CHAPTER 17

Improving thunderstorm prediction with neural networks using numerical weather and satellite data: a novel data fusion and validation approach

Hadi Mahdipour[1], Alireza Sharifi[2], and Dariush Abbasi-Moghadam[3]

[1]Centre for Cleantech and Biomass Resource Efficiency Centre at the Agricultural University of Plovdiv, Plovdiv, Bulgaria, [2]Department of Geomatics and Surveying Engineering, Faculty of Civil Engineering, Shahid Rajaee Teacher Training University, Tehran, Iran, [3]Department of Electrical Engineering, Shahid Bahonar University of Kerman, Kerman, Iran

17.1 Introduction

Thunderstorms present a significant hazard to aviation, with elements such as turbulence, strong winds, lightning, and hail posing risks to aircraft safety. These storms are particularly prevalent in Europe during the summer months, coinciding with peak air traffic, and frequently disrupt air traffic management (ATM) operations, causing widespread delays across the network. In 2018 alone, weather-related delays in Europe were estimated to have cost approximately €0.48 billion (Jardines et al., 2020, 2021). The ability to predict thunderstorms allows traffic managers to plan effectively for adverse weather conditions, thereby enhancing ATM operations and reducing associated costs.

Despite its importance, accurate thunderstorm forecasting remains a complex challenge. While certain meteorological conditions are known to contribute to storm development, predicting the precise timing and location of convective activity remains difficult. Short-term forecasting, or nowcasting, is commonly used for thunderstorm prediction within a timeframe of 1–3 hours. This technique relies on real-time data from sources like Doppler radars and satellites to identify potential storm formations (Wilson et al., 1998). However, nowcasting becomes less accurate over time. Numerical weather prediction (NWP) models, bolstered by advancements in weather science and computational capabilities, have emerged as a promising alternative. These models offer improved reliability and accuracy for severe weather forecasting, enabling proactive decision-making to mitigate disruptions and reduce costs (Jardines et al., 2020, 2021).

Utilizing Earth Observation Data in Reaching Sustainable Development Goals. DOI: https://doi.org/10.1016/B978-0-443-30204-6.00016-4

The integration of artificial intelligence (AI) techniques with a deep understanding of atmospheric physics has shown great potential for improving climate predictions, including thunderstorms (Naik et al., 2024; Pandey et al., 2023; Rodriguez-Rodriguez et al., 2021; Zhou et al., 2020, 2022). Machine learning (ML) methods, combined with data from satellites, Doppler radars, and NWP models, have successfully supported short-term storm predictions within a 24-hour timeframe (Li et al., 2019; Mecikalski et al., 2015). More advanced models, such as neural networks (NNs) and deep NNs, have extended the predictive horizon, improving forecasts for longer periods (Collins & Tissot, 2015; He & Loboda, 2020; Šaur, 2017; Simon et al., 2018). Additionally, convolutional neural newtorks CNNs have enhanced NWP-based forecasts of convective weather, offering timeframes ranging from 6 hours to as much as three days (Zhou et al., 2019). Although these methods have demonstrated success, existing models often lack the high temporal and spatial resolution required for pretactical ATM operations. Current high-resolution forecasting tools typically rely on physics-based models rather than ML, which limits their scalability and geographical scope (Baldauf et al., 2011; Spiridonov et al., 2020). Jardines et al. (2020, 2021) utilized multilayer perceptron neural networks (MLP NNs) in combination with NWP data to address this gap. Although this approach proved effective for pretactical ATM operations, it faced challenges related to significant computational complexity due to the large volume of input data. To mitigate this, the training dataset size was reduced to 50% of the available data, despite studies indicating that utilizing around 70% generally produces more accurate results. Further limitations in Jardines et al.'s methodology include the lack of a pre-trained initial model, the absence of k-fold validation, and reliance on a dataset representing only one month, which restricts model generalizability. These shortcomings underscore the need for further improvements, such as refining data subset selection, employing unbiased initial conditions, and incorporating validation strategies that enhance model performance (Beheshti et al., 2019). Jardines et al. (2024) further enhanced their model by integrating CNNs to extract spatial features from weather data, leveraging these features to improve the accuracy of storm predictions. This approach demonstrated the potential of combining traditional NWP methods with advanced ML techniques, particularly CNNs, to refine thunderstorm forecasting. However, their work also highlighted certain limitations, such as the absence of k-fold validation, reliance on a dataset from only one month, and issues related to computational complexity. These limitations underscore the need for additional refinements to improve model generalizability and efficiency.

On the other hand, Data fusion has emerged as a critical method for optimizing the processing and utilization of large datasets by combining diverse pieces of information to create more accurate and compatible datasets (Nazarko, 2002). Its applications span various fields, including military target tracking and recognition, machine vision, robotics, medical imaging, and climate prediction (Mahdipour et al., 2016; Nazarko, 2002). The objective of data fusion techniques is to integrate temporal information, synthesize dissimilar data, or combine similar data from different sources to enhance reliability and accuracy. Inherent challenges, such as inaccuracies and uncertainties in data due to sensing errors or the digitization of real-world

phenomena, further highlight the importance of robust fusion methods (Mowrer & Congalton, 2003). By handling such uncertainties, these techniques mitigate the effects of outliers and noise, ensuring more reliable outputs (Mahdipour et al., 2016, 2024).

As mentioned, satellite data stands out as a powerful tool for continuous monitoring of Earth features, hazard management, change detection, and various other applications, as indicated by numerous studies (Abaspur Kazerouni et al., 2021; Ghaderizadeh et al., 2022; Kosari et al., 2020; Mahdipour et al., 2020a, 2020b; Mahdipour et al., 2024; Mohammadi et al., 2021; Sharifi & Amini, 2015; Sharifi et al., 2015, 2016, 2022; Sharifi, 2021; Tariq et al., 2022; Zamani et al., 2022). Studies have demonstrated its effectiveness in enhancing storm prediction accuracy through its integration with ML and NWP models (Jardines et al., 2020, 2021, 2024). Leveraging these insights, the fusion approach adopted in this paper combines information from diverse sources to address data redundancies and improve thunderstorm prediction performance.

Building on the foundation laid by Jardines et al. (2020, 2021, 2024), this study addresses the key limitations of their storm prediction model by introducing several methodological advancements aimed at reducing computational complexity and improving prediction accuracy. The primary innovation involves applying a data fusion technique to preprocess the input data by minimizing redundancies and extracting essential information. This step is critical, as the vast and often repetitive input data from NWP models significantly leads to computational inefficiencies. By employing a fusion method, the data is streamlined before being fed into the fixed-structure NN, thereby reducing the computational burden without sacrificing predictive capability. Additionally, this study proposes further enhancements to optimize model performance. First, a more rigorous selection of training, validation, and testing subsets ensures balanced data representation, a factor crucial for effective generalization. Second, initializing NN responses without preassigned weights helps to avoid biases during the early training stages. This unbiased starting point is particularly advantageous for minimizing the risk of converging on suboptimal solutions. Third, the application of nonlinear normalization techniques, such as logarithmic and exponential transformations, to specific input features addresses disparities in data scale and improves the learning efficiency of the network. These techniques enhance the model's ability to capture subtle patterns and relationships within the data that linear normalization methods might overlook. Finally, a k-fold cross-validation strategy is incorporated to enhance the robustness of the proposed method further. This strategy, crucial when working with limited datasets such as the one-month data used in this study, ensures the model's performance is consistently evaluated across different subsets of the data. By rotating training and validation sets through k iterations, this method not only reduces the risk of overfitting but also provides a more comprehensive assessment of the model's predictive capabilities. Together, these innovations aim to strike a balance between reducing computational demands and maximizing the accuracy and reliability of thunderstorm forecasting, addressing the challenges that limited

previous methods. By combining advancements in data preprocessing, feature scaling, and model validation, this proposed approach aims to establish a new benchmark in computational efficiency and predictive accuracy for pre-tactical ATM.

