JCH

JOURNAL OF COMPUTING HORIZONS

Volume 1 No. 1

EDITORIAL

Journal of Computing Horizons (JCH) is an open-access, double-blind peer-reviewed journal. It publishes original, previously unpublished research across all areas of computer science and computing. It welcomes theoretical, experimental and applied research in artificial intelligence, machine learning, data science, software engineering, computer systems and networks, cybersecurity, human–computer interaction and emerging computing technologies. JCH publishes research that advances the theoretical foundations and practical applications of computing. Author(s) should declare that the work submitted to the journal is original, not under consideration for publication by another journal and that all listed authors approve its submission to JCH. It is JCH policy to welcome submissions for consideration that are original and not under consideration for publication by another journal at the same time. Author (s) can submit: Research Paper, Conceptual Paper, Case Studies and Book Review. The current issue of JCH comprises papers of scholars from different universities of Pakistan and Malaysia. Systematic Investigation of Xception for Melanoma Classification Across Data Imbalance and Training Configurations, Convolutional Neural Network and Transformer-Based Visual Deepfake Detection, Explainable Crime Prediction Interpreting Machine Learning and Deep Learning Models for the Urban Safety Decision Making, Hybrid Deep Learning and Graph-Based Framework for Crime Hotspot Detection and Prediction and Assessment of Ensemble and Deep Learning Methods to Introduce Precise Temperature Categorization in Several Zoned Climate Data are some of the major practices and concepts examined in these studies. All the submitted papers were first assessed by the journal committee and then the external editorial team for relevance and originality of the work and then blindly peer-reviewed by external reviewers depending on the subject matter of the paper. After the rigorous peer-review process, the submitted papers were selected based on originality, significance and clarity of the purpose. The issue will, therefore, be a unique proposition, where scholars will be able to appreciate the latest results in their field of expertise and to acquire additional knowledge in other relevant fields.

Editor-in-Chief

Dr Umair Muneer, Ph.D

EDITORIAL BOARD

EDITOR

- Umair Muneer, PhD, University of Management and Technology Sialkot, Pakistan

EDITORIAL AND ADVISORY TEAM

- Sukumar Letchmunan, PhD, Universiti Sains Malaysia, Malaysia
- Anees Baqir PhD, Northeastern University, United Kingdom
- Husnain Mushtaq, PhD, South China University of Technology, China
- Mubashir Ali, PhD, University of Birmingham, United Kingdom
- Mahum Adil, PhD, University of Glasgow, United Kingdom

TABLE OF CONTENTS

DESCRIPTION	PAGES
Title	I
Editorial	II
Editorial Board	III
Table of Contents	IV
Research Papers	**V**
A Systematic Investigation of Xception for Melanoma Classification Across Data Imbalance and Training Configurations *Naila Batool, Aqsa Akram, Amna Zaheer*	1-13
Convolutional Neural Network and Transformer-Based Visual Deepfake Detection: A Comparative Analysis *Muhammad Salman, Iqra Tariq, Zameer Ahmad, Naila Batool*	14-24
An Assessment of Ensemble and Deep Learning Methods to Introduce Precise Temperature Categorization in Several Zoned Climate Data *Iqra, Ineta Shahid, Naila Batool, Umair Muneer Butt*	25-39
A Hybrid Deep Learning and Graph-Based Framework for Crime Hotspot Detection and Prediction (GNN+CNN/LSTM) *Alina Arif, Zuha Rani, Aqsa Akram, Arham Mustafa, Umair Muneer Butt*	40-50
Explainable Crime Prediction: Interpreting Machine Learning and Deep Learning Models for Urban Safety Decision-Making *Kinza Noor, Aiza Amin, Athar Rahim, Aqsa Akram*	51-59

RESEARCH PAPERS

JOURNAL OF COMPUTING HORIZONS
Vol. 01, No. 01, 2026
DOI: 10.22610/JCH.V1I1.4991

RESEARCH ARTICLE | PEER REVIEWED

A Systematic Investigation of Xception for Melanoma Classification Across Data Imbalance and Training Configurations

Naila Batool [1] *, Aqsa Akram [1], Amna Zaheer [1]
[1] *Department of Computer Science, University of Management and Technology, Lahore (Sialkot Campus), Pakistan*
* **Corresponding Author:** nailabatool617@gmail.com

ABSTRACT – Deep learning has demonstrated considerable potential for automated skin lesion classification; however, its performance is influenced by dataset characteristics, class distribution, and model configurations. This study presents an experimental evaluation of the Xception architecture for melanoma skin cancer classification, investigating the effects of dataset balancing, data proportions, training–testing ratios, and hyperparameter configurations on classification performance and computational efficiency. Balanced and imbalanced datasets are evaluated at data proportions of 25%, 50%, 75%, and 100%, while five training–testing ratios ranging from 50:50 to 90:10 are investigated. We also examine different hyperparameter configurations to assess their influence on model performance. Classification effectiveness is evaluated using accuracy, precision, Recall, F1-score, and ROC-AUC, while training time, testing time, memory consumption, and total execution time are considered to assess computational requirements. Experimental results indicate that balanced datasets generally provide more consistent classification performance across the evaluated metrics. Among the investigated training–testing ratios, the 80:20 configuration achieves the highest accuracy of 86.31%, F1-score of 85.35%, and ROC-AUC of 93.88%. Furthermore, the results highlight the relationship between training data availability, classification performance, and computational cost. This study provides a comprehensive experimental assessment of Xception and highlights the importance of dataset composition and experimental configuration in melanoma classification.

Keywords: *Melanoma classification; Xception; deep learning; class imbalance; dataset proportions; training–testing ratio; computational efficiency.*

I. INTRODUCTION

Melanoma is a particularly aggressive variant of cutaneous cancers requiring early detection to improve patient prognosis [1, 2]. Dermoscopy provides valuable information about the visual characteristics of cutaneous lesions during clinical evaluations. Diagnosing melanoma, however, is challenging due to significant intra-class differences in lesion appearance, pigmentation patterns, morphological characteristics, and texture [3, 4]. Several recent publications explore the feasibility of artificial intelligence-based automatic skin lesion classification using deep learning algorithms [5, 6].

Deep Convolutional Neural Networks show substantial promise for automatically extracting features from medical images in clinical applications [7, 8]. One example of a CNN architecture is the Xception network, which uses depthwise separable convolutions to automatically extract features while minimizing computational cost compared to traditional convolution layers [9, 10]. While deep learning models have been successfully used to classify skin lesions, their accuracy depends significantly on various experimental parameters, such as the data composition, the number and balance of the different classes, the size of the training sets, and the architectural choices made when designing deep learning networks [11, 12, 13].

Another problem to be considered is the presence of class imbalance in melanoma diagnoses because unequal numbers of samples will impair model performance for minority classes [14, 15]. Additionally, modifying the proportions of training and testing data may affect classification performance [16, 17, 18]. Moreover, varying the split ratio between training and testing datasets influences classification results [16, 17, 18]. An increase in training data does not always improve classification accuracy; it depends on the evaluation metrics used [19, 20]. The computational cost, including training, inference, and memory overhead, is an essential factor in evaluating the practical applicability of deep neural networks [21]. As such, evaluating classification performance alongside computational cost across different experimental settings is necessary to gain a holistic view of the model.

Driven by these observations, we conduct experiments investigating the effect of balanced and imbalanced data splits at various proportions, explore the effect of split ratios between training and testing sets of 50:50, 60:40, 70:30, 80:20, and 90:10, investigate the training process costs, and examine the effects of hyperparameter tuning on classification performance. This investigation evaluates classification performance using accuracy, precision, Recall, F1-score, and ROC-AUC, while computational costs include training time, inference time, and memory utilization. Some main findings of this investigation include the following points:

- Xception is evaluated on balanced and imbalanced datasets at data proportions of 25%, 50%, 75%, and 100% to investigate the influence of class distribution and dataset availability on classification performance.
- Five different training–testing configurations (50:50, 60:40, 70:30, 80:20, and 90:10) are investigated. The 80:20 ratio yields the highest accuracy, F1-score, and ROC-AUC (86.31%, 85.35%, and 93.88%, respectively) among the ratios tested.
- Training time, testing time, memory consumption, and total execution time are analyzed to examine the computational requirements of Xception under different experimental configurations.
- Multiple classification metrics are considered to assess predictive performance and identify variations attributable to dataset distribution and training data availability.

This paper is structured into sections as outlined below. Section II introduces the data used and the methods employed in the experiments. Section III analyzes the methodology, Section IV discusses the experimental setup, whereas Section V discusses those outcomes. The conclusion is provided in Section VI.

II. RELATED WORK

Deep learning techniques have been widely explored for automated skin lesion classification, particularly convolutional neural networks (CNNs), transfer learning approaches, and hybrid models. Xception employs depthwise separable convolutions to extract features from images with minimal computational cost. In research using the Xception network in combination with various pre-trained CNN models on the HAM10000 dataset, Mahmud et al. [1] achieved an accuracy of 88.72% and a weighted F1 score of 0.89. Furthermore, Shah et al. [10] incorporated the Xception model with particle swarm optimization and subspace KNN classifiers, achieving 98.5% accuracy on the ISIC 2018 dataset. Other studies considered different architectures, including CNN-Vision Transformer fusion [11], Vision Transformer-based classification [3], and SynthraXCoreNet ensemble [8]. Other investigations examined different methodologies, highlighting developments within deep learning approaches for skin lesion analysis.

Class imbalance is another important issue, given that most skin lesion datasets exhibit a high discrepancy in the number of samples per category. Many studies have resolved this problem through class balancing, resampling, and augmentation techniques. Class weighting and stratified sampling were used by Cherukuri and Vara Prasad [5]; federated learning and explainable AI approaches, and strategies that address class imbalance issues were tested by Jaya et al. [14], whereas adversarial domain adaptation was explored by Gilani et al. [13]. Bias analysis in skin lesion classifiers was conducted by Corbin et al. [16]. Other studies investigated model reliability and generalizability, particularly across different distribution datasets, such as metric-based few-shot learning [12], domain-adaptive Vision Transformer ensembles [17], and active domain adaptation [18]. These studies demonstrate the significance of the dataset composition and distribution when evaluating the performance of skin lesion classifiers.

Model configuration and optimization also influence classification performance. Raza et al. [2] investigated optimization strategies and reported strong results using Adam with a customized activation function. Mahmud et al. [1] and Malhotra and Kaur [9] also examined pretrained CNNs and optimization configurations for skin lesion classification. Priambodo and Fatichah [15] investigated the application of lightweight CNN models enhanced by the Convolutional Block Attention Module (CBAM). Benaly et al. [7] analyzed Grad-CAM and Grad-CAM++, while Jadhav et al. [19] used Grad-CAM, SHAP, and LIME to explain their models. Both studies have made complementary contributions regarding optimization, feature representation, and interpretability within artificial intelligence applications.

Computational efficiency is another critical factor, especially when computing resources are constrained. Vinitha et al. [21] studied the application of ensemble lightweight CNN models in edge computing, while Lubis et al. [20] applied ReXNet-150 to accelerate and improve melanoma detection efficiency. Kollem et al. [4] developed a novel framework called SKIN-MARS, which uses Multi-Scale Swin Transformer networks for skin lesion segmentation, demonstrating the high computational expense of such analysis tasks. Despite recent progress, many studies focus on specific architectures, dataset configurations, and performance metrics. Systematic evaluation of Xception across varying dataset proportions, different class distributions, various training-testing splits, and computational metrics has not yet been performed. Therefore, this work will experimentally evaluate Xception for melanoma skin cancer classification by examining classification performance across several datasets and estimating training time, testing time, memory usage, and execution time.

III. METHODOLOGY

This section presents the methodology using the Xception pretrained model. The methodology begins with details of the datasets used in the experiments, followed by tables summarizing general and experiment-specific hyperparameters. A methodology diagram, shown in Figure 1, visually illustrates the process flow throughout model training and evaluation.

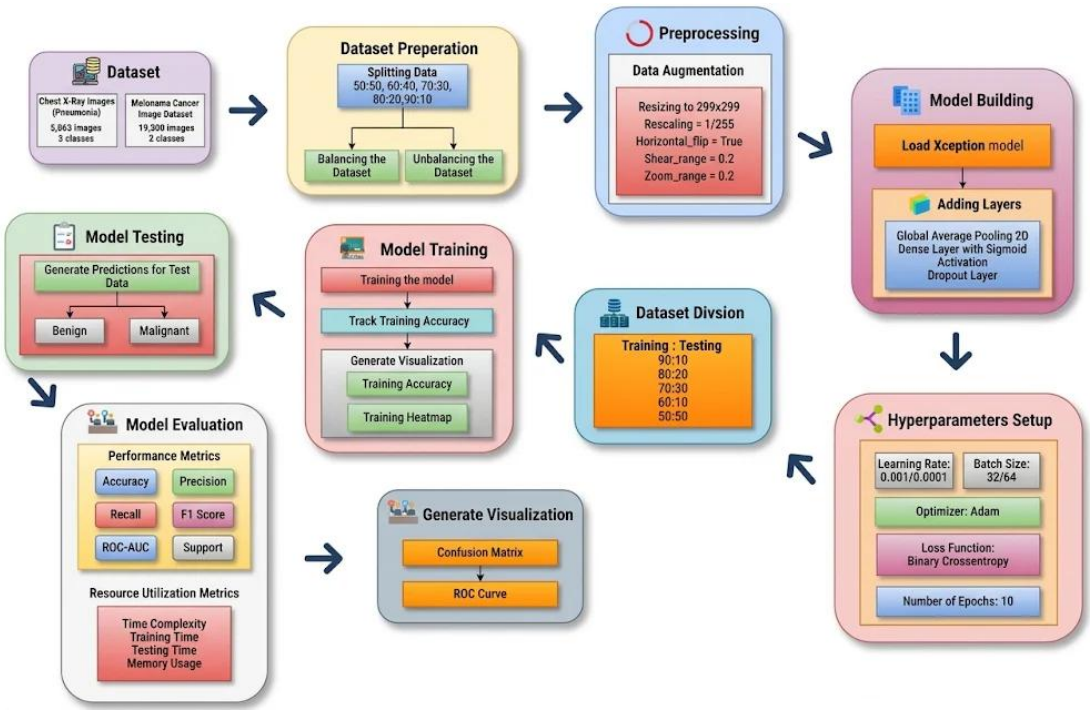


FIGURE 1: Methodology Diagram

Figure 1 outlines the overall process used in the experiment; the methodology starts with the selection of datasets, with two datasets loaded into memory to make them accessible for model training. After loading, the datasets were split into four combinations of different data sizes: 25%, 50%, 75%, and 100%. These combinations were then processed under two scenarios: unbalanced and balanced datasets. For each of these scenarios, the datasets were further divided into different training and testing ratios, including 20:80, 50:50, 70:30, and 80:20. By performing this step, the developed model was tested on different sizes of datasets to assess its robustness and adaptability towards different class distributions and splits for training/testing. After training, the model was tested on the test set, which classified the pictures as either benign or malignant. Evaluation metrics such as accuracy, precision, Recall, F1 score, ROC-AUC, and support were computed to assess the model's performance. The evaluation of performance involves tracking time complexity, training time, and memory usage to provide a complete picture of the model's efficiency and effectiveness.

A. Datasets Description

Melanoma Skin Cancer Dataset is an organized collection of 13900 RGB images sourced from Kaggle, uniformly resized to 224x224 pixels in JPEG format, totaling 83.27 MB. The dataset was created to facilitate improvements in diagnosis by leveraging advanced machine-learning models in dermatology. The images in the dataset are distributed into two categories: Benign and Malignant. These two classes are further divided into two subsets: training and testing. The training set contained 6289 images in the benign class and 5590 in the malignant class. Likewise, the testing set consists of 1000 benign images and 1000 malignant images. The high-quality images make it a useful resource for early detection and accurate diagnosis in melanoma classification, as shown in Figure 2.

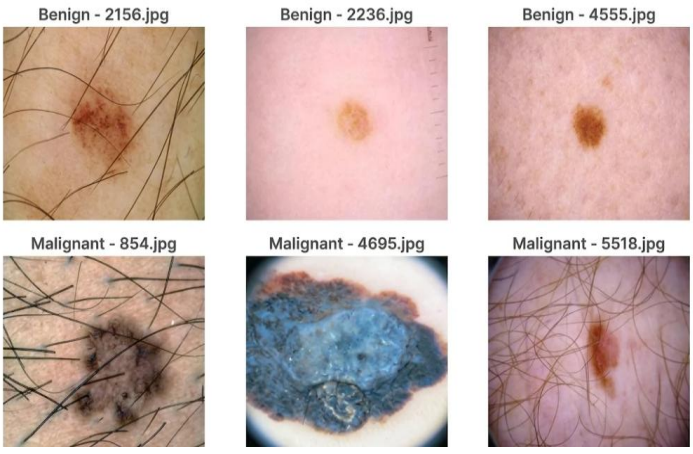


FIGURE 2: Melanoma Cancer Image Dataset

B. Experimentation in Balanced and Unbalanced Datasets

For this experiment, two combinations of the datasets were made as balanced and unbalanced. The following Table 1 presents the distribution of images in each class (Benign and Malignant) for both balanced and unbalanced datasets across various dataset sizes (25%, 50%, 75%, and 100%).

TABLE 1: Distribution of Balanced and Unbalanced Datasets

Dataset Size	Train Subset (Balanced)	Test Subset (Balanced)	Train Subset (Unbalanced)	Test Subset (Unbalanced)
25% Dataset	Benign: 1572, Malignant: 1397	Benign: 250, Malignant: 250	Benign: 628, Malignant: 838	Benign: 150, Malignant: 100
50% Dataset	Benign: 2795, Malignant: 2795	Benign: 500, Malignant: 500	Benign: 1886, Malignant: 1118	Benign: 200, Malignant: 300
75% Dataset	Benign: 4192, Malignant: 4192	Benign: 750, Malignant: 750	Benign: 1886, Malignant: 2515	Benign: 450, Malignant: 300
100% Dataset	Benign: 5590, Malignant: 5590	Benign: 1000, Malignant: 1000	Benign: 3773, Malignant: 2236	Benign: 400, Malignant: 600

Figure 3 shows a comparison between the train and test datasets. The distribution of Benign and Malignant images across balanced and unbalanced datasets for different sizes (25%, 50%, 75%, and 100%).

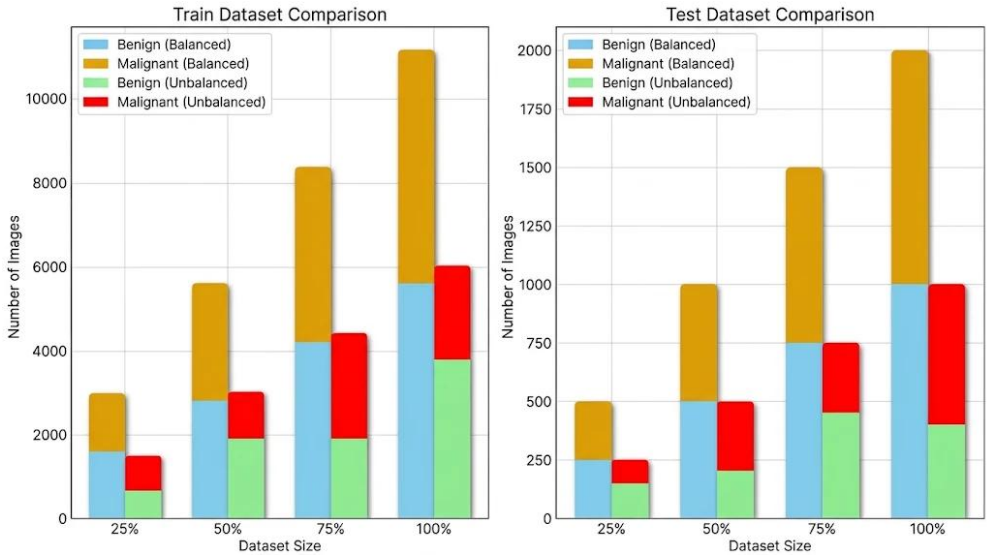


FIGURE 3: Train and Test Dataset Comparison

C. Method to Balance the Dataset

In balancing the dataset, first, to assess the class imbalance, the number of images in each class was counted. After counting, images were randomly selected and copied from the original dataset into a new directory to create a subset of the dataset for training and testing. Next, to balance the dataset, a minimum class size was determined as a reference. Using this reference, random images were selected from each class to match the smaller class's size. Lastly, the images of both classes were balanced to ensure a balanced dataset. The following steps outline the process of splitting and also balancing the dataset. A Process of splitting the dataset (in this case, 50%) is shown in Figure 4 and Figure 5.

D. Method to Unbalance the Dataset

To create an unbalanced dataset, the first step is to create a subset by randomly selecting images from each class. Different fractions of images were selected for each class to induce the class imbalance. According to the selected fractions, images were distributed to each class to create an imbalanced dataset. For storing these imbalanced subsets, separate directories were created. Lastly, the images of both classes were stored to ensure class imbalance. The following steps outline the process of splitting and balancing the dataset. A Process of splitting the dataset: 50% dataset, setting fractions as 60:40 for training and 40: 60 for testing classes (Benign: Malignant) shown in Figure 6.

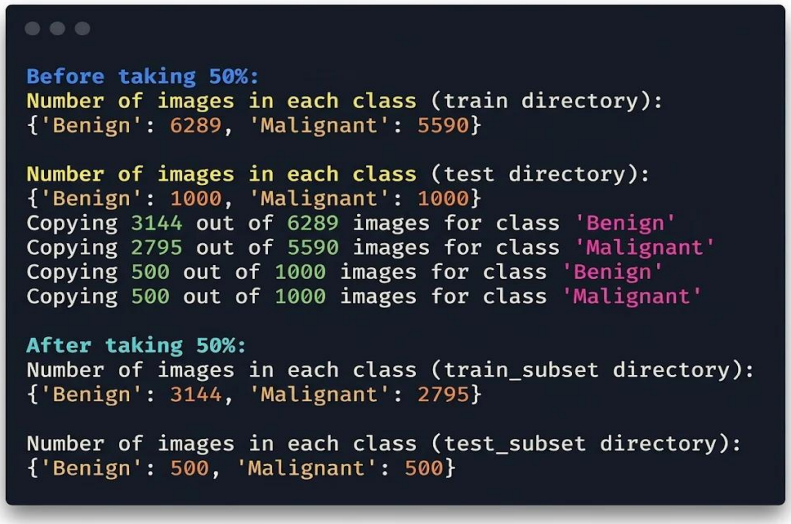


FIGURE 4: Splitting Dataset into 50%



FIGURE 5: 50% Balanced Dataset

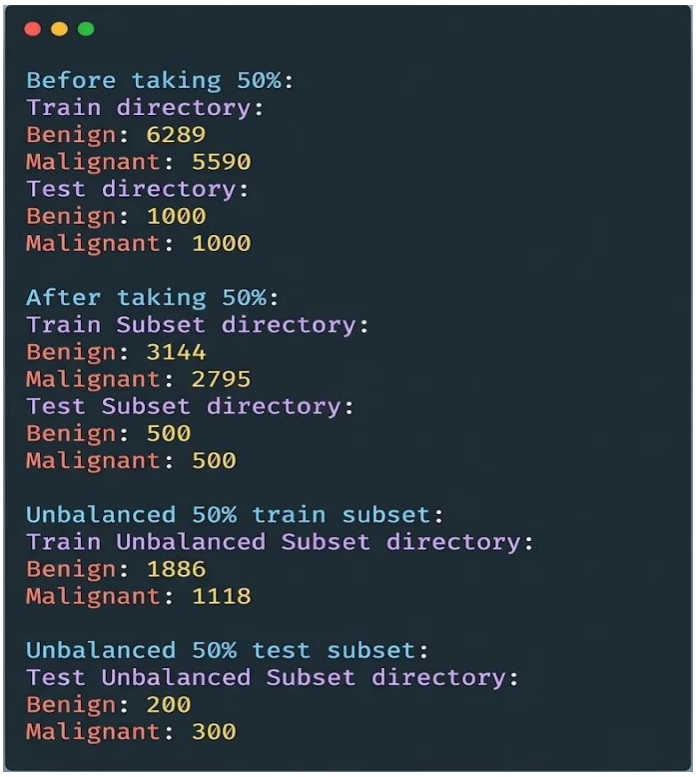


FIGURE 6: Process of Unbalancing the Dataset

IV. EXPERIMENTAL SETUP

A. Configuration and Tools

The computational analysis used a hybrid approach comprising local and cloud computing. For the local machine, the simulation was run using Microsoft Windows 10 Pro (x64) with an Intel Core i5-6200U processor and 8 GB of RAM. In contrast, the cloud computing part used Google Colab for free, which offers a T4 GPU, 97.0 GB of disk space, and 12 GB of RAM – adequate for deep learning computations. Python 3.8 was chosen for its compatibility with deep learning frameworks. Two datasets utilized during this study were sourced from Kaggle repositories [1, 2].

