Postgraduate Certificate in AI in Weather Prediction

Machine Learning For Weather Forecasting

Supervised learning is the foundation of most weather-forecasting models that rely on historical observations paired with known outcomes. In this paradigm the algorithm learns a mapping from input variables – such as temperature, humidity, wind speed, and satellite-derived cloud indices – to a target variable, for example next-day precipitation amount. The quality of the training data, the choice of loss function, and the model architecture together determine how well the system can generalise to unseen weather events.

Unsupervised learning techniques are employed when labeled data are scarce or when the goal is to uncover hidden structures within atmospheric datasets. Clustering algorithms such as K-means or hierarchical clustering can group similar weather patterns, aiding in the identification of regimes like blocking highs or tropical cyclones. Dimensionality-reduction methods, for instance principal component analysis (PCA), compress high-dimensional satellite imagery into a smaller set of orthogonal components that retain the most variance, simplifying downstream modelling tasks.

Reinforcement learning (RL) differs from the previous two approaches by framing weather prediction as a sequential decision-making problem. An RL agent interacts with an environment – often a numerical weather prediction (NWP) model – and receives rewards based on forecast skill scores. Although still experimental, RL shows promise for adaptive model-bias correction, where the agent learns to adjust NWP outputs in real time to minimise forecast error.

Regression models predict continuous quantities such as temperature, wind speed, or accumulated rainfall. Linear regression offers a simple baseline, while more sophisticated algorithms – e.G., gradient boosting or random forest – capture non-linear relationships. A common application is the prediction of hourly surface temperature from a combination of synoptic variables, satellite-derived land-surface temperature, and soil moisture measurements.

Classification tasks assign discrete labels, such as “rain” versus “no rain” or “tornado-watch” versus “no watch”. Logistic regression, support vector machines, and deep neural networks are all viable classifiers. For instance, a convolutional neural network (CNN) can ingest multi-spectral satellite imagery and output a pixel-wise probability map of convective storm initiation.

Time-series analysis is central to weather forecasting because atmospheric variables evolve continuously over time. Autoregressive models (AR), moving-average models (MA), and their combinations (ARMA, ARIMA) capture temporal dependencies. Modern deep learning architectures, such as recurrent neural networks (RNNs), long short-term memory (LSTM) cells, and gated recurrent units (GRU), extend this capability by learning long-range temporal patterns from sequences of radar reflectivity, satellite brightness

temperatures, or model fields.

Neural networks are universal function approximators that have reshaped many aspects of meteorology. A simple feed-forward network with a few hidden layers can map surface observations to a forecast of 24-hour precipitation. More complex architectures tailor the network topology to the data modality: CNNs excel at extracting spatial features from gridded fields; RNNs handle sequential data; and transformer models, with their attention mechanisms, enable the integration of heterogeneous inputs across space and time.

Convolutional neural networks (CNNs) process data that possess a grid-like topology, such as satellite images or NWP output fields. By applying learned filters, CNNs detect edges, textures, and higher-order patterns that correspond to atmospheric phenomena. A typical example is the detection of mesoscale convective systems from infrared satellite imagery: The network learns to recognise the characteristic cold cloud-top signatures and spatial organisation that precede heavy rainfall.

Recurrent neural networks (RNNs) maintain an internal state that evolves with each time step, allowing the model to remember past information. However, vanilla RNNs suffer from vanishing gradients, which limits their ability to capture long-range dependencies. LSTM and GRU cells mitigate this issue through gated mechanisms that regulate the flow of information. An operational use case involves feeding a sequence of hourly radar reflectivity fields into an LSTM to predict the location and intensity of a thunderstorm 6 hours ahead.

Attention mechanisms, popularised by transformer models, compute a weighted sum of input representations, enabling the model to focus on the most relevant parts of the data. In weather forecasting, attention can be used to dynamically select which atmospheric layers, geographic regions, or satellite channels contribute most to a given prediction. For example, a transformer that ingests multi-level NWP variables may allocate higher attention weights to the 850 hPa level when forecasting surface temperature anomalies.

Ensemble methods combine multiple base learners to improve predictive performance and robustness. Bagging (bootstrap aggregating) creates diverse models by training each learner on a different random subset of the data; random forest is a classic bagging implementation that also randomises feature selection at each split. Boosting iteratively focuses on the errors of previous learners; algorithms such as XGBoost and LightGBM have become standard tools for post-processing NWP outputs because they efficiently handle large feature sets and can produce calibrated probability estimates.

