**Methods**

**Overview of the NPF classification approach**

The primary objective of this study is to evaluate and compare the performance of various supervised machine learning models in identifying the atmospheric New Particle Formation (or NPF in short) events. These events, while playing a crucial role in aerosol climate interactions, are perhaps rare and often embedded in large volumes of non-event data. To automate the detection and understand the precursor conditions, we designed a pipeline that stimulates an early warning system and applies five commonly available and used machine learning models to forecast NPF occurrences based on the surface-level atmospheric data.

The dataset used in this work was collected from the Southern Great Plains (SGP) atmospheric observatory and includes several environmental and chemical measurements that previous studies have linked to NPF processes (e.g., BLH, , RH, and surface area concentrations). Our method consists of the following steps:

1. Preprocessing and labelling time series data to stimulate early warning of NPF events.
2. The implementation of the Synthetic Minority Oversampling Technique (SMOTE) is utilized to rectify the issue of class imbalance.
3. Furthermore, various machine learning classifiers are trained and evaluated on the identical dataset to ensure a fair comparison.
4. Interpreting each model’s results using metrics, confusion matrices, and feature importance analysis.

**Input Environmental Features**

We feed the models a diverse set of metrological and chemical features known to influence NPF. Each feature is derived from our time-aligned environmental dataset/ Their roles in the NPF are as follows:

* **Solar Radiation Intensity (BestEstimate\_down\_short\_hemisp\_interp)** – Sunlight drives photochemistry (e.g. oxidizing SO₂ into H₂SO₄), which is the primary nucleating vapor in NPF. High solar flux (especially UV) thus strongly correlates with NPF onset.
* **Relative Humidity (rh\_ambient\_interp)** – Water vapor affects cluster stability and the “condensation sink.” Lower RH generally favors NPF, because high humidity means more water and wet aerosols that scavenge precursors, suppressing nucleation. (Conversely, very dry air has less vapor loss to existing particles.)
* **Turbulent Kinetic Energy (turbulent\_kinetic\_energy\_interp)** – Turbulence enhances vertical and horizontal mixing of air. Stronger turbulence can dilute local precursor concentrations but also distribute fresh precursors (and any newly formed clusters) more widely. In practice, moderate turbulence often aids NPF by entraining clean air aloft and mixing in sulfuric acid and organics.
* **Total Particle Surface Area (total\_SA\_conc\_smps\_interp)** – This measures the existing aerosol load (condensation sink). A large surface area of preexisting particles removes trace vapors, hindering new cluster formation. In other words, a high condensation sink suppresses NPF. (This is why very clean air or morning conditions often precede strong NPF.)
* **Wind Direction (wind\_direction\_interp)** – This is a proxy for regional air mass source. Certain wind sectors may bring pollution or biogenic vapors that influence NPF. For example, winds from industrial regions might supply SO₂ or NOₓ, whereas marine air might be clean and favor nucleation.
* **Wind Speed (wind\_speed\_interp)** – Faster winds enhance horizontal advection and mixing, which can dilute or spread precursors. Calm conditions can allow local buildup of vapors, while higher winds mix the boundary layer. The effect on NPF is context-dependent (e.g., very low wind may trap pollutants, while moderate wind brings new sources).
* **Boundary Layer Height (bl\_height\_1\_interp)** – A shallow boundary layer concentrates pollutants (and nucleating vapors) near the surface, often enhancing NPF potential. A high boundary layer dilutes concentrations. Thus, low BLH can coincide with stronger NPF signals due to higher local precursor levels.
* **Total Organics (total\_organics\_interp)** – Volatile organic compounds (VOCs) – especially low‐volatility or highly oxidized organics – contribute to particle growth and sometimes to nucleation. Laboratory and field studies show that oxidized organics can participate in clustering and especially in post-nucleation growth. High organic vapor load can thus sustain or intensify NPF.
* **Sulfate (sulfate\_interp)** – Sulfate aerosol (e.g., from H₂SO₄ condensation) is both a product of photochemistry and part of the condensation sink. Elevated sulfate indicates active SO₂ oxidation. While sulfate particles themselves don’t initiate nucleation, they reflect photochemical activity and contribute to the total sink.
* **Nitrate (nitrate\_interp)** – Typically as nitrate aerosol (often ammonium nitrate), this relates more to growth than to nucleation. For instance, rapid condensation of nitrate onto newly formed particles can accelerate their growth. Recent work shows fast growth rates when nitrate is abundant.
* **Sulfur Dioxide (so2\_interp)** – SO₂ is the gaseous precursor to sulfuric acid, the key NPF driver. Sunny days with available SO₂ are well-known to yield frequent NPF. Thus, SO₂ concentration is a direct input relating to nucleation potential.
* **Ambient Temperature (temperature\_ambient\_interp)** – Colder temperatures generally enhance NPF because they lower vapor pressures of nucleating compounds, stabilizing small clusters. In polluted conditions, studies show lower temperature correlates with higher nucleation rates (clusters of H₂SO₄-ammonia form more readily at low T).