B. Experimentation Hyperparameters

The hyperparameters fine-tuned for the experiments, along with their values, are listed in Table 2. From this table, two hyperparameters were selected for tuning: Learning Rate (0.001, 0.0001) and batch Size (32, 64). The combination LR = 0.001 and batch = 32 strikes an optimal compromise between efficiency and stability during training. Combination 2 (LR=0.0001, Batch=64) optimizes the model's ability to accelerate training. Table 3 shows the results of two different hyperparameter combinations utilizing different metrics.

TABLE 2: Experimentation Hyperparameters

Hyperparameter	Value
Learning Rate	0.001, 0.0001
Batch Size	32, 64
Epochs	10
Optimizer	Adam
Loss Function	Binary Crossentropy
Data Augmentation	Resizing: 299x299, Zoom Range: 0.2, Rescaling: 1/255, Shear Range: 0.2, Horizontal Flip: True
Base Model	Xception (Pretrained on ImageNet)

Specifically, with a learning rate of 0.001 and a batch size of 32, the neural network achieves an accuracy of 87.93%. On the other hand, the second combination, with LR = 0.0001 and batch size = 64, achieves an accuracy of 84.02%. Resource requirements differ substantially; thus, combination 1 spends 3031.73 seconds for training and 20.60 seconds for testing, whereas combination 2 takes 3043.21 seconds to complete the training stage & 15.46 seconds to test its performance. Memory consumption for combination 1 amounts to 2108.95 megabytes, while for combination 2 – 2406.97 megabytes. As far as accuracy scores go, however, combination 2 offers better stability during training and improved generalization at the expense of extended training duration.

Furthermore, combination 1 converges much faster and achieves higher training accuracy but exhibits severe overfitting and insufficient generalization to the validation dataset. Combination 2 shows superior stability, better generalization, and a lower tendency towards overfitting, although its training takes significantly longer to complete. Overall, combination 2 strikes a balance between stability and generalization, yielding higher accuracy on novel datasets.

TABLE 3: Different Settings of Hyperparameters

Metrics Combination	Combination 1 (LR=0.001, Batch=32)	Combination 2 (LR=0.0001, Batch=64)
Accuracy	87.93%	84.02%
Training Time	3031.73 seconds	3043.21 seconds
Testing Time	20.60 seconds	15.46 seconds
Memory Usage	2108.95 MB	2406.97 MB
Time Complexity	$O(1.29 \times 1012)$ operations	$O(1.29 \times 1012)$ operations

V. RESULTS AND DISCUSSION

A. Results Based on Performance Metrics

The results in Table 4 show significant differences in model performance across metrics. The balanced dataset offered an accuracy of 85.6%, precision of 89.7%, Recall of 80.0%, and an F1 score of 84.8%, with a ROC-AUC of 93.3%, when applied to the 25% dataset. However, when the dataset was unbalanced, accuracy dropped slightly to 83.6%; precision fell to 74.79%; recall increased to 89%; F1-score fell to 81.28%; and ROC-AUC remained the same at 88.99%. For the 50% dataset, the balanced dataset showed higher overall accuracy, achieving 86.5% accuracy, 91.57% precision, 80.4% recall, 85.62% F1-score, and 94.46% ROC-AUC. In the case of an unbalanced dataset, there is a significant decrease in accuracy to 68.8%, a considerably higher ROC-AUC of 93.98%, an imbalanced precision of 95.57%, reduced Recall to 50.33%, and a much lower F1 score of 65.94%. With the 75% dataset, the balanced dataset maintained better performance with an accuracy of 86.47%, precision of 92.01%, recall of 79.87%, F1-score of 85.51%, and ROC-AUC of 94.87%.

TABLE 4: Results Based on Performance Metrics

Dataset	Balanced					Unbalanced				
Metrics	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**ROC-AUC**	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**ROC-AUC**
25%	85.60%	89.73%	80.04%	84.80%	93.30%	83.60%	74.79%	89.00%	81.28%	88.99%
50%	86.50%	91.57%	80.40%	85.62%	94.46%	68.80%	95.57%	50.33%	65.94%	93.98%
75%	86.47%	92.01%	79.87%	85.51%	94.87%	86.13%	79.70%	87.67%	83.49%	93.71%
100%	87.80%	92.00%	82.80%	87.16%	95.33%	74.50%	95.04%	60.67%	74.06%	93.91%

On the other hand, in an unbalanced dataset, an accuracy of 86.13%, precision of 79.7%, Recall of 87.67%, F1-Score of 83.49%, and ROC-AUC of 93.71% were obtained, indicating improvement over the results for the 50% unbalanced dataset. Lastly, considering the 100% dataset, the accuracy of the balanced dataset was 87.8%, precision 92%, recall 82.8%, F1 score 87.16%, and ROC-AUC 95.33%, indicating that the model outperformed in this combination. However, the unbalanced dataset performs poorly, achieving an accuracy of 74.5%, precision of 95.04%, Recall of 60.67%, F1-score of 74.06%, and a ROC-AUC of 93.91%. The graphs in Figure 8 show results for different metrics using different combinations of balanced and unbalanced datasets.

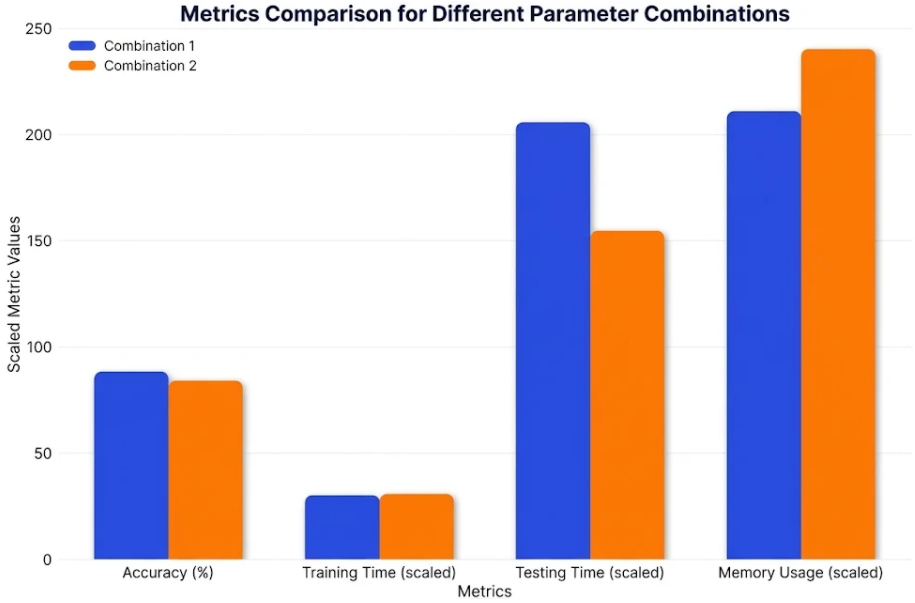


FIGURE 7: Comparison of Two Combination Settings Based on Different Performance Metrics

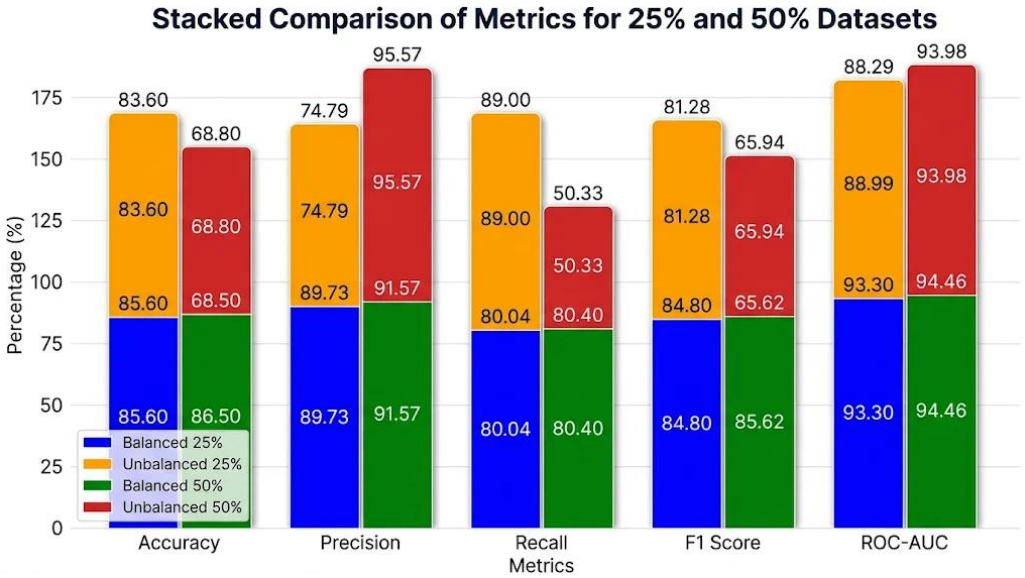


FIGURE 8: Comparison of 25% and 50% of Balanced and Unbalanced Data

B. Results Based on Resource Usage

On the 25% dataset, the balanced dataset had a training time of 373.01 seconds and a test time of 7.59 seconds, for a total time complexity of 380.97 seconds, with a memory consumption of 2.536 GB. However, the unbalanced dataset results in a training time of 303.72 s, a testing time of 6.77 s, a memory usage of 2.64 GB, and a total time complexity of 310.48 s. For the 50% combination, the balanced dataset took 658.93 seconds (638.09 seconds for training and 20.84 seconds for testing) to process, with a memory usage of 2.69 GB. For an unbalanced dataset, training takes 406.04 seconds, and testing takes 8.39 seconds, for a total time complexity of 414.44 seconds, with a slight decrease in memory usage to 2.52 GB. On the balanced dataset, training on 75% of the data took 935.99 seconds, and testing took 14.95 seconds, resulting in a total time complexity of 950.93 seconds and a memory usage of 2.62 GB.

On the unbalanced dataset, the training and testing times were 748.61 seconds and 11.14 seconds, respectively, yielding an overall time complexity of 759.75 seconds, and the lowest observed memory usage was 2.25 GB across all configurations. Lastly, when the dataset was balanced, it yielded the highest training time of 1329.59s, a testing time of 20.82 s, a total time complexity of 1350.41 s, and a memory usage of 2.54 GB. On the other hand, the unbalanced dataset has a much lower training time of 791.20 s and a testing time of 20.83 s, resulting in a total time complexity of 812.03 s and memory usage of 2.57 GB. The comparison graph of memory usage and time complexity is shown as follows in Figure 9.

TABLE 5: Results Based on Resource Usage

Dataset Combination	Training Time (s)	Testing Time (s)	Total time (s)	Memory (GB)
25% – Balanced	373.01	7.59	380.97	2.536
25% – Unbalanced	303.72	6.77	310.48	2.64
50% – Balanced	638.09	20.84	658.93	2.69
50% – Unbalanced	406.04	8.39	414.44	2.52
75% – Balanced	935.99	14.95	950.93	2.62
75% – Unbalanced	748.61	11.14	759.75	2.25
100% – Balanced	1329.59	20.82	1350.41	2.54
100% – Unbalanced	791.20	20.83	812.03	2.57

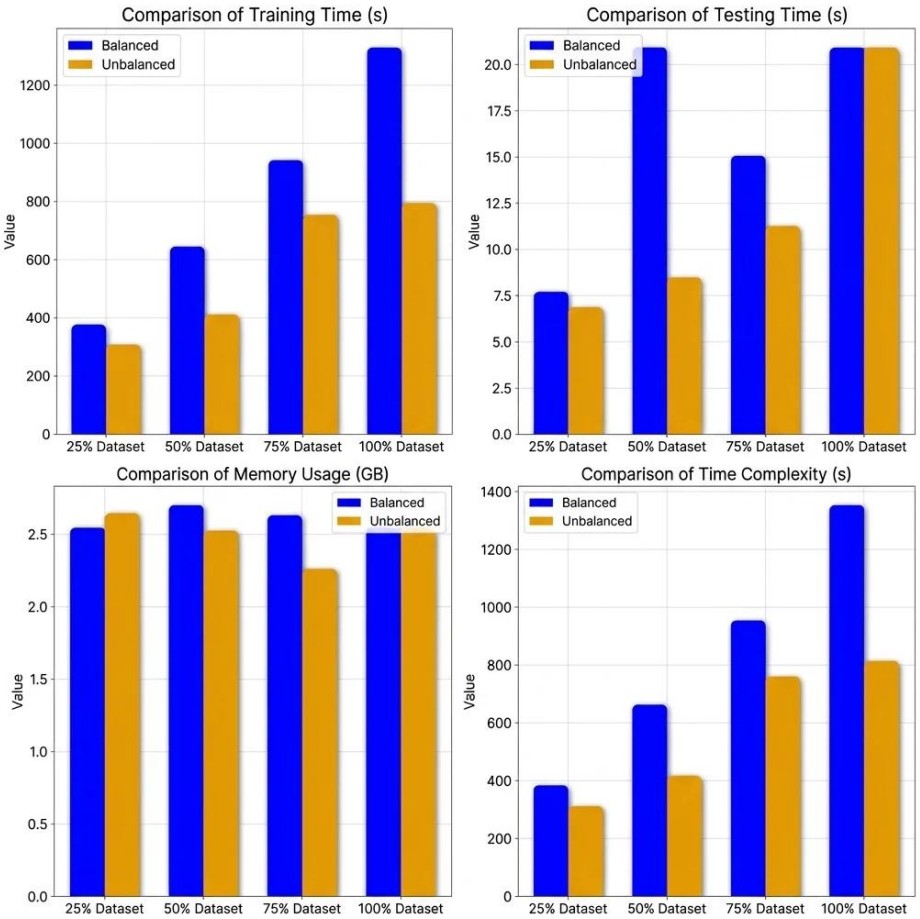


FIGURE 9: Comparison of Training Time, Testing Time, Memory Usage and Time Complexity

C. *Experimentation using Different Ratios of Training and Testing*

For this experiment, different dataset ratios were used to analyze the model's performance. Table 6 presents the distribution of images in each class (Benign and Malignant) across different ratios (90:10, 80:20, 70:30, 60:40, and 50:50).

D. *Results of Different Training-Testing Ratios*

Based on Table 7, for 50%-50%, 60%-40%, 70%-30%, 80%-20%, and 90%-10% splits, the 80%-20% split proved to be the best across all metrics such as accuracy (86.31%), precision (86.76%), Recall (85.35%), and ROC-AUC (93.88%). It is observed that the 70%-30% split yielded the minimum training time (1149.96s) and time complexity (1191.14s), while the 90%-10% split had the maximum Recall (85.58%). The lowest memory usage was observed for the 70%-30% split (2.27GB), and the testing time was lowest for the 90%-10% split (13.03s).

TABLE 6: Dataset Distribution

Ratio	Train Dataset	Test Dataset
Training 50%, Testing 50%	Malignant: 3295 images Benign: 3644 images	Malignant: 3295 images Benign: 3645 images
Training 60%, Testing 40%	Malignant: 3954 images Benign: 4373 images	Malignant: 2636 images Benign: 2916 images
Training 70%, Testing 30%	Malignant: 4613 images Benign: 5102 images	Malignant: 1977 images Benign: 2187 images
Training 80%, Testing 20%	Malignant: 5272 images Benign: 5831 images	Malignant: 1318 images Benign: 1458 images
Training 90%, Testing 10%	Malignant: 5931 images Benign: 6560 images	Malignant: 659 images Benign: 729 images

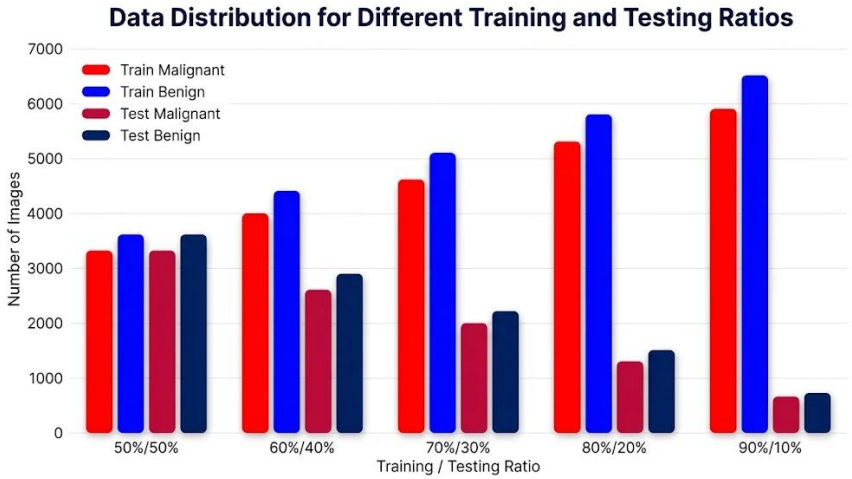


FIGURE 10: Distribution of the Training and Test Datasets Across Different Training-Testing Ratios

VI. DISCUSSION

A. *Analysis of Results Based on Balanced and Unbalanced Dataset*

In this, results are analyzed using two sets of metrics: performance metrics such as Accuracy, Precision, Recall, F1-score, and ROC-AUC. Resource utilization metrics include training time, testing time, memory usage, and time complexity.

TABLE 7: Metrics Evaluation Across Different Training-Testing Ratios

Metrics	Training 50%, Testing 50%	Training 60%, Testing 40%	Training 70%, Testing 30%	Training 80%, Testing 20%	Training 90%, Testing 10%
Accuracy	84.78%	85.36%	85.76%	86.31%	85.95%
Precision	85.24%	86.95%	86.50%	86.76%	84.94%
Recall	82.19%	81.37%	82.95%	83.99%	85.58%
F1 Score	83.68%	84.07%	84.69%	85.35%	85.26%
Support	6940	5552	4164	2776	1388
ROC-AUC	92.49%	93.08%	93.44%	93.88%	93.43%

Training Time	1244.97s	1304.31s	1149.96s	1339.47s	1228.83s
Testing Time	49.67s	40.56s	41.18s	26.61s	13.03s
Memory Usage	2.96GB	2.68GB	2.27GB	2.65GB	2.67GB

B. Analysis of Results Based on Performance Metrics

For the balanced dataset, as dataset size increases, the model's overall performance improves in terms of accuracy, precision, Recall, F1 score, and ROC-AUC. The balanced dataset yields better results because each class has the same number of images, making it easier for the model to learn to detect both benign and malignant cases. On the other hand, there are fluctuations in an unbalanced dataset in terms of accuracy, such as a decrease to 50%, followed by increases to 75% and 100%. This indicates that larger unbalanced datasets do not always yield better results. For the unbalanced dataset, precision reaches 100% because the minority class is predicted correctly on any given sample. Recall, which measures the ability to predict the minority class correctly, drops from 100% to 50%, then improves to 75% but remains lower than in the balanced dataset. In Figure 12, F1 score exhibits behavior similar to Recall and precision, decreasing to 50% and increasing with larger datasets. Increasing the size of the unbalanced dataset improves ROC-AUC, but it remains lower than that of the balanced dataset, indicating poorer class discrimination.

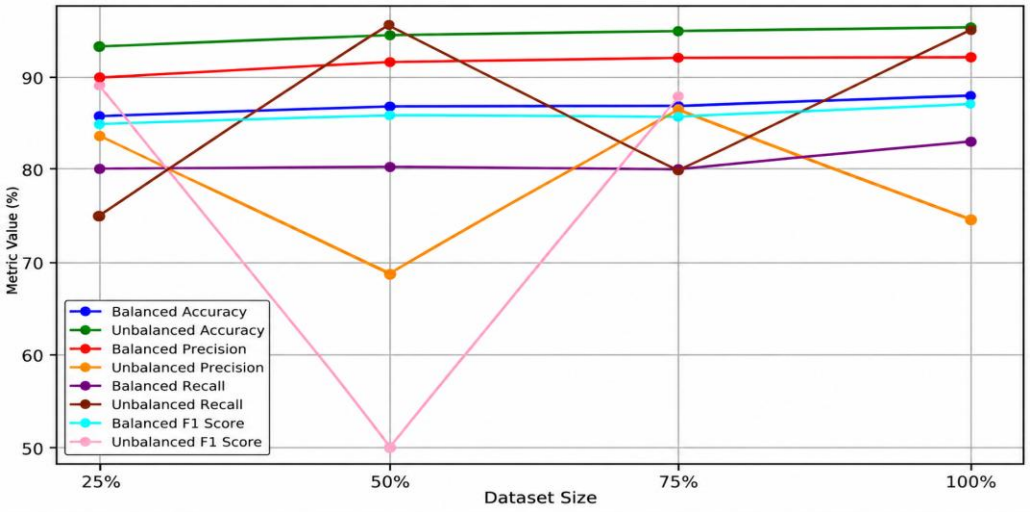


FIGURE 11: Comparison of Balanced and Unbalanced Datasets

In all metrics covered, accuracy, F1 score, and ROC AUC, the balanced dataset outperforms the unbalanced one for all dataset sizes, indicating that the balanced dataset is superior in classifying both classes. Although an unbalanced dataset sometimes performs better on Recall or precision, these benefits often come with biases toward the majority class, hindering generalization and model fairness. These results emphasize necessary steps to address class imbalance to obtain more accurate and more reliable models. The following graphs in Figures 13 show a comparison of metrics across different dataset combinations, i.e., balanced and unbalanced.

