A Comparison of One-Class Versus Two-Class Machine Learning Models for Wildfire Prediction in California

Fathima Nuzla Ismail[1](✉), Abira Sengupta[2], Brendon J. Woodford[2], and Sherlock A. Licorish[2]

[1] Department of Biochemistry, University of Otago, Dunedin, New Zealand
nuzla.ismail@otago.ac.nz

[2] School of Computing, University of Otago, Dunedin, New Zealand
{abira.sengupta,brendon.woodford,sherlock.licorish}@otago.ac.nz

Abstract. Due to climate change, forest regions in California are increasingly experiencing severe wildfires, with other issues affecting the rest of the world. Machine learning (ML) and artificial intelligence (AI) models have emerged to predict wildfire hazards and aid mitigation efforts. However, the wildfire prediction modelling domain faces inconsistencies due to database manipulations for multi-class classification. To help to address this issue, our paper focuses on creating wildfire prediction models through One-class classification algorithms: Support Vector Machine, Isolation Forest, AutoEncoder, Variational AutoEncoder, Deep Support Vector Data Description, and Adversarially Learned Anomaly Detection. To minimise bias in the selection of the training and testing data, Five-Fold Cross-Validation was used to validate all One-class ML models. These One-class ML models outperformed Two-class ML models using the same ground truth data, with mean accuracy levels between 90 and 99 percent. Shapley values were used to derive the most important features affecting the wildfire prediction model, which is a novel contribution to the field of wildfire prediction. Among the most important factors were the seasonal maximum and mean dew point temperatures. In providing access to our algorithms, using Python Flask and a web-based tool, the top-performing models were operationalized for deployment as a REST API, with the potential to strengthen wildfires mitigation strategies.

Keywords: One-class SVM · ANN-AutoEncoder · ANN-Variational Auto-Encoder · Isolation Forest · `scikit-learn` · `PyOD`

1 Introduction

Wildfires have become a significant issue, destroying thousands of square kilometres of forest yearly. This type of disaster has a global impact on environments, the economy, and health. Natural wildfires are caused primarily by lightning, volcanic eruptions, dry climate, and vegetation. However, it has been documented

D. Benavides-Prado et al. (Eds.): AusDM 2023, CCIS 1943, pp. 239–253, 2024.
https://doi.org/10.1007/978-981-99-8696-5_17

that at least 90% of wildfires are caused by human behaviour, such as smoking in public, camping fires, and garbage burning [28]. As a result, continuous monitoring is required to address this serious issue and, more importantly, to forecast the possibility of widespread and intense wildfires. This brings us to the fundamental challenge that the public and fire management authorities inevitably face: the possibility of predicting wildfires well in advance to take timely action to mitigate damages.

ML and AI methods may aid researchers in developing models for monitoring and predicting wildfire anomalies in advance. However, technical limitations and environmental issues impede the process of monitoring and detecting wildfire occurrences and spread. Furthermore, the specific characteristics that may influence wildfire ignition remain as a research gap. This is due primarily to significant changes in atmospheric conditions, which frequently include air temperature, relative humidity, wind speed and direction, and spatial and temporal time-bounded features [21].

Many ML solutions for wildfire prediction have been developed by researchers, but only a few solutions make it to the deployment stage when it comes to practical use. Incorporating ML models into an Application Programming Interface (API) to develop user-friendly applications would improve the wildfire prediction domain. We look to address this opportunity, and provide the following contributions:

1. Using a fire incidence data set, we demonstrate how the application of appropriate One-class classification algorithms are better suited towards fire risk prediction than Two-class models.
2. The use of Shapley values identify features from the One-class ML models that significantly influence the risk of a wildfire event, providing explainability for our models.
3. A proposed architecture for the development and deployment of a web-based wildfire prediction tool that adopts the best One-class ML model.

For this study, the state of California was selected as the context for predicting the occurrence of wildfires. The experiment generated a set of historical fire data from California (2012 to 2016). Multiple One-class ML algorithms: Support Vector Machine (SVM), Isolation Forest (IF), Autoencoder (AE), Variational AutoEncoder (VAE), Deep Support Vector Data Description (DeepSVDD), and Adversarially Learned Anomaly Detection (ALAD) were investigated in these experiments. Repeated Five-Fold Cross-Validation (CV) was applied to the training data set to generate these models, yielding accuracy ranging from 90% to 99%.

The rest of the paper is organized as follows. In Sect. 2 we provide the study background and describe the One-class ML algorithms used in our experiments. Next, in Sect. 3 we describe the data set for the Californian case study. Our methodology is then provided in Sect. 4. The results of applying our methodology are presented in Sect. 5. In Sect. 6, the deployment of the ML models is discussed, followed by the web-based prototype evaluation in Sect. 7. Finally, in Sect. 8 we summarise our findings and outline opportunities for future work.

2 Background

Defining a negatively unbiased sample data set for complex events such as wildfires is difficult. Without properly validating these data points, a slew of non-fire data points could be generated for a given location, date, and time. This problem can be solved by using a One-class classification model that defines a class boundary based on positive data labels [15].

