17. ROC curve plots of FC Autoencoder (Model A). Data set 1 on the left, Data set 2 on the right.

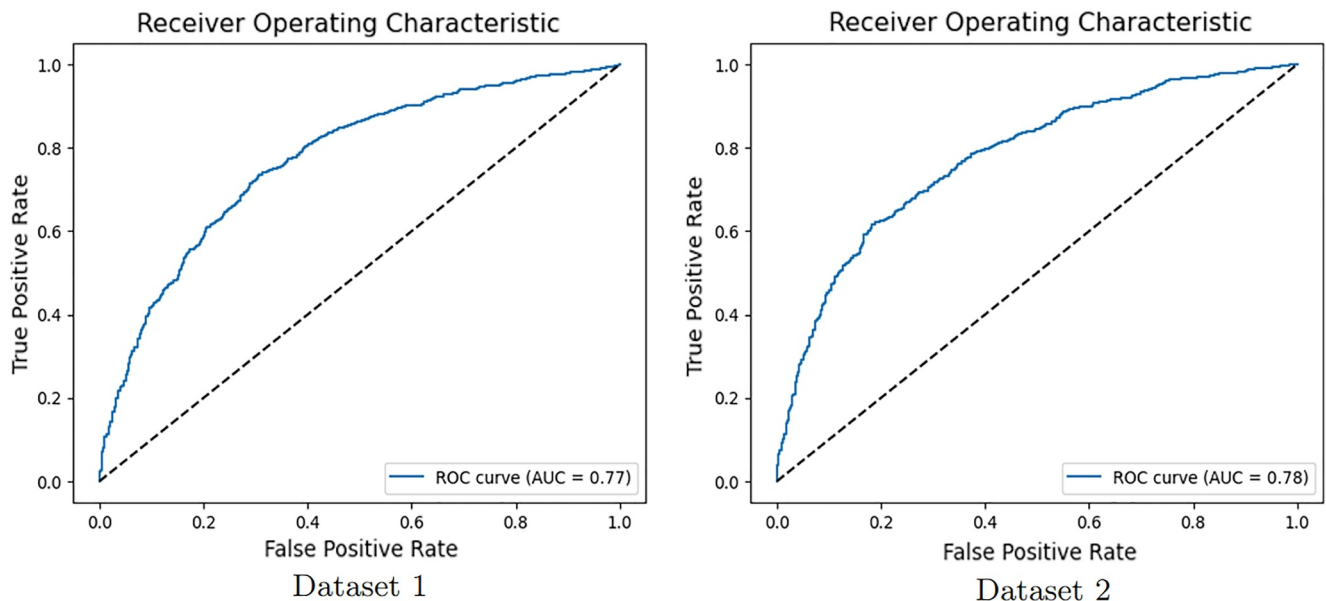


Figure 18. ROC curve plots of FC Autoencoder (Model B). Data set 1 on the left, Data set 2 on the right.

wildfire prediction domain using unsupervised ML and extend the applicability of the wildfire prediction methods to a global scale.

Data Availability Statement

The main data source is the data set released by IBM for the Call for Code Spot Challenge for Wildfires competition (IBM, 2020). It is available at <https://github.com/Call-for-Code/Spot-Challenge-Wildfires>. The specific elements from the previous data set used in this article have been processed from the following original sources: ERA5-Land hourly data from 1950 to present data set (Muñoz Sabater, 2019) available at the Copernicus Climate Change Service (C3S) Climate Data Store <https://cds.climate.copernicus.eu/datasets/reanalysis-era5-land?tab=download>; MOD13Q1 MODIS/Terra Vegetation Indices 16-Day L3 Global 250 m SIN Grid V006 (Didan, 2015) available at the NASA EOSDIS Land Processes Distributed Active Archive Center <https://lpdaac.usgs.gov/products/mod13q1v006/>; and MODIS/Aqua+Terra Thermal Anomalies/Fire locations 1km FIRMS V0061 (NRT, 2020) available at NASA FIRMS <https://www.earthdata.nasa.gov/learn/find-data/near-real-time/firms/mcd14dl-nrt>.