**Early Warning Labeling of the NPF events**

Since the main motive was to find the occurrence of the NPF before it actually happened, or to simulate the practical forecasting scenario, we do not train the models to detect the NPF events as they occur but rather anticipate them. Following prior approaches, we label each timestep as a positive class (NPF = 1) if it occurs within a predefined time window(e.g., 3 hours = 36 time steps, with data points available after every 5 minutes) prior to the actual onset of the NPF event, which is denoted by *flag3 = 1*(just a name) in the dataset. All other timesteps are labeled as zero (NPF = 0). This labeling scheme mimics the real-time warning signals for atmospheric field campaigns.

**Addressing the Class Imbalance with SMOTE:**

Events classified as NPF constitute a minor portion of the overall dataset, presenting a challenge for machine learning models that often exhibit bias towards the majority class (non-NPF). To address this challenge, we have employed the Synthetic Minority Oversampling Technique (SMOTE) on the training dataset. SMOTE generates synthetic instances of the minority class by interpolating between existing examples, thereby facilitating a balanced training dataset while maintaining the distribution of the test set.

**Machine Learning Models**

We’ll evaluate five commonly used supervised learning algorithms for binary classification:

**Random Forest (RF)**

Random Forest is an ensemble learning method that builds multiple decision trees and combines their predictions to improve generalization and robustness. Each individual tree in the forest is trained on a bootstrap sample of the dataset, a randomly sampled subset with replacement, and each split in the tree considers only a random subset of input features. This dual randomness (in data and feature selection) helps to decorrelate the trees, reduce overfitting, and improve model stability.

At its core, each decision tree makes a series of binary decisions (e.g., “Is solar radiation > threshold?”) to assign a class label (NPF event or not). For a new input instance , let represent the prediction of the tree . The final output of the Random Forest is the class with the majority vote among all trees:

This voting mechanism reduces variance: while individual trees may overfit to noise or small patterns, aggregating their predictions yields a more robust overall estimate.

Random Forest is particularly suitable for this study because it naturally models nonlinear relationships and interactions between atmospheric variables, such as how elevated solar radiation and high SO₂ levels jointly increase the likelihood of NPF. Furthermore, it handles mixed feature types, missing values, and redundant information with minimal preprocessing. For example, different trees in the forest may split on solar radiation, relative humidity (RH), boundary layer height (BLH), or turbulent kinetic energy, allowing the ensemble to capture diverse patterns that precede NPF events.

Additionally, Random Forest models offer interpretability via feature importance scores, allowing us to assess which environmental factors (e.g., , solar radiation, RH) are most influential in predicting upcoming NPF events. This is particularly useful in atmospheric science applications, where domain knowledge and model transparency are both essential.

**eXtreme Gradient Boosting (XGBoost)**

XGBoost (eXtreme Gradient Boosting) is a powerful tree-based ensemble method designed to improve prediction accuracy through a process called **gradient boosting**. Unlike Random Forest, which builds multiple trees independently and aggregates their votes, XGBoost constructs decision trees **sequentially**. Each new tree focuses on correcting the mistakes made by the previous ones. This “boosting” approach helps the model learn complex patterns and reduce errors incrementally.

Where is a regression tree, and is the space of all possible trees. In classification tasks, this score is passed through a logistic function to yield a probability for the positive class (e.g., an NPF event). The model is trained to minimize the following objective:

Here, is the classification loss (e.g., logistic loss) for prediction , and is a regularization term that penalizes overly complex trees, helping to prevent overfitting.

n practical terms, XGBoost learns by focusing more attention on **hard-to-predict cases**, such as borderline or rare NPF events. For example, if the first tree misclassifies observations with elevated SO₂ but low solar radiation, the next tree will be optimized to better distinguish those cases. This iterative refinement process makes XGBoost especially powerful when the target class (like NPF events) is rare and when subtle combinations of features are predictive.