When the model outcome probability is greater than the threshold value in One-class binary classification, it is labelled as an inlier (•) and when the model outcome probability is less than the threshold value in One-class binary classification, it is labelled as an outlier (?), which are based on the model output probability and the threshold value, as shown in Fig. 1a. Choosing an accurate threshold is critical for correctly classifying inliers and outliers. In principle, the classification boundary of One-class learning accepts many positive data labels while rejecting only a few outliers (see Fig. 1a). Positive data labels are used to train the model in One-class learning, whereas outliers are considered negative data labels or non-fire events.

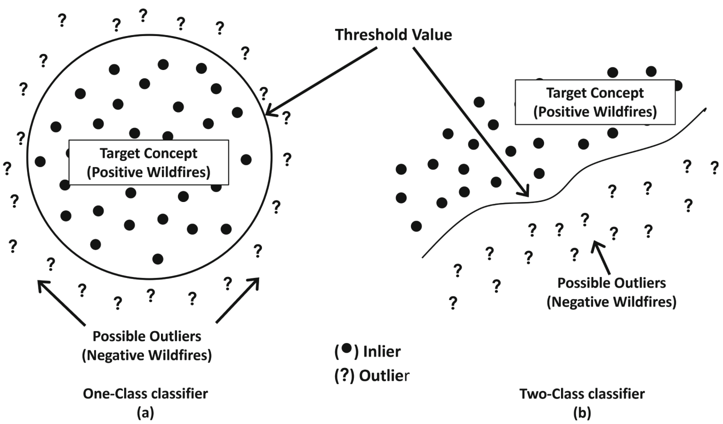

Fig. 1. The distinction between One-class classification (a) and Two-class classification (b). Compared with a One-class classification model, a Two-class classification model accepts inlier (positive) data labels but rejects outlier labels.

As noted above, the model's outcome probabilities, as well as its inlier and outlier predictions, are affected by the threshold value. When the threshold is greater than a certain value for a given prediction instance [10, p. 159], it will be detected as a fire (inlier). For our experiments, each One-class ML algorithm's functionality is described below.

In terms of our approaches, the Support Vector Machine (SVM) is a supervised learning model that analyses data and identifies patterns for both classification and regression tasks [6]. The One-class variant refers to two types of One-class SVM (OCSVM). The standard OCSVM uses a sphere of minimal volume to contain a specified proportion of training instances [31]. The other OCSVM trains objects using a hyperplane in a kernel feature space. Data is transformed to a higher dimensional space in order to investigate the possibility of constructing a hyperplane decision boundary, with the assumption that all training points belong to one class and all non-training points belong to another. To generate the ML model in the experiments, the latter OCSVM algorithm was used. The One-class OCSVM operates in such a way that standard data clustered into a single region has a high density, while outliers are detected as low-density regions. New data points can be tested based on density regions to detect normal or outlier cases.

Isolation Forest (IF) is a binary random forest approach in which each node randomly chooses a dimension, and then a splitting threshold [17]. It will keep going until each node has a single sample. This method is used to build an ensemble of trees. A sample with exceptional values has a higher chance of being isolated early in the growth of the tree by chance than samples in clusters; as a result, the average depth of samples in the ensemble of trees directly affects the abnormality score [17].

A recent overview of ML and AI algorithms used in wildfire prediction are summarised in [2] which discussed models that are based on Artificial Neural Networks (ANN) including ones based on Radial Basis Function ANNs [23]. For our experiments we similarly adopted an AutoEncoder (AE) which is a type of multi-layer ANN for unsupervised learning that copies input values to output values, allowing mapping from high-dimensional space to lower-dimensional representation [26]. To reduce reconstruction errors, input data is encoded in the hidden layers. This method forces the hidden layers to learn the most patterns in the data while ignoring the "noise". Anomalies are defined as input data with a high reconstruction error. In contrast to an AE, the Variational AutoEncoder (VAE) learns the parameters of a probability distribution representing the data, which could make the model more adept at spotting anomalies [26].

Like AE and VAE, the goal of DeepSVDD [29] is to learn network parameters collaboratively while minimising the average distance from all data representations to the center for this algorithm. Normal data are closely mapped to the center for this algorithm, whereas anomalous data are mapped farther from the centre or outside a hypersphere [16]. In DeepSVDD, ANN are used as One-class classifiers, where any data points which the neural network rejects is categorised as an outlier. Network weights are derived from the training data. These trained network weights are then used in the process of testing new data instances. We have selected DeepSVDD [29] and ALAD [34] due to their popularity in performing prediction domains.

Finally, ALAD [34] is a reconstruction-based anomaly detection technique that assesses how well a sample is reconstructed by a Generative Adversarial Network (GAN). GANs are adopted as they can model complex high-dimensional distributions of real-world data, implying that they could be useful in anomaly detection. ALAD is a promising approach in complex, high-dimensional data.

ALAD is based on bidirectional GANs and contains an encoder network that maps data samples to latent variables. During training, this learns an encoder from the data space to the latent space, making it significantly more efficient at test time. ALAD assesses how far a sample is from its reconstruction by the GAN, where normal samples should be accurately reconstructed while anomalous samples are likely to be poorly reconstructed.