Acknowledgments

None.

References

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., et al. (2016). TensorFlow: A system for large-scale machine learning. arXiv. (arXiv: 1605.08695 [cs])
- Abid, F. (2021). A survey of machine learning algorithms based forest fires prediction and detection systems. *Fire Technology*, 57(2), 559–590. <https://doi.org/10.1007/s10694-020-01056-z>
- Aggarwal, C. C., & Reddy, C. K. (2013). *Data clustering: Algorithms and applications, ser. Data mining and knowledge discovery series*. CRC Press/Taylor & Francis Group.
- Arif, M., Alghamdi, K., Sahel, S., Alosaimi, S., Alsaft, M., Alharthi, M., & Arif, M. (2021). Role of machine learning algorithms in forest fire management: A literature review. *IEEE Journal on Robotics and Automation*, 5(1), 212–226. <https://doi.org/10.36959/673/372>
- Arpaci, A., Malowerschnig, B., Sass, O., & Vacik, H. (2014). Using multi variate data mining techniques for estimating fire susceptibility of Tyrolean forests. *Applied Geography*, 53, 258–270. <https://doi.org/10.1016/j.apgeog.2014.05.015>
- Bank, D., Koenigstein, N., & Giryas, R. (2020). Autoencoders. arXiv. arXiv preprint arXiv:2003.05991.
- Borchers Arriagada, N., Palmer, A. J., Bowman, D. M., Morgan, G. G., Jalaludin, B. B., & Johnston, F. H. (2020). Unprecedented smoke-related health burden associated with the 2019–20 bushfires in eastern Australia. *Medical Journal of Australia*, 213(6), 282–283. <https://doi.org/10.5694/mja2.50545>
- Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). Lof: Identifying density-based local outliers. In *Proceedings of the 2000 acm sigmod international conference on management of data* (pp. 93–104).
- Canada, N. R. (2023). Canadian wildland fire information system. Retrieved from <https://cwfis.cfs.nrcan.gc.ca/home>
- Chang, Z., Du, Z., Zhang, F., Huang, F., Chen, J., Li, W., & Guo, Z. (2020). Landslide susceptibility prediction based on remote sensing images and gis: Comparisons of supervised and unsupervised machine learning models. *Remote Sensing*, 12(3), 502. <https://doi.org/10.3390/rs12030502>

