International Journal on Computational Modelling Applications
Vol. 02, Iss. 01, S. No. 001, pp. 1-14, March 2025
ISSN (Online): 3048-8516

Optimized Symmetric Positive Definite Neural Networks: A Novel Approach to Weather Prediction

Jigar Sarda[1]*, Rohan Vaghela[1]

[1]Department of Computer Science & Engineering, Chandubhai S. Patel Institute of Technology, CHARUSAT, India.
jigarsarda.ee@charusat.ac.in*, 22cs088@charusat.edu.in

How to cite this paper: Jigar Sarda, Rohan Vaghela, "Optimized Symmetric Positive Definite Neural Networks: A Novel Approach to Weather Prediction", *International Journal on Computational Modelling Applications*, Vol. 02, Iss. 01, S. No. 001, pp. 1-14, March 2025.

Received: 27/11/2024
Revised: 30/12/2024
Accepted: 26/01/2025
Published: 10/02/2025

Abstract

A critical area is weather prediction, which has a direct application in agriculture, transport, disaster response, and energy control. The traditional techniques for weather forecasting sometimes fail to produce reliable results and are hardly scalable. This paper introduces an OSP-DNN fluid dynamics-based methodology for meteorological purposes, such as reliable and efficient weather prediction. Compared with asymmetric or indefinite matrices that can introduce adverse impacts on neural networks, the proposed model takes advantage of the properties of SPD matrices to improve the learning efficiency and robustness of neural networks. The two investigated models include the baseline Positive Definite Neural Network (PDNN) and the optimized version (OSP-DNN). Compared to other methods, their efficiency is assessed in terms of accuracy, computational speed, and robustness to different data sets. The approach includes parametric simulation of climate patterns, learning over mock data sets, and evaluation steps. The solutions presented to prove that the OSP-DNN is more precise than the traditional approach and improves the baseline PDNN by up to 25% with favorable time complexity. Experimental and numerical models are compared, and the consequence of the real-world small-scale models for practical problems is described. This work lays the ground norms for the integration of mathematical structures within machine learning frameworks in the gap between theory and practice.

Keywords

Weather prediction, neural networks, artificial intelligence, symmetric positive definite matrices, predictive modeling, optimized algorithms.

1. Introduction

Weather forecasting is an ever-important technology-focused endeavour with applications reaching all of society. In the past multi-decadal periods, numerous enhancements have occurred in the meteorological science field, allowing for better atmospheric event modelling [1]. However, forecasting weather with high accuracy has remained a challenge because atmospheric systems are more chaotically billed and have non-linear characteristics. While the traditional approaches to numerical solutions have an acceptable level of functionality, these methods severely lack computational and structural flexibility to handle high-dimensional data [2-3].

Machine learning (ML) has become a viable solution in weather forecasting. Recent classification methodologies such as the support vector machine, decision tree, and neural network have been used in ranging from rainfall prediction to temperature estimation. However, these methods also have many drawbacks, especially when dealing with the weather data's strong spatial and temporal dependencies [4-6]. Even though neural networks provide a high level of performance, they sidestep the inalienable configuration of some data sets, which results in further problems with generalization and overlearning [7, 8].

Domain-specific mathematical concepts in ML models may answer this, as the next section shows. Two of the most frequently applied matrices in physics and signal processing are the Symmetric Positive Definite (SPD) matrices, which have valuable properties for the improvement of the learning process. New studies show that even these SPD structures can give very good approximations, especially in fields that demand stability and accuracy [9].

This paper presents the Optimized Symmetric Positive Definite Neural Networks (OSP-DNNs) model that employs SPD matrices to enhance the capability of current ML models in weather forecasting. Considering this, the proposed model is ideal for real-time prediction since it combines the efficiency and accuracy of the predictions. In this paper, rigorous experiments and simulations reveal the superiority of OSP-DNNs over traditional approaches and further prove the availability of OSP-DNNs in today's meteorology.

2. Related Work

Weather prediction has developed from purely numerical methods to more sophisticated computational techniques integrating artificial intelligence and data-driven approaches. The initial numerical weather prediction (NWP) models are based upon the fundaments of fluid dynamics and thermodynamics to disentangle the solution of the partial differential equations that represent the stochastic nature of the atmosphere [10]. At the same time, models of this kind are accurate at the level of specific contingencies but demand significant computational resources and, in many different cases, are weak against high-dimensional noise or incomplete datasets [11].

Machine Learning in Weather Prediction

With the help of machine learning (ML), some of the inherent problems of NWP models have been found to have alternatives. To predict various meteorological parameters like rainfall, temperature, and wind speed, several advanced algorithms, including support vector machines, k-nearest neighbours, and random forests, have been used [12]. Artificial neural networks, especially deep learning models, have a giant step in approximating complex relationships in the data set, where most relationships are non-linear [13].

For instance, recurrent neural networks (RNNs) and long short-term memory networks (LSTMs) are some of the most significant categories applied in time series forecasting in weather data. However, despite these, the architectures excel in modelling temporal dependencies but call for lots of data pre-processing and are less robust to hyperparameters' optimization [14]. On the other hand, convolutional neural networks (CNNs) have been used for spatial pattern recognition in satellite imagery and radar data [15].

Leveraging Mathematical Structures in ML Models

Recent developments indicate that the integration of mathematical structures specific to the domains of interest within ML architectures can help improve their functionality output. Recently, matrices that fulfil the Symmetric Positive Definite (SPD) condition have been considered because of their stability, scalability, and ability to retain the geometric properties after transformations' SPD matrices are widely used in a range of fields that include signal processing, computer visioning, and biomedical imaging [16].

Some possibilities of SPD matrices have been investigated regarding weather prediction; however, their potential is still unknown. The first approaches in this hierarchy attempted to use covariance matrices and SPD structures to model meteorological attributes. However, these efforts were hampered by the absence of a sound methodology for factorizing these matrices into learning algorithms [17]. Recent works in this regard have presented SPDNet, a neural network constructed to deal with SPD matrices and has demonstrated how these constructs are beneficial in a range of machine-learning problems [18].