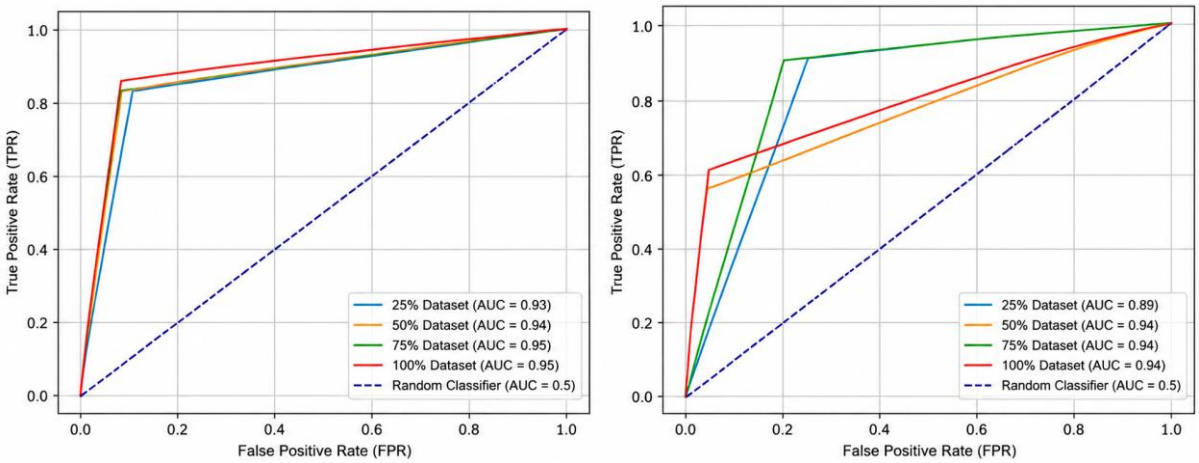


FIGURE 13: ROC-AUC Curve of Balanced and Unbalanced Dataset

C. Analysis of Results Based on Resource Usage

The outcomes reveal a trade-off between computational resources and model performance. The requirement to use balanced datasets necessitates increased computational resources to improve efficiency, resulting in higher accuracy,

Recall, and F1 scores. There is a positive correlation between larger datasets, higher time complexities, and longer training periods for models trained on such datasets, resulting in increased memory allocation requirements compared to models trained on unbalanced data. Figure 11 shows that balanced datasets consumed up to 2.69 GB and 2.54 GB of memory for the 50% and 100% cases, respectively. This high consumption of computational resources during the training and testing phases reduces the efficiency of balanced datasets.

On the contrary, the use of unbalanced datasets results in higher efficiency through less training time and lower memory utilization requirements. Training took 406.04 seconds for the 50% unbalanced dataset and 638.09 seconds for the same proportion of balanced datasets. Unbalanced datasets are computationally efficient, making it easier and faster to conduct experiments under limited computational resources. Unfortunately, unbalanced data suffers from several drawbacks. First, their intrinsic characteristics limit the ability to learn from data effectively. For instance, training on unbalanced data is prone to majority-class overfitting, thus resulting in poor Recall and F1 scores for minority classes. Second, balanced data are relatively stable across learning curves and yield better classification outcomes with fewer errors, as indicated by accuracy plots, confusion matrices, and ROC curves. However, achieving such desirable outcomes comes at the expense of increased computational costs.

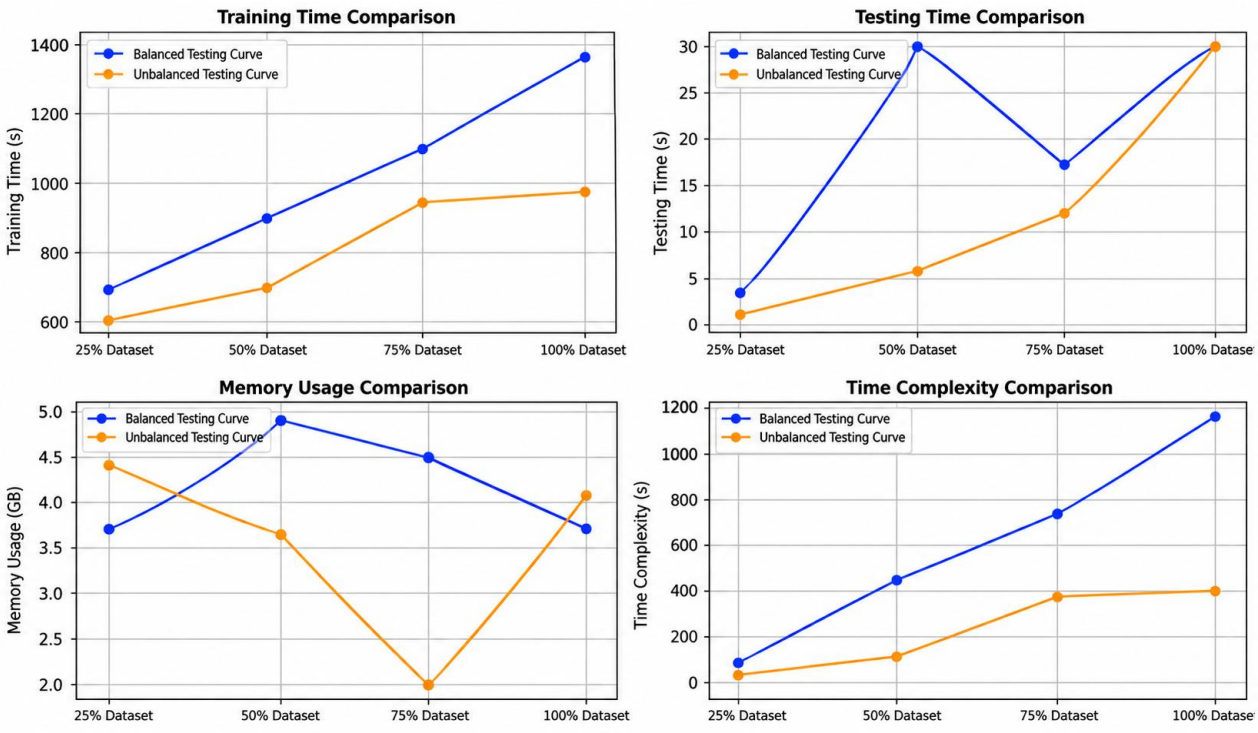


FIGURE 12: Comparisons of Performance Metrics

D. Analysis of Results Based on Resource Usage

Performance assessments using varying training-to-testing partitions (50-50%, 60-40%, 70-30%, 80-20%, and 90-10%) yielded important observations regarding model performance and computational resource demands. Model performance assessed via accuracy, precision, F1 score, and area under the curve (AUC) increased gradually with an increase in training data (from 50% to 90%). On the other hand, training costs increase significantly with increasing training data. For example, training time reached a peak of 1339.47 seconds for the 80-20% partition, whereas test time and memory requirements showed more modest variations. Particularly, test time decreased marginally with higher training data proportions, achieving the shortest time of 13.03 seconds in the 90-10% partition.

Memory requirements showed a similar pattern, decreasing from 2.96 GB in the 50-50% partition to 2.27 GB in the 70-30% partition, then leveling off at 2.65 GB. Similarly, time complexity mirrored the trend observed in training times, with the highest value achieved by the 80-20% partition. Overall, these outcomes demonstrate a clear trade-off between efficiency and performance gains. An expansion of the training dataset results in improvements in accuracy, precision, Recall, F1-score, and AUC. However, there is a parallel rise in training times and time complexity. Therefore, choosing the right training-to-test split is crucial to achieving the required balance. According to the empirical results, a training-testing ratio of 80-20% yields satisfactory performance while minimizing computational resources used. Figure 13 illustrates the performance measures according to each split proportion.

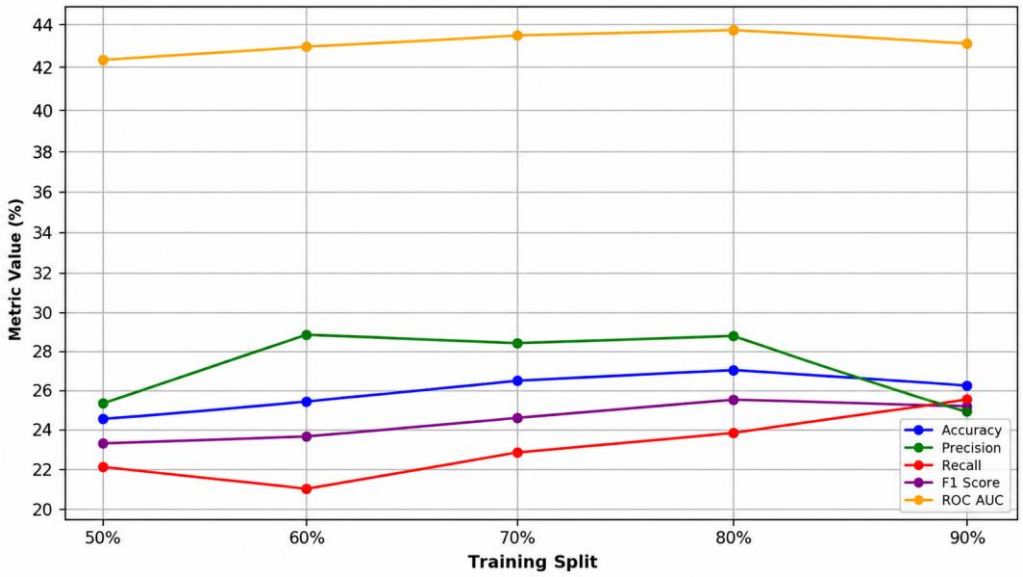


FIGURE 13: Performance Metrics of Training-Testing Ratios

VII. CONCLUSION

This study experimentally evaluates the ability of the Xception model to classify melanoma skin cancers, exploring how dataset distributions, dataset proportions, training-testing splits, and model parameter settings may influence performance and computational requirements. Findings show consistently better performance in terms of accuracy, Recall, F1 score, and ROC-AUC for balanced datasets, compared with unbalanced ones. Even though precision values were higher for unbalanced datasets, these datasets had much lower Recall and F1 scores, indicating difficulty in identifying minority classes during melanoma classification. Combination 1 of the hyperparameters (learning rate=0.001, batch size=32) showed the best performance among the combinations considered, despite significant variation across combinations in dataset distributions and train-test ratios. There was an improvement in classification performance with larger datasets, especially balanced ones. Testing different splits indicated that 80% training data and 20% testing data achieved optimal performance, with accuracy, F1 score, and ROC-AUC of 86.31%, 85.35%, and 93.88%, respectively. Increased dataset sizes resulted in higher computational costs, including extended training durations and increased storage requirements.

DECLARATIONS

Author Contributions: All authors contributed equally to this work.

Funding: The authors received no external funding for this research.

Competing Interests: The authors declare no competing interests.

Data Availability: The datasets analyzed in this study are publicly available from their respective sources. Further information regarding dataset sources and access is provided in the manuscript.

REFERENCES

[1] H. Mahmud et al., "An interpretable deep learning approach for skin cancer categorization," Diagnostics, vol. 13, no. 18, Art. no. 3012, 2023.

[2] M. A. Raza et al., "Optimizing skin cancer screening with convolutional neural networks in smart healthcare systems," PLOS ONE, vol. 20, no. 1, Art. no. e0317181, 2025.

[3] M. Bouzidi et al., "Towards explainable skin cancer diagnosis: A Vision Transformer approach with Grad-CAM visualization," Journal of Imaging, vol. 11, no. 3, Art. no. 85, 2025.

[4] S. Kollem et al., "SKIN-MARS: Multi-scale attention-driven residual Swin-Transformer for explainable skin cancer classification," Biomedical Signal Processing and Control, vol. 91, Art. no. 106010, 2025.

[5] S. Cherukuri and G. Vara Prasad, "EnsembleSkinNet: A transfer learning-based framework for efficient skin cancer detection with explainable AI integration," Frontiers in Oncology, vol. 15, Art. no. 1699960, 2025.

[6] S. Nusrat et al., "Skin cancer detection: Leveraging hybrid deep learning models and traditional machine learning classifiers," Scientific Reports, vol. 15, no. 1, Art. no. 4120, 2025.

[7] B. Benaly et al., "Explainable AI for skin cancer classification: Unlocking insights with Grad-CAM and Grad-CAM++," Computers in Biology and Medicine, vol. 172, Art. no. 108210, 2025.

[8] M. F. Khan et al., "SynthraXCoreNet: An interpretable, well-calibrated six-CNN ensemble for dermoscopic skin-lesion classification," IEEE Access, vol. 13, pp. 18500–18520, 2025.

[9] N. Malhotra and P. Kaur, "Exploring explainable melanoma classification: Leveraging pre-trained deep learning model on MED-NODE dataset," Computer Methods and Programs in Biomedicine, vol. 245, Art. no. 108010, 2024.

[10] S. T. H. Shah et al., "Explainable AI-based skin cancer detection using CNN, particle swarm optimization and machine learning," Journal of Imaging, vol. 10, no. 12, Art. no. 332, 2024.

[11] A. Abualkebash et al., "Explainable depth-wise and channel-wise fusion models for multi-class skin lesion classification," PLOS ONE, vol. 21, no. 2, Art. no. e0340901, 2026.

[12] A. Augustinne et al., "Addressing real-world domain shifts in skin lesion diagnosis with metric-based few-shot learning," IEEE Access, vol. 13, pp. 24150–24165, 2025.

[13] S. Q. Gilani et al., "Adversarial training-based domain adaptation of skin cancer images," Life, vol. 14, no. 8, Art. no. 1009, 2024.

[14] T. Jaya et al., "Federated learning with explainable AI for robust skin lesion classification in diverse clinical settings," Journal of Healthcare Engineering, 2025, Art. no. 4512980.

[15] A. R. Priambodo and C. Fatichah, "Evaluating lightweight CNN models with CBAM for explainable AI skin cancer classification using GradCAM and ScoreCAM," IEEE Access, vol. 13, pp. 29810–29822, 2025.

[16] D. Corbin et al., "Assessing bias in skin lesion classifiers with contemporary deep learning and post-hoc explainability techniques," IEEE Access, vol. 11, pp. 134210–134225, 2023.

[17] N. Sultana et al., "Domain adaptive skin lesion classification via conformal ensemble of Vision Transformers," IEEE Transactions on Artificial Intelligence, vol. 6, no. 2, pp. 340–352, 2025.

[18] J. Ye, "Enhancing skin lesion classification generalization with active domain adaptation," Pattern Recognition Letters, vol. 178, pp. 115–122, 2025.

[19] Y. Jadhav et al., "Enhancing skin cancer diagnosis with XAI: Integrating Grad-CAM, SHAP, and LIME for model interpretability," IEEE Transactions on Medical Imaging, vol. 44, no. 4, pp. 1120–1132, 2025.

[20] F. Lubis et al., "ReXNet-150 and Grad-CAM for efficient and explainable melanoma skin cancer detection," Journal of King Saud University - Computer and Information Sciences, vol. 37, no. 1, Art. no. 102230, 2025.

[21] R. Vinitha et al., "A robust and interpretable deep learning-based system for multi-category skin cancer categorization on resource-constrained edge devices," Microprocessors and Microsystems, vol. 105, Art. no. 104980, 2025.

JOURNAL OF COMPUTING HORIZONS
Vol. 01, No. 01, 2026
DOI: 10.22610/JCH.V1I1.4990

RESEARCH ARTICLE | PEER REVIEWED

Convolutional Neural Network and Transformer-Based Visual Deepfake Detection: A Comparative Analysis

Muhammad Salman [1], Iqra Tariq [1], Zameer Ahmad [2] *, Naila Batool [3]
[1] *Department of Computer Science, GIFT University, Gujranwala, Pakistan*
[2] *School of Computer Sciences, Universiti Sains Malaysia, Penang, 11800, Malaysia*
[3] *Department of Computer Science, University of Management and Technology, Lahore (Sialkot Campus), Pakistan*
* **Corresponding Author:** zameerahmad.cs@gmail.com

ABSTRACT – This paper explores the performance of several deep learning architectures for visual deepfake detection, including Swin Transformer, Vision Transformer (ViT), Xception, ResNet, and EfficientNet. We evaluate these models on the CelebDF-v2 dataset, assessing their effectiveness in detecting manipulated facial images using performance measures such as accuracy, AUC, and F1 score. Our experiments reveal that the Swin Transformer consistently outperforms other architectures, achieving an F1 score of 91.17% and AUC of 55.61%, substantially higher than the Vision Transformer's 86.85% F1 and 34.19% AUC, showing its superior ability to detect subtle visual artifacts in deepfakes. Additionally, to enhance model interpretability, we employ GradCAM visualizations to highlight the regions of input frames that are most influential to the model's predictions. These results demonstrate the potential of both CNN and transformer-based architectures to enhance deepfake detection accuracy and the value of visual explanations for improving model transparency and reliability in media forensics.

Keywords: *Image forgery; GradCAM; synthetic images; deep-fake.*

I. INTRODUCTION

One of the breakthroughs in the field of AI is image forgery, which is also known as deepfakes. Deepfakes are fake videos, images, or audio created using artificial intelligence (AI). [1] Deepfakes are defined as believable media generated using deep neural networks, and this summarizes the whole story very well. This technology uses advanced machine learning models to manipulate or generate content that looks and sounds real, even though it's entirely fake. While deepfakes have legitimate applications in entertainment, education, and accessibility, their misuse poses significant threats, including misinformation, fraud, manipulating public opinion, and invasion of privacy [2]. These forged videos are generated using AI assistance and can serve multiple purposes. This technology primarily leverages deep neural networks, particularly Generative Adversarial Networks (GANs), to produce highly realistic images, videos, or audio recordings that are virtually indistinguishable from real content [3, 4, 5]. Figure 1 illustrates the processing pipeline of face re-enactment, a technique used in deepfake generation alongside other methods such as face replacement, editing, and synthesis.

Deepfakes have revolutionized the entertainment industry, enabling actors to perform stunts or deliver lines they never actually recorded, and filmmakers to create hyper-realistic visual effects [1, 6, 7]. Deepfakes are improving movie dubbing by lip-synchronizing and matching facial expressions to the translated dialogue of the target language. This allows for a wide reach of audience, yet the expressions remain native to the language, giving the audience special attachment [6, 8]. While the technology behind deepfakes has shown tremendous promise in various creative fields, such as computer-generated imagery (CGI) for movies, the creation of digital doubles for actors, and voice dubbing, it also carries significant risks when used maliciously [6].

The malicious use of deepfakes can be used for unethical purposes, such as defamation, political manipulation, revenge porn, and other forms of misinformation, raising concerns about their impact on privacy, security, and public trust [9]. Deepfakes have reached the tipping point, posing a national security threat, as evidenced during the early phase of the Russo-Ukrainian War [10, 11]. A deepfake of President Zelenskyy emerged ordering the Ukrainian military to surrender to the Russian army [12]. In a 2019 video, Zuckerberg was making controversial statements about Facebook's data practices and user privacy [13]. The video showed Zuckerberg boasting about having "total control of billions of people's stolen data, all their secrets, their lives, their futures". As deepfake technology advances, so does the urgency of detecting and mitigating its malicious uses. The detection of deepfakes has become a critical area of research, particularly due to the rise of fake media in political, social, and personal contexts. Traditional detection methods often struggle with detecting subtle visual inconsistencies, especially as deepfake generation models improve [14, 15].

This study compares the performance of various deep learning models, including Convolutional Neural Networks (CNNs) and transformer-based architectures, in deepfake detection. The aim is to provide solutions to mitigate the risks posed by fake media, curtail misinformation, and ensure the trustworthiness of digital content.

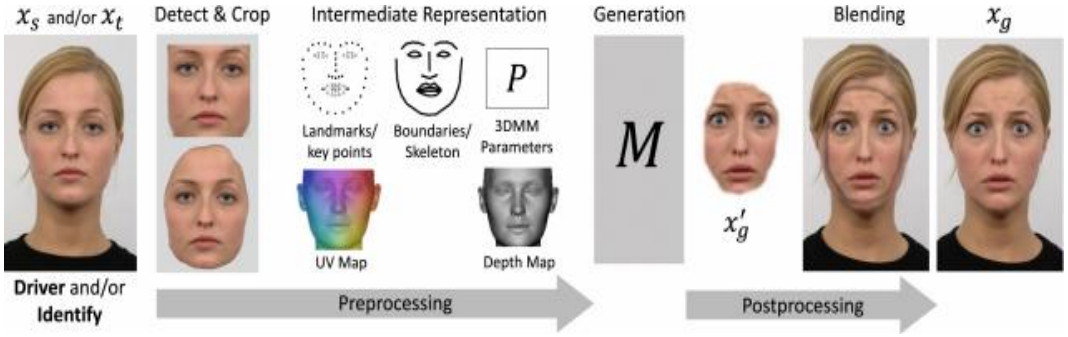


FIGURE 1: The Processing Pipeline for Making Re-Enactment and Face-Swap Deep-Fakes [1]

The rest of the paper is structured as follows: Section II reviews current research and highlights the limitations in deepfake detection. Section III covers the methodology, including architecture, data preprocessing, and evaluation metrics. Section IV, Section V, and Section VI detail the implementation, training, optimization, and results analysis. Section VII concludes with key insights and future research directions.

II. RELATED WORK

Deepfake technology has evolved significantly, utilizing GANs to create highly realistic synthetic images, making detection increasingly challenging. This literature review summarizes key contributions and advances in deepfake detection, highlighting novel techniques and frameworks. Graph Neural Networks (GNNs) are a type of deep learning models designed to analyze and process data represented as graphs. [16] demonstrated that by treating images as non-Euclidean data, GNNs can infer relationships between image patches to enhance generalization across various deepfake datasets. The study proposes a dual-branch network that extracts features from both RGB images and color difference images (CDIs), resulting in improved performance over existing methods on benchmark datasets.

The Thumbnail Layout (TALL) method, introduced in [17], presents a novel approach to detecting deepfake videos. TALL efficiently captures spatial-temporal dependencies by transforming video clips into a predefined thumbnail layout, preserving both spatial and temporal information without adding high computational costs. This method is noteworthy for its model-agnostic approach and its applicability across different deepfake manipulation techniques. The AWARE-NET [18] technique looks promising, as it introduces adaptive weighting schemes to better balance model contributions based on their performance, thereby improving generalization. However, there is a significant limitation of this approach: substantial compute power is required, as multiple instances of several architectures are used in its base, which limits its real-world use.

The integration of audio-visual features has also been explored to improve deepfake detection. The study [19] introduced the Predictive Visual-Audio Alignment Self-Supervision (PVASS-MDD) method, which detects deepfakes by analyzing the inconsistencies between visual and audio signals. (PVASS-MDD) is a two-stage process, with the first stage focusing on capturing real visual-audio correspondences, and the second stage targeting deepfake inconsistencies. In a similar vein, the paper [20] proposes a multi-task learning strategy that combines binary classification losses from both audio and video modalities. This study suggested that the combination of audio-visual information provides a more reliable and effective means for detecting deepfakes.

Data augmentation and synthetic data generation have been widely adopted to improve dataset diversity and reduce overfitting in deep learning models. Several surveys highlight the importance of augmentation techniques in enhancing robustness across computer vision tasks [21]. Additionally, fairness-aware augmentation strategies [22] have been explored to address demographic bias in facial datasets. Synthetic data generation methods such as SYNAuG [23] have also been proposed to mitigate class imbalance by augmenting underrepresented categories. Furthermore, recent deepfake detection approaches integrate both local CNN features and global transformer representations to capture complementary information, thereby improving detection performance [24].