Hyperparameter tuning involves searching for the optimal configuration of model parameters that are not learned during training, such as learning rate, number of layers, or tree depth. Grid search, random search, and Bayesian optimisation are common strategies. In the context of weather forecasting, careful tuning is essential to avoid overfitting to a limited historical period, which could degrade performance during extreme events that lie outside the training distribution.

Cross-validation provides a systematic way to assess model generalisation by partitioning the data into training and validation folds. For time-dependent data, a simple random split can leak future information into the training set; therefore, techniques like rolling-origin or blocked cross-validation are preferred. These methods preserve temporal ordering, ensuring that the validation set always follows the training set chronologically, which mirrors the real-world forecasting scenario.

Overfitting occurs when a model captures noise instead of the underlying signal, leading to poor performance on unseen data. In weather applications, overfitting can manifest as a model that predicts the exact location of a past storm but fails to generalise to a new storm track. Regularisation techniques – L1/L2 penalties, dropout, early stopping – and the use of larger, more diverse training datasets are standard remedies.

Underfitting is the opposite problem, where a model is too simple to represent the complexity of atmospheric processes. A linear regression trained on a single surface temperature variable may underfit the relationship between upper-air dynamics and surface extremes, resulting in high bias. Adding non-linear features, increasing model capacity, or incorporating additional data sources can alleviate underfitting.

Bias-variance trade-off summarises the tension between model simplicity (high bias, low variance) and model complexity (low bias, high variance). Selecting the appropriate balance is critical for operational forecasting, where reliability and skill must be maintained across a wide range of weather regimes. Ensemble methods, such as bagged trees, often reduce variance without substantially increasing bias, making them attractive for post-processing pipelines.

Feature engineering refers to the process of creating informative input variables from raw data. In meteorology, this may involve computing derived quantities such as convective available potential energy (CAPE), wind shear, or moisture flux convergence. Temporal features (e.G., Lagged values, moving averages) and spatial features (e.G., Distance to the coast, elevation) enrich the dataset and enable the model to capture physically relevant relationships.

Dimensionality reduction techniques compress high-dimensional data while preserving essential information. PCA, independent component analysis (ICA), and autoencoders are common choices. For example, a satellite sensor that provides 12 spectral channels can be reduced to a handful of principal components that capture the majority of variance, speeding up training and reducing the risk of overfitting.

Autoencoders are neural networks trained to reconstruct their input, forcing the intermediate bottleneck layer to learn a compact representation. Variational autoencoders (VAEs) add a probabilistic element, enabling the generation of synthetic atmospheric fields that respect learned statistical properties. These synthetic samples can augment training sets, particularly for rare events like severe hailstorms.

Data assimilation combines observations with model forecasts to produce an optimal estimate of the atmospheric state. Classical techniques such as three-dimensional variational assimilation (3D-Var) and

four-dimensional variational assimilation (4D-Var) are integral to NWP systems. Machine-learning-based assimilation methods, including neural-network approximations of the analysis step, aim to reduce computational cost while preserving accuracy.

Reanalysis datasets are retrospective reconstructions of the atmosphere that blend observations with a consistent NWP model. Products like ERA5, JRA-55, and NCEP-CFSR provide gridded fields of temperature, wind, humidity, and many derived quantities over decades. They serve as a primary source of training data for ML models, offering a homogeneous, quality-controlled record that can be sliced into training, validation, and test periods.

Satellite imagery delivers frequent, global observations of cloud cover, water vapour, sea-surface temperature, and other radiative properties. Multi-spectral and hyperspectral sensors (e.G., GOES-16, Himawari-8, Sentinel-3) generate massive data streams that are ideal for deep-learning models. Pre-processing steps include radiometric calibration, cloud masking, and georeferencing before the images can be fed into a CNN or transformer.

Radar data provide high-resolution, near-real-time measurements of precipitation intensity and motion. Reflectivity fields are routinely used as inputs for short-range (Numerical weather prediction (NWP) is the physics-based simulation of the atmosphere using the Navier-Stokes equations and parameterisations for processes such as radiation, convection, and surface fluxes. Modern NWP models (e.G., WRF, ICON, UM) produce forecasts at various spatial resolutions, from global 0.25° Grids to convection-permitting 1-km meshes. ML can be applied to NWP outputs in several ways: As a direct surrogate model, as a post-processor to correct systematic biases, or as a component within a hybrid system that blends physics-based and data-driven predictions.