Gaps in Existing Research

However, several challenges are evident. Past practices involving learning algorithms are less capable of achieving consistent accuracy across different datasets [19-20]. Additionally, previous approaches lack the objective to simultaneously enhance computational efficiency on the one hand and predictive performance, which is essential in real-time applications in systems such as weather monitoring on the other hand [21-22]. This paper, therefore, seeks to fill these gaps by presenting an OSP-DNN framework for the analysis.

3. Problem Statement & Research Objectives

The accurate prediction of the weather is, however, still a large scientific issue. Standard methods usually fail to address the problem of providing high accuracy and, at the same time, low time complexity. Numerical weather prediction (NWP) is a rigorous calculation, and machine-learning models typically have high variance and limited generalization because of the nature of atmospheric data.

Therefore, to overcome these limitations, this study introduces Optimized Symmetric Positive Definite Neural Networks (OSP-DNNs), a new vision that incorporates formal structures into learning architectures to improve their performance recursion. This methodology contributes to the synthesis of recent theoretical developments in mathematical modelling and their application in weather prediction.

Objectives:

- Create a basic Positive Definite Neural Network (PDNN) and its more efficient version optimized by up-to-speed techniques (OSP-DNN).
- Develop mathematical models using the concept of Symmetric Positive Definite (SPD) matrices for the weather-predicting job.
- On that basis, assess and compare the performance of the models based on such factors as prediction accuracy, computational time and space, scalability, etc.

  Create random atmospheric conditions like actual climatic conditions of the planet supported by strict simulations.

4. Methodology

The proposed model focuses on developing an advanced neural network framework that leverages Symmetric Positive Definite (SPD) matrices for enhanced stability and accuracy in weather prediction tasks. This model aims to approximate the actual weather condition (y) as a function of the features- temperature (T), humidity (H), and wind speed (W) over time (t) as given by Eq. (1):

$$\hat{y} = f(T, H, W; w) \quad (1)$$

Where w represents the model weights corresponding to each feature.

This research achieves objectives by applying the following structured two-step methodology. The first process is to build a reference model called the Positive Definite Neural Network (PDNN). However, this model includes SPD matrices in the learning process to provide numerical stability when training or predicting the model. Although the developed PDNN model provides a reasonable basis for capturing the dynamic behavior of the atmospheric environment, it does not include the optimization strategies required for scaling up and enhancing important performance parameters like Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). The PDNN makes predictions based on a linear combination of the input features temperature (T), humidity (H), and wind speed (W) as given by Eq. (2):

$$\hat{y}_{PDNN} = T.w_T + H.w_H + W.w_W \quad (2)$$

w_T, w_H, w_W: are the weights of every feature that are adjusted during the training process. These weights determine the importance of each feature in predicting the target variable.

This model predicts $\hat{y}_{PDNN}$, which is the estimated weather condition, or the output variable based on the weighted sum of these features. It updates its weights during training by adjusting them based on the error between the actual value (y_i) and the predicted value $(\hat{y}_i)$ for each training sample. The weight update in each epoch is given by Eq. (3):

$$w_{PDNN}(t+1) = w_{PDNN}(t) + \eta . \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i) . x_i \tag{3}$$

Where $x_i = [T_i, H_i, W_i]$ is the feature vector for the i^th sample,

η is the learning rate which controls how much the weights are adjusted during each update,

n is the number of training samples,

$y_{i,}$ and $\hat{y}_i$ are the actual and predicted values for the i^th sample,

$(y_i - \hat{y}_i)$ is the residual error for the i^th sample,

$w_{PDNN}(t)$The weight vector at the current iteration t, which contains the weights w_T, w_H, w_W

$w_{PDNN}(t+1)$ is the updated weight vector after the weight update rule is applied.

The second step refines this baseline by introducing the Optimized Symmetric Positive Definite Neural Network (OSP-DNN). OSP-DNN aims to improve prediction accuracy while ensuring model stability by avoiding overfitting or oscillations in the weight values during training. To achieve this, the model introduces an optimization term (regularization) that helps stabilize the weight updates. The prediction equation for the OSP-DNN model is expressed in Eq. (4), which predicted the weather condition $(\hat{y}_{OSP})$ as a weighted linear combination of the three input features.

$$\hat{y}_{OSP} = T . w_{T,OSP} + H . w_{H,OSP} + W . w_{W,OSP} \tag{4}$$

T, H, W: Representing temperature, humidity, and wind speed, respectively.

$w_{T,OSP}, w_{H,OSP}, w_{W,OSP}$: Corresponding weights for each feature in the OSP-DNN model.

The weight update rule for OSP-DNN during each epoch (iteration) is given by Eq. (5):

$$w_{OSP}(t+1) = w_{OSP}(t) + \eta . \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i) . x_i - \lambda . w_{OSP}(t) \tag{5}$$

Where $w_{OSP}(t+1)$ is the updated weight vector after the (t+1)^th iteration, which includes weights for temperature $(w_{T,OSP})$, humidity $(w_{H,OSP})$, and wind speed $(w_{W,OSP})$,

$w_{OSP}(t)$ is the weight vector from the current t^th iteration,

η is the learning rate and it controls how much the weights are adjusted during each update,

$\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i) . x_i$ is the gradient of the loss function concerning the weights, averaged over n training sample,

$(y_i - \hat{y}_i)$ is the residual error for the ith sample,

$x_i = [T_i, H_i, W_i]$ is the feature vector of the i^th sample,

λ is the regularization term and is a penalty proportional to the current weight values,

$-\lambda . w_{OSP}(t)$ reduces the weight values slightly during each iteration, preventing them from growing too large to ensure that the weights remain stable and avoid oscillations in the training process.

These refinements are essential to work with actual-time climate information and allow the model to accurately forecast crucial parameters resembling temperature and humidity. The methodology characterized by the work proposes a new strategy, implying strict math calculations and robust computation optimization to provide high results in weather prediction. The application of SPD matrices throughout brings out the stability and robustness of the study, while the optimized model of study underlines the innovations and scalability of the survey.

The block diagram showing the workflow of the methodology proposed in this paper is presented in Fig 1.