- Cheng, D., Fan, Y., Fang, S., Wang, M., & Liu, H. (2022). Resnet-ae for radar signal anomaly detection. *Sensors*, 22(16), 6249. <https://doi.org/10.3390/s22166249>
- Cheng, Z., Wang, S., Zhang, P., Wang, S., Liu, X., & Zhu, E. (2021). Improved autoencoder for unsupervised anomaly detection. *International Journal of Intelligent Systems*, 36(12), 7103–7125. <https://doi.org/10.1002/int.22582>
- Coca, M., & Datcu, M. (2021). Anomaly detection in post fire assessment. In *2021 IEEE international geoscience and remote sensing symposium igarss* (pp. 8620–8623).
- Demir, F. (2022). Deep autoencoder-based automated brain tumor detection from MRI data. In *Artificial intelligence-based brain-computer interface* (pp. 317–351). Elsevier.
- Didan, K. (2015). Mod13q1 modis/terra vegetation indices 16-day l3 global 250m sin grid v006 [dataset]. *NASA EOSDIS Land Processes Distributed Active Archive Center*. <https://doi.org/10.5067/MODIS/MOD13Q1.006>
- Esmaili, F., Cassie, E., Nguyen, H. P. T., Plank, N. O., Unsworth, C. P., & Wang, A. (2023). Anomaly detection for sensor signals utilizing deep learning autoencoder-based neural networks. *Bioengineering*, 10(4), 405. <https://doi.org/10.3390/bioengineering10040405>
- Gholamnia, K., Gudiyangada Nachappa, T., Ghorbanzadeh, O., & Blaschke, T. (2020). Comparisons of diverse machine learning approaches for wildfire susceptibility mapping. *Symmetry*, 12(4), 604. <https://doi.org/10.3390/sym12040604>
- Haar, L., Anding, K., Trambitckii, K., & Notni, G. (2019). Comparison between supervised and unsupervised feature selection methods. *Icram*, 582–589. <https://doi.org/10.5220/0007385305820589>
- Hasan, M., Choi, J., Neumann, J., Roy-Chowdhury, A. K., & Davis, L. S. (2016). Learning temporal regularity in video sequences. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 733–742).
- IBM. (2020). Call for code spot challenge for wildfires competition dataset [dataset]. <https://github.com/Call-for-Code/Spot-Challenge-Wildfires>
- Jain, P., Coogan, S. C., Subramanian, S. G., Crowley, M., Taylor, S., & Flannigan, M. D. (2020). A review of machine learning applications in wildfire science and management. *Environmental Reviews*, 28(4), 478–505. <https://doi.org/10.1139/er-2020-0019>
- Kondylatos, S., Prapas, I., Ronco, M., Papoutsis, I., Camps-Valls, G., Piles, M., et al. (2022). Wildfire danger prediction and understanding with deep learning. *Geophysical Research Letters*, 49(17), e2022GL099368. <https://doi.org/10.1029/2022gl099368>
- Kucharski, D., Kleczek, P., Jaworek-Korjakowska, J., Dyduch, G., & Gorgon, M. (2020). Semi-supervised nests of melanocytes segmentation method using convolutional autoencoders. *Sensors*, 20(6), 1546. <https://doi.org/10.3390/s20061546>
- Lehr, J., Philipps, J., Hoang, V. N., von Wrangel, D., & Krüger, J. (2021). Supervised learning vs. unsupervised learning: A comparison for optical inspection applications in quality control. *IOP Conference Series: Materials Science and Engineering*, 1140(1), 012049. <https://doi.org/10.1088/1757-899x/1140/1/012049>
- Li, T., Wang, Z., Liu, S., & Lin, W.-Y. (2021). Deep unsupervised anomaly detection. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision* (pp. 3636–3645).
- Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2012). Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data*, 6(1), 1–39. <https://doi.org/10.1145/2133360.2133363>
- Ma, H. (2022). Achieving deep clustering through the use of variational autoencoders and similarity-based loss. *Mathematical Biosciences and Engineering*, 19(10), 10344–10360. <https://doi.org/10.3934/mbe.2022484>