The rest of this paper is organized as follows: Section 17.2 describes the dataset and outlines the proposed methods and extensions. Simulation results and comparisons with existing methods are presented in Section 17.3, followed by the conclusions in Section 17.4.

17.2 Materials and methods

This section explains the dataset, the architecture of the NN, data preparation, and the proposed enhancements aimed at improving the prediction of thunderstorms. The improvements focus on reducing computational complexity while maximizing prediction accuracy.

17.2.1 Data set

The data used for this research comes from previous studies (Jardines et al., 2020, 2021) and is based on meteorological forecasts and satellite observations from June 2018. The geographical domain spans large parts of Western Europe and northern Africa (see Fig. 17.1).

EUMETSAT, the European organization that operates weather satellites, provides the observational satellite data used in this study. EUMETSAT's satellites are equipped with sophisticated instruments that collect data critical for understanding atmospheric conditions. This information is fundamental to the development of accurate WFs, and in this case, to model and predict thunderstorms. For this research, the focus is on convection events in the atmosphere, which are closely linked with thunderstorm formation. To predict such events, data from ensemble NWP forecasts combined with satellite observations of thunderstorms are utilized.

NWP Forecasts: NWP models are used to simulate the behavior of the atmosphere and generate WFs by solving partial differential equations that describe fluid motion and thermodynamic properties of the atmosphere. These simulations account for a vast array of physical parameters at multiple grid points over time. The NWP ensemble here comprises predictions from 50 different members, and each member provides 23 features (see Fig. 17.2) per grid point for the 25,521 locations being observed. The 23 features provide valuable information regarding atmospheric properties, such as temperature, wind speed, pressure, and moisture content. Each of these features plays a key role in forecasting thunderstorms by identifying convective processes in the atmosphere.

Satellite Storm Observations: Satellite data collected through the Rapid Development Thunderstorm (RDT) product provide high-resolution storm observations. This product, developed by Météo-France under the EUMETSAT NWC-SAF framework, processes cloud data from geostationary satellites. The RDT provides details on cloud structures, including location,

Figure 17.1
Geographical region covered by the forecast and observational data for storm prediction (Jardines et al., 2020, 2021). Map lines delineate study areas and do not necessarily depict accepted national boundaries.

movement, shape, and intensity of convective cells. To train the model, binary storm images are created from these observations. A binary image represents areas where thunderstorms occurred within a specified 15-minute interval. When creating training data for each hour, four 15-minute images are merged to display stormy regions. Each pixel in the binary target image represents whether a storm occurred within that timeframe (see Fig. 17.3 for an example).

17.2.2 Methodology

This study maintains the NN construction, parameters, and settings utilized in the research by Jardines et al. (2020, 2021). The objective is to ensure fairness in comparisons while proposing specific enhancements to improve the performance of the storm prediction model. The architecture of the utilized NN is a MLP with two hidden layers, each containing 16

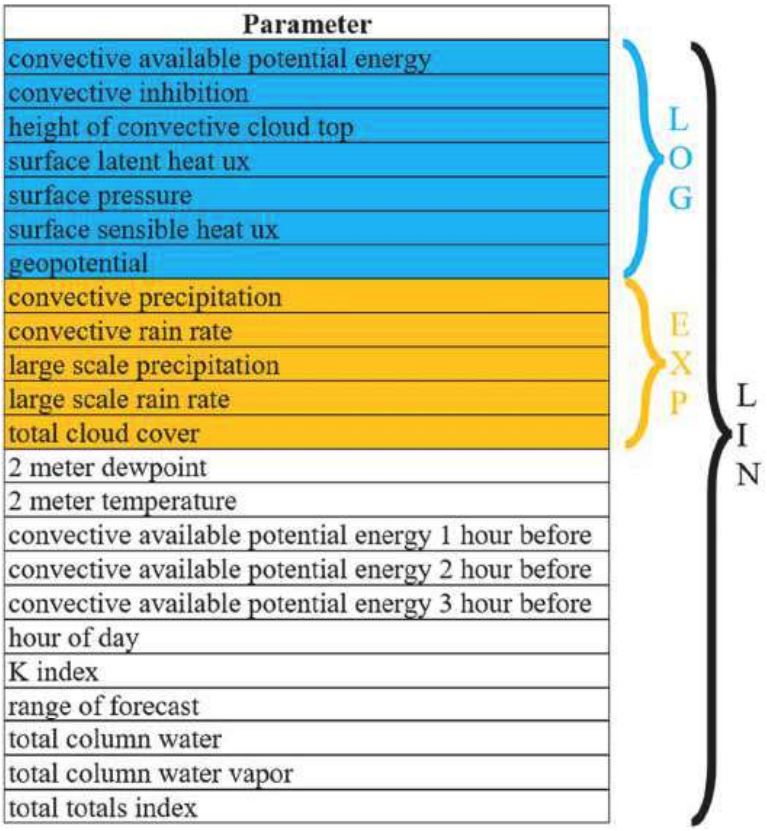

Figure 17.2
The list of parameters used for the model input (Jardines et al., 2020, 2021), with normalization techniques based on the features' dynamic range. The features are categorized into three groups: low dynamic range (using exponential normalization, as demonstrated by EXP), high dynamic range (using logarithmic normalization, as illustrated by LOG), and normal range (using linear normalization, as shown by LIN).

neurons, as depicted in Fig. 17.4. To reduce redundancy and computational overhead, a mean-based fusion function is employed, which significantly simplifies the dataset by aggregating the input values of the 50 ensemble members. While preserving the NN structure and other critical parameters from the original study, additional configurations are determined through an iterative trial-and-error approach. This ensures the network is optimized for the specific dataset and prediction task. Below, the proposed innovations are systematically categorized into minor and major methodological improvements. Mathematical formulations accompany these enhancements to ensure transparency and reproducibility.

17.2.2.1 Minor proposed suggestions

The minor suggestions focus on improving dataset utilization, input feature processing, and evaluation methodologies.

Figure 17.3
Binary storm target for June 17, 2018, at 19:00, showing polygons where storms are located. Map lines delineate study areas and do not necessarily depict accepted national boundaries.

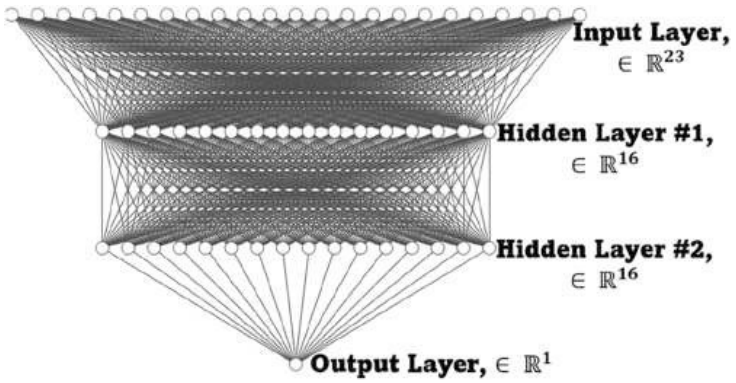

Figure 17.4
Structure of the utilized multilayer perceptron (MLP) model to predict the storm.

1. **Improved Data Partitioning**: The dataset used in Jardines et al.'s previous works (Jardines et al., 2020, 2021, 2024) was divided in a non-traditional way. The dataset was divided into subsets spanning four days, employing a unique distribution strategy.