The OCSVM algorithm from the Python `scikit-learn` package [27] was used for the experiments. All the remaining methods including an alternative implementation of the OCSVM algorithm were taken from the Python `PyOD` package [35].

3 Data Set

The case study is based on California in the United States of America, which spans a land area of 423,970 square km[1]. From 2012 to 2016, 7,335 wildfire events were recorded in California by US Federal land management agencies, NOAA, the American Scientific Agency, MODIS 500m resolution satellite images, and the US Census Bureau[2]. The variables for the Californian data set were acquired accordingly, and are listed in Table 1. The collected data were combined into a data set that was geolocated and transformed into an appropriate format for further analysis[3]. These procedures were followed for the implementation of the use case in California.

4 Methodology

As demonstrated in Fig. 2, developing a decision support system for wildfire prediction involves a number of steps, including data preparation, processing, modelling, validation of ML models, and the potential for deployment of ML models.

Wildfire features, weather features, Live Fuel Moisture Content (LFMC) features, and social features are the four input categories that are used. The data set was encoded and scaled to test the ML models based on One-class classification. Below is a more thorough explanation of these steps.

The relevant classifier function calls were used during model training to fit the model to the data. Hyper-parameter tuning was used to configure the function's hyper-parameters, eventually producing one ML model for each classifier type that performed the best. During the tuning process, the hyper-parameters of the models were adjusted to achieve the best accuracy based on the most

[1] https://www.fire.ca.gov/our-impact/statisticsStatistics on CA wildfires and CAL FIRE activity.

[2] A different case study with 2.2 million acres burned in Western Australia was conducted as the second case study. However, due to page limitations, we are unable to discuss this data set and its associated results in this paper.

[3] This thesis provides more detail on the steps involved in data pre-processing [10].

Table 1. Variables used for ML models - Californian data set (7,335 Events)

No.	Feature	Description	Prior Research
1	IDATE	Fire Occurrence Date (Month & Date as an Integer)	[1]
2	LAT	Fire location latitude (degrees)	[1,11,33]
3	LON	Fire location longitude (degrees)	[1,11,33]
4	ELEVATION_m	Fire location elevation (in meters)	[1,7,11]
5	ACRES	Acres burnt (in acres)	
6	PPT_mm	Precipitation (in mm for the fire incident date)	[1,11,13]
7	TMIN_c	Minimum temperature (in Celsius for the fire incident date)	[11,13]
8	TMEAN_c	Mean temperature (in Celsius for the fire incident date)	[11,13]
9	TMAX_c	Maximum temperature (in Celsius for the fire incident date)	[1,11,13]
10	TDMEAN_c	Mean dew point temperature (in Celsius for the fire incident date)	[11,13]
11	VPDMIN_hpa	Minimum vapor pressure (in hectopascals) - Californian use case	[7]
12	VPDMAX_hpa	Maximum vapor pressure (in hectopascals) - Californian use case	[7]
13	lfmc_mean	Mean fuel moisture for a particular day (numeric)	[11]
14	lfmc_stdv	Standard deviation of fuel moisture for a particular day (numeric)	[11]
15	Mean_Sea_Level _Pressure	Mean sea level pressure of the nearest weather station to the wildfire event (in hectopascals) - (Universal Kriging)	[25]
16	Mean_Station _Pressure	Nearest mean weather station pressure to the wildfire event (in hectopascasl) - (Universal Kriging)	[25]
17	Mean_Wind _Speed	Mean wind speed for a given location (numeric mph) - (Universal Kriging)	[1,7,11]
18	Maximum_sustained _wind_speed	Maximum sustained wind speed for a given location (numeric MPH) - (Universal Kriging)	[7,11]
19	NAMELSAD	County name (string)	[13]
20	Population	Number of residents living in the respective county (numeric)	[13,24]

significant features determined by the ML algorithm. This procedure used the Python `hyperopt` package [4]. To elaborate, the first step was to specify relevant hyper-parameters for the ML models with predefined options and a range of values. The ML models were then trained for 80 iterations using various combinations of those hyper-parameters. Within each iteration, each model was trained using Five-Fold CV, and the average performance of that model was used to tune the hyper-parameters for the following model. Target values were predicted using testing data and then on the entire data set using the best-performing ML model via 20 times Five-Fold CV. This process produced mean Accuracy, Precision, Recall, and F1-Score classification metrics.