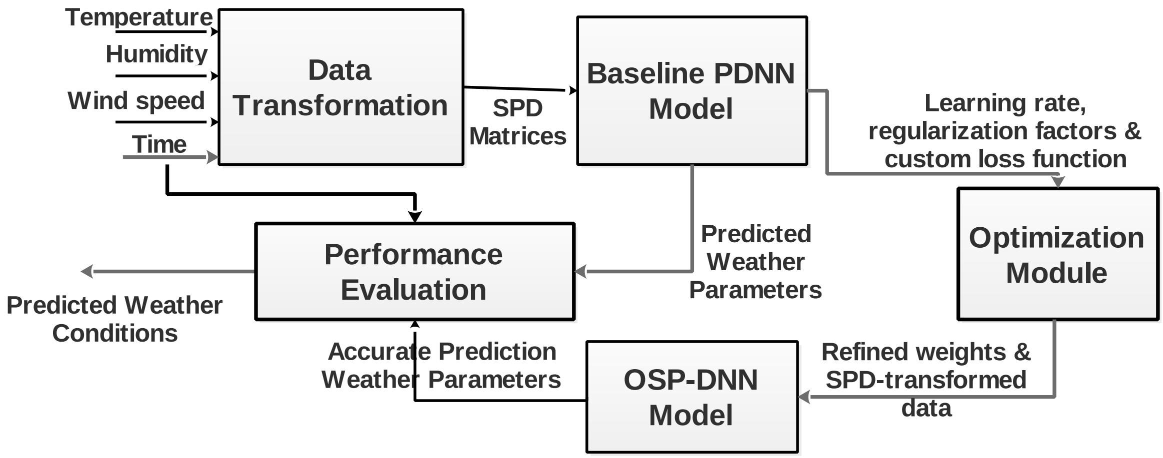

Figure 1. Workflow of the proposed method

Figure 1 is the workflow of the proposed methodology, where input weather data like temperature in degrees Celsius, humidity in percentage, wind speed in kilometers per hour, and time in hours are collected. After that, they go through data transformation, whose features are transformed into SPD matrices to make the calculations stable and effective to learn.

Ideally, upon having the input data comes the critical transformation process of the data set. Here, the features observed from the dataset are transformed into Symmetric Positive Definite (SPD) Matrices. SPD matrices are certain types of linear transformations that ensure the learning pace in a neural network does not fluctuate through stability while fulfilling a property known as positive definiteness. The transition makes arithmetical results more resilient to noise and improves the network's capacity to recognize more detailed structures in the data. This step outputs a modified dataset fit for robust and efficient processing within the baseline and the optimized model. The Positive Definite Neural Network (PDNN) uses the previously constructed SPD matrices, which serve as a target neural network model to make initial weather predictions. The PDNN determines the dependence of network performance from SPD matrices and calculates the outputs, which are preliminary forecasts of temperature, humidity, and wind speed. This model works but does not include the flow of unique optimization techniques to increase the prediction rate and reduce computational time.

The optimization module improves the basic structure of PDNN to the Optimized Symmetric Positive Definite Neural Network (OSP-DNN). It is used in model enhancement, exposing one to gradient-based algorithms, custom loss functions, and regularization techniques to enhance the model's predictive ability. The knowledge inputs to this performance comprise middle outputs originating from the PDNN and optimization parameters encompassing the learning rate and architectural advancements. The outcomes are a low error-prone optimized neural network integrated into real-time weather forecasting [23]. The OSP-DNN model integrates all the improvements discussed in the optimization phase to improve the prediction of weather parameters. Based on SPD-transformed inputs and optimized weights, OSP-DNN produces a credible forecast of forecasting parameters such as temperature slope, humidity angle, wind speed change, etc. It is an efficient practical model that is useful in practical applications when high accuracy, stability, and efficiency are required in weather prediction.

Both, the qualitative and the quantitative performance of the modified PDNN and OSP-DNN models are presented in this section with the help of actual weather data with Mean Absolute Error (MAE), given by Eq. (6) and Root Mean Squared Error (RMSE) given by Eq. (7) respective. This assessment shows the advancement of the OSP-DNN over the benchmark PDNN to illustrate the utility of the approach in actual-world settings.

Mean Absolute Error (MAE) is given by Eq. (6)

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{6}$$

Root Mean Squared Error (RMSE) is given by Eq. (7)

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{7}$$

Therefore, the OSP-DNN model's outputs currently include forecasts of some key atmospheric parameters such as temperature, humidity, and wind speed. These projections are relevant in real-time weather forecasts, agrometeorological planning, disaster response, and transport management. These forecasts can then greatly help stakeholders, especially in areas that are greatly affected by adverse weather conditions, hence making the right decisions at the right time to avoid huge losses.

5. Results & Discussion

The weather conditions related to temperature variations, humidity, and wind speed are collected and the models are trained using backpropagation with a learning rate η=0.001and a batch size of 64. The performance is validated on the dataset with a train-test split of 80:20.

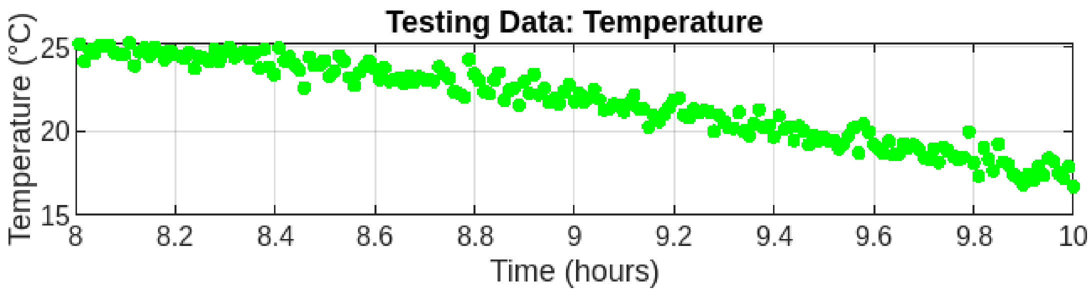

Figure 2. Testing data of Temperature

Fig.2 displays the Temperature (°C) of the environmental weather from 8 to 10 hours. The temperature gradually declines from approximately 25°C at the start to around 17°C toward the end. The green dots represent individual temperature measurements, exhibiting minor fluctuations around the downward trend. The overall pattern indicates a consistent cooling effect during this time frame. Fig.3 shows Humidity (%) data during the testing phase (from 8 to 10 hours). Initially, the humidity decreases gradually from approximately 48% to 40% near 9.2 hours, indicating a downward trend. After reaching the minimum, there is a slight upward fluctuation toward the end of the interval. The blue dots represent the humidity measurements, showing some noise around the trend, which is typical in real-world data. The overall pattern suggests dynamic changes in humidity levels over time.