Specifically, 50% of the data was designated for training purposes, 25% for validation, and the remaining 25% constituted the test set. Such proportions risk under-representing critical patterns in training. A more standard division would be 60%–80% for training, with 10%–20% each for validation and testing (Alqahtani & Whyte, 2016; de Carvalho Paulino et al., 2020; Guimarães & Shiguemori, 2019; Mahdipour et al., 2013; Sharifi et al., 2022). In this study, the data for one month is split into weekly subsets. It means considering days #3 to #9 as subset #1, days #10 to #16 as subset #2, days #17 to #23 as subset #3, and days #24 to #30 as subset #4. For each subset, five days (about 71.4%) are used for training, one day (14.3%) for validation, and one day (14.3%) for testing. Indeed, the sizes of the training, validation, and testing datasets can be expressed as follows:

$$|T| = 0.741 \times |S|, \quad |V| = |E| = 0.143 \times |S| \tag{17.1}$$

where $|T|$, $|V|$, and $|E|$ are the sizes of the training, validation, and testing datasets, respectively, and $|S|$ is the size of a subset. This redistribution ensures greater generalizability by maximizing the training data while preserving separate datasets for unbiased evaluation. The revised methodology aligns more closely with established best practices (Alqahtani & Whyte, 2016; de Carvalho Paulino et al., 2020; Guimarães & Shiguemori, 2019; Mahdipour et al., 2013; Sharifi et al., 2022).

2. **Normalization Strategies**: Feature variability poses challenges for NN optimization by causing gradients to skew during backpropagation. To address this, a two-stage normalization strategy is applied. Firstly, nonlinear normalization will be applied to some input features. Then, a linear normalization will be applied to all input features.

In the nonlinear normalization step, for features with low dynamic ranges, exponential normalization enhances sensitivity. In this step, for features with high dynamic ranges, logarithmic normalization reduces the influence of outliers. This nonlinear normalization step can be expressed mathematically as follows:

$$x_{\text{exp}-\text{norm}} = \exp(x), \quad x \geq 0 \tag{17.2}$$

$$x_{\log-\text{norm}} = \log(x + 1), \quad x \geq 0 \tag{17.3}$$

After applying the nonlinear normalization to some input features, all of the input features will have a medium and acceptable dynamic range. Then, to standardize all features to zero mean and unit variance, the linear normalization will be applied as follows:

$$x_{\text{normalized}} = \frac{x - \mu}{\sigma}, \mu = \frac{1}{N}\sum_{i=1}^{N} x_i, \sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{17.4}$$

These techniques ensure improved convergence during training by mitigating the uneven contributions of individual features. Fig. 17.2 illustrates this normalization process, categorizing features into three groups based on their dynamic ranges. Using nonlinear normalization for both low- and high-dynamic range features ensures that the data is better

suited for gradient-based algorithms. For example, exponential normalization prevents minor variations from being overshadowed, while logarithmic normalization controls high dynamic range. This balanced approach minimizes skewed gradient updates, improving model convergence and overall accuracy.

17.3 Robust evaluation using k-fold cross-validation

Given the limited dataset (which spans only 1 month), k-fold validation is employed to maximize data utilization while ensuring rigorous evaluation. In each fold, the dataset is divided into k equal-sized subsets. The model is trained on k–1 subsets, and validation is performed on the remaining one. This process repeats k times, and performance metrics are averaged across folds:

$$\text{Performance}_{\text{avg}} = \frac{1}{k}\sum_{i=1}^{k}\text{Performance}_i \tag{17.5}$$

This systematic evaluation reduces bias and variance, ensuring reliable conclusions about model performance and leveraging all available data efficiently.

17.3.1 Major proposed suggestions

The major suggestions emphasize optimizing computational efficiency and improving the representational capacity of the NN.

1. Data Fusion for Redundancy Reduction: As illustrated in

Fig. 17.5 shows data from one month, comprising 30 days, with each day containing 24 hours of information. WFs are generated at midday and midnight each day by 50 members. For storm prediction, only the forecast information from the previous 36 hours is used. Consequently, for each hour, three WFs from three different times are available, referred to as information from three different ranges. For example, as shown in

Fig. 17.5, for the hour of 21:00, we consider weather forecasts (WFs) from today's midday, yesterday's midnight, and yesterday's midday, labeled as WF#1 with Range=9, WF#2 with Range=21 (21=9+12), and WF#3 with Range=33 (33=9+24), respectively. Each WF consists of predictions from 50 members for 25,521 grid points, with each member's prediction for each point containing 23 features. Consequently, the total number of features and digits to process exceeds 63 billion, which is exceptionally high. Regarding the files used to train the NN, which are from Python's Pandas DataFrames type, including forecast predictions and processed RDT observations as the target, each file is 142.063 MB in size. With three different files corresponding to different ranges, the total data size for each hour (sample) is 3 × 142.063 MB = 426.189 MB, roughly equivalent to 0.4 GB per sample. Since the NWP data in this research's dataset is available for each point at each time by 50 members, it results in

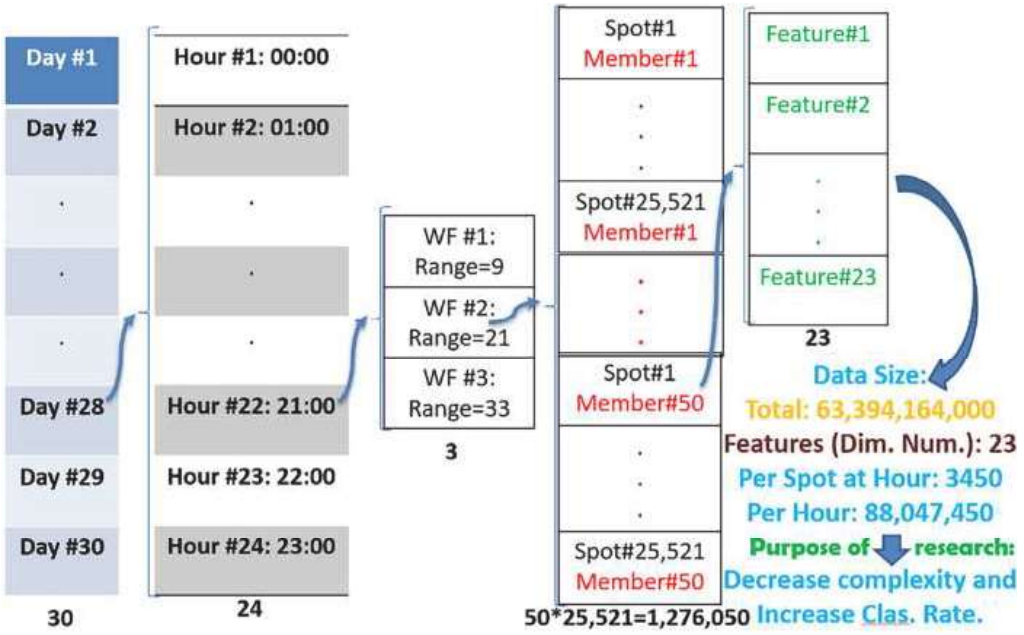

Figure 17.5
Structure of utilized input database in the research.

very redundant input data for storm prediction. Therefore, to reduce computational complexity and improve classification performance, the first major idea involves fusing the predictions of the 50 members for each point at each time and using the mean value of these 50 values. To this purpose, the 23 input features are categorized into two groups. The first group of features, containing "hour," "range," and "z," are fixed for all points of NWP corresponding to each member at a specific time. The second group of input features, containing 20 remaining input features that are changing for each point in the members' predictions for each time, will be fused as follows:

$$x_{\text{fused}} = \frac{1}{50} \sum_{i=1}^{50} x_i \tag{17.6}$$

This process reduces the input data size by 50-fold while maintaining robustness by mitigating outliers (Mahdipour et al., 2016, 2024; Nazarko, 2002). The block diagram of this proposed suggestion is illustrated in detail in