We selected XGBoost for this study due to its robustness on **tabular datasets with heterogeneous environmental inputs** (e.g., boundary layer height, , solar radiation, RH, etc.). It can naturally handle missing values, nonlinear interactions, and a variety of feature scales without extensive preprocessing. Additionally, the model includes built-in hyperparameter tuning options, such as learning rate, tree depth, and subsample ratios, which allow us to optimize performance specifically for early NPF detection.

XGBoost combines the strengths of decision trees and gradient-based optimization, making it an effective and interpretable model for capturing complex relationships between meteorological conditions and the likelihood of new particle formation.

**Light Gradient Boosting Machine (LightGBM)**

LightGBM is a high-performance gradient boosting algorithm tailored for efficiency and scalability on large, high-dimensional tabular datasets, making it well-suited for environmental data analysis. Like XGBoost, LightGBM builds an ensemble of decision trees sequentially, where each new tree is trained to correct the residual errors of the current model. However, its **key distinction lies in its tree-growing strategy**.

While traditional gradient boosting (including XGBoost) expands trees **level-wise**, growing all nodes at a given depth before moving deeper, LightGBM adopts a **leaf-wise (best-first) approach**. At each iteration, LightGBM identifies the leaf with the highest potential to reduce loss and expands that branch alone. This can produce deeper, more specialized trees that fit complex patterns more quickly.

Formally, LightGBM models the predicted output as:

Where each is a decision tree chosen to minimise a loss function ん, such as a binary cross-entropy for classification:

To accelerate training, LightGBM uses **histogram-based binning**, grouping continuous input values into discrete bins, significantly reducing computational cost while retaining predictive power.

In this study, LightGBM is selected because of its ability to:

* Efficiently process large datasets with many environmental predictors (e.g., SO₂, solar radiation, RH, wind speed).
* Automatically handle missing values and sparse features.
* Capture complex interactions in fewer iterations due to its leaf-wise growth strategy.

Practically, LightGBM enables faster training without major sacrifices in accuracy. For example, it might detect that a specific combination, such as high solar radiation, low relative humidity, and moderate , strongly indicates an impending NPF event. Its flexible architecture allows for real-time or near-real-time forecasting, which is essential for time-sensitive atmospheric applications.

An intuitive analogy for LightGBM is that of a physician prioritizing patients by severity: instead of treating all equally, it focuses resources on the “most uncertain” predictions (i.e., the leaves with the highest loss), improving those cases first. This targeted optimization makes it particularly effective for our NPF prediction task, where rare but important events must be captured with high recall and minimal delay.

**Support Vector Machines (SVM)**

Support Vector Machines (SVMs) are supervised learning algorithms used for binary classification tasks, particularly effective when the goal is to find a clear boundary between two classes. In the context of NPF (New Particle Formation) prediction, the SVM algorithm attempts to distinguish between time periods where NPF is likely to occur versus those where it is not, based on environmental variables such as solar radiation, relative humidity (RH), , boundary layer height (BLH), and others.

At the core of an SVM is the concept of a **hyperplane** — a decision boundary that best separates the two classes in the feature space. Among the infinitely many possible hyperplanes that could separate the data, SVM selects the one with the **maximum margin**, meaning the one that leaves the largest distance between the hyperplane and the nearest points from each class (called **support vectors**). This margin maximization leads to better generalization on unseen data and robustness to noise.

Mathematically, given training data where and , the SVM solves the following optimisation problem:

Here, is the weight vector defining the hyperplane, and is the bias term. This formulation assumes the data is linearly separable. For more complex cases, a **soft margin** SVM introduces slack variables to allow some misclassifications, and a **kernel function** is applied to implicitly map the data to a higher-dimensional space where linear separation may become possible.

In our study, we used the **Radial Basis Function (RBF) kernel**, which can capture **nonlinear relationships** among atmospheric variables. This is particularly important in modeling NPF, which depends on complex interactions, e.g., elevated SO₂ and high solar radiation might jointly indicate favorable conditions, but not always independently.

The **kernelized SVM decision function** becomes:

where:

* is the kernel function (e.g., Gaussian RBF),
* ​ are the learned coefficients,
* are the support vectors.

We are using SVM because:

* It is highly effective for binary classification tasks like NPF/non-NPF detection.
* It works well in high-dimensional feature spaces and is robust to overfitting when properly regularized.
* It can model **complex, nonlinear decision boundaries** using kernels, crucial for aerosol dynamics where environmental features interact in nonlinear ways.
* It focuses on **boundary-defining data points (support vectors)**, which makes it particularly good at identifying difficult-to-classify conditions.