Model output statistics (MOS) are statistical techniques that translate raw NWP fields into calibrated forecasts for specific variables and locations. Traditional MOS uses linear regression or logistic regression, but contemporary MOS pipelines often employ machine-learning algorithms such as gradient-boosted trees. An example MOS application is the generation of site-specific temperature forecasts from a coarse-resolution global model, where the ML model learns location-specific correction factors.

Bias correction adjusts systematic errors in model forecasts. Simple methods include mean bias subtraction or scaling; more advanced techniques use regression, quantile mapping, or neural networks to correct both the mean and distributional shape. For precipitation, quantile mapping is popular because it preserves the observed probability distribution while aligning the forecast quantiles with the observed quantiles.

Post-processing refers to any manipulation of raw model output before dissemination. Machine-learning-based post-processing can produce probabilistic forecasts, categorical probabilities (e.G., “Probability of exceeding 10 mm rain”), or deterministic point forecasts. The post-processing stage is crucial for operational centres because it bridges the gap between the raw, often biased model fields and the end-user requirements for accuracy and reliability.

Probabilistic forecasting yields a full probability distribution or a set of quantiles rather than a single deterministic value. This approach is valuable for risk-aware decision making, such as flood management or aviation planning. Ensemble methods, Bayesian neural networks, and quantile regression forests are all capable of producing probabilistic outputs. A typical product is a 24-hour precipitation forecast expressed as the probability of exceeding thresholds of 1 mm, 5 mm, and 20 mm.

Ensemble forecasting generates multiple forecasts by perturbing initial conditions, model physics, or parameters, thereby sampling the uncertainty in the atmospheric state. The resulting ensemble spread provides a measure of forecast confidence. Machine learning can be employed to calibrate ensemble forecasts, reducing dispersion errors and improving reliability. Techniques such as Bayesian model averaging or neural network calibration functions transform raw ensemble members into well-calibrated probability forecasts.

Quantile regression directly predicts specific quantiles (e.G., 0.1, 0.5, 0.9) Of the target distribution. Unlike mean regression, quantile regression is robust to skewed distributions, which are common in precipitation where many zero observations are mixed with heavy tails. Gradient-boosted quantile regression trees and deep quantile regression networks have been successfully applied to produce calibrated precipitation forecasts that capture both the median and extreme events.

Probabilistic calibration assesses how well forecast probabilities align with observed frequencies. A perfectly calibrated forecast would have, for example, a 30 % probability of rain on days that actually experience rain 30 % of the time. Calibration techniques include isotonic regression, Platt scaling, and ensemble model output statistics (EMOS). Proper calibration is essential for users who rely on probability thresholds for decision making.

Verification metrics quantify forecast quality. Common deterministic metrics include root-mean-square error (RMSE) and mean absolute error (MAE). For probabilistic forecasts, the Brier score, continuous ranked probability score (CRPS), and reliability diagrams are standard. Skill scores, such as the Brier skill score or the equitable threat score, compare a model's performance against a reference, often climatology or a persistence forecast.

Deterministic vs probabilistic forecasting distinguishes between a single-valued prediction (deterministic) and a distributional prediction (probabilistic). Deterministic forecasts are easier to communicate but can be misleading when the underlying uncertainty is large. Probabilistic forecasts convey uncertainty explicitly, enabling users to perform cost–loss analyses and make informed risk assessments.

Climatology provides a baseline forecast derived from historical averages or percentiles for a given location and time of year. In verification, climatology often serves as the reference model for skill-score calculations. Machine-learning models can be trained to outperform climatology by learning the deviations caused by dynamic atmospheric processes.

Reforecast (or hindcast) datasets consist of model runs performed retrospectively over a long historical period, using the same model configuration as the operational system. Reforecasts are valuable for training and validating ML models because they provide a consistent set of forecasts and observations, enabling robust skill assessment across multiple years and seasons.

Preprocessing steps prepare raw data for model ingestion. Common tasks include missing-value imputation, normalization, scaling, and outlier detection. For atmospheric variables, physical constraints guide preprocessing: For example, specific humidity must be non-negative, and wind speed is often transformed using a log-scale to reduce skewness. Proper preprocessing improves convergence and model stability.