Fig. 17.6. Totally, fusion serves multiple purposes:

- **Noise Reduction:** Averaging dampens the influence of outliers, stabilizing predictions.
- **Dimensionality Reduction:** Input data size is reduced by a factor of 50, leading to exponential reductions in processing time (Arora & Barak, 2009; Cormen et al., 2022):

$$T_{\text{processing}} \propto V_{\text{data}}^{l},\ l > 1 \tag{17.7}$$

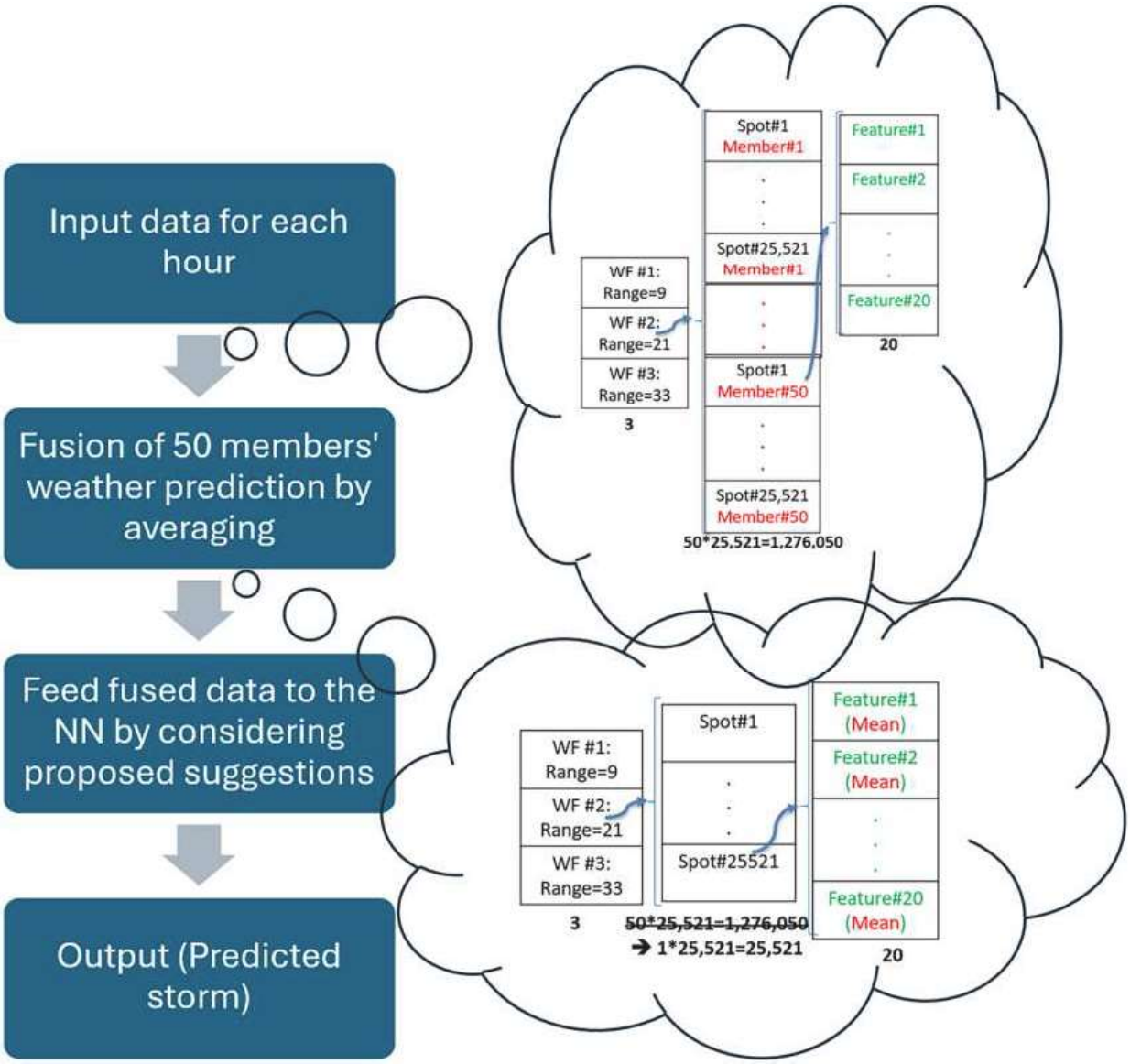

Figure 17.6

Block diagram of the major proposed method, where the utilized data in each hour is expressed in detail, where the effect of fusion is illustrated by showing the input and the output data structures.

Here, $T_{\text{processing}}$ is the computational time, and V_{data} is the data volume. For example, if the computational time for raw data (V_{raw}) is 100 hours, reducing V_{data} to $V_{\text{raw}}/50$ can decrease the time to approximately $V_{\text{raw}}/50^{l}$. For instance, an initial data processing time of 100 hours could decrease to less than 20 minutes after fusion when l = 1.5.

2. **Addressing Sensitivity to Initial Weights**: NNs often produce inconsistent results due to sensitivity to initialization. To address this, the network is trained multiple times, each with

random initialization of weights and biases (Li & Yeh, 2002; Mahdipour et al., 2013). The final output is obtained as the mean or median of individual outputs:

$$\text{Output}_{\text{ensemble}} = \frac{1}{M} \sum_{i=1}^{M} \text{Output}_i \tag{17.8}$$

where M is the number of models' runs. This method ensures robustness by minimizing dependency on specific initialization conditions. It is important to note that this idea, besides some other ideas like k-fold validation strategy, can not be applied to the application without the fusion idea due to a very high amount of computational complexity and duration time.

17.4 Results and discussion

The results presented in this study are derived from simulations conducted using a GPU-accelerated high-performance computing system. The computational environment comprises an Intel(R) Xeon(R) Gold 6238 R CPU with 64GB of RAM and an NVIDIA RTX A5000 GPU with 32GB memory, leveraging Python 3 alongside the TensorFlow deep learning framework. GPU acceleration was implemented using Windows Subsystem for Linux (WSL), enabling efficient utilization of computational resources. Despite these optimizations, certain proposed methodologies necessitated extensive runtimes, particularly when applied to the full dataset. For instance, the integration of initial-weight-free initialization and data rearrangement strategies, applied consistently across PM-1, PM-2, and PM-3, requires over 101 days of computation when applied to the full dataset. As a result, some extensions are evaluated using a smaller subset of data to manage computational complexity.

For benchmarking purposes, two existing methods are considered: a baseline approach based on an existing NWP-based convection indicator (González-Arribas et al., 2017) and Method-1, a robust storm prediction algorithm proposed in (Jardines et al., 2020, 2021). The baseline provided a minimal-complexity reference, while Method-1 represented a state-of-the-art comparison with significant computational demands.

Three proposed methods, referred to as PM-1, PM-2, and PM-3, are developed and evaluated. Each method initially incorporated data rearrangement to ensure consistent and representative partitioning into training, validation, and testing datasets. PM-1 was defined as applying the NN to the unfused dataset. PM-2 utilized a fusion process, where 50 ensemble members of weather predictions were averaged to create a reduced and more robust input dataset. PM-3 extended PM-2 by running the NN multiple times (10 repetitions) and selecting the median output to enhance stability and accuracy.

The methodologies are examined in two phases. In the first phase, the effects of fusion, initial-weight-free initialization, and data rearrangement are evaluated using the full dataset. In the

second phase, k-fold validation and nonlinear normalization techniques are assessed using a subset of the data due to their high computational requirements.