- Malik, A., Jalin, N., Rani, S., Singhal, P., Jain, S., & Gao, J. (2021). Wildfire risk prediction and detection using machine learning in San Diego, California. In *2021 IEEE smartworld, ubiquitous intelligence & computing, advanced & trusted computing, scalable computing & communications, internet of people and smart city innovation (smartworld/scalcom/uic/atc/iop/sci)* (pp. 622–629).
- Masci, J., Meier, U., Cireşan, D., & Schmidhuber, J. (2011). Stacked convolutional auto-encoders for hierarchical feature extraction. In *Artificial neural networks and machine learning—ICANN 2011: 21st international conference on artificial neural networks, Espoo, Finland, June 14–17, 2011, proceedings, part i 21* (pp. 52–59).
- Maxion, R. A., & Roberts, R. R. (2004). Proper use of roc curves in intrusion/anomaly detection. *School of Computing Science Technical Report Series*.
- Meddour-Sahar, O. (2015). Wildfires in Algeria: Problems and challenges. *iForest: Biogeosciences and Forestry*, 8(6), 818–826. <https://doi.org/10.3832/for1279-007>
- Melo, F. (2013). Area under the roc curve. *Encyclopedia of systems biology*, 2013.
- Modis. (2020). Modis/aqua+terra thermal anomalies/fire locations 1km firms v0061 nrt [dataset]. *NASA FIRMS*. <https://doi.org/10.5067/FIRMS/MODIS/MCD14DL.NRT.0061>
- Muñoz Sabater, J. (2019). Era5-land hourly data from 1950 to present [dataset]. *Copernicus Climate Change Service (C3S) Climate Data Store (CDS)*. <https://doi.org/10.24381/cds.e2161bac>
- Nhangumbe, M., Nascetti, A., Georganos, S., & Ban, Y. (2023). Supervised and unsupervised machine learning approaches using sentinel data for flood mapping and damage assessment in Mozambique. *Remote Sensing Applications: Society and Environment*, 32, 101015. <https://doi.org/10.1016/j.rsase.2023.101015>
- Norman, B., Newman, P., & Steffen, W. (2021). Apocalypse now: Australian bushfires and the future of urban settlements. *npj Urban Sustainability*, 1(1), 2. <https://doi.org/10.1038/s42949-020-00013-7>
- Ogie, R., Moore, A., Wickramasuriya, R., Amirghasemi, M., James, S., & Dilworth, T. (2022). Twitter data from the 2019–20 Australian bushfires reveals participatory and temporal variations in social media use for disaster recovery. *Scientific Reports*, 12(1), 16914. <https://doi.org/10.1038/s41598-022-21265-6>
- Pang, Y., Li, Y., Feng, Z., Feng, Z., Zhao, Z., Chen, S., & Zhang, H. (2022). Forest fire occurrence prediction in China based on machine learning methods. *Remote Sensing*, 14(21), 5546. <https://doi.org/10.3390/rs14215546>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2018). Scikit-learn: Machine learning in python. *arXiv*. (arXiv:1201.0490 [cs])
- Pérez-Porras, F.-J., Triviño-Tarradas, P., Cima-Rodríguez, C., Meroño-de Larriva, J.-E., García-Ferrer, A., & Mesas-Carrascosa, F.-J. (2021). Machine learning methods and synthetic data generation to predict large wildfires. *Sensors*, 21(11), 3694. <https://doi.org/10.3390/s21113694>
- Pradhan, P., Meyer, T., Vieth, M., Stallmach, A., Waldner, M., Schmitt, M., et al. (2021). Computational tissue staining of non-linear multimodal imaging using supervised and unsupervised deep learning. *Biomedical Optics Express*, 12(4), 2280–2298. <https://doi.org/10.1364/boe.415962>
- Qiang, Z., Ze Meka, E., Anderson, C., & Kakabadse, Y. (2011). Our planet: Forests nature at your service. Retrieved from <https://www.unep.org/resources/report/our-planet-forests-nature-your-service>
- Sakr, G. E., Elhaji, I. H., & Mitri, G. (2011). Efficient forest fire occurrence prediction for developing countries using two weather parameters. *Engineering Applications of Artificial Intelligence*, 24(5), 888–894. <https://doi.org/10.1016/j.engappai.2011.02.017>
- Salehi, M., Rusu, L. I., Lynar, T., & Phan, A. (2016). Dynamic and robust wildfire risk prediction system: An unsupervised approach. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 245–254). ACM.