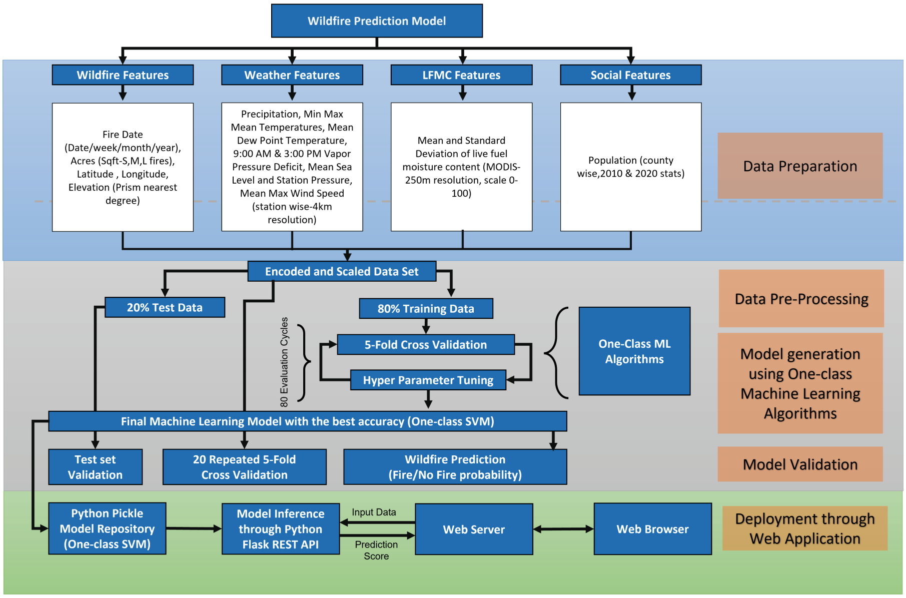

Fig. 2. The process of building a wildfire prediction model involves various steps from data preparation to deploying ML models through a web-based tool

5 Results

Here we compare the results of the ML models on the California wildfire data set as described in Sect. 5.1 with Two-class classification problems, which are described in the earlier research cited in Sect. 5.2. Additionally, Sect. 5.3 illustrates the most important features and their impact on One-class ML models for the two subtypes of OCSVM.

5.1 One-Class Machine Learning Model Results

Table 2 summarises the performance of the One-class ML models for the Californian data set. The number of inliers (fire positive) predictions, which correspond to the number of actual wildfire events, and the number of outliers (fire negative) predictions are used to assess the effectiveness of the applied ML approaches. The results from Table 2 highlight that the OCSVM (`PyOD`) model was the best performing One-class classifier, achieving a mean test Accuracy of 0.99, mean Precision of 1.00, mean Recall of 0.99, and mean F1-Score of 0.99. A more objective assessment of the OCSVM (`PyOD`) model through 20 $\times$ Five-Fold CV resulted in its performance being higher than the other ML models observed.

With mean test Accuracy of 0.99, Precision of 1.00, Recall of 0.99, and F1-Score of 1.00, the IF model produced results comparable with the other ML models validating its outstandng performance. Mean test Accuracy, Precision,

Table 2. California ML Model Results Summary

ML Technique	Data set Type	Data set Count	Inliers	Outliers	Mean Accuracy	Mean Precision	Mean Recall	Mean F1-Score	20 × Five-Fold CV
OCSVM	Train (80%)	5,868	5,806	62	0.989	1.000	0.989	0.994	0.990
(`sklearn`)	Test (20%)	1,467	1,443	24	0.983	1.000	0.983	0.991	±0.0030
OCSVM	Train (80%)	**5,868**	**5,809**	**59**	**0.989**	**1.000**	**0.990**	**0.990**	**0.990**
(`PyOD`)	Test (20%)	**1,467**	**1,458**	**9**	**0.993**	**1.000**	**0.990**	**1.000**	**±0.0028**
AE	Train (80%)	5,868	5,809	59	0.989	1.000	0.990	0.990	0.989
(`PyOD`)	Test (20%)	1,467	1,454	13	0.991	1.000	0.990	1.000	±0.0030
VAE	Train (80%)	5,868	5,809	59	0.989	1.000	0.990	0.990	0.989
(`PyOD`)	Test (20%)	1,467	1,454	13	0.991	1.000	0.990	1.000	±0.0028
IF	Train (80%)	**5,868**	**5,809**	**59**	**0.989**	**1.000**	**0.990**	**0.990**	**0.989**
(`PyOD`)	Test (20%)	**1,467**	**1,458**	**9**	**0.993**	**1.000**	**0.990**	**1.000**	**±0.0030**
DeepSVDD	Train (80%)	5,868	5,281	587	0.899	1.000	0.900	0.950	0.897
(`PyOD`)	Test (20%)	1,467	1,316	151	0.897	1.000	0.900	0.950	±0.0101
ALAD	Train (80%)	5,868	5,281	587	0.899	1.000	0.900	0.950	0.900
(`PyOD`)	Test (20%)	1,467	1,272	195	0.867	1.000	0.870	0.930	±0.0081

Recall, and F1-Score for the AE and VAE models ranged from 0.99 to 1.00. Additionally, the mean test Accuracy, Precision, Recall, and F1-Score values for the DeepSVDD and ALAD ML models were lower ranging from 0.87 to 1.00.

It should be noted that both OCSVM ML models, despite being less complex than an ALAD and DeepSVDD model, perform better on all mean test metrics providing adequate support for the outcomes of adopting simpler One-class ML models.

5.2 Two-Class Machine Learning Outcomes for the Same Ground-Truth Data

In assessing the One-class ML approach using the same ground truth data and a randomly generated equal amount of false data [32], created by applying Two-class ML models for the California region. Sayad [30] used a similar approach in representing negative samples using random timestamps and locations. Hence, the same approach was followed in creating a false data set. Furthermore, commonly used wildfire prediction models using Two-class ML models were investigated and chosen for this use case. As shown in Table 3, the Two-class ML algorithms were used with supporting literature for predicting wildfires.