17.4.1 Effects of fusion, initial-weight-free initialization, and data rearrangement

As mentioned, in this subsection, two extensions of initial weights, free and rearranging data for two states, fusing 50 members' NWPs by the mean function (PM-2 and PM-3) and not-fusing (PM-1), are implemented. At the same time, they are applied to the entire dataset. Simulation results show that the proposed methods perform better in terms of area under the curve (AUC), true positive rate (TPR), and false positive rate (FPR), as presented in Table 17.1 and Fig. 17.7. As can be seen, the proposed methods have better performance in terms of storm detection compared to other methods. Although the computational complexity in the PM-1 case is very high, PM-2 is faster, and its detection performance is also better than

Table 17.1 Performance of the proposed methods in comparison to other methods.

Method	%AUC	%FPR	%TPR	Run time
Baseline (González-Arribas et al., 2017)	80.83913	12.44321	72.25637	Couple of minutes
Method-1 (Jardines et al., 2020, 2021)	95.28233	11.96384	90.74992	10 days
PM-1	95.40195	12.10917	90.94430	30 days
PM-2	95.75532	11.55979	91.76360	5:20'
PM-3	95.77761	11.04367	91.35065	53:10'

AUC, area under curve; *TPR*, true positive rate; *FPR*, false positive rate.

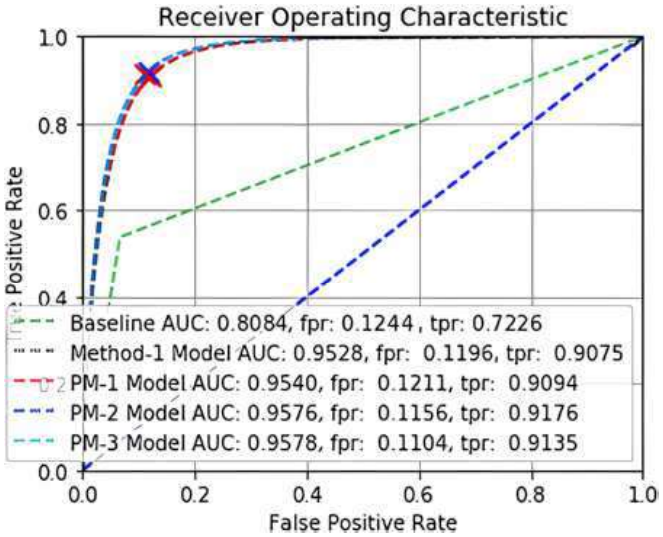

Figure 17.7
Operating characteristic of baseline (González-Arribas et al., 2017), Method-1 (Jardines et al., 2020, 2021), and proposed methods (PM-1, PM-2, and PM-3).

that of PM-1. The cause of this performance improvement in all terms of comparison relates to the benefits of fusion. The fusing operation reduces the effect of outliers on the output and decreases redundancy very efficiently. Due to a decrease in the volume of input data by approximately 98% (50 times), leading to substantial runtime reductions, the runtime in PM-2 has also decreased significantly. The results for PM-3 are very similar to those for PM-2, with a slight improvement in storm detection and significant degradation in runtime.

In the case of range sensitivity, results are reported in Fig. 17.8. As can be seen visually, increasing the range from 0–12 to 12–24 hours and then to 24–36 hours, the RUC performance is decreased for all models (e.g., for PM-3 it decreases from 0.9598 to 0.9586 and then to 0.9547), except the "Baseline" method. This is because the "Baseline" method is very simple and cannot accurately predict the total storm. It appears that "Baseline" has the worst performance, and the performance degradation becomes more pronounced as the range increases. While the performances of other methods for different time periods are very similar to each other, it is challenging to distinguish the differences visually. The sensitivity of Method-1 to range increases is slightly less than that of the Baseline. However, PM-1 is more robust to range increment compared to both the Baseline and Method-1 methods. It seems it relates to more training of the NN and better search by running multiple times. This improvement is more pronounced in PM-2, as it utilizes fusion and reduces the outliers' effect on the output by averaging the results. And finally, PM-3 has the best robustness to range increases and is even better than PM-2, due to its more thorough search for NN weights.

In terms of normalized output, reported in Fig. 17.9, the ideal output is the case of having two separate and sharp distributions on 0 and 1 for the nonconvective (gray color) and convective (red color) classes, respectively. Naturally, if the distributions have minimum coverage (complete separately in the ideal case) and are sharper on points of 0 and 1 (or nearer to them), it means better output in this term of comparison.

As can be seen in Fig. 17.9, the baseline method has the worst performance in this term too. The performances of Method-1, PM-1, PM-2, and PM-3 seem the same, although the distributions in the proposed suggestions cases (PM-1, PM-2, and PM-3) seem more separate (slightly) and sharper (obviously). The sharpness is more sensitive for the nonconvective class's distribution and at the 0 point. The rank of understudying methods in this term is the same as the AUC case (see Table 17.1).

For June 17, 2018, at 19:00 while the maximum storms are occurring, just to see the visual results of the simulated methods, target and resulting storm prediction images of all simulated methods (where they are binarized by considering the threshold value of 0.5) are presented in Fig. 17.10. Although the baseline has the worst performance as can be seen in Fig. 17.10B, the benefits of the proposed method cannot be found easily and visually in this case too. Indeed, this term of comparison is a sample from the overall comparison that is done by AUC, FPR, and TPR in Table 17.1, but the slight differences cannot be determined in this specific case.