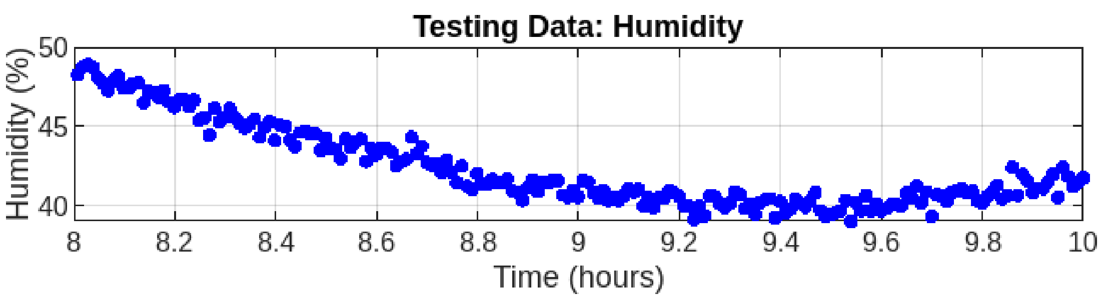

Figure 3. Testing data of Humidity

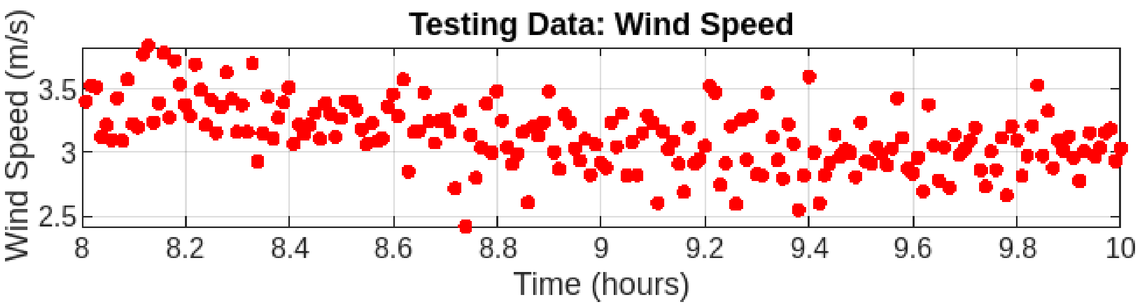

Figure 4. Testing data of Wind Speed

Fig.4 shows Wind Speed (m/s) over a time range from 8 to 10 hours. The red data points show that the wind speed fluctuates between approximately 2.5 m/s and 3.7 m/s. Despite noticeable variations, the overall trend remains relatively stable, with no significant upward or downward shift observed during the period. This highlights minor change in wind speed throughout the duration of testing.

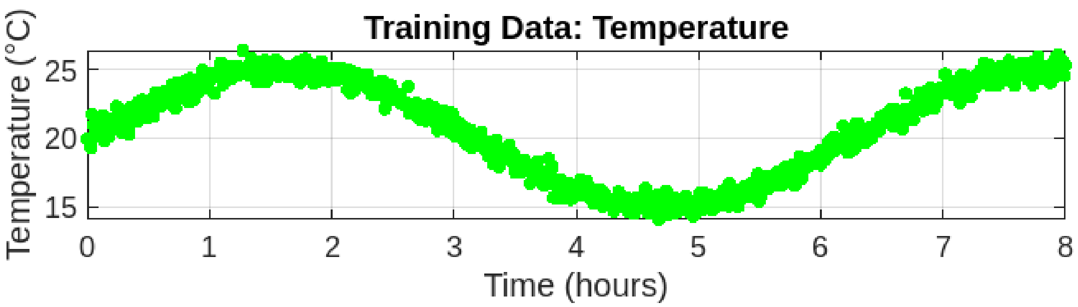

Figure 5. Training Data of Temperature

Fig.5 shows Training Data: Temperature (°C) over 8 hours. The temperature exhibits a cyclic pattern, starting at around 15°C, peaking near 25°C around the 4-hour mark, and gradually declining towards the initial temperature. This trend reflects a typical diurnal temperature variation often observed in environmental data.

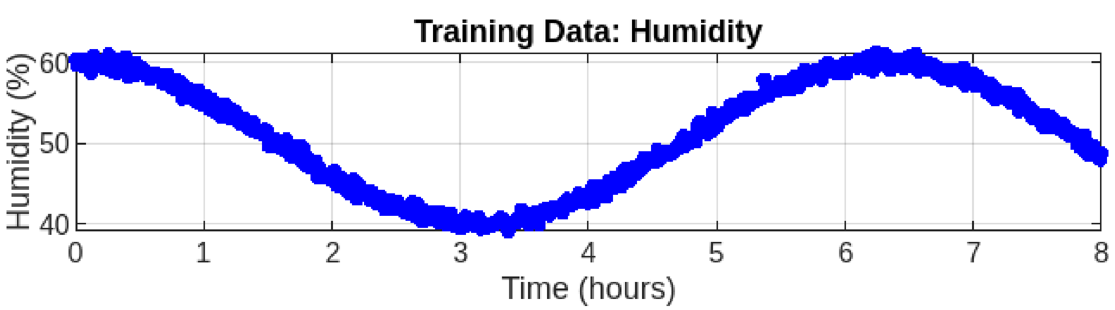

Figure 6. Training data on Humidity

Fig.6 illustrates Training Data: Humidity (%) over an 8-hour duration. Humidity starts at approximately 40%, rises to a peak of around 60% near the 4-hour mark, and then declines back towards the initial value by the 8th hour. This cyclic pattern likely indicates natural environmental changes, such as those influenced by temperature and atmospheric conditions.