Table 3. Performance of Two-class ML models

ML Algorithm	Supporting Literature	Mean Accuracy	Mean Precision	Mean Recall	Mean F1-Score
SVM	[8,11,22]	0.628	0.657	0.763	0.706
RF	[8,11,20]	0.679	0.664	0.724	0.693
Logistic Regression	[3,9,22]	0.676	0.651	0.756	0.697
XGBoost Regression	[19,20]	0.675	0.660	0.717	0.688
ANN	[8,9,22]	0.682	0.665	0.732	0.697

The outcome shows that similar Two-class-based ML models achieved mean test Accuracies from 0.63 to 0.68 for the test data set. Mean test Precision recorded values from 0.65 to 0.66 and average mean Recall values ranged from

0.73 to 0.76. Mean test F1-Score values recorded a range from 0.69 to 0.72. Hence, these results suggest that the Two-class models exhibit reduced performance for the selected data sets compared to One-class ML models using the same ground truth data. Therefore, One-class ML models can serve as good alternatives in prediction models such as wildfire risk, which has limited ground truth data over the period in question.

5.3 Feature Importance Derived Using Shapley Values

This section examines the results obtained through the application of Shapley values, which emphasize the most crucial features and their impact on One-class ML models. These values are obtained by using game theory principles and coefficients from the internal linear regression [18].

The Shapley value is a metric used to determine the average marginal contribution of each feature when considering all possible combinations (coalitions) of features [18]. To illustrate, to calculate the Shapley value of mean wind speed, one needs to evaluate all possible combinations of mean wind speed observations. For each combination, the marginal contribution of ignition probability will be assessed. By aggregating all the marginal contributions to ignition probability, the mean marginal contribution of ignition probability can be determined as the Shapley value's outcome.

Using Shapley values [18], the plot on the left in Fig. 3 shows the average impact of the features on the One-class OCSVM PyOD ML models' outputs. The most influential attributes included the maximum and average dew point temperatures associated with different seasons. Then `Mean_Sea_Level_Pressure`, `PPT_mm`, and `lfmc_mean` are the second set of essential features that influence wildfire prediction. For example, the temperature variables and `lfmc_mean` have a more significant impact on the model output for the risk of wildfire than does the population. Also, high LFMC is more susceptible to ignition and can signal more fire spread [11]. `Mean_Sea_Level_Pressure` is the average level of one or more bodies of water on Earth from which elevation can be calculated. With increasing elevation, sea level pressure decreases. Wind speed and direction are both factors in the wind effect. The dry wind is one of the primary causes of wildfire spreading. The rate of wildfire spread has been estimated to be around 8% of wind speed, regardless of fuel type, especially in dry fuel moisture conditions [12]. It can be noted that these same features are ranked highly across all the models, and hence, these top-ranked features should be given more importance in the modelling process. Furthermore, the Shapley value impact has been investigated in Fig. 3. The result of testing the features and models informed a web-based tool, which is presented in the following section to showcase the efficiency and practicality of One-class ML models.

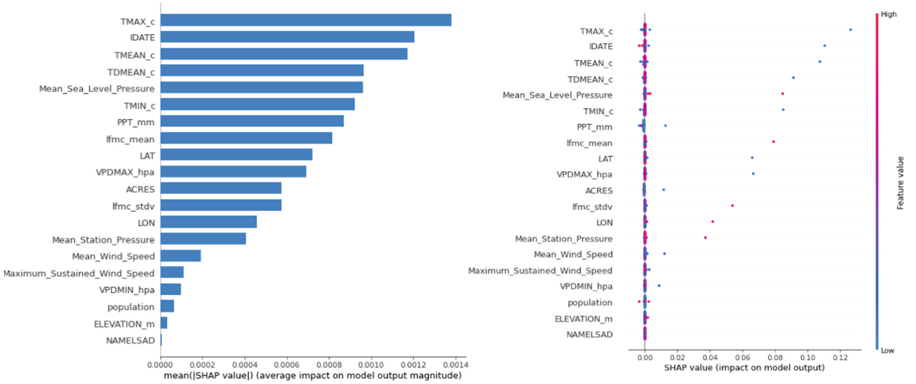

Fig. 3. Shapley values generated from the OCSVM PyOD model (right) shows that the mean and temperature values are high when the model output is predicting a positive fire occurrence

6 Deployment of Machine Learning Models

The web-based tool[4] is presented as a contribution to the state-of-the-art ML-based wildfire prediction domain (see Fig. 4). The main objective of this phase of research is to deploy the results of the selected ML model using a REST API, which will then be fed into a web-based tool. The web-based tool's goal is to provide long-term wildfire predictions based on One-class classification-based ML models that can predict the start of a wildfire one week in advance for any given location in California. This can also help international wildfire management authorities test wildfire prediction models across multiple geographies. The web-based tool is also useful for countries that do not have access to wildfire forecasting systems. However, this is not meant to replace current regional wildfire forecasting systems.