- Sanderfoot, O., Bassing, S., Brusa, J., Emmet, R., Gillman, S., Swift, K., & Gardner, B. (2022). A review of the effects of wildfire smoke on the health and behavior of wildlife. *Environmental Research Letters*, 16(12), 123003. <https://doi.org/10.1088/1748-9326/ac30f6>
- Saxena, A., Prasad, M., Gupta, A., Bharill, N., Patel, O. P., Tiwari, A., et al. (2017). A review of clustering techniques and developments. *Neurocomputing*, 267, 664–681. <https://doi.org/10.1016/j.neucom.2017.06.053>
- Sayad, Y. O., Mousannif, H., & Al Moatassime, H. (2019). Predictive modeling of wildfires: A new dataset and machine learning approach. *Fire Safety Journal*, 104, 130–146. <https://doi.org/10.1016/j.firesaf.2019.01.006>
- Schölkopf, B., Williamson, R. C., Smola, A., Shawe-Taylor, J., & Platt, J. (1999). Support vector method for novelty detection. In *Advances in neural information processing systems* (Vol. 12).
- Serra, A., & Tagliaferri, R. (2019). Unsupervised learning: Clustering. In S. Ranganathan, M. Gribskov, K. Nakai, & C. Schönbach (Eds.), *Encyclopedia of bioinformatics and computational biology*. Academic Press.
- Shin, H. J., Eom, D.-H., & Kim, S.-S. (2005). One-class support vector machines—An application in machine fault detection and classification. *Computers & Industrial Engineering*, 48(2), 395–408. <https://doi.org/10.1016/j.cie.2005.01.009>
- Shmuel, A., & Heifetz, E. (2022). Global wildfire susceptibility mapping based on machine learning models. *Forests*, 13(7), 1050. <https://doi.org/10.3390/f13071050>
- Sulova, A., & Jokar Arsanjani, J. (2020). Exploratory analysis of driving force of wildfires in Australia: An application of machine learning within google earth engine. *Remote Sensing*, 13(1), 10. <https://doi.org/10.3390/rs13010010>
- Sutera, A. (2021). Importance measures derived from random forests: Characterisation and extension. arXiv. (arXiv:2106.09473 [cs, stat])
- Thudumu, S., Branch, P., Jin, J., & Singh, J. (2020). A comprehensive survey of anomaly detection techniques for high dimensional big data. *Journal of Big Data*, 7, 1–30. <https://doi.org/10.1186/s40537-020-00320-x>
- Tian, F., Gao, B., Cui, Q., Chen, E., & Liu, T.-Y. (2014). Learning deep representations for graph clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 28). <https://doi.org/10.1609/aaai.v28i1.8916>
- Tuo, K., Li, W., Deng, S., & Zhu, Y. (2023). Supervised, semi-supervised, and unsupervised learning of the domany-kinzel model. arXiv preprint arXiv:2309.13990.
- Vardoulakis, S., Marks, G., & Abramson, M. J. (2020). Lessons learned from the Australian bushfires: Climate change, air pollution, and public health. *JAMA Internal Medicine*, 180(5), 635–636. <https://doi.org/10.1001/jamainternmed.2020.0703>
- Yu, P., Xu, R., Abramson, M. J., Li, S., & Guo, Y. (2020). Bushfires in Australia: A serious health emergency under climate change. *The Lancet Planetary Health*, 4(1), e7–e8. [https://doi.org/10.1016/s2542-5196\(19\)30267-0](https://doi.org/10.1016/s2542-5196(19)30267-0)
- Zenati, H., Romain, M., Foo, C.-S., Lecouat, B., & Chandrasekhar, V. (2018). Adversarially learned anomaly detection. In *2018 IEEE international conference on data mining (ICDM)* (pp. 727–736).
- Zhang, J., Sheng, Z., He, Y., Zuo, X., Jin, B., & He, M. (2021). Analysis of the impact of the 2019–20 Australian bushfire season on the atmospheric environment. *Frontiers in Earth Science*, 9, 566891. <https://doi.org/10.3389/feart.2021.566891>
- Zhao, Y., Deng, B., Shen, C., Liu, Y., Lu, H., & Hua, X.-S. (2017). Spatio-temporal autoencoder for video anomaly detection. In *Proceedings of the 25th acm international conference on multimedia* (pp. 1933–1941).



DSpace

<https://dspace.org/>

Deep autoencoders for unsupervised anomaly detection in wildfire prediction

Üstek, rem; Arana-Catania, Miguel; Farr, Alexander; Petrunin, Ivan

2024-11-28

American Geophysical Union (AGU)

<https://doi.org/10.1029/2024ea003997>

Üstek , AranaCatania M, Farr A, Petrunin I. (2024) Deep autoencoders for unsupervised anomaly detection in wildfire prediction. Earth and Space Science, Volume 11, Issue 11, November 2024, Article number e2024EA003997