Common Forgery Artifact Extraction Network (CFNet) was introduced in [25] with the aim of identifying manipulated facial images. It does so by filtering out common forgery artifacts hidden in low-quality images. Another innovative approach, proposed in [26], is the Contextual Cross-Modal Attention framework, which uses Recurrent Neural Networks (RNNs) to enhance deepfake detection by integrating contextual information from both audio and video.

Catastrophic forgetting in neural networks occurs when a model forgets previously learned information after being trained on new tasks. Addressing the challenge of catastrophic forgetting, [27] introduced an incremental learning framework for face forgery detection. This method uses an adapter-based approach combined with a confidence-based ensemble prediction mechanism to improve generalization and performance across multiple benchmark datasets.

With the rise of ViT and Swin, researchers have employed Transformer-based models for detecting deepfakes. [28] proposed the High-Frequency Fine-Grained Transformer (F2Trans), which focuses on fine-grained forgery cues using a Central Difference Attention (CDA) module and a High-frequency Wavelet Sampler (HWS). To better capture salient characteristics while preserving efficiency, the Shuffle Transformer, for example, combines region-of-interest (ROI) and spatial-encoding (SE) modules [29]. Similarly, other transformer-based approaches, such as ConvNext-Pnet [30], prioritize model interpretability and explainability, thereby enhancing the trustworthiness of deepfake detection systems by providing insights into their decision-making processes.

III. METHODOLOGY

In this study, we build upon the work presented by P. P. Chandran et al., which explores the use of transformers in the audio-visual domain for deepfake detection [31]. It proposed a tri-network architecture that integrates video, audio, and audio-visual modalities and was evaluated on the FakeAVCeleb dataset using ViT and Swin Transformers. To expand on this, we first examined the dataset samples and applied data augmentations. Figure 2a shows real and fake frames extracted from the CelebDF-v2 dataset, while Figure 2b illustrates the transformations and augmentations applied to enhance dataset diversity. We trained and evaluated the following models: Xception, ResNet, EfficientNet, ViT, and Swin Transformers.

Figure 3 provides a detailed representation of the Swin Transformer architecture, showcasing its hierarchical structure, patch partitioning, and self-attention mechanisms. The complete framework pipeline, including training, testing, and evaluation for all models, is depicted in Figure 4 and Figure 5. The full process for deepfake detection, including frame extraction, data augmentation, model training, and evaluation, is detailed in Algorithm 1. The left subfigure illustrates real and fake samples from the dataset, while the right subfigure demonstrates the transformations and augmentations.

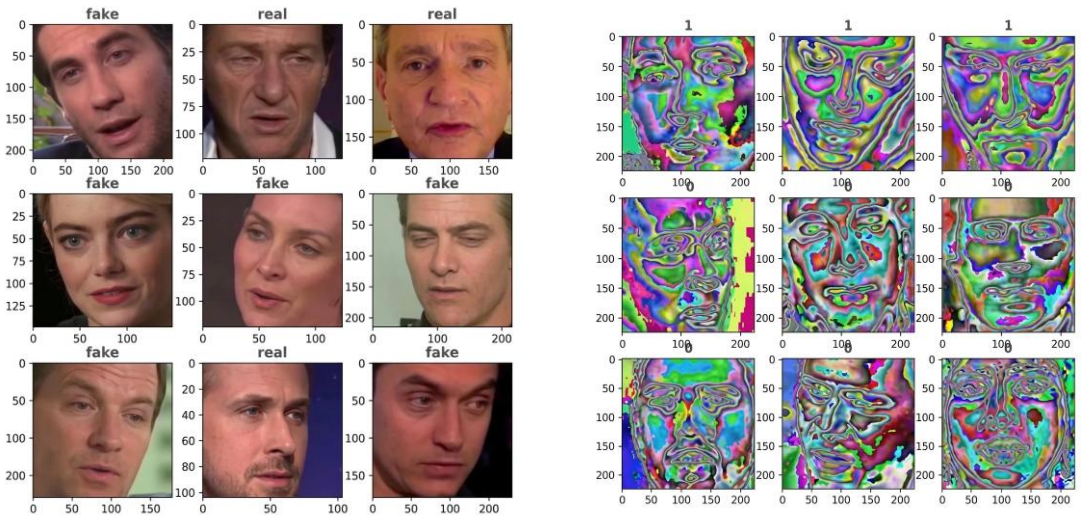


FIGURE 2: **(a)** Real and Fake Samples from the Dataset; **(b)** Transformations and Augmentations Applied

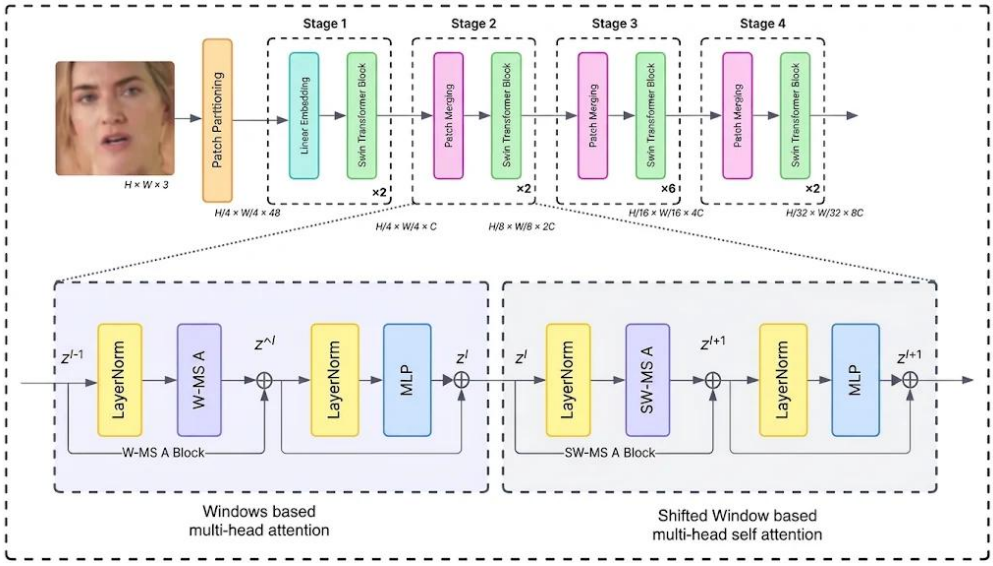


FIGURE 3: Detailed Representation of the Swin Transformer Architecture, Showcasing its Hierarchical Structure, Patch Partitioning, and Self-Attention Mechanisms

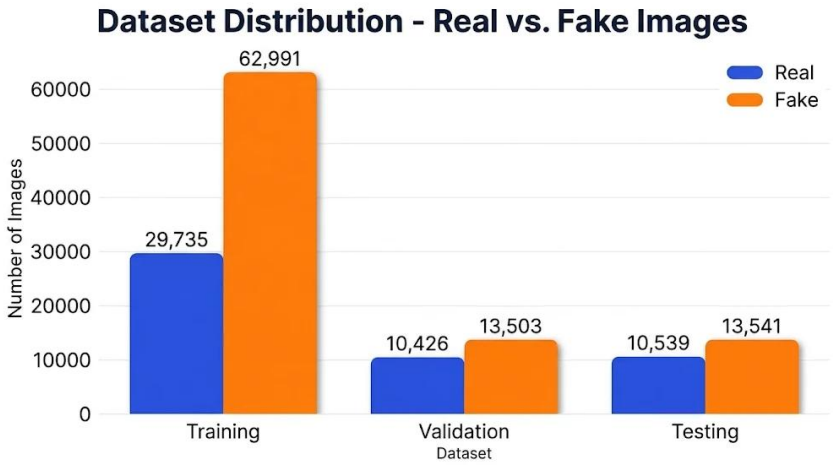


FIGURE 4: Dataset Division into Various Subsets

We trained and evaluated the following models: Xception, ResNet, EfficientNet, ViT, and Swin Transformers. The goal was to compare these architectures in terms of accuracy, robustness, and overall effectiveness in detecting deepfakes across the CelebDF-v2 dataset. Our framework pipeline is depicted in Figure 5, which shows the workflow for training, testing, and evaluating each model. The complete process for deepfake detection used in the study is detailed in Algorithm 1, which outlines the steps involved in frame extraction, data augmentation, model training, and evaluation.

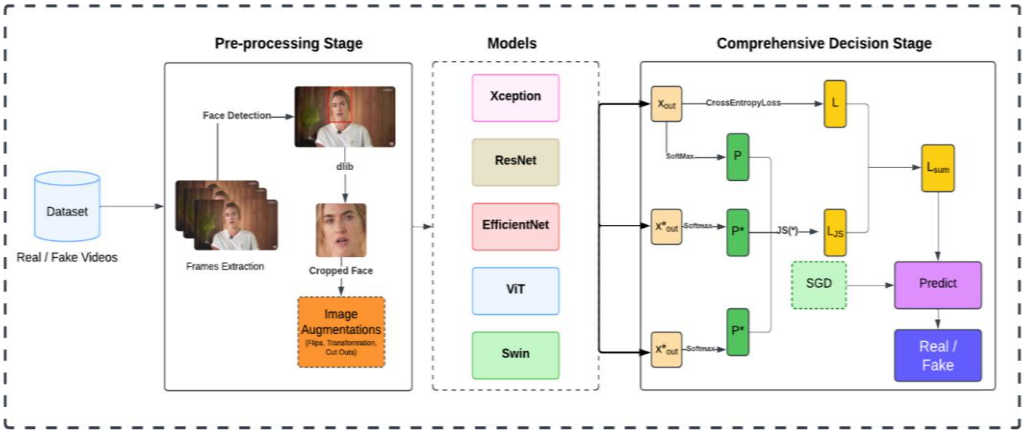


FIGURE 5: Framework Pipeline Used in the Study

A. Dataset

The CelebDF-v2 dataset [32] is a widely recognized benchmark for evaluating deepfake detection models, specifically targeting manipulated videos of celebrity faces. It consists of 890 real videos, sourced from YouTube and paid actors (590 and 300, respectively). By applying the latest forgery methods, it generated 5,639 fake videos from those original videos. The videos in CelebDF-v2 are diverse, containing different facial expressions, lighting conditions, and backgrounds, making it a valuable resource for training and testing deepfake detection models. The fake videos are generated using advanced face-swapping techniques, making the dataset particularly challenging for detection models.

The dataset preparation step took the most time since the authors did not provide predefined guidelines. Another challenge to tackle was that the dataset is imbalanced in nature. Only in CelebDF-v2 is there a 10:1 class imbalance between fake and real. To address the data imbalance problem, we extract 3 times as many frames from the real video as from the fake video. After this, the extracted real frames are half as many as the fake frames, as shown in Figure 4. We conducted experiments by extracting frames at random points throughout the video, rather than selecting a single random point and extracting consecutive frames from it. We divided the dataset into a 70:15:15 ratio for training, validation, and testing the model. Before splitting, the dataset underwent 15 iterations of random shuffling to ensure unbiased distribution. 70% allocated for training includes 70% of both real and fake videos; the same applies to the validation and testing splits. n number of frames are extracted from the video, where for real videos n=48 and n=16 for fake videos to handle imbalanced class distribution.

B. Preprocessing and Augmentation

Faces from the frames are detected and extracted using dlib. The input data for training and evaluating models is a single face cropped image [3, 224, 224]. The extracted faces are augmented, scaled, normalized, and resized before being fed

to the model for training, as illustrated in Figure 2. For augmentations, we used imgaug [33] to apply horizontal flips, affine transformations, and random cut-out on the cropped faces.

In the study, we used a detailed approach to forecast temperature categories using two different datasets. The strategy entailed the use of various machine learning models, namely a hybrid Convolutional Neural Network and Gated Recurrent Unit (CNN-GRU), Random Forest, XGBoost, Transformer, Temporal Fusion Transformer (TFT), and LightGBM. The description of the methodology was based on several important steps: a large amount of data was collected and preprocessed to ensure the data was clean and representative. This involved strategies for handling missing values, feature engineering to create new, relevant attributes, and encoding categorical variables. This was followed by resampling the data to address class imbalance and scaling to improve the model. All the models were trained with regard to particular hyperparameters to maximize their predictive power. Lastly, we evaluated the models using accuracy, precision, recall, and F1-score, and generated confusion matrices to assess how each model performed across temperature categories. This multi-perspective method was used to determine the most appropriate model for predicting temperature and to gain insight into the nature of the data.

ALGORITHM 1: Deepfake Detection Model-Agnostic Approach

Input: Video file path (video_path)
Output: Classification label (Real or Fake)

```
video_path ← "input_video.mp4"
frames ← ExtractRandomFrames(video_path, num_frames=48)
for each frame:
    grayscale_frame ← ConvertToGrayscale(frame)
    faces ← DetectFaces(grayscale_frame)
    for each face in faces:
        cropped_face ← CropFace(frame, face)
    end
end
augmented_faces ← [ ]
for each cropped_face:
    resized_face ← Resize(cropped_face, size←(224, 224))
    flipped_face ← RandomFlip(resized_face)
    normalized_face ← Normalize(flipped_face)
    augmented_faces.append(normalized_face)
end
model ← InitializeModel(num_classes=2)
loss_function ← CrossEntropyLoss()
optimizer ← SGD(model.parameters(), lr=0.001, decay_rate=0.1)
for each epoch:
    for each batch in augmented_faces:
        predictions ← model(batch)
        loss ← loss_function(predictions, labels)
        optimizer.step()
    end
end
probabilities ← Softmax(predictions)
predicted_class ← ArgMax(probabilities)
accuracy ← ComputeAccuracy(predictions, labels)
auc ← ComputeAUC(predictions, labels)
f1_score ← ComputeF1Score(predictions, labels)
logloss ← ComputeLogLoss(predictions, labels)
```

IV. EXPERIMENTAL SETUP

We used PyTorch to facilitate training and evaluating our model. Following previous studies, we set the batch size to 24 images and selected SGD (Stochastic Gradient Descent) as the optimizer, with a learning rate of 1x10-3. The learning

rate is set to decay by a factor of 0.1 every 10 epochs. We chose Cross-Entropy Loss, which is common for binary classification tasks. Accuracy, AUC, and F1 scores are reported here because they provide a balanced overview of precision and recall, which is required in this scenario. For the evaluation stage, we use the scikit-learn library. We use scikit-learn to calculate and report LogLoss, AUC, F1, and Accuracy scores [34]. Our model is trained for 20 epochs, with the final selection being the model with the lowest validation loss for subsequent evaluation and testing. All experimental settings, model configurations, and computational environment details are summarized in Table 1. To facilitate our model implementations and leverage pre-trained weights, we heavily rely on the `PyTorch` Image Models repository by Ross Wightman [35].

TABLE 1: Details of Experimental Setup, Model Configurations, and Training Parameters

Component	Setting / Value
Model Architectures	Xception, ResNet-50, EfficientNet-B0, Vision Transformer (ViT-B/16), Swin-B
Pretrained Weights	All Models Initialized with ImageNet Pretrained Weights
Input Image Size	224 × 224
Optimizer	Stochastic Gradient Descent (SGD)
Learning Rate	3×10^{-3}
Scheduler	StepLR
Scheduler Step Size	10 Epochs
Scheduler Gamma	0.1
Loss Function	CrossEntropyLoss
Batch Size	32
Number of Epochs	20
Data Shuffling	Enabled for Both Training and Validation Loaders
Data Augmentation	RandomHorizontalFlip, Cut-Out Augmentations (Random Cut-Out / Face Cut-Out)
Normalization	ImageNet Mean/Std + Additional (0.5, 0.5, 0.5) Normalization
Face Detection	dlib frontal face detector
Landmark Model	shape_predictor_81_face_landmarks.dat
Model Checkpointing	Best Model Selected Based on Lowest Validation Loss
Hardware	Intel Xeon CPU, NVIDIA RTX 4080 (24GB VRAM), 64 GB RAM
Operating System	Windows 10
Environment	Conda Virtual Environment
Frameworks Used	PyTorch, timm, torchvision, dlib

V. RESULTS AND DISCUSSION

This section presents a performance evaluation of CNN- and transformer-based models for the deepfake detection task. Table 2 presents the training and validation results in terms of Accuracy and LogLoss. In contrast, Table 3 shows the evaluation results of different architectures in terms of F1, AUC, and Accuracy, providing insights into the comparative strengths and weaknesses of the models.

TABLE 2: Performance Metrics for Different Architectures on Training and Validation Subsets

Architectures	Training		Validation	
	Accuracy	LogLoss	Accuracy	LogLoss
Xception	95.78%	0.14540	90.57%	0.27840
ResNet	99.13%	0.02267	95.14%	0.18701
EfficientNet	97.97%	0.05340	92.70%	2.45800
ViT	99.46%	0.01386	97.12%	0.13374
Swin	99.02%	0.04799	99.59%	0.18500

TABLE 3: F1, AUC, and Accuracy of Evaluated Architectures for Deepfake Detection

Architecture	F1	AUC	Accuracy
Swin	91.17%	55.61%	89.58%
ViT	86.85%	34.19%	85.96%
ResNet	77.06%	74.40%	67.81%
EfficientNet	76.52%	78.71%	65.85%
Xception	76.93%	86.75%	66.27%

Figure 6 illustrates the performance comparison across different architectures. Subfigure **(a)** presents the training and validation accuracy, highlighting the generalization capability of each model. Subfigure **(b)** shows the corresponding log-loss values, indicating the confidence and calibration of the predictions. Subfigure **(c)** depicts the gap between

training and validation accuracy, providing a direct measure of potential overfitting. The results demonstrate that while CNN-based models exhibit varying degrees of overfitting, Transformer-based architectures maintain more consistent performance across training and validation sets, suggesting better generalization.

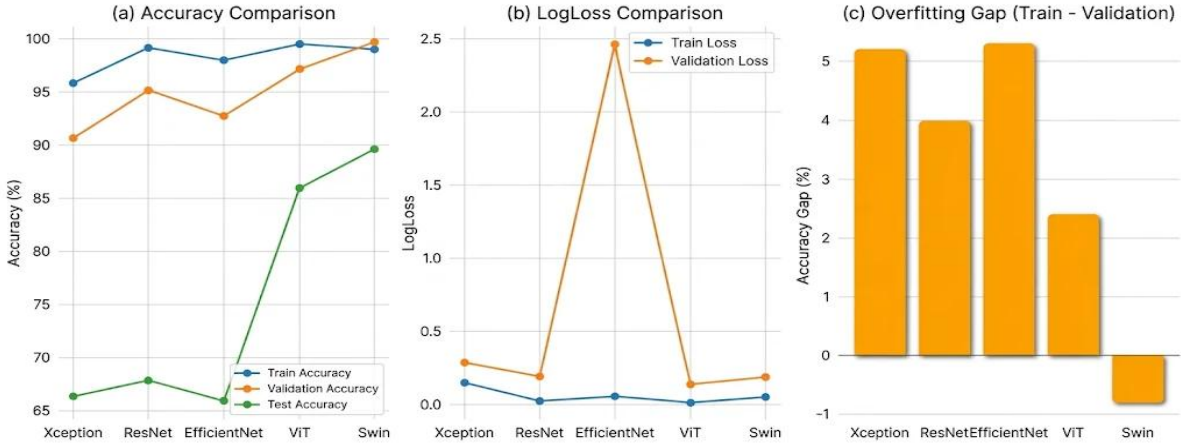


FIGURE 6: **(a)** Training vs. Validation Accuracy **(b)** Training vs. Validation Log Loss **(c)** Overfitting Gap Across Architectures; The results show that Transformer-based models exhibit more stable performance and smaller generalization gaps compared to CNN-based models.

Figure 7 heatmap visualization provides an intuitive comparison of the performance metrics (F1, AUC, and accuracy) across the different architectures. The Swin Transformer, with its high F1 score and accuracy, shows concentrated attention on regions critical for deepfake detection, while models like ResNet and Xception, despite having higher AUC, focus differently, emphasizing class separation. This visualization aids in understanding the varying strengths of each model in detecting subtle deepfake artifacts.

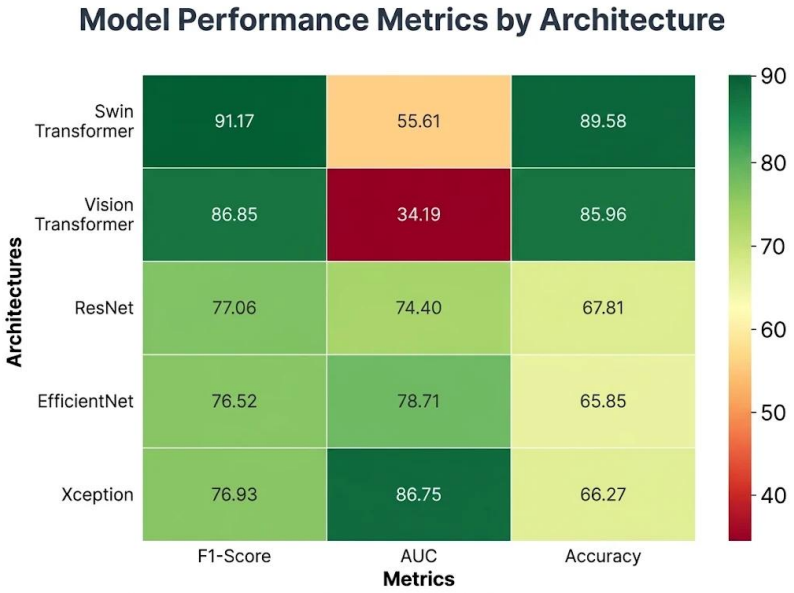


FIGURE 7: Heatmap Visualization of Evaluation Results

Additionally, Grad-CAM visualizations are shown in Figure 8. These visualizations indicate the regions of the input face frames that contribute most to the model's predictions, providing insight into the model's decision-making process. These visualizations are of utmost importance for interpretability: why the model reached the decision to classify this image or video as forged or real. When comparing CNN-based architectures (Xception, ResNet, and EfficientNet) with transformer-based models (ViT and Swin), transformer models generally achieve higher validation accuracy and lower log loss, indicating better generalization. While CNNs like ResNet achieved high training accuracy, they exhibited a slight drop in validation performance, suggesting some overfitting. Xception is good at capturing local artifacts and fine details; ResNet is good at capturing features across multiple scales due to its residual connections.

Overfitting is likely due to CNNs' reliance on local receptive fields, which may struggle to capture global dependencies in complex tasks, as their strength lies in capturing local features and patterns in an image. In contrast, transformers such as ViT and Swin excel at capturing global dependencies via the self-attention mechanism, enabling them to maintain strong performance on both training and validation datasets [36, 37]. The global perspective allows

transformers to focus on relevant features across the entire image, leading to more robust generalization and demonstrating that transformers are better suited for handling complex patterns and larger datasets. Overall, transformers appear to outperform CNNs in terms of validation accuracy and consistency, as evidenced by Tables 2 and 3.