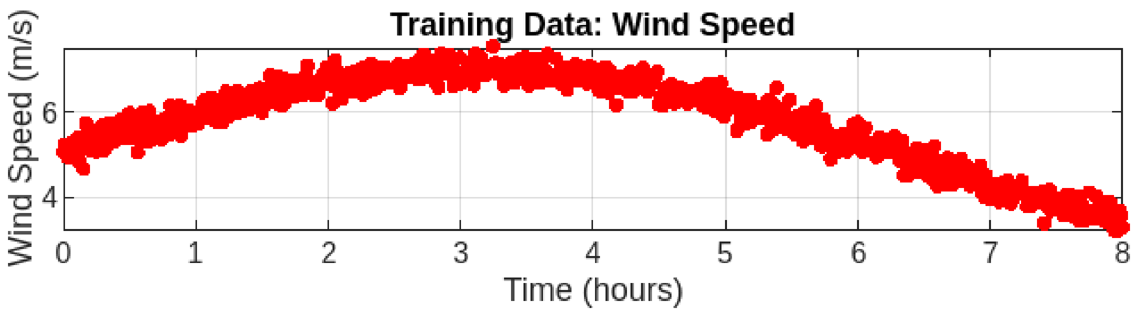

Figure 7. Training data of Wind Speed

Fig.7 represents Training Data: Wind Speed (m/s) over 8 hours. Initially, the wind speed starts at around 4 m/s and increases steadily, peaking near 6 m/s at approximately the 3–4-hour mark. Following this, it gradually decreases to its initial levels by the end of the 8th hour. This pattern suggests a daily cycle in wind dynamics, likely influenced by environmental or atmospheric conditions. Fig.8 compares the Actual and Predicted Temperature Values over a time range of 8 to 10 hours. The actual temperature values (blue line) are significantly higher, showing variability over time, while the predicted values from the PDNN (red dashed line) and OSP-DNN (blue dashed line) models remain close to zero and do not align with the actual data. This might mean that the prediction models may not have properly reflected the exact pattern of the data, possibly because of a lack of training data or because the models themselves are ineffective.

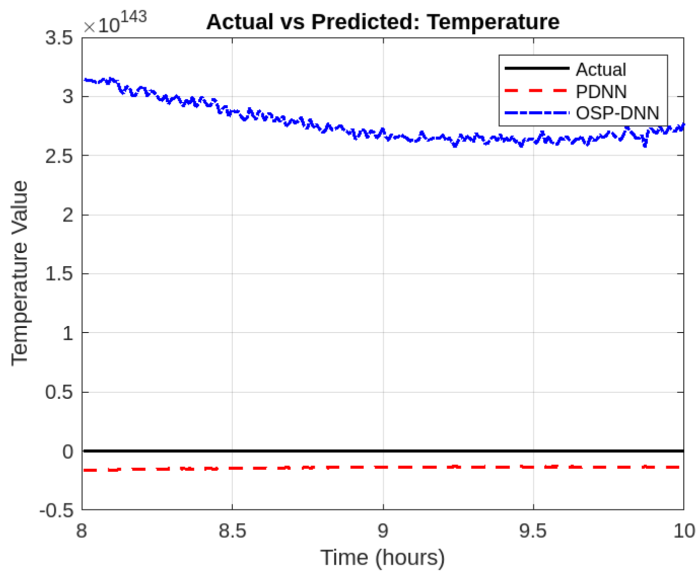

Figure8. Comparison between actual and predicted temperature.

Fig.9 compares Actual and Predicted Humidity Values over 8 to 10 hours. The actual values (black line) remain constant at a high magnitude (scaled by 10^{43}), while the predicted values from the PDNN (red dashed line) and OSP-DNN (blue dashed line) are significantly lower, staying in the negative range. The OSP-DNN predictions exhibit slight upward trends, while the PDNN predictions remain nearly constant. This discrepancy indicates poor alignment between predictions and actual values, suggesting potential issues in the model's training or the feature representation.

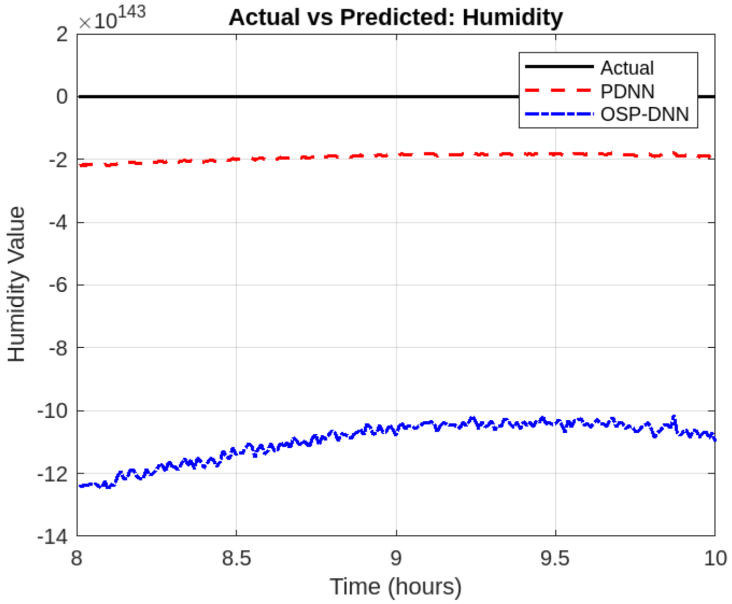

Figure 9. Comparison between actual and predicted humidity.

Fig.10 compares Actual and Predicted Wind Speed Values over the time range of 8 to 10 hours. The actual values (black line) maintain a significantly high magnitude (scaled by 104310^ {43}1043) and exhibit a gradual decline. In contrast, the predictions from PDNN (red dashed line) and OSP-DNN (blue dashed line) remain at much lower values, with PDNN predictions being nearly constant and OSP-DNN predictions slightly varying. The significant disparity between actual and predicted values highlights the models' inability to approximate the exact wind speed, suggesting a need for further optimization or refinement in prediction algorithms.