Missing data imputation addresses gaps caused by sensor outages, satellite swaths, or quality-control flags. Simple approaches fill gaps with climatological means or nearest-neighbour values; more sophisticated methods employ spatio-temporal interpolation, Gaussian process regression, or deep-learning inpainting techniques that respect the underlying physical relationships.

Normalization rescales variables to a common range, typically $[0, 1]$ or $[-1, 1]$, facilitating gradient-based optimisation. Standardisation (z-score) subtracts the mean and divides by the standard deviation, preserving the distribution shape while centering the data. For variables with heavy tails, a log-transform before normalisation can improve model performance.

Scaling is particularly important when combining heterogeneous features such as temperature (in Kelvin) and wind speed (in $\mathrm{m\,s^{-1}}$). Scaling ensures that the learning algorithm does not implicitly prioritise variables with larger numeric ranges. In practice, scaling parameters are computed on the training set and applied unchanged to validation and test sets to avoid data leakage.

Outlier detection identifies anomalous observations that may stem from instrument errors, data-entry mistakes, or rare atmospheric events. Robust statistical methods (e.G., Median absolute deviation) and machine-learning methods (e.G., Isolation forest) can flag outliers. Depending on the context, outliers may be removed, corrected, or retained to preserve extreme-event information.

Data augmentation artificially expands the training set by applying transformations that preserve the physical meaning of the data. For satellite images, rotations, flips, and slight spatial jittering are permissible because atmospheric patterns are isotropic over small angular displacements. Augmentation helps mitigate overfitting, especially when the original dataset contains few examples of severe weather.

Transfer learning leverages knowledge acquired from one task or domain to improve performance on another, often with limited data. A CNN pretrained on global cloud-classification tasks can be fine-tuned on a regional hail-prediction problem, reducing the required training data and accelerating convergence. Domain adaptation techniques further align the feature distributions between source and target domains, handling differences in sensor characteristics or climate regimes.

Explainability addresses the “black-box” nature of many machine-learning models. Techniques such as SHAP values (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) assign importance scores to input features for individual predictions. In weather forecasting, explainability can reveal, for example, that a high SHAP value for CAPE contributed strongly to a predicted thunderstorm, providing confidence to forecasters.

Uncertainty quantification assesses the confidence of model predictions. Bayesian neural networks place probability distributions over network weights, yielding predictive intervals that reflect epistemic uncertainty. Monte-Carlo dropout approximates Bayesian inference by randomly dropping units at inference time, producing an ensemble of stochastic forward passes. Distinguishing between aleatoric (intrinsic) and epistemic (model) uncertainty helps users interpret forecast reliability.

Bayesian methods incorporate prior knowledge and update beliefs in light of new data. In the context of weather forecasting, Bayesian hierarchical models can blend observations, reanalysis, and NWP outputs, producing probabilistic estimates that respect physical constraints. Gaussian process regression offers a non-parametric Bayesian approach that yields closed-form predictive distributions, useful for small-sample problems like localized extreme-event modelling.

Gaussian processes define a prior over functions, characterised by a mean function and a covariance kernel that encodes assumptions about smoothness and length scales. When applied to meteorological time series, kernels can be designed to capture periodicity (e.G., Diurnal cycles) and long-range dependence. Although computationally intensive for large datasets, sparse approximations enable their use in regional forecasting applications.

Stochastic processes model the evolution of random variables over time. Markov chains, hidden Markov models (HMMs), and stochastic differential equations provide frameworks for representing weather dynamics in a probabilistic manner. HMMs have been employed to classify weather regimes based on sequences of atmospheric indices, while stochastic differential equations underpin ensemble Kalman filter formulations.

Kalman filter is a recursive estimator that combines a forecast from a dynamical model with observations to produce an optimal state estimate. The ensemble Kalman filter (EnKF) extends this concept to high-dimensional systems by representing the error covariance with an ensemble of model states. Machine-learning-enhanced filters replace the linear observation operator with a learned non-linear mapping, improving the assimilation of complex satellite products.