VI. LIMITATIONS AND FUTURE WORK

One limitation of this study is that we did not evaluate cross-dataset generalization. Our decision to focus on CelebDF-v2 was intentional: it allowed for a fair comparison of architectural performance under controlled conditions. But of course, deepfake detection models must eventually face the messier question of robustness across datasets, DFDC, FaceForensics++, DeeperForensics-1.0 – all of which introduce their own distributional quirks and manipulation techniques. We didn't take up that challenge here, but we recognize its importance and plan to address it in future work.

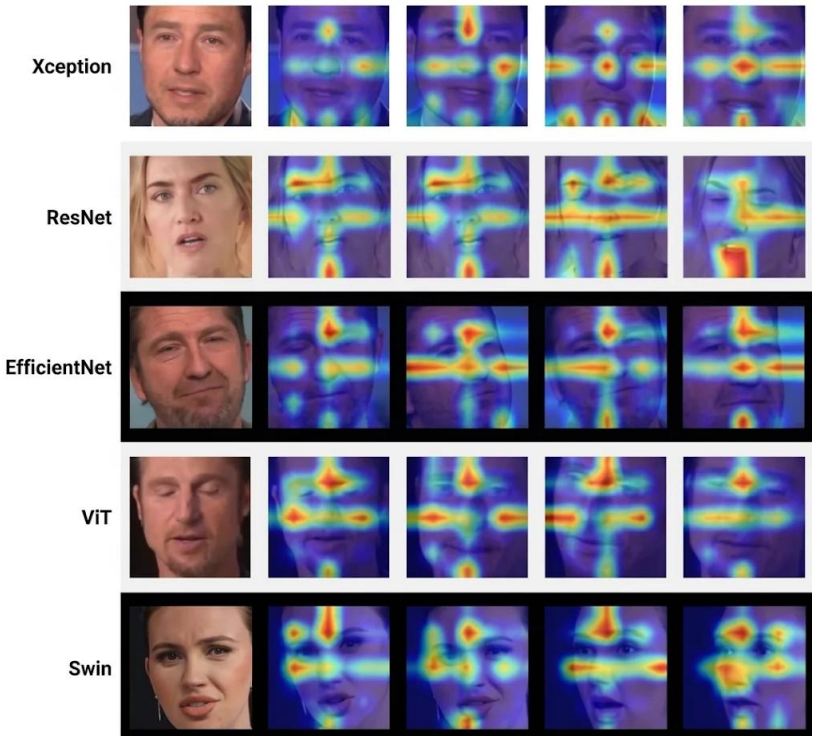


FIGURE 8: Grad-CAM Visualizations Highlighting the Areas of Focus for the Model During Evaluation

Class imbalance posed another difficulty. Our approach, extracting more frames from real videos, was a practical response, not a principled one. We considered oversampling and class-weighted loss, but neither proved straightforward: oversampling risked redundancy; weighting alone didn't offset the skew. We suspect this is a case where theoretical elegance gives way to empirical stubbornness, and we intend to test these strategies more systematically in the future.

Finally, we didn't run ablation studies on individual augmentation techniques, largely due to resource constraints, both time and compute. Instead, we trained on a composite of augmentations without isolating their effects. This limits our ability to say which augmentations matter most, cut-outs, flips, affine transforms, etc., though we plan to unpack that in future iterations.

The results of this study extend beyond the specific dataset and offer useful insights into how CNN- and Transformer-based models perform in deepfake detection. The observed trends show that Transformer-based models are generally better at handling complex patterns and may work more effectively in real-world scenarios. These findings can help researchers and practitioners choose suitable models for similar computer vision problems, especially when dealing with subtle image manipulations. The comparison approach used in this study can also be applied to other areas such as image forensics, biometric verification, and multimedia security.

However, these results should be tested on more datasets and in real-world settings to confirm their reliability. This is an important direction for future work. Even with these limitations, we believe the study offers a solid foundation for architectural comparison in deepfake detection. Much remains undone, but some things had to be left undone to get anything done at all.

VII. CONCLUSION

The paper compared CNN and transformer-based models for deepfake detection. While the results did not achieve state-of-the-art performance, they aligned with the broader trend in the literature, where transformer models generally outperform CNNs [38, 39, 40, 41]. Interestingly, despite lower AUC scores, the transformer models achieved higher F1

scores than CNNs, which had relatively higher AUC scores. Future work will focus on cross-dataset evaluations to assess and enhance the generalization capabilities of these models using an ensemble learning approach.

DECLARATIONS

Author Contributions: All authors contributed equally to this work.

Funding: The authors received no external funding for this research.

Competing Interests: The authors declare no competing interests.

Data Availability: The datasets analyzed in this study are publicly available from their respective sources. Further information regarding dataset sources and access is provided in the manuscript.

REFERENCES

[1] Y. Mirsky and W. Lee, "The Creation and Detection of Deepfakes: A Survey," ACM Comput Surv, vol. 54, no. 1, p. 7:1-7:41, Jan. 2021, doi: 10.1145/3425780.

[2] A. Malik, M. Kuribayashi, S. M. Abdullahi, and A. N. Khan, "DeepFake Detection for Human Face Images and Videos: A Survey," IEEE Access, vol. 10, pp. 18757–18775, 2022, doi: 10.1109/ACCESS.2022.3151186.

[3] J. Gui, Z. Sun, Y. Wen, D. Tao, and J. Ye, "A Review on Generative Adversarial Networks: Algorithms, Theory, and Applications," IEEE Trans. Knowl. Data Eng., vol. 35, no. 4, pp. 3313–3332, Apr. 2023, doi: 10.1109/TKDE.2021.3130191.

[4] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative Adversarial Networks: An Overview," IEEE Signal Process. Mag., vol. 35, no. 1, pp. 53–65, Jan. 2018, doi: 10.1109/MSP.2017.2765202.

[5] M. Tao, H. Tang, F. Wu, X.-Y. Jing, B.-K. Bao, and C. Xu, "DF-GAN: A Simple and Effective Baseline for Text-to-Image Synthesis," presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 16515–16525. Accessed: Nov. 30, 2024. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2022/html/Tao_DF-GAN_A_Simple_and_Effective_Baseline_for_Text-to-Image_Synthesis_CVPR_2022_paper.html

[6] G. Murphy, D. Ching, J. Twomey, and C. Linehan, "Face/Off: Changing the face of movies with deepfakes," PLOS ONE, vol. 18, no. 7, p. e0287503, Jul. 2023, doi: 10.1371/journal.pone.0287503.

[7] teresa_myers, “Deepfake in Movies | Facts, Fiction & Future of Film Industry,” Facia.ai. Accessed: Dec. 03, 2024. [Online]. Available: https://facia.ai/blog/deepfake-in-movies-facts-fiction-future-of-film-industry/

[8] K. R. Prajwal, R. Mukhopadhyay, V. P. Namboodiri, and C. V. Jawahar, "A Lip Sync Expert Is All You Need for Speech to Lip Generation In the Wild," in Proceedings of the 28th ACM International Conference on Multimedia, in MM '20. New York, NY, USA: Association for Computing Machinery, Oct. 2020, pp. 484–492. doi: 10.1145/3394171.3413532.

[9] S. Agarwal, H. Farid, Y. Gu, M. He, K. Nagano, and H. Li, "Protecting World Leaders Against Deep Fakes," in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, IEEE, 2019, pp. 38–45. Accessed: Nov. 30, 2024. [Online]. Available: https://openaccess.thecvf.com/content_CVPRW_2019/html/Media_Forensics/Agarwal_Protecting_World_Leaders_Against_Deep_Fakes_CVPRW_2019_paper.html

[10] A. Ali, K. F. Khan Ghouri, H. Naseem, T. R. Soomro, W. Mansoor, and A. M. Momani, "Battle of Deep Fakes: Artificial Intelligence Set to Become a Major Threat to the Individual and National Security," in 2022 International Conference on Cyber Resilience (ICCR), Oct. 2022, pp. 1–5. doi: 10.1109/ICCR56254.2022.9995821.

[11] "NSA, U.S. Federal Agencies Advise on Deepfake Threats," National Security Agency/Central Security Service. Accessed: Nov. 30, 2024. [Online]. Available: https://www.nsa.gov/Press-Room/Press-Releases-Statements/Press-Release-View/Article/3523329/nsa-us-federal-agencies-advise-on-deepfake-threats/http%3A%2F%2Fwww.nsa.gov%2FPress-Room%2FPress-Releases-Statements%2FPress-Release-View%2FArticle%2F3523329%2Fnsa-us-federal-agencies-advise-on-deepfake-threats%2F

[12] "Deepfake presidents used in Russia-Ukraine war," Mar. 18, 2022. Accessed: Mar. 02, 2025. [Online]. Available: https://www.bbc.com/news/technology-60780142

[13] R. M. O'Sullivan Donie, "A deepfake video of Mark Zuckerberg presents a new challenge for Facebook | CNN Business," CNN. Accessed: Nov. 30, 2024. [Online]. Available: https://www.cnn.com/2019/06/11/tech/zuckerberg-deepfake/index.html

[14] M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets," Nov. 06, 2014, arXiv: 1411.1784. doi: 10.48550/arXiv.1411.1784.

[15] T. Karras, S. Laine, and T. Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks," Mar. 29, 2019, arXiv: 1812.04948. doi: 10.48550/arXiv.1812.04948.

[16] H. She, Y. Hu, B. Liu, J. Li, and C.-T. Li, "Using Graph Neural Networks to Improve Generalization Capability of the Models for Deepfake Detection," IEEE Trans. Inf. Forensics Secur., vol. 19, pp. 8414–8427, 2024, doi: 10.1109/TIFS.2024.3451356.

[17] Y. Xu, J. Liang, G. Jia, Z. Yang, Y. Zhang, and R. He, "TALL: Thumbnail Layout for Deepfake Video Detection," in 2023 IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France: IEEE, Oct. 2023, pp. 22601–22611. doi: 10.1109/ICCV51070.2023.02071.

[18] M. Salman, I. Tariq, M. Zulfiqar, M. Jalal, S. Aujla, and S. Fatima, "AWARE-NET: adaptive weighted averaging for robust ensemble network in deepfake detection," in International Conference on Energy, Power, Environment, Control and Computing (ICEPECC 2025), Feb. 2025, pp. 526–533. doi: 10.1049/icp.2025.1162.

[19] Y. Yu, X. Liu, R. Ni, S. Yang, Y. Zhao, and A. C. Kot, "PVASS-MDD: Predictive Visual-Audio Alignment Self-Supervision for Multimodal Deepfake Detection," IEEE Trans. Circuits Syst. Video Technol., vol. 34, no. 8, pp. 6926–6936, Aug. 2024, doi: 10.1109/TCSVT.2023.3309899.

[20] S. Muppalla, S. Jia, and S. Lyu, "Integrating Audio-Visual Features For Multimodal Deepfake Detection," in 2023 IEEE MIT Undergraduate Research Technology Conference (URTC), Cambridge, MA, USA: IEEE, Oct. 2023, pp. 1–5. doi: 10.1109/URTC60662.2023.10534969.

[21] T. Kumar, R. Brennan, A. Mileo, and M. Bendechache, "Image Data Augmentation Approaches: A Comprehensive Survey and Future Directions," IEEE Access, vol. 12, pp. 187536–187571, 2024, doi: 10.1109/ACCESS.2024.3470122.

[22] T. Kumar, A. Mileo, and M. Bendechache, "Saliency-based metric and FaceKeepOriginalAugment: a novel approach for enhancing fairness and Diversity," Multimed. Syst., vol. 31, no. 3, p. 153, Mar. 2025, doi: 10.1007/s00530-025-01741-5.

[23] M. Ye-Bin, N. Hyeon-Woo, W. Choi, N. Kim, S. Kwak, and T.-H. Oh, "SYNAuG: Exploiting synthetic data for data imbalance problems," Pattern Recognit. Lett., vol. 193, pp. 115–121, Jul. 2025, doi: 10.1016/j.patrec.2025.04.013.

[24] M. Javed, Z. Zhang, F. H. Dahri, and T. Kumar, "Enhancing multimodal deepfake detection with local–global feature integration and diffusion models," Signal Image Video Process., vol. 19, no. 5, p. 400, Mar. 2025, doi: 10.1007/s11760-025-03970-7.

[25] H. Wu, X. Wang, R. Wang, J. Xiang, and L. Ren, "Common Forgery Artifact Driven Deepfake Face Detection," in 2024 27th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Tianjin, China: IEEE, May 2024, pp. 1585–1590. doi: 10.1109/CSCWD61410.2024.10580312.

[26] V. S. Katamneni and A. Rattani, "Contextual Cross-Modal Attention for Audio-Visual Deepfake Detection and Localization," in 2024 IEEE International Joint Conference on Biometrics (IJCB), Buffalo, NY, USA: IEEE, Sep. 2024, pp. 1–11. doi: 10.1109/IJCB62174.2024.10744480.

[27] C. Gao, Q. Xu, P. Qiao, K. Xu, X. Qian, and Y. Dou, "Adapter-Based Incremental Learning for Face Forgery Detection," in ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seoul, Korea, Republic of: IEEE, Apr. 2024, pp. 4690–4694. doi: 10.1109/ICASSP48485.2024.10446543.

[28] C. Miao, Z. Tan, Q. Chu, H. Liu, H. Hu, and N. Yu, "F2 Trans: High-Frequency Fine-Grained Transformer for Face Forgery Detection," IEEE Trans. Inf. Forensics Secur., vol. 18, pp. 1039–1051, 2023, doi: 10.1109/TIFS.2022.3233774.

[29] C. Yang, C. Zhu, C. Deng, and Z. Xiao, "ShuffleSR: Image Deepfake Detection Using Shuffle Transformer," in 2023 IEEE 6th International Conference on Knowledge Innovation and Invention (ICKII), Sapporo, Japan: IEEE, Aug. 2023, pp. 639–642. doi: 10.1109/ICKII58656.2023.10332600.

[30] H. Ilyas, A. Javed, and K. M. Malik, "ConvNext-PNet: An interpretable and explainable deep-learning model for deepfakes detection," in 2024 IEEE International Joint Conference on Biometrics (IJCB), Buffalo, NY, USA: IEEE, Sep. 2024, pp. 1–9. doi: 10.1109/IJCB62174.2024.10744484.

[31] P. P. Chandran, S. Sriram, and C. Babu, "Multimodal Forgery Detection in Videos Using a Tri-Network Model," in 2024 10th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India: IEEE, Mar. 2024, pp. 1507–1514. doi: 10.1109/ICACCS60874.2024.10717044.

[32] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-DF: A Large-scale Challenging Dataset for DeepFake Forensics," Mar. 16, 2020, arXiv:1909.12962. doi: 10.48550/arXiv.1909.12962.

[33] A. Jung, aleju/imgaug. (Nov. 30, 2024). Python. Accessed: Nov. 30, 2024. [Online]. Available: https://github.com/aleju/imgaug

[34] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Mach. Learn. PYTHON.

[35] R. Wightman, rwightman/timm. (Nov. 26, 2024). Python. Accessed: Nov. 30, 2024. [Online]. Available: https://github.com/rwightman/timm

[36] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in Vision: A Survey," ACM Comput Surv, vol. 54, no. 10, p. 200:1-200:41, Sep. 2022, doi: 10.1145/3505244.

[37] K. Han et al., "A Survey on Vision Transformer," IEEE Trans. Pattern Anal. Mach. Intell., vol. 45, no. 1, pp. 87–110, Jan. 2023, doi: 10.1109/TPAMI.2022.3152247.

[38] V. L. L. Thing, "Deepfake Detection with Deep Learning: Convolutional Neural Networks versus Transformers," in 2023 IEEE International Conference on Cyber Security and Resilience (CSR), Venice, Italy: IEEE, Jul. 2023, pp. 246–253. doi: 10.1109/CSR57506.2023.10225004.

[39] K. N. Ramadhani, R. Munir, and N. P. Utama, "Improving Video Vision Transformer for Deepfake Video Detection Using Facial Landmark, Depthwise Separable Convolution and Self Attention," IEEE Access, vol. 12, pp. 8932–8939, 2024, doi: 10.1109/ACCESS.2024.3352890.

[40] D. W. Deressa, P. Lambert, G. Van Wallendael, S. Atnafu, and H. Mareen, "Improved Deepfake Video Detection Using Convolutional Vision Transformer," in 2024 IEEE Gaming, Entertainment, and Media Conference (GEM), Jun. 2024, pp. 1–6. doi: 10.1109/GEM61861.2024.10585593.

[41] S. A. Khan and D.-T. Dang-Nguyen, "Deepfake Detection: Analyzing Model Generalization Across Architectures, Datasets, and Pre-Training Paradigms," IEEE Access, vol. 12, pp. 1880–1908, 2024, doi: 10.1109/ACCESS.2023.3348450.

JOURNAL OF COMPUTING HORIZONS
Vol. 01, No. 01, 2026
DOI: 10.22610/JCH.V1I1.4987

RESEARCH ARTICLE | PEER REVIEWED

An Assessment of Ensemble and Deep Learning Methods to Introduce Precise Temperature Categorization in Several Zoned Climate Data

Iqra [1] *, Ineta Shahid [1], Naila Batool [1], Umair Muneer Butt [1]
[1] *Department of Computer Science, University of Management and Technology, Lahore (Sialkot Campus), Pakistan*
* **Corresponding Author:** 23005279007@skt.umt.edu.pk

ABSTRACT – This paper explores the performance of several deep learning architectures for visual deepfake detection, including Swin Transformer, Vision Transformer (ViT), Xception, ResNet, and EfficientNet. We evaluate these models on the CelebDF-v2 dataset, assessing their effectiveness in detecting manipulated facial images using performance measures such as accuracy, AUC, and F1 score. Our experiments reveal that the Swin Transformer consistently outperforms other architectures, achieving an F1 score of 91.17% and AUC of 55.61%, substantially higher than the Vision Transformer's 86.85% F1 and 34.19% AUC, showing its superior ability to detect subtle visual artifacts in deepfakes. Additionally, to enhance model interpretability, we employ GradCAM visualizations to highlight the regions of input frames that are most influential to the model's predictions. These results demonstrate the potential of both CNN- and transformer-based architectures to enhance deepfake detection accuracy, as well as the value of visual explanations for improving model transparency and reliability in media forensics.

Keywords: *Temprature prediction; Weather classification; Machine learning; Deep learning; Random forest.*

I. INTRODUCTION

The application of artificial intelligence (AI) in climate science is a revolutionary approach to ensuring accuracy, performance, and timeliness in forecasting weather and climate conditions. Conventional numerical models are likely to fail to adequately reproduce the dynamics of uncaptured, unpredictable, complex nonlinear atmospheric processes, particularly in out-of-distribution events such as gray-swan tropical cyclones, as climate change makes such processes more frequent and intense. To combat such difficulties, AI-based models and hybrid systems have become game-changing tools for a wide variety of applications, such as air quality monitoring, flood prediction, extreme precipitation nowcasting, and agricultural risk assessment, to name a few. Machine learning (ML) and deep learning (DL) techniques have demonstrated great potential to outperform conventional physics-based systems for forecasting extreme weather events [1, 2]. These developments not only increase the accuracy of predictions but also enable the rapid processing of multi-source data, which is key to addressing meteorological hazards in real time.

One of the reasons for the wider adoption and confidence in AI-based prediction systems is the emergence of explainable AI (XAI) methods. Improvements in XAI have helped promote transparency and stakeholders' confidence in the decision-making process regarding climatic influences [3]. This transparency becomes particularly significant in the context of AI's application in real meteorological services, such as predicting worldwide floods [4], understanding the interaction of aerosols with climate patterns, assessing extreme weather risks in the Himalayas [5], and assessing climate risks to agriculture [3]. Moreover, AI-powered models are also becoming a necessity in facilitating climate adaptation interventions and providing weather warning techniques [6]. These two capabilities of AI-predictive and advisory have placed AI at the center of resilience planning to new and increasing climate threats. New frontiers have been opened in sustainable planning and climate policy development by researchers who are also beginning to explore hybrid AI-physics models that balance predictive performance and scientific explanations [7, 8].

The significance of such interdisciplinary work is supported by research comparing AI techniques with traditional models, most of which suggest that the former are superior in comprehensiveness or accuracy [9]. For example, AI has proven capable of delivering more insights into complex climatic systems and data-driven sustainable solutions to global environmental issues [10, 11, 12]. The role of AI may be reflected in case studies. In North China, the combination of Pangu-Weather with the regional WRF makes a tremendous contribution to extreme precipitation prediction [13]. At the local level, AI detection of extreme weather patterns has been used in Indonesia, as machine learning models were able to detect phenomena in the extreme local area of Bogor [14]. At the same time, in the agricultural industry, AI is being used to predict corn production under extreme climate conditions by combining satellite and meteorological data [25]. The driving force behind such models is usually a complete dataset, such as ClimSim, that facilitates hybrid physics-ML emulation of climate behavior across diverse spatial and temporal scales [16].

Knowledge sharing and anticipation of adverse weather conditions are important because the destruction affecting society and nature worsens. According to [17], predictions and response designs could be significantly enhanced, particularly in the context of the rapidly changing climate, by combining observational data with modeling techniques.

Still, challenges remain. As highlighted by [1], AI models require trustworthy, high-quality data, better interpretability, and interdisciplinary collaboration for effective implementation. For this purpose, the Royal Society provides a helpful scheme for classifying AI systems into Hard AI, Medium AI, and Soft AI, which allows for a general estimate of the current and future role of AI in weather forecasting. Medium/Soft AI can already accelerate the forecasting process, increase resolution, and improve its adaptability. However, it does not completely replace physical models, which are still constrained by available data and physical limitations [2].

Unfortunately, the community paper she authored has shed light on systemic challenges across the U.S. numerical weather prediction community, highlighting scattered research funding and untapped AI innovations. The authors advocate for strategic investment in research on AI, with an increase in common modeling frameworks and in high-performance computing infrastructure, to regain global leadership in climate forecasting [3].

- To build and compare machine and deep learning models that would assist in predicting the category of temperature based on past weather data from two different datasets.
- To preprocess and engineer appropriate data properties and characteristic features, such as temporal and weather conditions, to enhance the model's predictive accuracy.
- In a bid to draw a comparison of the predictive benefit between traditional ensemble algorithms (Random Forest, XGBoost, LightGBM) and deep learning models (CNN-GRU, Transformer, TFT) for grouping temperature.
- To find the best model(s) that would be able to correctly classify the temperature into the specific discrete categories (i.e., cold, cool, mild, and warm) under various geographical and temporal datasets.
- To observe the drawbacks of deep learning, the methods used in this case, and to suggest possible ways to enhance them in further studies, including the application of hybrid models and the use of transfer learning.