However, SVMs are computationally more intensive compared to tree-based models and are sensitive to **feature scaling**. We normalized inputs and tuned hyperparameters (e.g., , ) to mitigate these issues.

**Multi-Layer Perception (MLP)**

The Multi-Layer Perceptron (MLP) is a type of feedforward artificial neural network that models complex, nonlinear relationships between input features and output classes. In the context of predicting New Particle Formation (NPF) events, the MLP learns patterns in environmental variables—such as solar radiation, relative humidity (RH), sulfur dioxide (SO₂), boundary layer height (BLH), and others—that typically precede particle bursts in the atmosphere.

An MLP is composed of an input layer, one or more **hidden layers** of artificial neurons, and an output layer. Each neuron in the network computes a weighted sum of its inputs and applies a nonlinear activation function such as ReLU (Rectified Linear Unit) or sigmoid. These nonlinearities allow the network to model highly complex and nonlinear interactions that may not be easily captured by traditional classifiers like SVM or Random Forest.

Mathematically, for an input vector , the MLP computes a series of transformations:

Where and are the weights and biases for the layer, and is the activation function. The final output is typically a probability indicating the likelihood of an NPF event occurring in the corresponding time window.

In our study, the MLP takes as input the same environmental features used in other models (e.g., RH, temperature, , solar radiation intensity, wind parameters, etc.). These features pass through multiple hidden layers, each of which progressively extracts higher-level representations of the input. For instance, the first hidden layer may detect basic interactions like “high radiation → likely photochemical activity,” while deeper layers combine these signals into more abstract predictors, such as “low RH and elevated SO₂ jointly suggest nucleation conditions.”

This **layered abstraction mechanism** makes MLPs particularly suitable for capturing **nonlinear and synergistic effects** among atmospheric variables, which are common in NPF events. Unlike decision trees that make discrete splits based on thresholds, MLPs use **continuous transformations**, allowing for smoother decision boundaries.

We included MLP in our algorithm comparison because:

* It is **flexible and powerful** for modeling **nonlinear relationships** and high-dimensional interactions among features.
* It can approximate any continuous function with sufficient depth and neurons (universal function approximation).
* It may detect **hidden patterns** in the data that are missed by tree-based or margin-based classifiers.

However, MLPs come with trade-offs:

* They require **larger training datasets** and are more sensitive to **hyperparameter tuning** (e.g., learning rate, architecture, activation functions).
* They are generally **less interpretable** than models like Random Forest or XGBoost.
* They can be **computationally expensive** to train and are prone to overfitting without regularization.

All models were implemented using scikit-learn, XGBoost, and LightGBM libraries with default parameters unless specified otherwise. For fairness, all models used the same training and test splits and were trained on the same balanced data after SMOTE.

**Evaluation Metrics**

We evaluated model performance on the test set using the following metrics:

* **Precision**: the proportion of predicted positives that are actual positives.
* **Recall**: the proportion of actual positives that are correctly identified.
* **F1 Score**: the harmonic mean of precision and recall.
* **ROC AUC**: the Area Under the Receiver Operating Characteristic Curve, which measures the model’s ability to discriminate between classes.

ROC AUC (Receiver Operating Characteristic – Area Under Curve) the calculation involves plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) across different thresholds and determining the area beneath this curve.

* **Confusion Matrix**: a 2×2 table summarizing correct and incorrect predictions.
* **Training and Inference Time**: recorded to compare computational efficiency.
* **Number of Tunable Parameters**: counted as a proxy for model complexity.

**Model Interpretation:**

For models that support feature importance (RF, XGB, LGBM), we computed the relative importance of each input variable. We also generated Partial Dependence Plots (PDPs) for the top six features to visualize how each predictor influences the model’s output while marginalizing over other variables. These plots helped confirm the role of known NPF precursors such as , solar radiation intensity, and relative humidity.

**Model Outputs**Each model produces a probability or class label indicating the likelihood of an NPF event in the near future. In practice, we threshold the output probability at 0.5 to assign a binary prediction. A prediction of “1” (NPF) means the model believes conditions now are within the typical precursor window of an upcoming event. This is effectively an *early warning*: it signals that, given current solar radiation, humidity, winds, etc., an NPF burst is likely within the next *x* hours. A “0” means no event is expected under present conditions. We evaluate performance by how well the model’s “1” predictions cover actual events (recall/sensitivity) and avoid false alarms (precision).