Big data challenges arise from the sheer volume, velocity, and variety of meteorological observations. Satellite constellations, radar networks, and dense surface sensor arrays generate petabytes of data annually. Distributed computing frameworks (e.G., Apache Spark) and cloud-based storage solutions enable the ingestion, processing, and training of ML models at scale. Efficient data pipelines are essential for turning raw observations into ready-to-train datasets.

Cloud computing offers elastic resources that can be provisioned on demand, facilitating large-scale model training and hyperparameter optimisation. GPU-accelerated instances dramatically reduce training time for deep neural networks, while serverless functions can host inference services for real-time forecasting. Cost-effective cloud workflows often combine spot instances for batch training with reserved instances for high-availability serving.

GPU acceleration leverages parallel processing units to speed up matrix operations central to deep learning. Modern frameworks such as TensorFlow and PyTorch automatically allocate tensors to GPU memory, enabling rapid training of CNNs on high-resolution satellite imagery. For operational forecasting, GPU inference can deliver sub-second predictions, meeting the tight latency requirements of nowcasting services.

Model deployment translates a trained model into a production-ready service. Containerisation technologies (e.G., Docker) package the model, its dependencies, and a lightweight web server into a portable unit. APIs expose prediction endpoints that ingest real-time observations and return forecasts. Continuous integration pipelines automate testing, monitoring, and updating of the deployed model, ensuring reliability in an operational environment.

Real-time forecasting imposes stringent latency constraints: Data ingestion, preprocessing, inference, and post-processing must all occur within minutes. Stream-processing architectures ingest radar and satellite feeds, apply feature extraction, and feed the result into a pre-trained neural network that outputs a nowcast. Techniques such as model quantisation (reducing precision from 32-bit to 8-bit) and pruning (removing redundant weights) further accelerate inference while preserving accuracy.

Operational constraints include computational budget, data latency, and regulatory compliance. Forecast centres must balance the desire for high-resolution, data-intensive models with the need for timely delivery. Model complexity is often limited by available hardware, and data pipelines must be robust to missing or delayed observations. Understanding these constraints guides the selection of appropriate ML techniques and system architectures.

Ethical considerations arise when automated forecasts influence public safety or economic decisions. Biases in training data—such as under-representation of certain geographic regions—can lead to inequitable forecast quality. Transparency, explainability, and rigorous validation are essential to maintain trust. Moreover, data privacy regulations must be respected when incorporating crowdsourced observations or mobile-sensor data.

Reproducibility ensures that scientific results can be independently verified. In ML for weather, reproducibility demands version-controlled code, documented data preprocessing steps, fixed random seeds, and archived model weights. Sharing model artefacts through repositories (e.G., Zenodo, GitHub) and adhering to FAIR (Findable, Accessible, Interoperable, Reusable) principles facilitates collaboration and accelerates progress.

Open data initiatives, such as the European Centre for Medium-Range Weather Forecasts (ECMWF) Copernicus programme, provide free access to satellite, radar, and reanalysis products. Leveraging open datasets reduces barriers to entry for research groups and enables the community to benchmark ML models on common testbeds. When combined with open-source ML libraries, these resources foster an ecosystem of collaborative development.

FAIR principles guide the management of scientific data. Making datasets findable through persistent identifiers, accessible via standard protocols, interoperable by using common formats (e.G., NetCDF, Zarr), and reusable through clear licensing and metadata, enhances the value of weather data for ML research. Compliance with FAIR standards streamlines the creation of training pipelines and facilitates cross-institutional collaborations.

Feature selection reduces dimensionality by retaining only the most informative variables. Techniques such as recursive feature elimination, mutual information, and embedded methods (e.G., Feature importance from random forests) identify a subset of predictors that maximises skill while minimising computational cost. In practice, a reduced set of physically relevant features—such as geopotential height at 500 hPa, surface temperature, and humidity—often yields comparable performance to a full feature set.

Regularisation penalises model complexity to prevent overfitting. L1 regularisation (lasso) promotes sparsity by driving coefficients toward zero, effectively performing feature selection. L2 regularisation (ridge) shrinks coefficients uniformly, improving numerical stability. In deep learning, weight decay implements L2 regularisation, while dropout randomly disables neurons during training, fostering robustness.

Dropout is a stochastic regularisation technique that temporarily removes a fraction of neurons during each training iteration. By preventing co-adaptation of features, dropout encourages the network to learn redundant, distributed representations, which enhances generalisation. At inference time, dropout is typically disabled, or Monte-Carlo dropout is used to estimate predictive uncertainty.