The ML model is fed with 20 features (see Sect. 3) from four categories and six probability rates of danger levels, which are mapped by the decision scores (d) of the One-class ML models: No Danger ($d \leq 0$), Low ($0 < d \leq 60$), Moderate ($60 < d \leq 80$), High ($80 < d \leq 90$), Very High ($90 < d \leq 97$) and Severe ($97 < d \leq 100$). These fire danger rating breakpoints used were similar to fire spread probabilities modelled by the US Wildland Fire Decision Support System [14] to create these threshold classes. The selection of these fire danger rating class thresholds was informed by historical fire danger outcomes, as documented in [5]. The Flask REST service for the web-based tool uses three other external REST APIs for generating wildfire prediction outcomes:

1. The publicly available free Open Topo Data elevation REST API which gives elevation data of any location when latitude and longitude are given.
2. The OpenWeather REST API provides historical, current and forecasted weather details through REST APIs for any point on the world.

[4] https://www.bushfirepredict.com.

3. USGS Earth Explorer Website hosts LFMC data as different vegetation indexes carry a file format of all locations of California based on a MODIS grid.

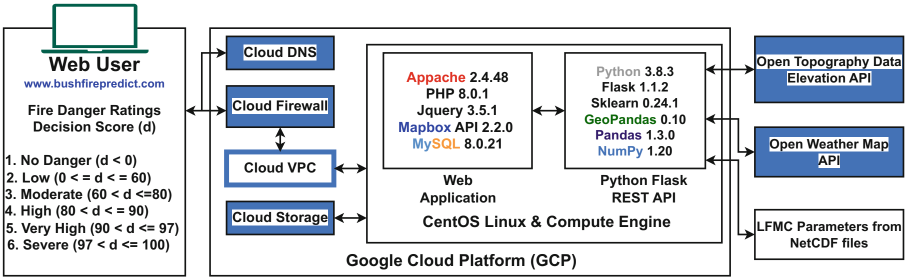

Fig. 4. The deployed ML model's architecture deriving wildfire prediction outcomes using six fire danger rating levels

The four main functionalities listed below were thus identified as the outcome of this web-based tool:

1. Choose some historical wildfire events to train the ML models and validate the model output. Users can also alter the input parameters and analyse and explore the most important features of the ML models.
2. Select any location in California using the map, manually enter the input features, and use a probability to predict wildfire susceptibility.
3. Search all input features for the next 7 d.
4. View historical yearly wildfire heat-maps based on ML model training and testing data.

Several technological advancements in the field of wildfires have emerged in recent years as a result of the high costs and practical difficulties of fighting wildfires. Technology training is at the top of the list because it is crucial for sharing common resources and standards for information on fire danger and communication between fire authorities. The use of technological systems to forecast and predict the occurrence of wildfires has enabled emergency response teams to plan ahead and take preventative action. Firefighters must receive the necessary training to deal with such emergencies. Implementing a straightforward, inexpensive prototype can significantly lower the price of intricate training. Large financial budgets can also be set aside for public education campaigns about wildfire prevention and natural wildfire occurrences. The infrastructure cost of the Google Cloud Platform (GCP) (virtual machine and domain name) constitutes the sole cost element for the implementation of this web-based tool. The remaining programmes and services are either open source, free, or have a free usage tier. Hosting this in an on-premise local area network may thus result in an infrastructure cost saving in terms of infrastructure. The web-based tool has a monthly fee of $42 NZD and can forecast wildfires up to a week in advance[5]. To assess the utility of this tool, a user-based questionnaire evaluation has been carried out, the results of which are reported in the next section.

7 Web-Based Prototype Evaluation

The utility of the web-based tool was assessed by administering an 18-question questionnaire to New Zealand computing practitioners covering general design, performance, and content. The questionnaire was completed in an average of 15 min by 11 participants. More than 81 percent of respondents used Chrome, while Firefox was also used by some, according to the findings. Additionally, over 63% of respondents reported feeling extremely satisfied, with the remaining respondents rating their satisfaction as “somewhat”. Experience with the mobile phone view using different browsers produced mixed results, with the majority of respondents being satisfied, 18% being neither satisfied nor dissatisfied, and the remaining 9% being extremely dissatisfied. The mobile phone version needs to be enhanced further as a result. The overall design of the tool resulted in an above-average ranking for all feedback. Performance-wise, the speed of information retrieval from the input feature fields, the speed of ML prediction, and the response time were all very quick. Additionally, the overall performance was rated as being better than satisfactory, with 80% stating that it performed excellently and with positive feedback exceeding 63% for the tool’s content when measuring the understandability of input and output features of the accuracy of wildfire prediction outcomes. This results in a high-performance rating for the web-based tool.