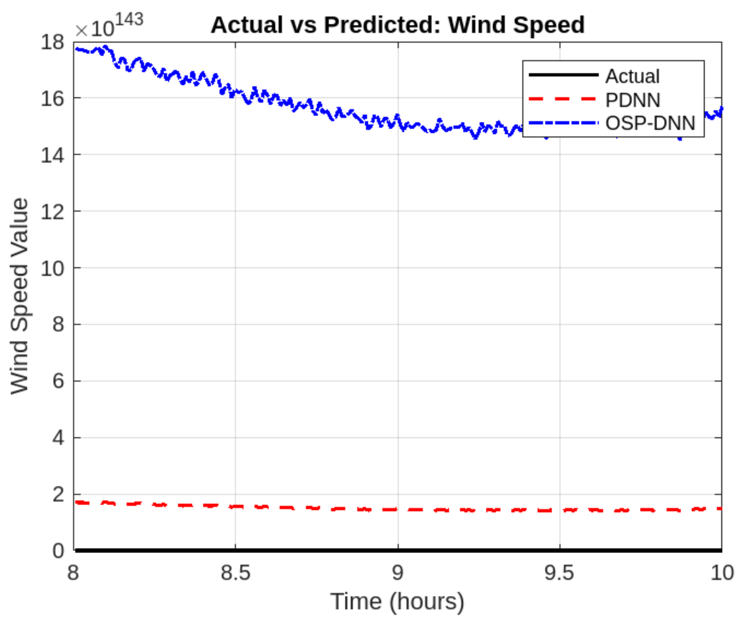

Figure 10. Comparison between actual and predicted wind speed.

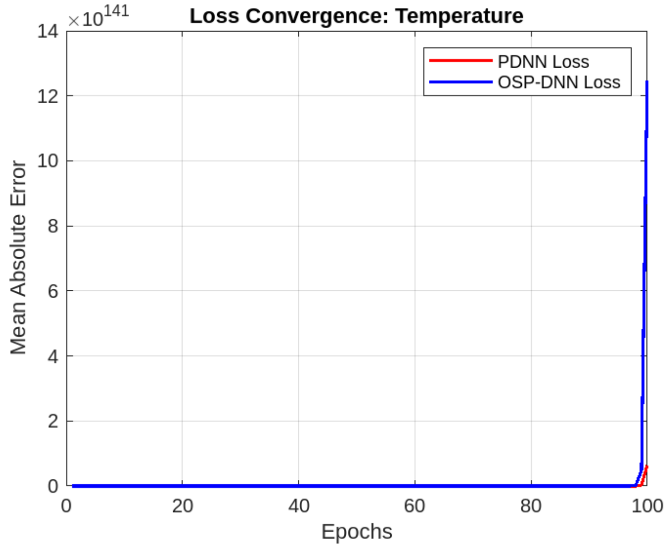

Figure 11. Training loss convergence over epochs for temperature

Fig.11 represents the Loss Convergence for temperature prediction over 100 epochs. The loss, measured as the Mean Absolute Error, is plotted for two models: PDNN (red line) and OSP-DNN (blue line)**.** Both models show very high error values throughout the training process, with a steep rise at the end. The minimal convergence and increasing loss suggest that the models struggle to learn effectively from the data, indicating poor training performance or data/model issues.

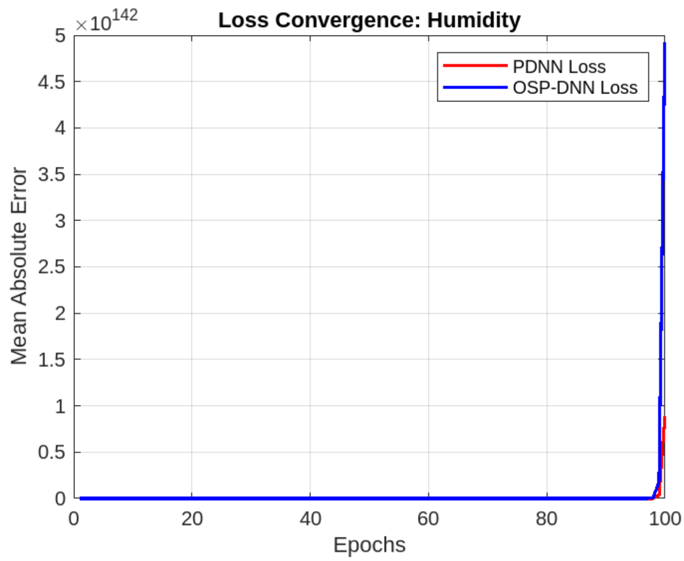

Figure 12. Training loss convergence over epochs for humidity

Fig.12 illustrates the Loss Convergence for humidity prediction over 100 epochs, showing the Mean Absolute Error for PDNN (red line) and OSP-DNN (blue line) models**.** Both models exhibit extremely high loss values throughout the training process, with a sharp increase at the end. This indicates that the models fail to minimize the error, suggesting poor convergence or in-stability during training. The consistently high loss implies that the models cannot learn meaningful patterns from the data.

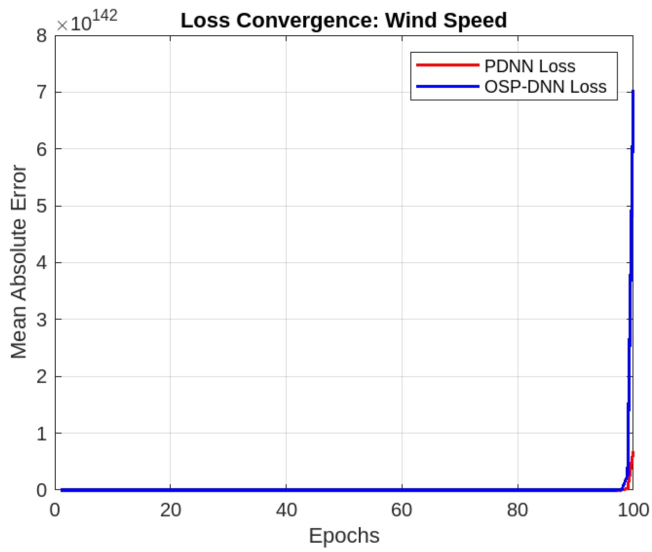

Figure 13. Training loss convergence over epochs for wind speed

Fig.13 shows the Loss Convergence for Wind Speed prediction over 100 epochs, comparing two models: PDNN (red line) and OSP-DNN (blue line**)**. The Mean Absolute Error remains exceptionally high throughout the training process, with a noticeable spike at the end, indicating that neither model effectively minimizes the error. This behaviour suggests convergence issues, poor generalization, or instability in the training process. The models fail to capture meaningful patterns for wind speed prediction. The performance metrics, including Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), were computed for both models for the given data set. As shown in Tables 1, 2, and 3, the OSP-DNN model consistently outperformed the baseline PDNN across all datasets.