Early stopping monitors validation loss during training and halts the optimisation when performance ceases to improve, avoiding overfitting to the training data. In weather forecasting, early stopping is often combined with a learning-rate scheduler that reduces the step size after a plateau, ensuring fine-grained convergence on complex loss landscapes.

Learning rate controls the magnitude of weight updates during gradient descent. A learning-rate schedule —such as step decay, cosine annealing, or adaptive methods (Adam, RMSprop)—helps navigate the loss surface efficiently. Selecting an appropriate learning rate is critical; too large a value can cause divergence, while too small a value leads to excessively long training times.

Batch size determines the number of samples processed before the model parameters are updated. Larger batches provide smoother gradient estimates but require more memory, while smaller batches introduce noise that can help escape local minima. In meteorological applications, batch sizes are often limited by the

high dimensionality of gridded inputs (e.G., Multi-channel satellite images).

Loss functions quantify the discrepancy between predictions and observations. For regression, common choices include mean squared error (MSE) and mean absolute error (MAE). For probabilistic forecasting, the CRPS and negative log-likelihood are preferred because they evaluate the entire predictive distribution. Classification tasks often use cross-entropy loss, possibly weighted to address class imbalance (e.G., Rare severe-weather events).

Class imbalance is a pervasive issue when forecasting extreme events, where the number of positive cases (e.G., Tornadoes) is orders of magnitude smaller than negatives. Strategies to mitigate imbalance include resampling (oversampling the minority class, undersampling the majority), synthetic data generation (SMOTE), and loss weighting (assigning higher penalty to misclassifying the minority class). Proper handling of imbalance improves detection rates without inflating false alarms.

Calibration aligns forecast probabilities with observed frequencies. A calibrated model will, for example, produce a 70 % probability of rain that indeed rains on 70 % of such days. Calibration techniques such as isotonic regression, Platt scaling, and Bayesian model averaging transform raw model outputs into well-calibrated probabilities, enhancing decision-maker confidence.

Reliability diagrams visualise calibration by plotting observed frequencies against forecast probabilities. A perfectly reliable system lies on the diagonal. Deviations indicate over- or under-confidence, guiding post-processing adjustments. In practice, reliability diagrams are generated for each forecast lead time to assess how calibration degrades with increasing horizon.

Skill scores compare model performance against a reference forecast. The Brier skill score (BSS) measures improvement over climatology for binary events, while the equitable threat score (ETS) assesses categorical forecasts while accounting for hits due to chance. Skill scores are essential for communicating the added value of ML-enhanced forecasts to stakeholders.

Spatial verification evaluates forecast quality over a domain rather than at a single point. Metrics such as the Fractions Skill Score (FSS) and the Object-Based Diagnostic Evaluation (OBDE) compare spatial patterns of predicted and observed precipitation fields, accounting for location errors and displacement. Spatial verification is particularly relevant for high-resolution convective forecasts, where exact placement is challenging.

Temporal verification assesses how well the model captures the evolution of a variable over time. Time-series metrics (e.G., Temporal correlation, lag-1 autocorrelation) and skill scores for lead-time series (e.G., The mean absolute skill score) evaluate the consistency of forecasts across multiple horizons. Temporal verification highlights whether a model systematically under- or over-estimates the rate of change.

Model interpretability goes beyond explainability by providing a holistic understanding of how the model functions. Techniques such as saliency maps, which highlight regions of an input image that most influence

the output, help forecasters see which cloud structures drive a severe-weather prediction. Layer-wise relevance propagation (LRP) offers a more detailed decomposition of contributions across network layers.

Hybrid modelling combines physics-based NWP with data-driven ML components. One common architecture couples a conventional NWP core with a neural-network bias-correction module that ingests the NWP fields and outputs a corrected forecast. Another approach integrates ML directly into the NWP time-step, replacing computationally expensive parameterisations (e.G., Convection) with surrogate neural networks that approximate the same physical processes at a fraction of the cost.

Surrogate modelling creates a computationally cheap approximation of a complex model. In weather forecasting, surrogate models can emulate high-resolution NWP runs, enabling rapid ensemble generation for uncertainty quantification. Training a deep neural network on a large set of NWP simulations yields a surrogate that reproduces the original model's output with sub-second latency, suitable for ensemble-based decision support.