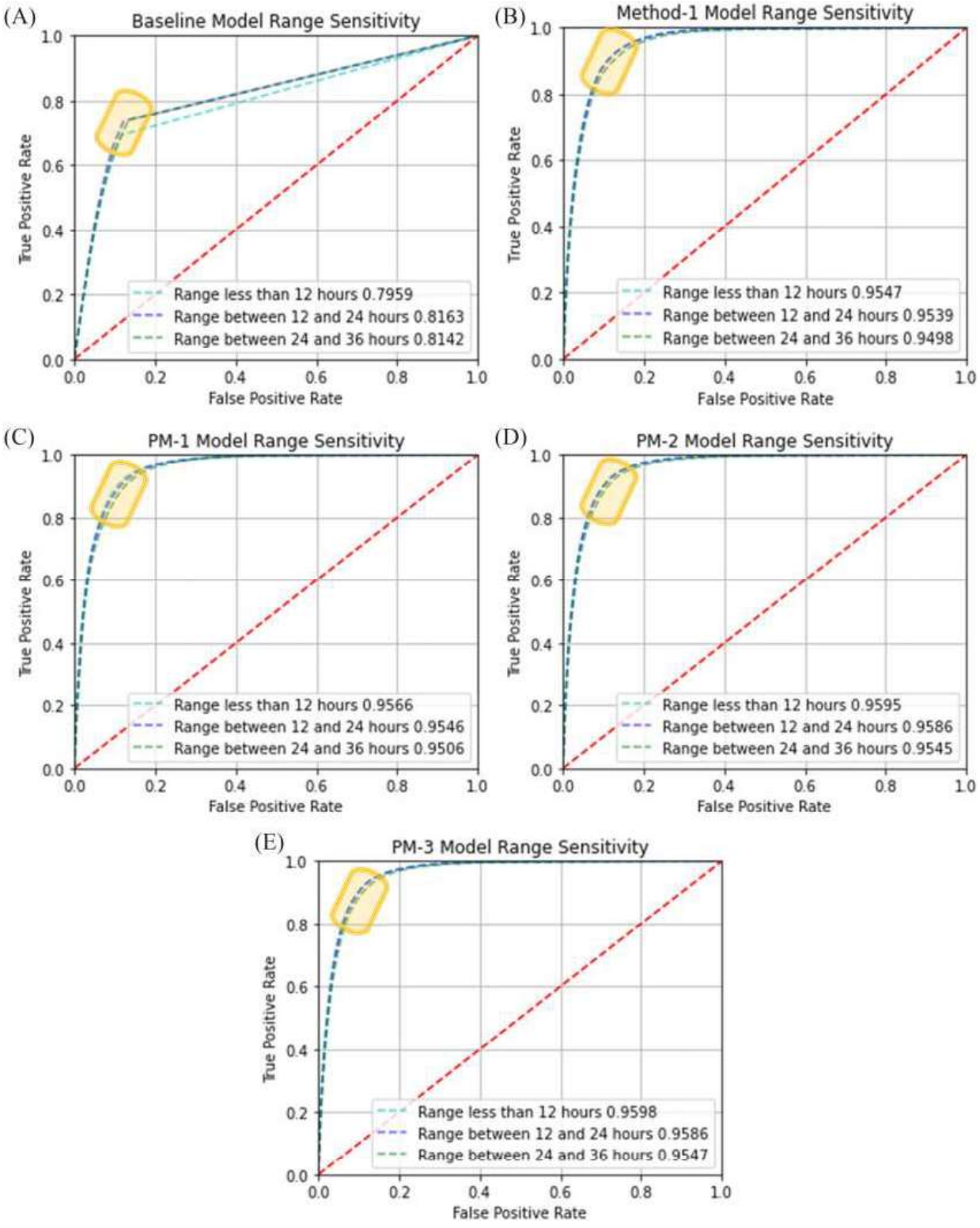

Figure 17.8

Range sensitivity of different range values in the methods of (A) Baseline (Nazarko, 2002); (B) Method-1; (C) PM-1; (D) PM-2, and (E) PM-3.

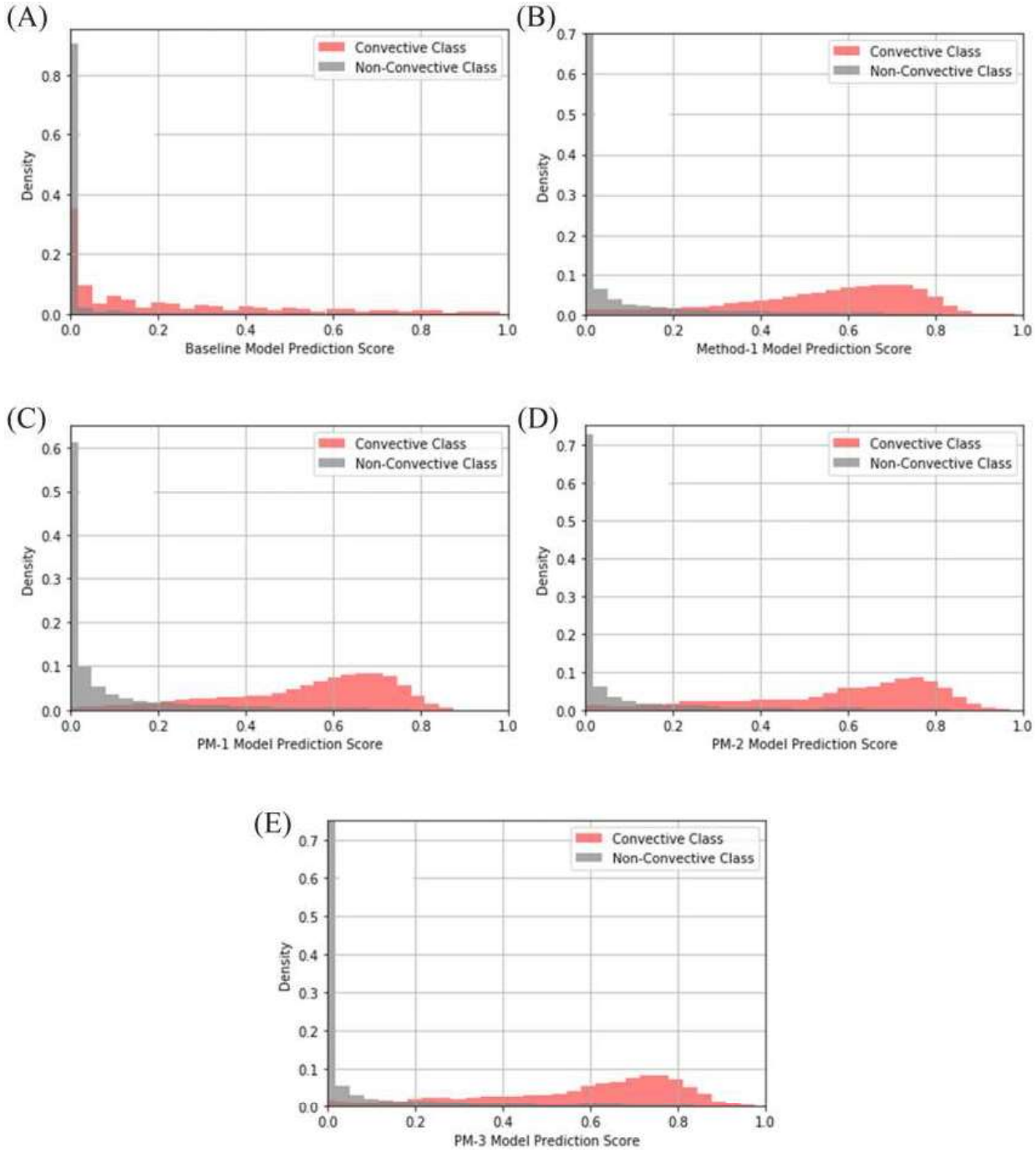

Figure 17.9

(A) Normalized output of Baseline; (B) Normalized output of Method-1; (C) Normalized output of PM-1; (D) Normalized output of PM-2; and E) Normalized output of PM-3.

Regarding the results reported in Fig. 17.10, it is important to note that our input data is unbalanced, with significantly less data related to storms compared to data without storms. In cases where the minority class (storm data) is of higher importance and high accuracy for this