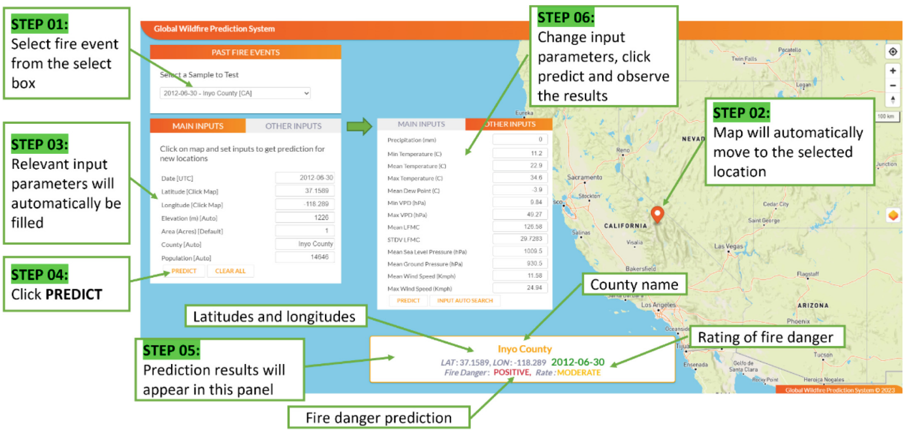

Fig. 5. Inyo 2012-06-30 Fire Event with main input features

[5] More information on the cost calculation can be found on [10, pp. 167–168].

8 Conclusion and Future Work

In summary, historical wildfire events in California were represented using a total of 20 features. Seven different One-class ML algorithms were used order to train multiple models. After the hyper-parameters of the ML models were tuned, the models were validated using repeated 20 × Five-Fold CV. The average test Accuracy of each ML model ranged from 0.90 to 1.00, demonstrating the ML models' high generic performance for the California data set. In addition, Precision, Recall, and F1-Score values were used to evaluate the effectiveness of the ML models.

Not only does our study address the need to create ML-based wildfire prediction models, but, more importantly, it identifies key features from these models that could influence wildfire ignition. As well as our findings being consistent with the outcomes of previous research we also showed the degree to which these identified features contribute to the risk of a wildfire event.

Finally, we described development of a web-based prototype that integrates the best performing ML algorithms and model of the sequence of wildfire events for wildfire occurrence mapping. The intended audiences for this tool are the general public and wildfire authorities.

However, as we only used one data set for this study, future work will involve the creation of more wildfire data sets from other countries, potentially using different features. Top-ranked features extracted from these ML models using Shapley values may be compared and contrasted with the findings from our existing work to show how the contribution of different features influence the risk of wildfire depending on the location.

References

1. Abdollahi, A., Pradhan, B.: Explainable artificial intelligence (XAI) for interpreting the contributing factors feed into the wildfire susceptibility prediction model. Sci. Total Environ. **879**, 163004 (2023). https://doi.org/10.1016/j.scitotenv.2023.163004
2. Alkhatib, R., Sahwan, W., Alkhatieb, A., Schütt, B.: A brief review of machine learning algorithms in forest fires science. Appl. Sci. **13**(14) (2023). https://doi.org/10.3390/app13148275, https://www.mdpi.com/2076-3417/13/14/8275
3. de Bem, P., de Carvalho Júnior, O., Matricardi, E., Guimarães, R., Gomes, R.: Predicting wildfire vulnerability using logistic regression and artificial neural networks: a case study in Brazil. Int. J. Wildland Fire **28**(1), 35–45 (2018). https://doi.org/10.1071/WF18018
4. Bergstra, J., Yamins, D., Cox, D.D.: Making a science of model search: hyperparameter optimization in hundreds of dimensions for vision architectures. In: Proceedings of the 30th International Conference on International Conference on Machine Learning, vol. 28, pp. I-115–I-123. ICML 2013, JMLR.org (2013)
5. Center, N.I.F.: National Wildfire Coordinating Group (NWCG). Interagency Standards for Fire and Fire Aviation Operations. Createspace Independent Publishing Platform, Great Basin Cache Supply Office: Boise, ID, USA (2019)