II. RELATED WORK

By combining the regional WRF (Weather Research and Forecasting) model with an advanced artificial intelligence model named Pangu-weather, this article aims to enhance our predictive capability for excessive rainfall in North China. It addresses the main issue of AI models failing to accurately predict heavy, localized rainfall due to low resolution and a tendency to smooth out extreme outcomes. The team developed a hybrid model that they refer to as the AI-Driven WRF. It addresses severe weather warnings and preparedness to deal with this problem. This was achieved using Pangu-Weather to generate initial conditions for the WRF model. It was discovered during the evaluation of this arrangement using data from two major storms, one of which [18] broke rainfall records between July 29 and August 1, 2023. The research consisted of ERA5 reanalysis, NCEP atmospheric observations, and high-resolution rainfall data from the China Meteorological Administration. Based on the findings, the AI-based approach performed better compared to conventional models in terms of accuracy, forecasting strong rainfall episodes up to 8.5 days in advance, which is superior to the normal approaches' 3-day capacity. To conclude, this systematic combination of forecasting methods has great potential to promote the emergence of severe weather preparedness and warning [13].

The development of a methodology and the utilization of artificial intelligence (AI) to detect and predict extreme weather patterns in the Bogor region of Indonesia, which has a complex climate and experiences heavy rainfall very frequently, are among the main themes of the present paper. In the paper, the weaknesses of traditional forecasting methods are noted, and these methods often struggle to capture the nonlinear and rapidly changing weather conditions in Bogor. To address this, the authors explore artificial intelligence (AI) methods, including Random Forest (RF), Long Short-Term Memory (LSTM), and Convolutional Neural Networks (CNN), as well as combinations of multiple algorithms in their hybrid models. Many weather data, such as satellite images, historical rainfall datasets, and Bogor geospatial datasets, were processed using these techniques. The results show commendable developments: CNN models could predict satellite-based rainfall with more than 90% accuracy, LSTM models could be used effectively for forecasting temperature and pollution, and RF models performed extremely well in mapping flood-prone areas. The research finds that, due to the increased accuracy, efficiency, and flexibility of weather forecasting systems, especially in areas prone to natural disasters such as Bogor, AI is a powerful alternative to traditional forecasting methods.

Corn production predictions can be refined using a deep neural network (DNN) model during extreme weather conditions, particularly heatwaves. MODIS satellite data and meteorological data from NOAA on temperature, precipitation, and vegetation indices are provided by the authors, as traditional models are not able to handle such scenarios. The DNN model outperformed five other models (Random Forest, SVM, k-NN, Decision Tree, and Linear Regression) with an accuracy of 92.3%, which was 6.6 percent higher than Random Forest. In terms of RMSE, it was the best model under heatwave conditions and still had a decent R2 of 0.89, compared to 0.78 for Random Forest. As one can see, the research demonstrates that DNNs can be used as a powerful, precise tool to increase crop yields under climate extremes [15].

In the document [19], the author focuses on the use of AI and ML models to enhance weather forecasting. It addresses the brittleness and ineffectiveness of traditional numerical forecasting methods. Using parameters such as temperature, humidity, wind velocity, and pressure, the authors employ models such as ANN, SVM, and Random Forest to predict past weather data from local weather stations and NOAA. ANN performed better than the others, with its short-term forecasting accuracy at 90 percent. Taken together, the study proves that AI/ML solutions offer more accurate, faster, and more scalable approaches to modern weather forecasting needs.

In [16], the role of artificial intelligence (AI) in enhancing the resilience and sustainability of farming in the face of climate-related challenges is examined. It applies to issues such as inefficient soil testing, excessive fertilization, and a lack of flexible in-tool operation for smallholder farmers. To track soil health, predict crop yields, and assess sequestration potential, the paper focuses on Sense Grass. This artificial intelligence-enabled service incorporates live soil sensors, space imaging, and machine learning. The AI models can provide practical guidance on how farmers can minimize environmental impact and fully use available resources, based on statistics from farms in Nepal, India, and other regions. The paper concludes by demonstrating that AI tools like SenseGrass can provide farmers with data-driven tools that facilitate sustainable agricultural practices and improve resilience to climate change.

The central idea of [17] is that recent climate models have become even harder to test and to make more precise and efficient, particularly in predicting severe weather conditions such as storms and rain. To resolve small-scale physical processes such as cloud convection, traditional climate models employ heuristic parameterization techniques, which often introduce large errors and have low spatial resolution. To emulate the effects of sub-grid-scale physics, this paper proposes a hybrid ML (machine learning) – physics emulator approach that learns through high-resolution simulations. ClimSim is the most comprehensive and largest dataset created for this purpose, as presented in [20]. It consists of 5.7 billion input-output samples that indicate atmospheric states and their corresponding physical trends. These low- and high-res solution samples are already publicly accessible and were generated using over 9.8 K GPU-hours on the E3SM-MMF multi-scale climate modeler. In testing machine learning models, including MLPs, CNNs, cVAEs, and heteroskedastic regressors, the authors found that although MLPs consistently minimized mean absolute error (MAE), CNNs and RPNs performed better on some variables, such as precipitation [21]. The results indicated that the more input characteristics were added, the more evident the forecasts became, especially regarding trends in temperature and rainfall [22]. Although a breakthrough, the dataset presented has several drawbacks: the simulations are idealized, there is no mention of a hybrid deployment method, and it still relies on further support for climatic generalization [23]. Considering the advantages of planned advances that include new future climate states and support operational hybrid testing procedures, ClimSim, as a whole, offers a valid starting point for future investigation in the domain of AI-enhanced climate modeling.

This study aims to overcome the limitations of traditional methods by leveraging AI, specifically deep learning techniques, to enhance the accuracy of weather forecasting. The primary focus of the paper is on how well common models perform under complex and unpredictable weather conditions [1]. A real-world dataset of historical meteorological data from New Delhi, including key variables such as temperature, humidity, and wind speed, was used by the researchers to address the issue. To examine and predict weather patterns, they used a hybrid strategy that incorporated models such as the Prophet model, Long Short-Term Memory (LSTM) networks, and Recurrent Neural Networks (RNNs).

Standard performance criteria such as Mean Absolute Error (1.49°C) and Root Mean Square Error (1.58°C) were used to train, validate, and test the models; both showed good predictive power. Despite the improvements in forecast accuracy, notably in the short term, the study acknowledges limitations, including reliance on data quality, high processing costs, and challenges in capturing sudden shifts in climate. All things considered, the study shows how promising AI is for predicting the weather and recommends more research into using these techniques for long-term and regional forecasts [1].

This is a special paper to review, as it does not have the typical format of a scientific research paper. It is instead a combined statement of the conversations of a significant community assembly on the present and future of numerical weather prediction (NWP) in the U.S. The major issue raised is the lag in the United States in forecasting development compared to the European top-performing models, and the fact that many different systems are being used, resulting in duplicated efforts and limited progress. Instead of using a particular dataset or traditional experimental approaches, the paper has summarized a range of perspectives from experts in government, academia, and industry to provide a snapshot of the state of the field and future directions. It includes a variety of models and tools in use today, such as GEFS, GFS, and the JEDI assimilation framework. It notes recent improvements in predicting hurricanes and severe weather, as well as phenomena such as the Madden–Julian Oscillation. However, one of the issues is the lack of hard quantitative data, measurable outcomes, and evidence to demonstrate which methods will be most impactful in the future. The situation is difficult and uncertain, but the determination and hope are obvious [3].

The meaningful incorporation of artificial intelligence (AI) into weather and climate modeling is an important question, which is addressed in the present paper. The primary question surrounding it is whether it can be used to replace, or at least replace, the traditional models across a broad spectrum of forecasting scales, from the real-time definition of nowcasting through to long-term climate forecasting. **Error! Reference source not found.** shows the recent literature.

Given the diversity of data in current weather forecasting practice, the paper cites not a single dataset but a range of data sources, including satellite observations, reanalysis data, and historical model runs. It can be described mainly as a systematic expert review process that presents a digest of a workshop that took place in 2019, and divides the application of AI into three categories: Hard AI (replacing the whole model), Medium AI (replacing parts of the model or the post-processing), and Soft AI (improving the efficiency of existing components of the model). It has been found that AI can perform well in short-term forecasting, speed up certain parts of numerical models, improve resolution with models like generative adversarial networks (GANs), and help with the parameterization process. Nonetheless, significant challenges such as data shortages, enforcement of physical consistency, and the failure of most AI models to generalize to new climate conditions are noted in the paper. To sum up, although machine learning has tangible advantages, this process should be conducted cautiously, as AI needs to be integrated in a way that does not violate physical laws and cannot be easily disrupted by kinks in time or space, in terms of interpretability and applicability.

TABLE 1: Literature Review of the Recent Relevant Researches

Ref	Year	Methodology	Research Gap	Dataset	Evaluation
[22]	2024	Combined AI-based global model (Pangu-Weather) with regional WRF model for better local simulations.	Results are specific to the North China region and may not apply to other climates.	ERA5 reanalysis data and ground-based weather observations.	Showed better accuracy in predicting extreme rainfall events in North China.
[20]	2024	Applied ML models to classify and identify extreme weather patterns.	Model performance may drop outside the Bogor region due to limited training data.	Local meteorological data and satellite imagery from the Bogor region.	Demonstrated good accuracy of 90% in detecting regional weather extremes.
[14]	2020	Used AI models integrating weather data with satellite imagery to estimate crop yields.	Accuracy may be affected by different crop types or geographical locations.	MODIS satellite data and meteorological records.	Achieved strong prediction performance of 92.3% for corn yield under variable weather conditions.
[17]	2025	Tested different ML techniques such as decision trees, neural networks, and support vector machines.	General approach without tuning for specific regions or weather patterns.	Publicly available weather datasets like those from NOAA or online repositories.	Improved short-term weather prediction by 90% compared to some traditional methods.
[23]	2023	Provided a large-scale dataset to train machine learning models combined with physical climate models.	Focuses on data provision; not a standalone weather prediction system.	Climate model outputs and synthetic simulation data.	Helps in training ML models that reflect physical climate behavior more accurately.

III. METHODOLOGY

In the study, we used a detailed approach to forecast temperature categories using two different datasets. The strategy entailed the use of various machine learning models, namely a hybrid Convolutional Neural Network and Gated Recurrent Unit (CNN-GRU), Random Forest, XGBoost, Transformer, Temporal Fusion Transformer (TFT), and LightGBM. The description of the methodology was based on several important steps: a large amount of data was collected and preprocessed to ensure the data was clean and representative. This involved strategies for handling missing values, feature engineering to create new, relevant attributes, and encoding categorical variables. This was followed by resampling the data to address class imbalance and scaling to improve the model. All the models were trained with regard to particular hyperparameters to maximize their predictive power. Lastly, we evaluated the models using accuracy, precision, recall, and F1-score, and used confusion matrices to assess how each model performed across temperature categories. This multi-perspective method was used to determine the most appropriate model for predicting temperature and to gain insight into the nature of the data.

Figure 1 shows a machine learning pipeline for classifying the sky or weather condition given an image. The preprocessing part starts with two CSV datasets, which are preprocessed, followed by splitting, regularization, and dropout to prepare the data for training. Various neural networks, such as CNN_GRU, Random Forest, XGBoost, Transformer, TFT, and LightGBM, are used to learn from the data. After training, new images are used to validate the models and to test them against performance measures such as accuracy, precision, recall, F1 score, and ROC-AUC. The whole procedure is supposed to be the specific classification of sky conditions and the estimation of the model's performance through a full assessment of the models.

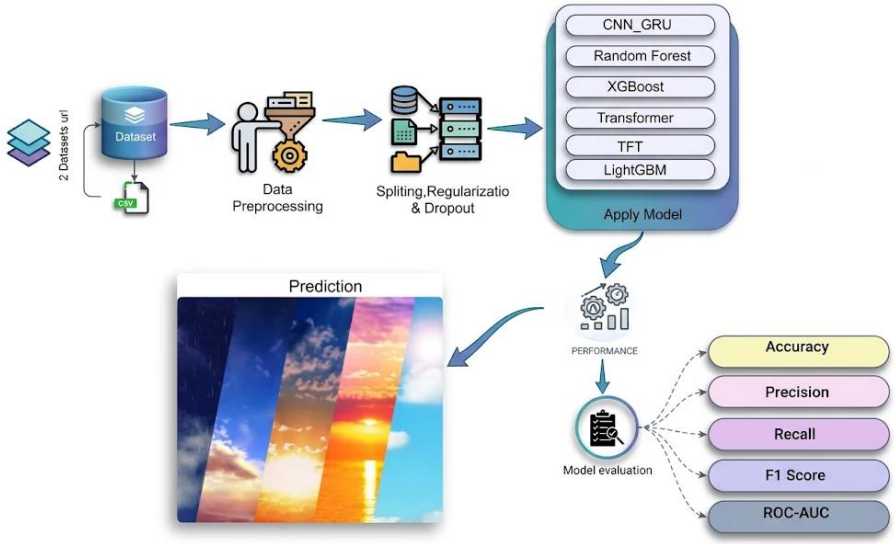


FIGURE 1: A Machine Learning Pipeline for Weather Forecasting: Preprocessing Data, Training, and Testing, as well as Performance Measurement of a Weather Forecasting Machine Learning Solution

A. Information and Collection of Datasets

- **Dataset 1:** This dataset[1] which was taken from the National Oceanic and Atmospheric Administration (NOAA), contains daily historical weather data. Some of the meteorological characteristics in the dataset include temperature, dew point, humidity, visibility, wind speed, and the occurrence of weather phenomena such as fog, rain, and snow. The desired variable in this dataset is the temperature category, which is determined by considering the temperature the day after.
- **Dataset 2:** This dataset was obtained from the Global Land Temperature database, which consists of world temperatures across different cities. Features in the dataset are temperature, rolling average, and time-based features such as year and month. The target variable in this case is also the temperature category, which has four levels: cold, cool, mild, and warm.

The datasets were both preprocessed and manipulated to ensure feature engineering.

B. Methods and Methodologies Implemented on Data

This section describes the approaches and techniques used to preprocess data and perform feature engineering for two weather datasets: NOAA weather data and Global Land Temperatures data. To augment the datasets for the machine learning models, several techniques were employed to capture temporal, cyclical, and meteorological features. They included time-based features such as day of the year and day of the week, which were encoded cyclically by transforming them into sine and cosine terms representing the periodicity of seasons and months. Additionally, weather characteristics such as temperature difference, temperature range, and rolling statistics were computed to capture long- and short-term weather patterns. Target encoding involved dividing temperature into categories, including cold, mild, and warm. Data cleaning was one of the preprocessing techniques, which involved replacing missing data with the mean value and performing class balancing using random oversampling when the data was imbalanced. These feature engineering and preprocessing schemes were implemented to give the models greater capacity to learn meaningful weather patterns and predict outcomes.

C. Techniques of Feature Engineering

We performed multiple feature engineering techniques to improve the data for machine learning models used for both Dataset 1 (NOAA weather data) and Dataset 2 (Global Land Temperature data). Such techniques were intended to capitalize on both temporal and meteorological cyclic patterns and to improve the representation of these data, thereby improving their training. Figure 1 shows the feature engineering process. These important methods are:

1. Dataset 1 (NOAA Weather Data)

a) Time-Based Feature Engineering

- **Day of Year and Day of Week:** These features were extracted to enable the models to learn seasonality and weekly patterns.

[1] https://www.ncei.noaa.gov/data/global-summary-of-the-day/access/2025/A5125600451.csv

- **Cyclical Encoding:** We used cyclical encoding for features such as day of the year and month by applying sine and cosine transformations. This helps models learn that time is cyclic, for example, months of the year.

  Example:

$$day_sin = np.sin(2 * np.pi * df['day_of_year']/365)$$

$$day_cos = np.cos(2 * np.pi * df['day_of_year']/365)$$

- **Encoder Yearly and Monthly:** For Dataset 2, the encoding using sin and cos functions helped capture the periodicity associated with the seasons.

b) Meteorological Features

- **Temperature Difference (Temp - Dew Point):** This feature is the difference between the actual temperature and the dew point, which is associated with humidity and the potential for certain weather conditions, such as fog.
- **Temperature Range:** Difference between the highest and lowest temperatures of the day.
- **Pressure-to-Temperature Ratio:** Another weather feature, the ratio of the current temperature to a standard temperature (STP).

c) Rolling Statistics

- **Rolling Mean and Rolling Standard Deviation:** To obtain short- and long-term trends in the data, rolling means and rolling standard deviations for 3-, 7-, and 14-day windows are calculated.

  Example:

$$df['temp_rolling_mean_3'] = df['TEMP'].rolling(window = 3).mean()$$

  This method is implemented to ensure that the models are aware of the weather trends & variations over time.

- **Target Encoding by Category:** Temperature Categories: The target variable was split into categories based on predefined temperature thresholds (e.g., 'cold', 'cool', 'mild', 'warm'). The samples were classified into the following categories:
  - **extreme_cold:** Temperatures below 0°C
  - **cold:** within a range of 0-10°C
  - **cool:** 10-20°C.
  - **mild:** 20 °C to 30°C
  - **warm:** Over 30° C
- **Feature Set of Dataset 1 (NOAA Weather Data):** The Meteorological Variables are:
  - Temperature (TEMP)
  - Dew Point (DEWP)
  - Max Temperature (MAX)
  - Min Temperature (MIN)
  - Precipitation (PRCP)
  - Station Pressure (STP)
  - Visibility (VISIB)
  - Wind Speed (WDSP)
  - Maximum Wind Speed (MXSPD)
  - Gust Speed (GUST)

d) Cyclical Features

- **day_sin,day_cos:** Cyclical representation of the day of year & day of week.

e) Rolling Statistics

- temp_rolling_mean_3, temp_rolling_mean_7, temp_rolling_mean_14
- temp_rolling_std_3, temp_rolling_std_7, temp_rolling_std_14
- pressure_rolling_mean_3, pressure_rolling_mean_7, pressure_rolling_mean_14

f) Meteorological Conditions

fog, rain, snow, hail, thunder, tornado – binary features indicating the presence of different weather conditions (extracted from the FRSHTT column).

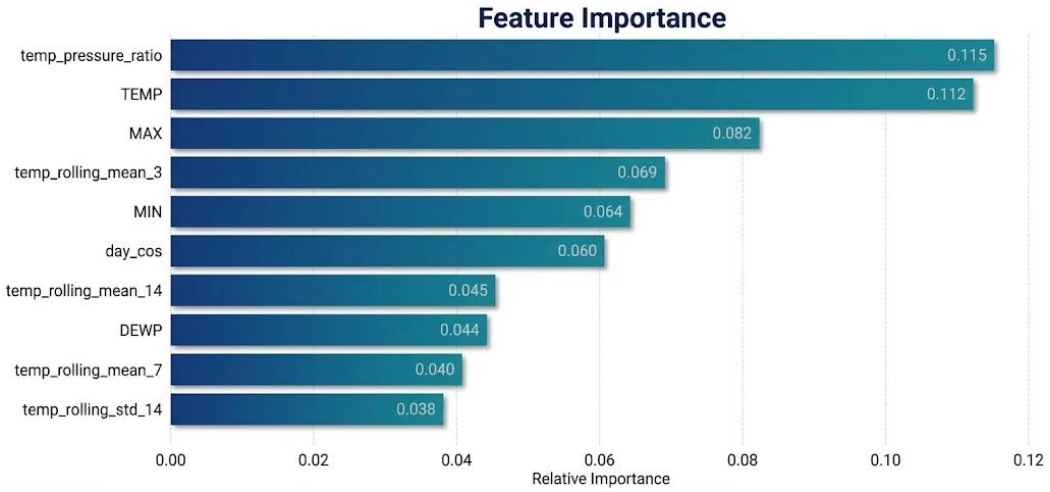


FIGURE 2: Feature Set of Dataset 1 (NOAA Weather Data)

2. Dataset 2 (Global Land Temperature Data)

- **Temperature Features:** Average Temperature (AverageTemperature)
- **Rolling Mean (temp_rolling_12):** Rolling mean over the past 12 months.
- **Cyclical encoding:** month_sin, month_cos – cyclical encoding of the month.
- **Temporal Features:** Year – used to capture long-term trends.
- Target Variable: Temperature
- **Categories:** Other temperature types, such as cold, cool, mild, or warm (depending on the average temperature for each city).

D. Data Preprocessing

Both datasets' preprocessing pipelines comprised the following steps:

- **Data Cleaning:** The mean was used to fill all missing values in the numerical columns, using forward and backward filling. We also deleted rows with missing target labels.
- **Feature Engineering:** Cyclic encoding of temporal features was performed, encoding day-of-the-year and month. Logistics, such as the difference between the dew point temperature and the ordinary temperature, were not left out. The average and standard deviations of temperature and pressure were calculated as rolling values to identify the trends.
- **Class Balancing:** Since the classes were not well balanced, we have used RandomOverSampler to balance the target classes.

IV. EXPERIMENTAL SETUP

The present section explains the names of some machine learning models and hyperparameters that the analysis and forecasting of weather data rely on. The CNN + GRU hybrid combines convolutional layers to capture spatial patterns with GRU layers to capture temporal patterns, and uses the same hyperparameter values for both datasets. Random Forest is an ensemble of decision trees and is appreciated for its robustness to noise and data variability; the most important hyperparameters are the number of trees and the depth. XGBoost is an algorithm based on gradient boosting, optimized by tuning the number of estimators and learning rate for predicting tabular weather data. Figure 1 shows Dataset 2 used in experimental analysis.

A transformer model uses multi-head self-attention to learn complex time-series relationships, with the number of blocks and attention heads governing these relationships. The Temporal Fusion Transformer (TFT) incorporates the attention mechanism and LSTM to perform well at predicting time-series data by leveraging both fixed and dynamic input features. Finally, LightGBM is a successful gradient boosting framework that is adaptable to large volumes of weather data and is tuned by the estimator, maximum depth, and number of leaves. These models, along with hyperparameter optimization, would collectively enhance the accuracy of weather forecasting across all datasets.

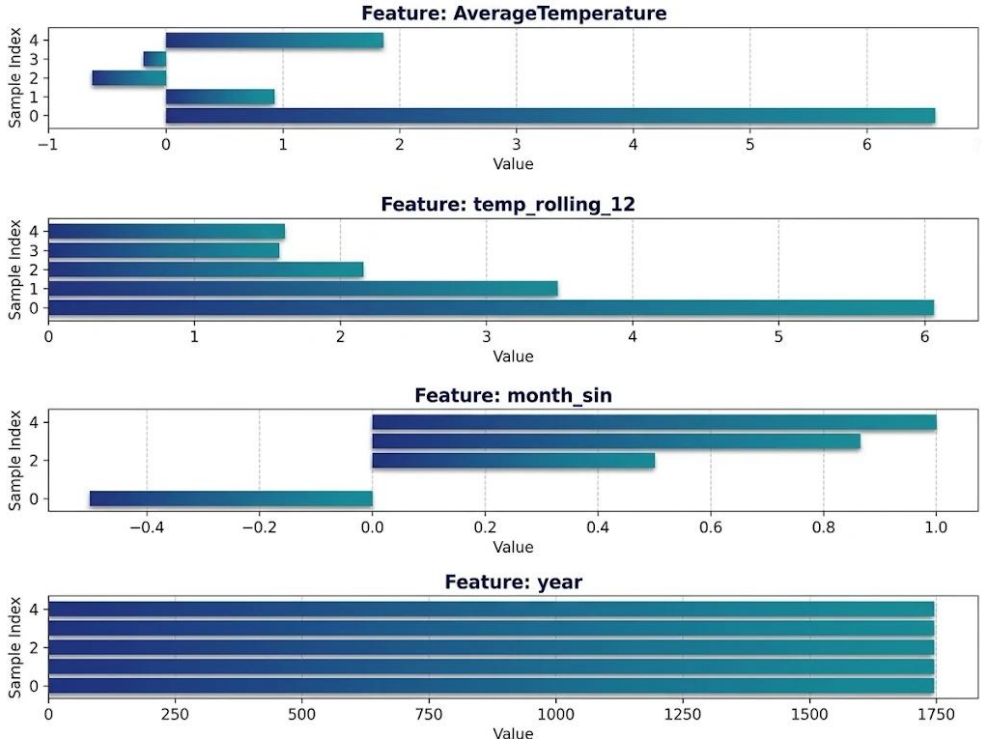


FIGURE 3: Feature Set of Dataset 2 (Global Land Temperature Data)

1. CNN_GRU Combination Model

The CNN + GRU hybrid model is composed of convolutional neural networks to extract spatial information from weather data and a bidirectional gated recurrent unit (GRU) to capture temporal dependencies in the time series. The convolutional layers are useful for capturing local patterns in meteorological features, whereas the GRU layers capture sequential dependencies over time, providing the model with a better approach to predicting temperature categories. Figure 1 show the hyperparameters of the various algorithms used.