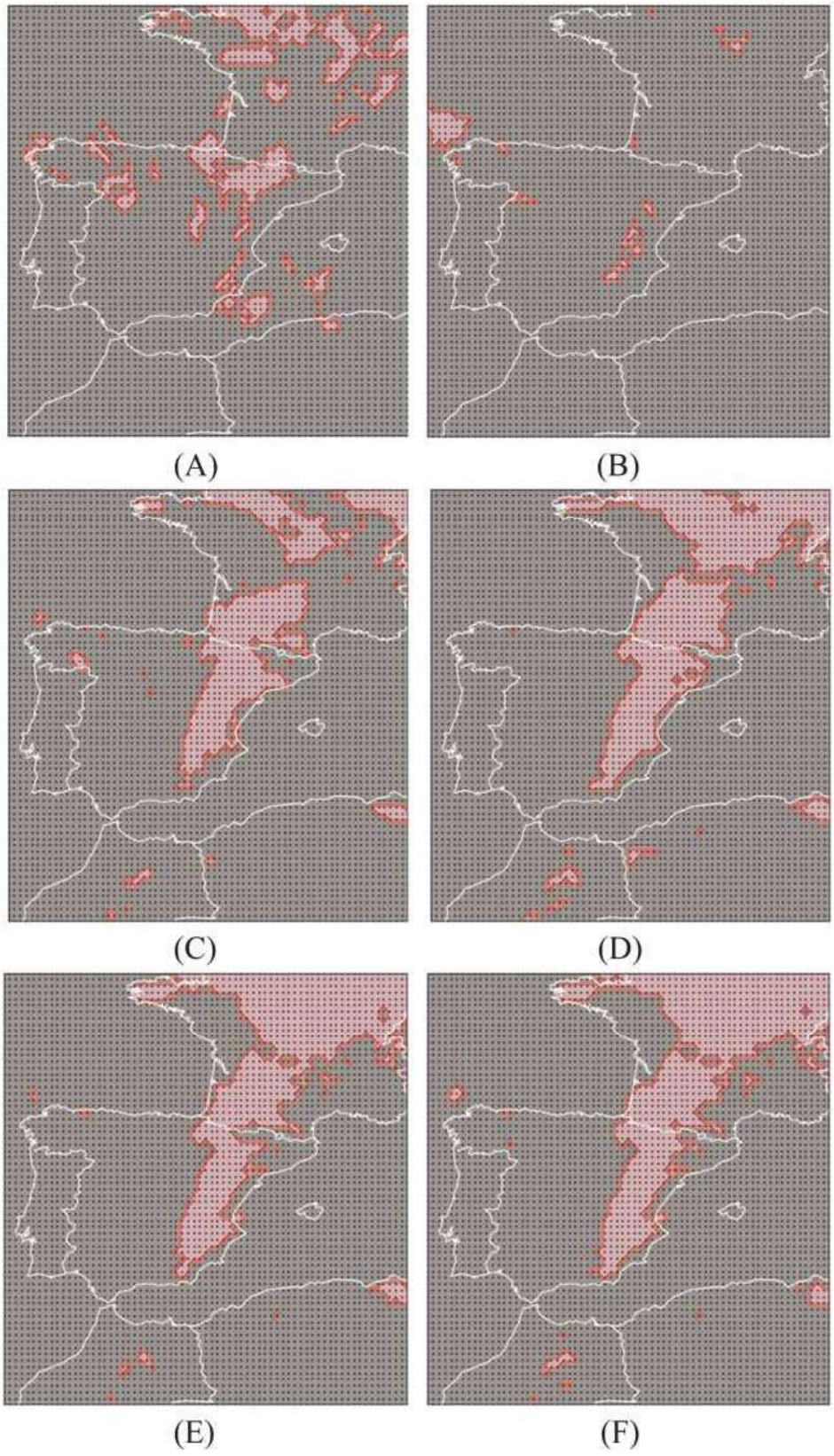

Figure 17.10
Binary image of storms:(A) target and predicted by (B) Baseline method, (C) Method-1, (D) PM-1, (E) PM-2, and (F) PM-3. Map lines delineate study areas and do not necessarily depict accepted national boundaries.

Table 17.2 Effect of k-fold validation and nonlinear normalization on the output, applied on a subset of data.

Method	%Classification accuracy	Run time
Method-1 (Jardines et al., 2020, 2021)	91.36543	22':49"
k-fold validation	93.42728	103':51"
Nonlinear Normalization	91.96320	39':22"

class is desired, there is a risk of overestimation. To address class imbalance, techniques such as over-sampling the minority class or assigning higher class weights can be employed. Implementing cost-sensitive learning, which involves rewarding the minority class more (by increasing the weight or importance of correctly predicting the minority class), can effectively reduce overestimation. This method compensates for the higher misclassification costs of the minority class without causing excessive overestimation. From another perspective, the dataset used in this scenario contains information only about weather and storm events for one month, rather than a large amount of redundant and computationally complex data. As described in Section 17.2.1, given the limited information available, it is essential to implement a k-fold validation strategy to ensure robust and reliable model performance. We have utilized these techniques to compensate for overestimation, achieving good predictive performance in storm prediction.

17.4.2 Effects of k-fold validation and nonlinear normalization

Since the remaining proposed suggestions, that is, *k*-fold validation and nonlinear normalization, increase the computational complexity severely, they are applied only on a mini subset of data. Therefore, in this case, the classification rate and running time are reported in Table 17.2.

In the *k*-fold validation process, k is 6. As mentioned, the linear normalization cannot be efficient for the features with very high or very low dynamic range. Therefore, in the first step of normalization, the logarithmic and exponential functions are applied to some features as demonstrated in Fig. 17.2. Then, in the second step of normalization, the linear normalization (to have features with zero-mean and unit standard deviation) is applied to all features.

17.5 Conclusions

Convective air, known as a significant hazard, can be catastrophic for aircraft. Developing a storm prediction model presents a new research opportunity where artificial NNs are increasingly utilized. In recent studies, NWP has been used as input, with satellite image

data serving as the desired output. This paper proposed several extensions and considerations for improving NN applications in this context. The suggested improvements include efficiently rearranging data to select training, validation, and test datasets; obtaining initial weights-free responses; implementing a *k*-fold validation strategy; and applying nonlinear normalization to certain input features. These strategies have been shown to enhance the storm prediction system's performance in terms of AUC, FPR, TPR, and classification accuracy. However, fully implementing these suggestions would require more than 500 days, making it impractical. To address this challenge, this paper proposed the use of a mean-based fusion function at the feature level as an initial solution. This method significantly reduced computational complexity by approximately 98% while also improving storm detection performance. For instance, the classification rate improved by about 2% when *k*-fold validation was applied.

Additionally, the mean-based fusion function reduced the model's sensitivity to noise and outliers and mitigated the impact of range increases. Future research directions include exploring other fusion methods, extending the dataset to cover more than one month, redesigning the NN using deep NNs, and incorporating spatial information. These approaches could further enhance storm prediction capabilities and address the limitations highlighted in this study.

References

Abaspur Kazerouni, Mahdipour, H., Dooly, G., & Toal, D. (2021). Vector fuzzy c-spherical shells (VFCSS) over non-crisp numbers for satellite imaging. *Remote Sensing, 13*(21), 4482.

Alqahtani, & Whyte, A. (2016). Estimation of life-cycle costs of buildings: Regression vs artificial neural network. *Built Environment Project and Asset Management*.

Arora, S., & Barak, B. (2009). *Computational complexity: A modern approach*. Cambridge University Press.

Baldauf, M., Seifert, A., Förstner, J., Majewski, D., Raschendorfer, M., & Reinhardt, T. (2011). Operational convective-scale numerical weather prediction with the COSMO model: Description and sensitivities. *Monthly Weather Review, 139*(12), 3887–3905.

Beheshti, Mahdipour Hossein-Abad, H., Matsuda, H., s, J.-A., & Initiative, D. N. (2019). Identification of Alzheimer's disease on the basis of a voxel-wise approach. *Applied Sciences, 9*(15), 3063.

de Carvalho Paulino, Â., Guimarães, L. N. F., & Shiguemori, E. H. (2020). Assessment of noise impact on hybrid adaptive computational intelligence multisensor data fusion applied to real-time uav autonomous navigation. *IEEE Latin America Transactions, 18*(02), 295–302.

Collins, W., & Tissot, P. (2015). An artificial neural network model to predict thunderstorms within 400 km^2 South Texas domains. *Meteorological Applications, 22*(3), 650–665.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). *Introduction to algorithms*. MIT Press.

Ghaderizadeh, S., Abbasi-Moghadam, D., Sharifi, A., Tariq, A., & Qin, S. (2022). Multiscale dual-branch residual spectral–spatial network with attention for hyperspectral image classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 15*, 5455–5467.

González-Arribas, D. et al. (2017). Robust optimal trajectory planning under uncertain winds and convective risk. In *ENRI International Workshop on ATM/CNS*, Springer, pp. 2–103.

Guimarães, L. N. F., & Shiguemori, E. H. (2019). Hybrid adaptive computational intelligence-based multisensor data fusion applied to real-time UAV autonomous navigation. *Inteligencia Artificial, 22*(63), 162–195.