6. Cortes, C., Vapnik, V.: Support vector machine. Mach. learn. **20**(3), 273–297 (1995)
7. Donovan, G.H., Prestemon, J.P., Gebert, K.: The effect of newspaper coverage and political pressure on wildfire suppression costs. Soc. Nat. Resour. **24**(8), 785–798 (2011)
8. Ghorbanzadeh, O., et al.: Spatial prediction of wildfire susceptibility using field survey GPS data and machine learning approaches. Fire **2**(3), 43 (2019)
9. Goldarag, Y., Mohammadzadeh, A., Ardakani, A.: Fire risk assessment using neural network and logistic regression. J. Indian Soc. Remote Sens. **44**, 1–10 (2016). https://doi.org/10.1007/s12524-016-0557-6
10. Ismail, F.N.: Novel machine learning approaches for wildfire prediction to overcome the drawbacks of equation-based forecasting, Ph. D. dissertation, University of Otago (2022)
11. Jaafari, A., Pourghasemi, H.R.: 28 - Factors influencing regional-scale wildfire probability in Iran: an application of random forest and support vector machine. In: Pourghasemi, H.R., Gokceoglu, C. (eds.) Spatial Modeling in GIS and R for Earth and Environmental Sciences, pp. 607–619. Elsevier (2019)
12. Jain, P., Coogan, S.C., Subramanian, S.G., Crowley, M., Taylor, S., Flannigan, M.D.: A review of machine learning applications in wildfire science and management. Environ. Rev. **28**(4), 478–505 (2020)
13. Jiménez-Ruano, A., Mimbrero, M.R., de la Riva Fernández, J.: Understanding wildfires in mainland Spain. a comprehensive analysis of fire regime features in a climate-human context. Appl. Geogr. **89**, 100–111 (2017)
14. Jolly, W.M., Freeborn, P.H., Page, W.G., Butler, B.W.: Severe fire danger index: a forecastable metric to inform firefighter and community wildfire risk management. Fire **2**(3), 47 (2019). https://doi.org/10.3390/fire2030047
15. Khan, S.S., Madden, M.G.: One-class classification: taxonomy of study and review of techniques. Knowl. Eng. Rev. **29**(3), 345–374 (2014)
16. Kim, S., Choi, Y., Lee, M.: Deep learning with support vector data description. Neurocomputing **165**, 111–117 (2015)
17. Liu, F.T., Ting, K.M., Zhou, Z.: Isolation forest. In: 2008 Eighth IEEE International Conference on Data Mining, pp. 413–422. IEEE (2008)
18. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 4768–4777. Curran Associates Inc. (2017)
19. Ma, J., Cheng, J., Jiang, F., Gan, V., Wang, M., Zhai, C.: Real-time detection of wildfire risk caused by powerline vegetation faults using advanced machine learning techniques. Adv. Eng. Inform. **44**, 101070 (2020). https://doi.org/10.1016/j.aei.2020.101070
20. Michael, Y., Helman, D., Glickman, O., Gabay, D., Brenner, S., Lensky, I.M.: Forecasting fire risk with machine learning and dynamic information derived from satellite vegetation index time-series. Sci. Total Environ. **764**, 142844 (2021). https://doi.org/10.1016/j.scitotenv.2020.142844
21. Miller, C., Hilton, J., Sullivan, A., Prakash, M.: SPARK – a bushfire spread prediction tool. In: ISESS 2015. IAICT, vol. 448, pp. 262–271. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-15994-2_26
22. Nhu, V.H., et al.: Shallow landslide susceptibility mapping: a comparison between logistic model tree, logistic regression, naïve bayes tree, artificial neural network, and support vector machine algorithms. Int. J. Environ. Res. Public Health **17**(8), 2749 (2020)

23. Ntinopoulos, N., Sakellariou, S., Christopoulou, O., Sfougaris, A.: Fusion of remotely-sensed fire-related indices for wildfire prediction through the contribution of artificial intelligence. Sustainability **15**(15), 1–24 (2023). https://doi.org/10.3390/su151511527
24. Nunes, A., Lourenço, L., Meira Castro, A.C.: Exploring spatial patterns and drivers of forest fires in Portugal (1980–2014). Sci. Total Environ. **573**, 1190–1202 (2016). https://doi.org/10.1016/j.scitotenv.2016.03.121
25. Papadopoulos, A., Paschalidou, A., Kassomenos, P., McGregor, G.: On the association between synoptic circulation and wildfires in the Eastern Mediterranean. Theoret. Appl. Climatol. **115**(3), 483–501 (2014)
26. Patterson, J., Gibson, A.: Deep Learning: A Practitioner's Approach. O'Reilly Media, Inc. (2017)
27. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
28. Reisen, F., Duran, S.M., Flannigan, M., Elliott, C., Rideout, K.: Wildfire smoke and public health risk. Int. J. Wildland Fire **24**(8), 1029–1044 (2015)
29. Ruff, L., et al.: Deep one-class classification. In: Dy, J., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 80, pp. 4393–4402. PMLR (2018)
30. Sayad, Y.O., Mousannif, H., Al Moatassime, H.: Predictive modeling of wildfires: a new dataset and machine learning approach. Fire Saf. J. **104**, 130–146 (2019). https://doi.org/10.1016/j.firesaf.2019.01.006
31. Tax, D.M., Duin, R.P.: Support vector domain description. Pattern Recogn. Lett. **20**(11–13), 1191–1199 (1999)
32. Tien Bui, D., Bui, Q.T., Nguyen, Q.P., Pradhan, B., Nampak, H., Trinh, P.T.: A hybrid artificial intelligence approach using GIS-based neural-fuzzy inference system and particle swarm optimization for forest fire susceptibility modeling at a tropical area. Agric. For. Meteorol. **233**, 32–44 (2017)
33. Tonini, M., D'Andrea, M., Biondi, G., Degli Esposti, S., Trucchia, A., Fiorucci, P.: A machine learning-based approach for wildfire susceptibility mapping. the case study of the liguria region in Italy. Geosciences **10**(3), 105 (2020)
34. Zenati, H., Romain, M., Foo, C.S., Lecouat, B., Chandrasekhar, V.: Adversarially learned anomaly detection. In: 2018 IEEE International Conference on Data Mining (ICDM), pp. 727–736. IEEE (2018)
35. Zhao, Y., Nasrullah, Z., Li, Z.: PyOD: a python toolbox for scalable outlier detection. J. Mach. Learn. Res. **20**(96), 1–7 (2019)