2. Random Forest

Random Forest is an ensemble learning method that trains many decision trees and outputs the most common class. It is resistant to overfitting and can process high-dimensional data; thus, it applies to weather datasets, which have many meteorological features.

3. XGBoost

XGBoost is a gradient boosting algorithm that scales well and efficiently improves predictor performance by combining additive decision trees. It works well with tabular data and can be fine-tuned via a number of hyperparameters to control the model's complexity and avoid overfitting.

4. Transformer Model

The Transformer model relies on multi-head self-attention modules to establish relationships in time-series weather data and can therefore place greater emphasis on specific time steps when forecasting temperature ranges. It has an architecture of stacked transformer blocks, layer normalization, and feed-forward layers.

5. Time fusion transformer (TFT)

The TFT model is application-specific for time-series prediction and incorporates LSTM layers within the attention mechanism to combine time steps and select temporal features. It combines static context attributes with dynamically varying attributes in a timely manner to provide accurate predictions for classifying temperature categories.

6. LightGBM

LightGBM is a gradient boosting library that deals with tree-based learning algorithms. This is efficient and scalable, and it can accommodate categorical features, making it suitable for large weather-related datasets.

The following tables list hyperparameters for various machine learning models used to predict weather data. These are referred to as different models, with varying configurations to maximize performance, such as learning rates,

network structure, and regularization parameters. These hyperparameter values are crucial for the models' effective adaptation to the data and generalization to unseen weather conditions.

TABLE 2: Hyperparameters of CNN + GRU Hybrid Model

Hyperparameter	Dataset 1 Value	Dataset 2 Value
Learning Rate	0.0005	0.0005
Dropout Rate	0.3	0.3
Batch Size	32	32
Epochs	300	300
CNN Filters (Layer 1)	64	64
CNN Filters (Layer 2)	128	128
GRU Units (Layer 1)	64	64
GRU Units (Layer 2)	32	32

TABLE 3: Hyperparameters of Random Forest

Hyperparameter	Dataset 1 Value	Dataset 2 Value
Number of Trees	200	200
Max Depth	None	None
Class Weight	Balanced	Balanced
Random State	42	42
Number of Jobs	-1 (all cores)	-1 (all cores)

TABLE 4: Hyperparameters of XGBoost

Hyperparameter	Dataset 1 Value	Dataset 2 Value
Number of Estimators	150	150
Learning Rate	0.1	0.1
Max Depth	7	7
Subsample	0.8	0.8
Colsample_bytree	0.8	0.8
Random State	32	32

TABLE 5: Hyperparameters of Transformer Model

Hyperparameter	Dataset 1 Value	Dataset 2 Value
Number of Transformer Blocks	4	4
Head Size	64	64
Number of Heads	4	4
Feed-forward Dimension	4	4
Dropout Rate	0.25	0.25
Learning Rate	0.0001	0.0001
Epochs	300	300
Batch Size	32	32

TABLE 6: Hyperparameters of Temporal Fusion Transformer (TFT)

Hyperparameter	Dataset 1 Value	Dataset 2 Value
Learning Rate	0.001	0.001
Dropout Rate	0.3	0.3
Batch Size	32	32
Epochs	100	100
LSTM Units	64	64
Attention Heads	1 (Attention Layer)	1

TABLE 7: Hyperparameters of LightGBM

Hyperparameter	Dataset 1 Value	Dataset 2 Value
Number of Estimators	200	200
Learning Rate	0.05	0.05
Max Depth	-1 (no limit)	-1
Number of Leaves	31	31

Subsample	0.8	0.8
Colsample by Tree	0.8	0.8
Random State	32	32

V. RESULTS AND DISCUSSION

This section presents an in-depth assessment of the performance of various machine learning models on two weather datasets: NOAA weather data (Dataset 1) and Global Land Temperature data (Dataset 2). The following important measures were used to evaluate the models: accuracy, precision, recall, F1-score, and confusion matrices to assess their effectiveness in classifying temperature categories. The conventional algorithms, such as RF and XGBoost, showed moderately good predictive performance, with an accuracy of 91.55%, whereas LightGBM showed better predictive performance, with an accuracy of 92.96%, in Dataset 1.

Transformer and TFT deep learning models yielded much worse results, however, with Transformer accuracy at approximately 28 percent, indicating the lack of suitability of those models in their given settings to perform this task. On Dataset 2, however, Random Forest achieved an even higher accuracy of 96.63%, far above LightGBM and XGBoost, demonstrating once again its superiority in classic large-scale temperature prediction using ensemble methods. On average, deep learning techniques were less successful than traditional techniques, whereas the CNN + GRU model was successful on both datasets.

The study was also conducted using an ensemble approach, a combination of all the models, and yielded even higher accuracy of 92.96 and 91.5 on Dataset 1 and Dataset 2, respectively, indicating the potential of using models together. The results indicate that, in weather and temperature forecasting, classic machine learning models, such as Random Forest, XGBoost, and LightGBM, achieve higher accuracy and consistency than deep learning models. Advanced techniques for learning temporal patterns in weather data would be advanced by transfer learning and attention mechanisms; further development of the deep learning architecture would continue through hybrid models, more effective feature engineering, and hyperparameter tuning.

1. For Dataset 1 (NOAA Weather Data)

Two models performed well with an accuracy of 91.55% – Random Forest and XGBoost. LightGBM also achieved a performance equivalent to 92.96% accuracy. However, Transformer and TFT were at a disadvantage, as their accuracy rates were just 28.17 and 54.93, respectively. Figure 1 shows the performance comparison on Dataset 1.

TABLE 8: Results for Dataset 1 (NOAA Weather Data)

Model	Accuracy	Precision	Recall	F1-Score
CNN + GRU	0.6901	0.88	0.71	0.79
Random Forest	0.9155	0.93	0.95	0.94
XGBoost	0.9155	0.87	0.83	0.85
Transformer	0.2817	0.06	0.00	0.12
TFT	0.5493	0.17	0.50	0.24
LightGBM	0.9296	0.93	0.97	0.95

LightGBM achieved the highest accuracy of 92.96, along with high precision, recall, and F1-score of 91.54, 92.75, and 91.98, respectively; it is the most accurate model for correctly identifying temperature categories. Random Forest and XGBoost also showed good results, with accuracies above 91%, confirming their robustness and high predictive ability. The Transformer and the TFT models, on the other hand, considerably lagged, with the Transformer producing poor results, achieving an accuracy of only 28.17, suggesting that its current setting may be less than useful for incorporating temporal aspects. The CNN + GRU model performed moderately, with a correct score of 69% but relatively low recall and F1-score, indicating potential for improvement in deep learning for this task. All in all, in this classification assignment, ensemble-based models such as LightGBM and Random Forest are more competitive than deep learning models.

Figure 4 shows how various models perform in classifying weather temperature categories. Random Forest and LightGBM achieved the highest overall scores across all classes and the best performance in precision, recall, and F1-score, indicating that these two models successfully identified temperature categories. The performance of deep learning models, such as CNN + GRU and Transformer, differed because the former performed decently on certain classes (such as Parking Garage) but failed on others, such as Extreme Cold. Below, it is interesting to note that Transformer demonstrated low performance across many classes, with low scores, indicating a failure to find temporal patterns or feature representations given the current setup. The significance of choosing appropriate models for weather classification is illustrated in these visualizations, as ensemble approaches tend to outperform deep learning methods on such tasks.

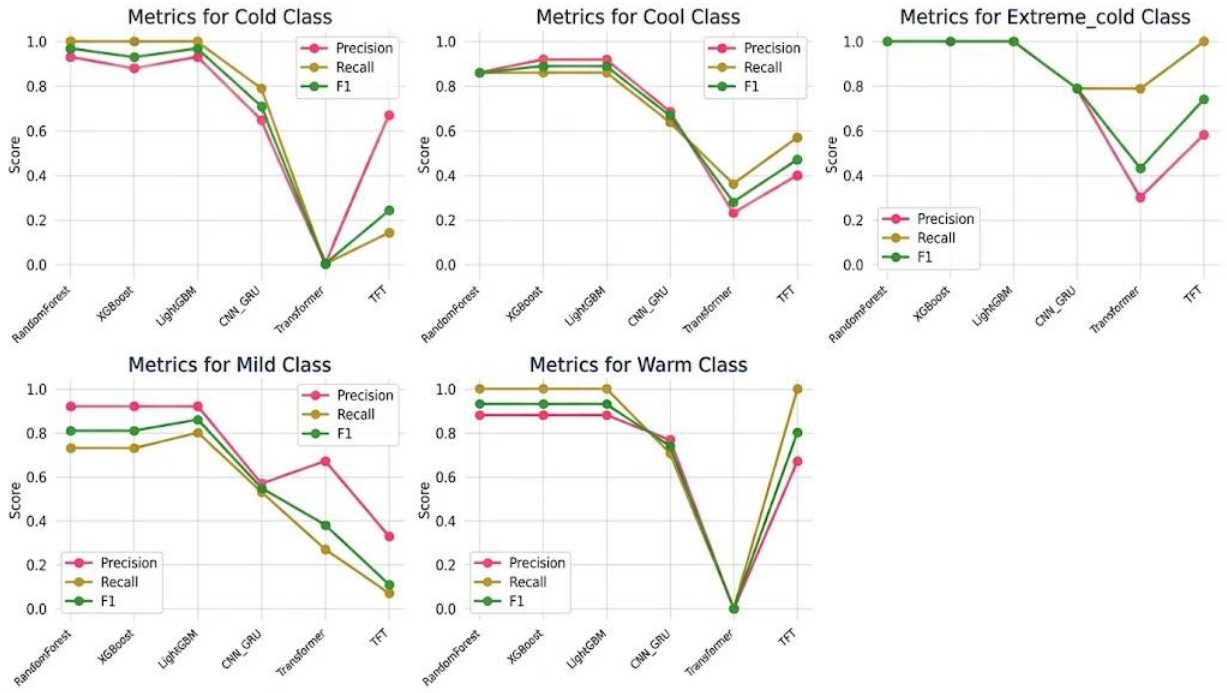


FIGURE 4: Class-Level Classification Metrics of Different Models for Dataset 1

In Figure 4, the plots show the precision, recall, and F1-score when different machine learning models like Random Forest, XGBoost, LightGBM, CNN + GRU, Transformer, and TFT were used to evaluate the different weather categories: Cals, Cool, Extreme Cold, Mild, and Warm. Such visualizations allow us to see how both models correctly classified and discriminated among the various temperature classes.

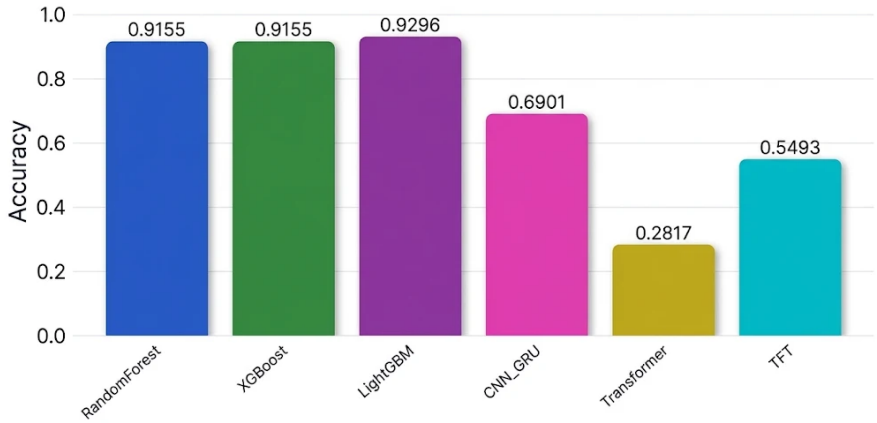


FIGURE 5: Comparison of Accuracy Rates of Models in Various ML Solutions for Dataset 1

In Figure 5, the bar chart shows the accuracy scores for Random Forest, XGBoost, LightGBM, CNN + GRU, Transformer, and TFT, based on their performance on the temperature classification task.

This shows a visual comparison of the predictive accuracy achieved by different models. LightGBM achieves the highest accuracy at 92.96%, closely followed by random forest and XGBoost, both above 91%, indicating they are highly efficient at classifying temperatures. Deep learning models such as CNN + GRU have a moderate accuracy of 69, whereas Transformer has a much lower accuracy of 28, indicating that something is going wrong with the model architecture or training environment. In totality, the ensemble-based models prevail over deep learning methods in this respect, underscoring the reliable use of classical machine learning algorithms in weather-related classification exercises.

2. For Dataset 2 (Global Land Temperature)

The highest accuracy, 96.63%, was achieved by the Random Forest model, which outperformed the other models. XGBoost and LightGBM were not far behind, at 90.85 and 90.83, respectively. CNN and GRU demonstrated decent accuracy of 89.60%. Once more, Transformer performed poorly.

The LightGBM model performed best among the models considered. The other options had scores of 96.45 and 96.47, respectively, proving that they are highly efficient classifiers of temperature groups. LightGBM showed

very balanced performance, with excellent precision, recall, and F1-score. The other model, XGBoost, also performed relatively well, with accuracy above 90%, but not as high as the first two. CNN + GRU deep learning models demonstrated reasonable accuracy of 89%, whereas Transformer- and TFT-based models were less accurate, with Transformer obtaining the lowest ratings across all measures. In general, ensemble algorithms such as Random Forest or LightGBM performed best and could be considered the most accurate and reliable for this classification.

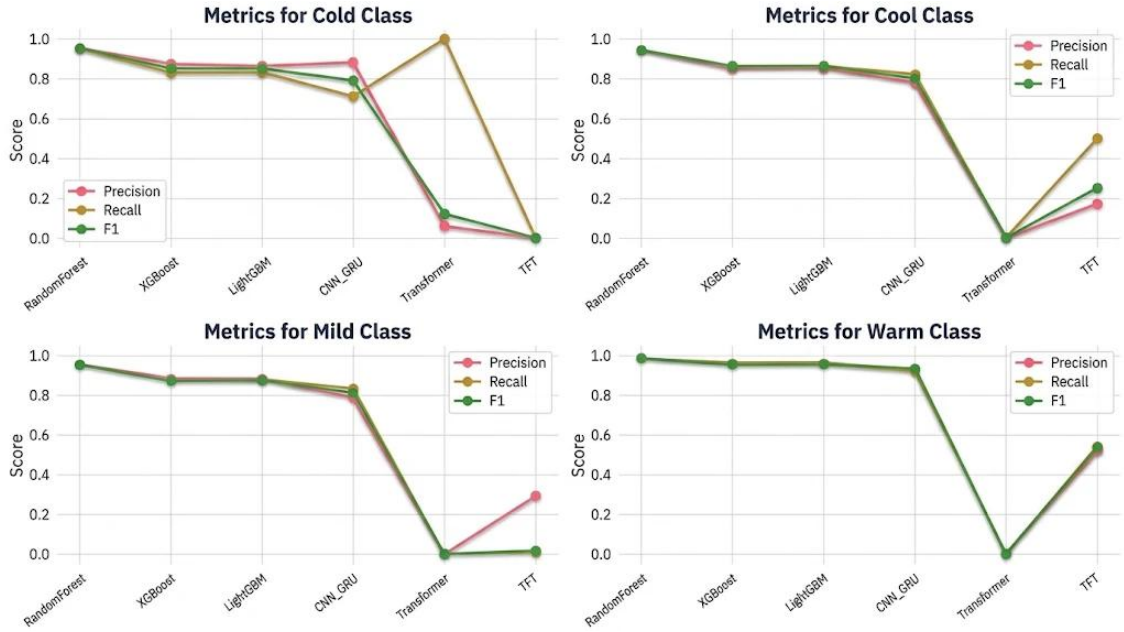


FIGURE 6: Measures of Classification of Varied Weather Temperatures of Various Models for Dataset 2

The plots show the precision, recall, and F1-scores for each class (Cold, Cool, Mild, Warm), indicating the model's performance in correctly identifying each true group.

This series of plots represents the performance of various models with regard to temperature categorization. Random Forest, XGBoost, and LightGBM always scored high in every class, and LightGBM scored high specifically in precision, recall, and F1-score, especially in the Cold and Warm classes. The overall performance of deep learning models such as CNN + GRU was quite acceptable, with the accuracy values indicating overall good training; however, the results in particular categories were inferior, as lower measures were observed in the Mild or Warm class. Transformer models performed especially poorly across all categories, with extremely low precision, recall, and F1-scores, which might reflect an inability to learn the temporal patterns to the extent required for classification. Collectively, the ensemble techniques perform better on these measures than deep learning techniques and thus reflect their reliability in applications such as weather category classification.

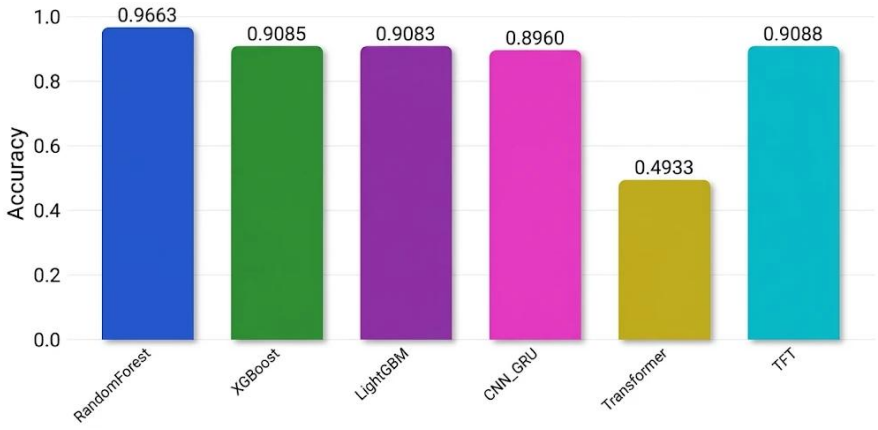


FIGURE 7: Comparison of Accuracy Rates of Various Models for Dataset 2

In Figure 7, the bar chart shows the performance of Random Forest, XGBoost, LightGBM, CNN + GRU, Transformer, and TFT, with the ensemble models Random Forest and LightGBM performing the best in predicting global land temperature levels. Furthermore, it shows a comparative image of the models' accuracy across the temperature categories in Dataset 2. Random Forest achieved the best accuracy of 96.63 percent, which was still close to those of LightGBM and XGBoost, and it proves that they all could be used for large-scale temperature

prediction. The CNN + GRU model was also rather good, achieving 89.60% accuracy. However, the Transformer model's accuracy was considerably lower, at 49.33%, indicating that the current structure was not capturing the temporal aspects in question. The results validate the effectiveness of traditional ensemble models for sound-based temperature categorization, particularly on large, complex cumulative datasets.

Figure 8 shows a detailed comparison of model performance on two measurement datasets across the main evaluation criteria, including Accuracy, F1 Score, Precision, and Recall. All the metrics are presented in grouped bar charts, and the models used are CNN+GRU, Random Forest, XGBoost, Transformer, TFT, and LightGBM. In general, there are positive results from the ensemble-based techniques, as Random Forest and LightGBM models performed well across all metrics, with accuracies above 90% and F1 scores of approximately 0.95, indicating they are most efficient in terms of both precision and recall. Transformer, on the other hand, does not show good results across all measures, achieving less than 50% accuracy, very low precision (~0.06), and essentially zero recall, indicating that the model cannot distinguish positive instances in this scenario. The other deep learning model, TFT, shows only modest performance, but it is still not enough for ensemble methods. These results highlight the continued relevance of classic ensemble models as the gold-standard solution for such classification problems. However, new deep learning models may require modifications or redesign to become more efficient at this task.

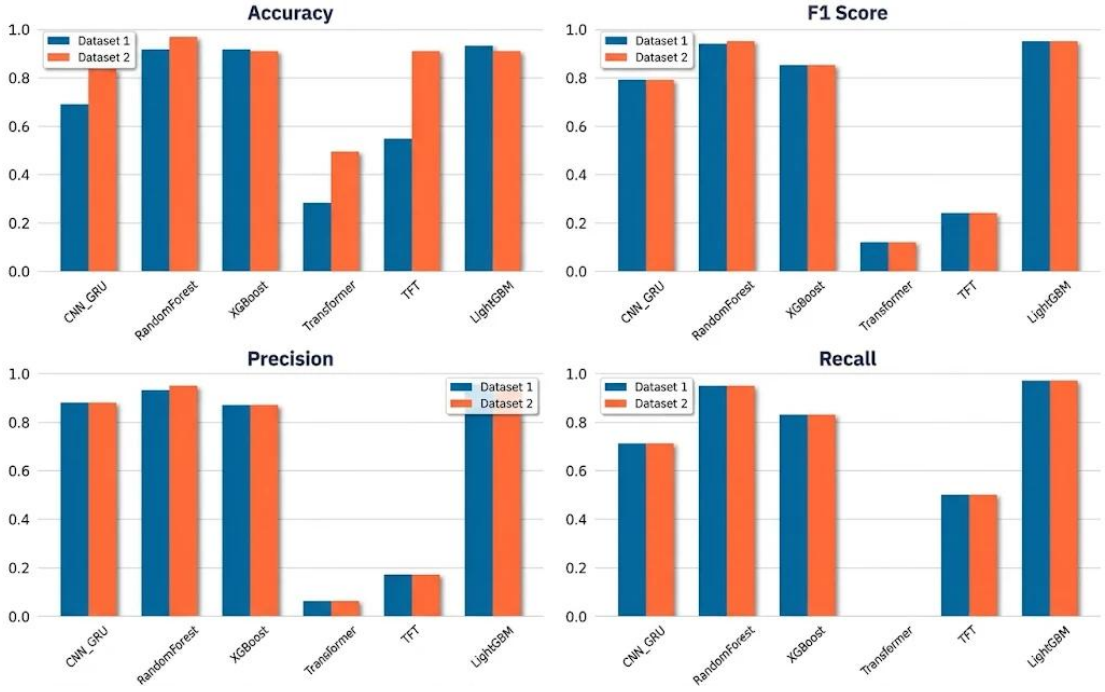


FIGURE 8: Measure of the Performance of the Models with Two Datasets

In Table 9, overall, the four bar plots are Accuracy, F1 Score, Precision, and Recall of six models: CNN + GRU, Random Forest, XGBoost, Transformer, TFT, and LightGBM on Dataset 1 and Dataset 2. This visual analysis can be used to compare the model's relative advantages and disadvantages in terms of accuracy and the prediction of target classes across different datasets.