Physics-informed neural networks (PINNs) embed differential equations into the loss function, forcing the network to satisfy known physical constraints (e.G., Conservation of mass). By penalising violations of the governing equations, PINNs produce predictions that are both data-consistent and physically plausible. Applications include learning sub-grid scale fluxes while respecting energy balance, or estimating atmospheric state variables from sparse observations.

Domain adaptation addresses the shift between training and deployment environments. For example, a model trained on European satellite data may need to be applied to an Asian satellite with different spectral characteristics. Techniques such as adversarial training align feature distributions across domains, reducing performance degradation when the model encounters new sensor modalities or climate regimes.

Transfer learning (repeated for emphasis) enables leveraging large-scale pretraining on global datasets to fine-tune on local, high-resolution tasks. A CNN pretrained on worldwide cloud classification can be adapted to predict hailstorm likelihood in a specific watershed, requiring far fewer local training samples while retaining the learned low-level visual features.

Model compression reduces the memory footprint and inference latency of large neural networks. Pruning removes redundant connections, while quantisation reduces numerical precision. Knowledge distillation transfers the behaviour of a large "teacher" model to a smaller "student" model, preserving performance while enabling deployment on edge devices or low-power servers.

Edge computing brings inference closer to the data source, such as deploying a lightweight storm-prediction model on a weather station or an unmanned aerial vehicle. By processing data locally, edge computing reduces latency and bandwidth requirements, which is crucial for real-time alerts in remote or bandwidth-constrained regions.

Model monitoring tracks performance metrics after deployment, detecting drift, degradation, or anomalies.

Continuous evaluation against incoming observations ensures that the model remains reliable over time. Alerting mechanisms can trigger retraining or rollback procedures when forecast skill falls below predefined thresholds.

Retraining pipelines automate the periodic update of models using the latest data. Automated workflows ingest new observations, update the training set, re-run hyperparameter optimisation, and redeploy the refreshed model. This continuous learning approach maintains relevance in a changing climate and adapts to sensor upgrades or new data sources.

Cost-loss analysis quantifies the economic trade-off between taking protective action based on a forecast and the potential loss from an adverse event. By integrating forecast probabilities with user-specific cost-loss ratios, decision makers can derive optimal thresholds for issuing warnings. ML-driven probabilistic forecasts enable more nuanced cost-loss assessments than deterministic forecasts.

Ensemble Kalman filter (EnKF) is a data-assimilation method that updates an ensemble of model states using observations, accounting for both model and observation error statistics. Machine-learning-enhanced EnKFs replace the linear observation operator with a neural network that maps raw satellite radiances to physical variables, improving the assimilation of complex sensor data.

Stochastic parameterisation introduces random perturbations to represent sub-grid processes that are not explicitly resolved. By sampling from probability distributions for parameters such as cloud-droplet number concentration, stochastic schemes generate ensemble spread that reflects model uncertainty. ML can learn these probability distributions from high-resolution simulations, informing more realistic stochastic parameterisations.

Hybrid ensemble-ML systems blend traditional ensemble forecasts with machine-learning post-processing. For instance, each ensemble member of a NWP model can be bias-corrected by a neural network trained on historical errors, after which the corrected members are combined to produce a calibrated probabilistic forecast. This approach leverages the physical diversity of the ensemble while reducing systematic errors.

Quantile mapping aligns the distribution of model forecasts with that of observations by matching quantiles. This non-parametric technique corrects both mean bias and distribution shape, making it especially useful for variables with skewed distributions like precipitation. When applied to each lead time separately, quantile mapping can significantly improve forecast reliability across the full forecast horizon.

Probabilistic neural networks output parameters of a probability distribution (e.G., Mean and variance) rather than a single point estimate. For regression tasks, the network may predict the parameters of a Gaussian distribution, allowing the calculation of prediction intervals. This approach naturally incorporates aleatoric uncertainty, which is valuable for high-variability variables such as wind speed.

Mixture density networks extend probabilistic neural networks by modelling the output as a mixture of several distributions (e.G., A Gaussian mixture). This enables the representation of multimodal predictive

distributions, which can occur in weather when multiple distinct outcomes are plausible (e.G., Rain versus no rain). Training involves maximising the likelihood of the observed data under the mixture model.