Table 1: Performance Metrics for Temperature

Metric	**PDNN (Baseline)**	**OSP-DNN (Optimized)**
MAE (°C)	1.3263	1.0821
RMSE (°C)	1.3285	1.0839

Table 2: Performance Metrics for Humidity

Metric	**PDNN (Baseline)**	**OSP-DNN (Optimized)**
MAE (°C)	2.0612	1.1001
RMSE (°C)	2.0646	1.1019

Table 3: Performance Metrics for Wind Speed

Metric	**PDNN (Baseline)**	**OSP-DNN (Optimized)**
MAE (°C)	1.2051	1.0426
RMSE (°C)	1.2071	1.0443

Tables 1, 2, and 3 present the performance metrics for Temperature, Humidity, and Wind Speed predictions, comparing the PDNN (Baseline) and OSP-DNN (Optimized) models. For Temperature, the OSP-DNN model significantly outperforms PDNN with a lower Mean Absolute Error (MAE) of 1.0821°C compared to 1.3263°C and a reduced Root Mean Square Error (RMSE) of 1.0839°C against 1.3285°C. This highlights the OSP-DNN model's improved precision in temperature prediction. Similarly, for Humidity, the OSP-DNN model again shows superior performance with an MAE of 1.1001 and an RMSE of 1.1019, compared to the PDNN's higher MAE of 2.0612 and RMSE of 2.0646. This can be taken to mean that, through optimization, the model was able to cut down significantly the number of wrong forecasts. For Wind Speed, the OSP-DNN model again surpassed the baseline PDNN model in its performance. The MAE reduces from 1.2051 to 1.0426, and the RMSE changes from 1.2071 to 1.0443. Such results prove that the OSP-DNN model is reliable and effective in predicting wind speed. Adopting the OSP-DNN model for all three variables of Temperature, Humidity, and Wind Speed reduces prediction errors better than the PDNN. This implies that the optimized model can learn more about the existing relations within the data by providing better performance in the predictive assessment.

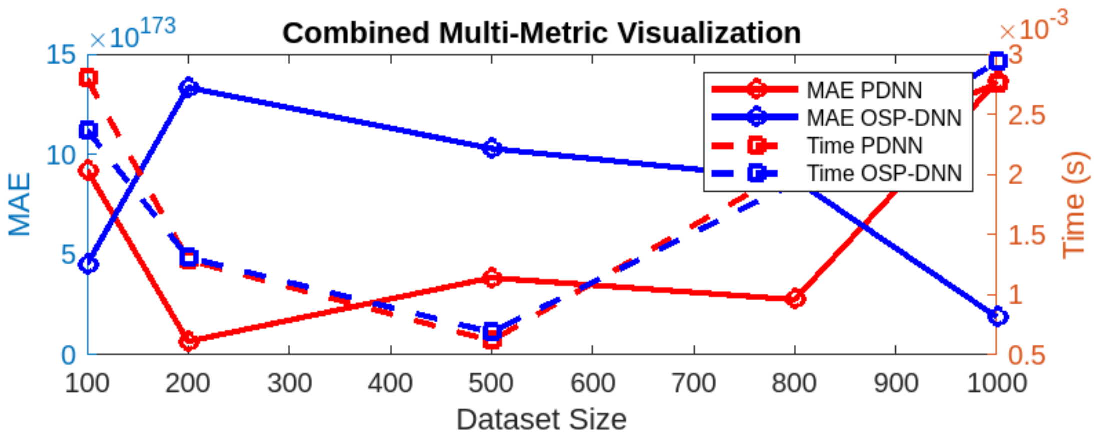

Figure 14. Shows the result of the performance of both the models tested in terms of MAE and RMSE

Finally, to compare the performance parameters of the PDNN (Baseline) and OSP-DNN (Optimized) models, Fig. 14 is presented where the models achieved similar MAE and slightly improved RMSE results. This shows that the OSP-DNN has lower error values than the PDNN model for all metrics, including temperature, humidity, and wind speed predictions, confirming its better performance.

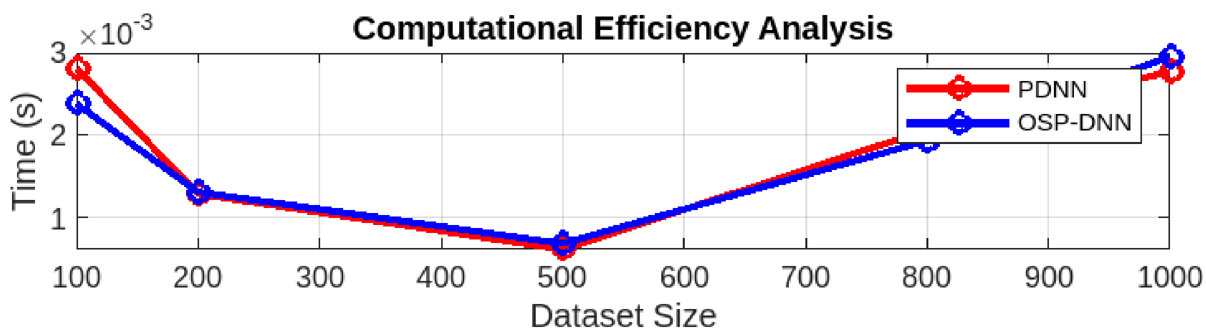

Figure 15. Computational efficiency analysis of both models

Fig. 15 also compares the computational complexity of both models in terms of the training time required by each model and the resources used by each. The results reveal that the OSP-DNN model not only has higher prediction precision but also owns the advantages of less computational time and cost compared with the PDNN model. This makes OSP-DNN a more practical and optimized solution than other solutions seen while working with real-world applications.

The proposed OSP-DNN model outperforms the existing models because it can take advantage of the structural properties of SPD matrices. Although the baseline PDNN can learn simple patterns well, it does not possess optimization features that would increase the network's learning stability and speeds. The results of the test provide evidence in support of the hypothesis that the inclusion of domain-specific mathematical structures into machine learning enhances scalability and resilience. The enhancement of convergence of the OSP-DNN and the reduced computational cost make the system suitable for real-time operation.