He, J., & Loboda, T. V. (2020). Modeling cloud-to-ground lightning probability in Alaskan tundra through the integration of weather research and forecast (WRF) model and machine learning method. *Environmental Research Letters, 15*(11), 115009.

Jardines, et al. (2024). Thunderstorm prediction during pre-tactical air-traffic-flow management using convolutional neural networks. *Expert systems with applications, 241*, 122466.

Jardines, M. S. A., Cervantesb, A., & Garcia-Heras, J. (2020). Convection indicator for pre-tactical air traffic flow management using neural networks. In *Presented at the international conference on recent advances in fluids and thermal sciences (iCRAFT)*.

Jardines, Soler, M., Cervantes, A., García-Heras, J., & Simarro, J. (2021). Convection indicator for pre-tactical air traffic flow management using neural networks. *Machine Learning with Applications, 5*, 100053.

Kosari, Sharifi, A., Ahmadi, A., & Khoshsima, M. (2020). Remote sensing satellite's attitude control system: Rapid performance sizing for passive scan imaging mode. *Aircraft Engineering and Aerospace Technology, 92*(7), 1073–1083.

Li, H., Li, Y., Li, X., Ye, Y., Li, X., & Xie, P. (2019). A comparative study on machine learning approaches to thunderstorm gale identification. In *Proceedings of the 2019 11th international conference on machine learning and computing*, pp. 12–16.

Li, X., & Yeh, A. G.-O. (2002). Neural-network-based cellular automata for simulating multiple land use changes using GIS. *International Journal of Geographical Information Science, 16*(4), 323–343.

Mahdipour, H., Nezamabadi-pour, H., Abbasi-Moghadam, D., & Khademi, M. (2013). Joint detection, channel estimation, and interference cancellation in downlink MC-CDMA communication systems using complex-valued multilayer neural networks. *annals of telecommunications-annales des télécommunications, 68*(7), 467–476.

Mahdipour, H., Shabanian, M., & Kazerouni, I. A. (2020a). Vectorized kernel-based fuzzy C-means: A method to apply KFCM on crisp and non-crisp numbers. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 28*(04), 635–659.

Mahdipour, H., Shabanian, M., & Kazerouni, I. A. (2020b). Fuzzy c-means clustering method with the fuzzy distance definition applied on symmetric triangular fuzzy numbers. *Journal of Intelligent & Fuzzy Systems, 38*(3), 2891–2905.

Mahdipour, H., Sharifi, A., Sookhak, M., & Medrano, C. R. (2024). Ultra-fusion: Optimal fuzzy fusion in land-cover segmentation using multiple panchromatic satellite images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*.

Mahdipour, H., Yazdi, H. S., & Khademi, M. (2016). Efficient land-cover segmentation using meta fusion. *Journal of Information Systems and Telecommunication (JIST)*.

Mecikalski, J. R., Williams, J. K., Jewett, C. P., Ahijevych, D., LeRoy, A., & Walker, J. R. (2015). Probabilistic 0–1-h convective initiation nowcasts that combine geostationary satellite observations and numerical weather prediction model data. *Journal of Applied Meteorology and Climatology, 54*(5), 1039–1059.

Mohammadi, M., Sharifi, A., Hosseingholizadeh, M., & Tariq, A. (2021). Detection of oil pollution using SAR and optical remote sensing imagery: A case study of the Persian Gulf. *Journal of the Indian Society of Remote Sensing, 49*(10), 2377–2385.

Mowrer, H. T., & Congalton, R.G. (2003). *Quantifying spatial uncertainty in natural resources: Theory and applications for GIS and remote sensing*. CRC Press.

Naik, K., et al. (2024). Current status and future directions: The application of artificial intelligence/machine learning for precision medicine. *Clinical Pharmacology & Therapeutics*.

Nazarko, S. (2002). Evaluation of data fusion methods using Kalman filtering and transferable belief model. Master's thesis, University of Jyväskylä.

Pandey, D. K., Hunjra, A. I., Bhaskar, R., & Al-Faryan, M. A. S. (2023). Artificial intelligence, machine learning and big data in natural resources management: A comprehensive bibliometric review of literature spanning 1975–2022. *Resources Policy, 86*, 104250.

Rodriguez-Rodriguez, Rodriguez, J.-V., Shirvanizadeh, N., Ortiz, A., & Pardo-Quiles, D.-J. (2021). Applications of artificial intelligence, machine learning, big data and the internet of things to the COVID-19 pandemic: A

scientometric review using text mining. *International Journal of Environmental Research and Public Health, 18*(16), 8578.

Šaur, D. (2017). Forecasting of convective precipitation through NWP models and algorithm of storms prediction. In *Computer Science On-line Conference*. Springer, 125–136.

Sharifi (2021). Development of a method for flood detection based on Sentinel-1 images and classifier algorithms. *Water and Environment Journal, 35*(3), 924–929.

Sharifi, & Amini, J. (2015). Forest biomass estimation using synthetic aperture radar polarimetric features. *Journal of Applied Remote Sensing, 9*(1), 097695.

Sharifi, Amini, J., Sri Sumantyo, J. T., & Tateishi, R. (2015). Speckle reduction of PolSAR images in forest regions using fast ICA algorithm. *Journal of the Indian Society of Remote Sensing, 43*, 339–346.

Sharifi, Amini, J., & Tateishi, R. (2016). Estimation of forest biomass using multivariate relevance vector regression. *Photogrammetric Engineering & Remote Sensing, 82*(1), 41–49.

Sharifi, Mahdipour, H., Moradi, E., & Tariq, A. (2022). Agricultural field extraction with deep learning algorithm and satellite imagery. *Journal of the Indian Society of Remote Sensing, 50*(2), 417–423.

Simon, T., Fabsic, P., Mayr, G. J., Umlauf, N., & Zeileis, A. (2018). Probabilistic forecasting of thunderstorms in the Eastern Alps. *Monthly Weather Review, 146*(9), 2999–3009.

Spiridonov, V., Baez, J., Telenta, B., & Jakimovski, B. (2020). Prediction of extreme convective rainfall intensities using a free-running 3-D sub-km-scale cloud model initialized from WRF km-scale NWP forecasts. *Journal of Atmospheric and Solar-Terrestrial Physics, 209*, 105401.

Tariq, et al. (2022). Flash flood susceptibility assessment and zonation by integrating analytic hierarchy process and frequency ratio model with diverse spatial data. *Water, 14*(19), 3069.

Wilson, J. W., Crook, N. A., Mueller, C. K., Sun, J., & Dixon, M. (1998). Nowcasting thunderstorms: A status report. *Bulletin of the American Meteorological Society, 79*(10), 2079–2100.

Zamani, Sharifi, A., Felegari, S., Tariq, A., & Zhao, N. (2022). Agro climatic zoning of saffron culture in miyaneh city by using WLC method and remote sensing data. *Agriculture, 12*(1), 118.

Zhou, H., Li, Y., Zhang, Q., Xu, H., & Su, Y. (2022). Soft-sensing of effluent total phosphorus using adaptive recurrent fuzzy neural network with Gustafson-Kessel clustering. *Expert Systems with Applications, 203*, 117589.

Zhou, H., Zhang, Y., Duan, W., & Zhao, H. (2020). Nonlinear systems modelling based on self-organizing fuzzy neural network with hierarchical pruning scheme. *Applied Soft Computing, 95*, 106516.

Zhou, K., Zheng, Y., Li, B., Dong, W., & Zhang, X. (2019). Forecasting different types of convective weather: A deep learning approach. *Journal of Meteorological Research, 33*(5), 797–809.