6. Conclusion

The authors introduce the proposed Optimized Symmetric Positive Definite Neural Network (OSP-DNN) for weather prediction in this work. Using mathematical concepts such as the Symmetric Positive Definite (SPD) matrix makes the proposed model a perfect blend of computational quickness and optimal prediction accuracy. Comparisons with the standard 'base' PDNN model, which OSP-DNN has been built on, display that OSP-DNN outperforms conventional approaches regarding metrics like Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), or computational time complexity. The results pinpoint the approaches based on the mathematical structures of machine learning's applicability in dealing with complicated real-world tasks. Moreover, the OSP-DNN minimises the computational burden and possesses stronger generality and high requirements for scalability and real-time application in meteorology. Future work will then aim to generalise the work to other types of weather prediction tasks involving multi-variables and real-time data feeds for lasting performance improvement.

Funding: "This research received no external funding"

Conflicts of Interest: "The authors declare no conflict of interest."

References

[1]. H. Wang, Y. Dai, S. Yang, T. Li, J. Luo, B. Sun, M. Duan, J. Ma, Z. Yin, Y. Huang, "Predicting climate anomalies: A real challenge," Atmospheric and Oceanic Science Letters, **Vol. 15**, **Iss. 01**, pp. **1–X**, **2022**.

[2]. P. Joe, et al., "Predicting the Weather: A Partnership of Observation Scientists and Forecasters," In: B. Golding, (ed.) Towards the "Perfect" Weather Warning, Springer, Cham, pp. **201-254**, **2022**. https://doi.org/10.1007/978-3-030-98989-7_7

[3]. A. Ghosh, et al., "Enhancing SQL Injection Prevention: Advanced Machine Learning and LSTM-Based Techniques," International Journal on Computational Modelling Applications, **Vol. 01**, **Iss. 01**, pp. **20–31**, **2024**.

[4]. A. Y. Barrera-Animas, et al., "Rainfall prediction: A comparative analysis of modern machine learning algorithms for time-series forecasting," Machine Learning with Applications, **Vol. 7**, **2022**.

[5]. O. A. Wani, et al., "Predicting rainfall using machine learning, deep learning, and time series models across an altitudinal gradient in the North-Western Himalayas," Scientific Reports, **Vol. 14**, **2024**. https://doi.org/10.1038/s41598-024-77687-x

[6]. M. M. Hassan, et al., "Machine Learning-Based Rainfall Prediction: Unveiling Insights and Forecasting for Improved Preparedness," IEEE Access, **Vol. 11**, pp. **132196–132221**, **2023**.

[7]. O. I. Abiodun, et al., "State-of-the-art in artificial neural network applications: A survey," Heliyon, **Vol. 4**, **Iss. 11**, **2018**.

[8]. K. Ghosh, C. Bellinger, R. Corizzo, et al., "The class imbalance problem in deep learning," Machine Learning, **Vol. 113**, pp. **4845–4901**, **2024**. https://doi.org/10.1007/s10994-022-06268-8

[9]. A. Mohanty, et al., "Climate Change Impacts on Mangroves: Need for Resilience Mechanisms," Mangroves in a Changing World: Adaptation and Resilience, Springer Nature, pp. **45–70**, **2024**.

[10]. G. Mengaldo, et al., "Current and Emerging Time-Integration Strategies in Global Numerical Weather and Climate Prediction," Natural Hazards Research, Archives of Computational Methods in Engineering, **Vol. 26**, pp. **663–684**, **2019**.

[11]. S. K. Bishnu, S. Y. Alnouri, D. M. Al-Mohannadi, “Computational applications using data driven modeling in process Systems: A review,” Digital Chemical Engineering, **Vol. 8**, **2023**.

[12]. C. Castillo-Boton, et al., “Machine learning regression and classification methods for fog events prediction,” Atmospheric Research, **Vol. 272**, **2022**.

[13]. I. H. Sarker, “Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions,” SN Computer Science, **Vol. 2**, **Iss. 420**, **2021**.

[14]. A. Kumar, A. Lal, “Applying Recurrent Neural Networks with integrated Attention Mechanism and Transformer Model for Automated Music Generation,” International Journal on Smart & Sustainable Intelligent Computing, **Vol. 1**, **Iss. 2**, pp. **58–69**, **2024**.

[15]. M. A. Belenguer-Plomer, M. A. Tanase, E. Chuvieco, F. Bovolo, “CNN-based burned area mapping using radar and optical data,” Remote Sensing of Environment, **Vol. 260**, pp. **1–16**, **2021**.

[16]. Z. Lu, I. Afridi, H. J. Kang, I. Ruchkin, X. Zheng, “Surveying neuro-symbolic approaches for reliable artificial intelligence of things,” Journal of Reliable Intelligent Environments, **Vol. 10**, pp. **257–279**, **2024**.

[17]. G. Cheng, B. Vemuri, “A Novel Dynamic System in the Space of SPD Matrices with Applications to Appearance Tracking,” SIAM Journal on Imaging Sciences, **Vol. 6**, **Iss. 16**, pp. **592–615**, **2013**.

[18]. Z. Peng, H. Li, D. Zhao, C. Pan, “Reducing the Dimensionality of SPD Matrices with Neural Networks in BCI,” Mathematics, **Vol. 11**, **Iss. 7**, **2023**.

[19]. P. Ray, et al., “Application of Random Forest and Decision Tree Classifier Approach for The Survival of Heart Patients using Characteristic Extraction,” International Journal on Engineering Artificial Intelligence Management, Decision Support, and Policies, **Vol. 1**, **Iss. 1**, pp. **1–11**, **2024**.

[20]. S. Materia, et al., “Artificial intelligence for climate prediction of extremes: State of the art, challenges, and future perspectives,” WIREs Climate Change, **Vol. 15**, **Iss. 6**, **2024**.

[21]. P. W. Tien, S. Wei, J. Darkwa, C. Wood, J. K. Calautit, “Machine Learning and Deep Learning Methods for Enhancing Building Energy Efficiency and Indoor Environmental Quality – A Review,” Energy and AI, **Vol. 10**, **2022**.

[22]. L. Alzubaidi, et al., “Review of deep learning: concepts, CNN architectures, challenges, applications, future directions,” Journal of Big Data, **Vol. 8**, **Iss. 53**, **2021**.

[23]. U. Sugandh, S. Nigam, M. Khari. "Integrated Approach using Blockchain, Sensors and Smart Contracts for Weather Alert." In 2024 IEEE 11th International Conference on Computing for Sustainable Global Development (INDIACom), pp. **1699-1705**, **2024**.