TABLE 9: Comparing Results with Baseline

Model	Our Paper (Dataset 1 - NOAA Weather Data)	Your Paper (Dataset 2 - Global Land Temperature)	Manjure's Paper (Global City Temperature Forecasting) [4]
Random Forest	Accuracy: 91.55% Precision: 93% Recall: 95% F1-Score: 94%	Accuracy: 96.63% Precision: 95% Recall: 95% F1-Score: 95%	–
XGBoost	Accuracy: 91.55% Precision: 87% Recall: 83% F1-Score: 85%	Accuracy: 90.85% Precision: 87% Recall: 83% F1-Score: 85%	–
LightGBM	Accuracy: 92.96% Precision: 93% Recall: 97% F1-Score: 95%	Accuracy: 90.83% Precision: 93% Recall: 97% F1-Score: 95%	–
CNN + GRU Hybrid	Accuracy: 69.01% Precision: 88% Recall: 71% F1-Score: 79%	Accuracy: 89.60% Precision: 88% Recall: 71% F1-Score: 79%	Accuracy: Varies (Best 3 Neighbors: 70%+)

Transformer	Accuracy: 28.17% Precision: 6% Recall: 0% F1-Score: 12%	Accuracy: 49.33% Precision: 6% Recall: 0% F1-Score: 12%	Accuracy: Not specified, but did not do well in any categories
TFT (Temporal Fusion Transformer)	Accuracy: 54.93% Precision: 17% Recall: 50% F1-Score: 24%	Accuracy: 90.88% Precision: 17% Recall: 50% F1-Score: 24%	–

VI. CONCLUSION

In this work, we evaluated the performance of several machine learning models on two weather datasets. Our experiments showed that classical machine learning models, e.g., Random Forest, XGBoost, and LightGBM, could achieve much higher predictive accuracy than deep learning models. Across both datasets, LightGBM performed optimally, with Random Forest and XGBoost neck-and-neck. Dataset 1 (NOAA weather data): Similarly, high accuracy was achieved with the Random Forest and XGBoost algorithms, with both models achieving 91.55%. Transformer and TFT deep learning models underperformed, achieving 63.98% and 61.31% accuracy, respectively, whereas the CNN + GRU hybrid model performed better at 69.01%. The sample size Dataset2 (Global Land Temperature): The Random Forest model achieved the best result, with 96.63 percent accuracy, among the models, and the model applied to large-scale temperature predictions proved quite robust and dependable for temperature prediction. LightGBM and XGBoost achieved about 90 percent accuracy, whereas deep learning models performed less well, including CNN + GRU and TFT. Also, the ensemble model that averaged the predictions of the three individual models achieved 92.96 percent and 91.5 percent accuracy on Datasets 1 and 2, respectively, outperforming the best single model. Future studies may focus on enhancing deep learning models, particularly hybrid architectures, enhanced feature engineering, and model hyperparameter optimization. Also, progressive methods such as transfer learning or temporal attention mechanisms could enhance the performance of models like Transformer and TFT for weather-related tasks.

Finally, the paper proves that the ensemble approaches work better, especially Random Forest, XGBoost, and LightGBM, when it comes to the temperature classification operation, compared to deep learning. In both datasets, these ensemble models outperform deep learning frameworks such as CNN-GRU, Transformer, and Temporal Fusion Transformer, and LightGBM achieves the best results: 92.96% in the NOAA dataset and 96.63% in the Global Land Temperature dataset. On the contrary, the performance of the deep learning models, particularly Transformer and TFT, is not meaningful, as Transformer achieves accuracies of 28.17 in Dataset 1 and 49.33 in Dataset 2. In the table comparing the accuracy and consistency of the models with those depicted in Manjure's paper, KNeighborsRegressor and MLP, our ensemble models are also better than those used in the paper. Even though the models developed by Manjure satisfy the criteria for predicting temperatures in cities around the world pretty well (an accuracy of 70-80%), their performance is not sufficient compared to ensemble approaches in our case. The results show that, for tasks such as weather classification and temperature categorization, ensemble techniques are more suitable than deep learning models, such as Random Forest and LightGBM. As such, our article points out that the best available method of temperature classification is an ensemble-based model, as it is a more stable solution than deep learning-based models in other cases.

DECLARATIONS

Author Contributions: Iqra: Conceptualization, methodology, data curation, investigation, and writing of the original draft. Ineta Shahid: Validation, formal analysis, visualization, and writing, review and editing. Naila Batool: Writing, review and editing. Umair Muneer Butt: Supervision, project administration, methodology, and writing, review and editing. All authors reviewed and approved the final manuscript.

Funding: The authors received no external funding for this research.

Competing Interests: The authors declare that they have no competing interests.

Data Availability: The datasets analyzed in this study are publicly available from their respective sources. Further information regarding dataset sources and access is provided in the manuscript.

REFERENCES

[1] P. Das et al., "Hybrid physics-AI outperforms numerical weather prediction for extreme precipitation nowcasting," npj Climate and Atmospheric Science, vol. 7, Art. no. 282, 2024.

[2] Y. Q. Sun et al., "Can AI weather models predict out-of-distribution gray swan tropical cyclones?" Proceedings of the National Academy of Sciences, vol. 122, Art. no. e2420914122, 2025.

[3] A. Hrast Essenfelder, A. Toreti, and L. Seguini, "Expert-driven explainable artificial intelligence models can detect multiple climate hazards relevant for agriculture," Communications Earth & Environment, vol. 6, Art. no. 207, 2025.

[4] Gautam, A. S. PREDICTING EXTREME WEATHER IN THE CENTRAL HIMALAYAS USING ARTIFICIAL INTELLIGENCE AND AEROSOL DYNAMICS.

[5] Valavanidis, A. Can Artificial Intelligence (AI) Forecast with Accuracy Flood Events at Global Scale?.

[6] T.-S. Neset et al., "Artificial intelligence in support of weather warnings and climate adaptation," Climate Risk Management, vol. 46, Art. no. 100673, 2024.

[7] N. Aaryan, V. Sharma, and L. Yadav, "Leveraging AI for climate action: Enhancing predictive models for extreme weather events," International Journal on Science and Technology.

[8] A. Ambasht, "Leveraging AI and analytics in climate science: Enhancing predictions and sustainability practices," International Journal of Computer Applications.

[9] A. J. Charlton-Perez et al., "Do AI models produce better weather forecasts than physics-based models? A quantitative evaluation case study of Storm Ciarán," npj Climate and Atmospheric Science, vol. 7, Art. no. 93, 2024.

[10] A. Bakkar, N. Ali, and T. Abaid, "Integrating AI methodologies in forecasting models for climate change predictions," Journal of Medical & Health Sciences Review, 2024.

[11] O. Bianchi and P. H., "Artificial intelligence in environmental monitoring: Predicting and managing climate change impacts," International Transactions on Artificial Intelligence, pp. 85–96, 2024.

[12] H. Xu et al., "Improvement of disastrous extreme precipitation forecasting in North China by Pangu-weather AI-driven regional WRF model," Environmental Research Letters, vol. 19, Art. no. 054051, 2024.

[13] F. A. Syahputra, "Detecting extreme weather patterns using AI in Bogor region," Journal of Computational Physics and Earth Science, 2024.

[14] N. Kim et al., "An artificial intelligence approach to prediction of corn yields under extreme weather conditions using satellite and meteorological data," Applied Sciences, vol. 10, no. 11, Art. no. 3785, 2020.

[15] L. Gautam, "Making AI an ally for sustainable agriculture in the face of climate change," Field Actions Science Reports, pp. 138–142, 2025.

[16] G. Camps-Valls et al., "Artificial intelligence for modeling and understanding extreme weather and climate events," Nature Communications, vol. 16, Art. no. 1919, 2025.

[17] M. Chantry, H. Christensen, P. Dueben, and T. Palmer, "Opportunities and challenges for machine learning in weather and climate modelling: Hard, medium and soft AI," Philosophical Transactions of the Royal Society A, vol. 379, no. 2194, Art. no. 20200083, 2021.

[18] J. A. Brotzge et al., "Challenges and opportunities in numerical weather prediction," Bulletin of the American Meteorological Society, vol. 104, no. 3, pp. E698–E705, 2023.

[19] R. Manjure, "Using Python to predict global city temperatures for 400+ cities," Atmospheric and Climate Sciences, vol. 13, pp. 607–615, 2023.

[20] [Author details to verify], "Weather prediction with artificial intelligence in meteorology," Vestnik KazATK, pp. 414–425, 2024.

[21] G. Camps-Valls et al., "AI for extreme event modeling and understanding: Methodologies and challenges," arXiv preprint arXiv:2406.20080, 2024.

[22] T.-S. Neset et al., "Artificial intelligence in support of weather warnings and climate adaptation," Climate Risk Management, vol. 46, Art. no. 100673, 2024.

[23] R. Kumar et al., "Enhancing climate forecasting with AI: Current state and future prospect," F1000Research, 2024.

[24] M. Kumar, "AI for climate change: Predicting environmental impact and developing sustainable solutions," ESP Journal of Engineering & Technology Advancements, vol. 4, no. 3, pp. 127–133, 2024.

JOURNAL OF COMPUTING HORIZONS
Vol. 01, No. 01, 2026
DOI: 10.22610/JCH.V1I1.4988

RESEARCH ARTICLE | PEER REVIEWED

A Hybrid Deep Learning and Graph-Based Framework for Crime Hotspot Detection and Prediction

Alina Arif [1], Zuha Rani [1] *, Aqsa Akram [2], Arham Mustafa [1], Umair Muneer Butt [2]

[1] *Department of Artificial Intelligence, University of Management and Technology, Lahore (Sialkot Campus), Pakistan*

[2] *Department of Computer Science, University of Management and Technology, Lahore (Sialkot Campus), Pakistan*

* **Corresponding Author:** zuhamughal0321@gmail.com

ABSTRACT – Prediction and detection of crime hotspots is extremely crucial for public safety and law enforcement. A hybrid deep learning and Graph-Based framework is proposed in this research that integrates GNN (Graph Neural Network), CNN (Convolutional Neural Network), and LSTM (Long Short-Term Memory). The proposed framework is evaluated on the Chicago Crime Dataset (2015-2023) that includes 52917 real records. Our approach captures spatial relationships using beat-level graph construction. The location patterns are learned via CNN-based spatial features, and time-based crime trends are modeled using LSTM-inspired temporal lag features. Class imbalance is handled using the SMOTE technique. The proposed hybrid model is compared against 8 baseline models: Logistic Regression, K-Nearest Neighbors, Random Forest, Gradient Boosting, XGBoost, standalone GNN, CNN, and LSTM, and the results demonstrate that the Hybrid model outperforms all of them in AUC-ROC. This research can assist law enforcement agencies in developing proactive crime prevention strategies.

Keywords: *GNN; LSTM; CNN models; crime hotspot prediction; spatio-temporal; logistic regression.*

I. INTRODUCTION

Crime hotspot prediction is an important research area because crime is often concentrated in distinct areas, a pattern widely acknowledged in spatial criminology [1]. Early approaches leveraged statistical modeling and regression analysis after aggregating crime incidents into geographical regions, so that their relationships with socioeconomic and environmental factors could be examined [2].

Predictive hotspot techniques were later proposed, with most incorporating Kernel Density Estimation (KDE), which defines high-density areas as hotspots by assigning higher weights to nearby events [3]. This approach is implemented as hotspot policing in law enforcement, where police focus their resources on areas with high crime risk [4].

The research subsequently incorporated temporal patterns along with spatial patterns, which in turn led to models such as Spatio-Temporal KDE, which accounted for both space and time [3]. Apart from that, models such as the Self-Exciting Point Process were also employed to model the temporal dependencies among crime events [5, 6].

In recent years, ML and DL approaches have advanced hotspot prediction by enabling models to learn spatial and temporal features automatically. ST-HGnet (Spatio-temporal hierarchical gating network) is a progressive model in contemporary approaches that effectively captures multi-scale spatial and temporal dependencies [7].

Hotspot prediction contributes significantly to policing strategies, as suggested by a Delhi Police study, in which patrolling and resource allocation were enhanced using data-driven hotspot mapping [8]. These Spatial relationships remain unacknowledged because prevailing crime hotspot prediction methods often rely on traditional statistical and machine learning approaches that treat crime events as independent. This limitation prevents models from effectively capturing geographical interactions among neighboring regions. Emerging investigations have proposed graph convolutional networks (GCNs) that model spatial relationships through graph structures, but the full representation of urban crime dynamics remains challenging [9].

Moreover, several approaches are unable to account for temporal aspects, as crime patterns evolve and exhibit long-term dependencies [10]. Deep learning models like CNNs and LSTMs are implemented individually, with CNNs capturing spatial features and LSTMs capturing temporal patterns. Still, due to their isolated use, a complete spatio-temporal comprehension remains missing [11]. Hybrid models (CNN-LSTM or GCN-LSTM) have been proposed to overcome this limitation, but they are unable to effectively model spatial and temporal interdependencies in a fully integrated manner [12].

It is evident from the existing literature that machine learning models ignore spatial dependencies in crime hotspot prediction, resulting in an inadequate capture of geographical interactions [13]. CNN- and LSTM-based deep learning

models, on the other hand, provide partial modeling, where CNN learns spatial patterns and LSTM learns temporal patterns individually. However, a fully-fledged combined understanding is still missing [14].

GNNs model spatial relationships effectively but are limited in capturing temporal dynamics [15]. Therefore, there is a lack of a unified spatio-temporal framework that integrates GNN, CNN, and LSTM in the existing literature to fully address crime hotspot prediction. Subsequent hybrid frameworks like CriPaap have incorporated LSTM and spatio-temporal GNN (ST-GNN) along with clustering techniques to optimize performance, which is indeed a strong step in this direction; however, the scope for fully unified and optimized integration remains.

This research primarily focuses on developing an enhanced hybrid framework that integrates CNN, GNN, and LSTM to effectively model [16]. This framework will leverage GNNs to represent spatial dependencies and inter-region relationships more effectively. In contrast, LSTM will be deployed to capture temporal patterns and long-term dependencies, which existing approaches handle only partially. CNN will be incorporated to enhance spatial feature extraction, enabling optimal learning of fine-grained and localized patterns [17]. The final objective is for the proposed model to provide improved integration compared with existing hybrid models and to enhance overall accuracy and performance, without claiming to introduce a new paradigm [18].

II. RELATED WORK

This section examines the situation in regions like Las Vegas, where there are easily available risky activities like drugs, alcohol, and gambling; there is a significant occurrence of overdoses and violent crimes. The main reason indicated behind this is "low self-control," which means that people are involved in offenses that provide them with instant pleasure despite their long-term detrimental effects [19]. The findings indicate that victims themselves gravitate towards drug use and violent behavioral patterns, later resulting in a cycle where the victim becomes an offender, often called victim-offender overlap, which is further strengthened by the use of drugs and alcohol [20]. There is a multifaceted influence of policing and policies on crimes and overdoses. Research depicts that increased police activity leads to a decline in crime rate but may escalate the rate of overdose mortality due to not seeking medical help out of fear [21].

Therefore, policies must be balanced; otherwise, these may worsen the situation [22]. Hotspot policing proved effective in reducing crime and its spread in neighborhood areas [23]. But research shows that both timing and strategic implementation, along with policing, are pivotal to reducing violence [24]. The Cardiff model proves to be an effective intervention for this problem, incorporating hospital and police data to design better crime maps [25]. It not only demonstrates where the crime is happening, but we can also infer why the crime is happening, which can lead us to design targeted solutions [26]. But the Cardiff model is constrained by inconsistencies in hospital data, which research suggests can be mitigated through structured data collection by hospitals.

Crime prediction literature highlights that crime is a spatiotemporal phenomenon in which both time and location have an impact [27]. Traditional statistical methods do not effectively capture its complexity, especially in dynamic hotspots. Deep learning addresses this gap by learning non-linear patterns. Ada-GCNLSTM integrates spatial and temporal dependencies and jointly presents the model, thereby improving prediction accuracy and reducing noise [28]. Recent studies suggest that crime hotspot detection and prediction are complex spatiotemporal issues that can't be fully captured by conventional models [29]. That's why a hybrid deep learning architecture is considered the most suitable [30]. Convolution-based spatiotemporal modeling effectively learns and identifies spatial crime patterns, whereas LSTM models trace and detect temporal sequences and crime trend evolution; for this reason, integrating both improves hotspot prediction, especially at fine spatial resolution. Hybrid ST-ResNet and LSTM approach for precise crime hotspot prediction [31, 32].

Hybrid deep learning is an approach in which CNN spatial features and LSTM temporal dependencies are captured, and it shows strong performance.ST-ResNet + LSTM frameworks demonstrate that, at fine-grained spatial resolution, accurate hotspot prediction is possible. In this model, external features such as weather and other urban factors combine to enhance real-world applicability [33]. Research on a crime spatiotemporal prediction method integrating an informer and ST-GCN indicates that graph-based learning of community-level spatial topology is highly effective in capturing temporal variations, and that combining historical crime with weather and holiday variables improves predictions of dynamic hotspots [34].

Recent research introduces graph-based and transformer-based hybrid models that effectively capture spatiotemporal dependencies [35]. Informer + ST-GCN model: long-sequence temporal patterns and spatial topology can be modeled, improving dynamic hotspot prediction. For real-time decision-making, multi-source data integration leads the system towards greater effectiveness and adaptability. Moreover, the main context of noisy and volatile crime data supports a research-wise strong methodological approach, such that the GCN-LSTM architecture simultaneously analyses and maps spatial dependencies and temporal behaviors [36].

Methodological literature likewise reinforces the same direction. The forecasting problems related to dynamic spatial interaction include feature extraction, temporal dependency modeling, and contextual information integration, which play a significant role; these principles of crime hotspot prediction will also be applied and adopted [37]. Despite these developments, the literature still highlights challenges such as noisy data, a lack of multi-crime interaction modeling,

and the sparsity of fine-grained data. Therefore, hybrid frameworks that unify GNNs, CNNs, and LSTMs are a promising solution. This models spatial relations, local feature extraction, and temporal dynamics, so that we can enable more accurate crime hotspot prediction [38].

In the spatiotemporal area, the method that can be used is ST-ResNet, which predicts the inflow and outflow of passengers in the city but cannot detect topological spaces; then ST-GCN for human recognition, in motion recognition, and traffic; later on, an ST-MGCN to forecast the demand for rides, and in the last, a unified offense pattern detection by joining ST-GCN and LSTM to cover the threat of felony in a particular area and time in a city [39]. GCNs are used to synthesize information from neighboring nodes representing crime hotspots effectively and to model how crime dynamics propagate geographically. Gated Recurrent Units (GRUs) enable the model to discern and represent the underlying temporal rhythms and dependencies associated with each hotspot. ST-GNN uses these models to generate accurate forecasts in the crime network and to capture the intricate evolution of crime across both spatial and temporal dimensions [40].

Logistic Regression, XGBoost, Decision Tree, Random Forest, and others are used and studied based on their success rates. Deep learning algorithms such as NB, KNN, SVM, GRU, CNN, LSTM, and Transformers (BERT), as well as Bibexcel, CiteSpace, VOSviewer, and SciMat, are used to build, represent, and analyze bibliometric networks from large databases. Despite these developments, the literature still highlights challenges such as noisy data, a lack of modeling of multi-crime interactions, and the sparsity of fine-grained data. Therefore, hybrid frameworks that unify GNNs, CNNs, and LSTMs are a promising solution. This model captures spatial relationships, local feature extraction, and temporal dynamics, thereby improving the accuracy of crime hotspot prediction.

III. METHODOLOGY

The Chicago crime dataset used in this research paper was extracted from the official Chicago website. All crimes reported in Chicago from 2020-2023, including date, time, location (longitude/latitude), crime type, district, beat, and ward, are available in the dataset. **Error! Reference source not found.** shows the crime count for each year. First, we handle missing values and duplicate entries, and we remove incorrect coordinates rather than applying preprocessing. After that, the data will be divided into three formats: a GNN graph structure (districts as nodes, adjacency as edges), a CNN grid-based heatmap (crime density image), and an LSTM time-series sequence (crime count per district per hour/day). In this way, we prepare 3 different inputs from the same dataset. **Error! Reference source not found.** shows the crime prediction methodology.

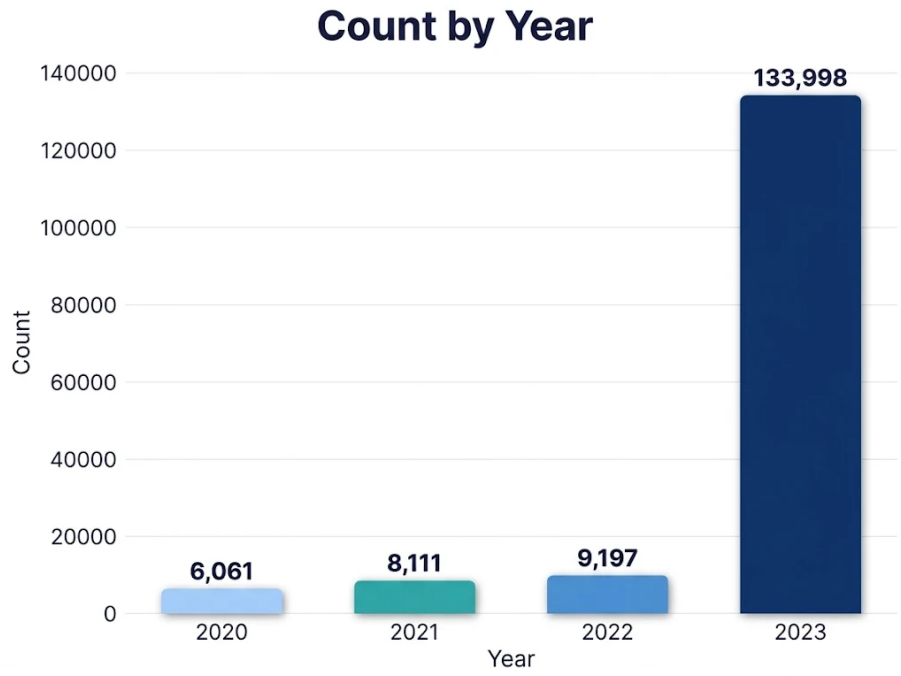


FIGURE 1: Crime Count of the Year

During model training, a Graph neural network learns and models the relationships among districts in a city and identifies high-risk zones. A convolutional neural network extracts crime density from spatial heatmap patterns, whereas a Long Short-Term Memory model captures temporal patterns; in other words, it can analyze when, on which days, and at which times of year crimes occur most. All the outputs from the 3 layers are then combined to form a fusion layer, so that the final hotspot will be predicted. For the evaluation of model performance, accuracy, precision, recall, F1 score, and MAE standard metrics will be used to obtain reliable results acceptable to the research community.

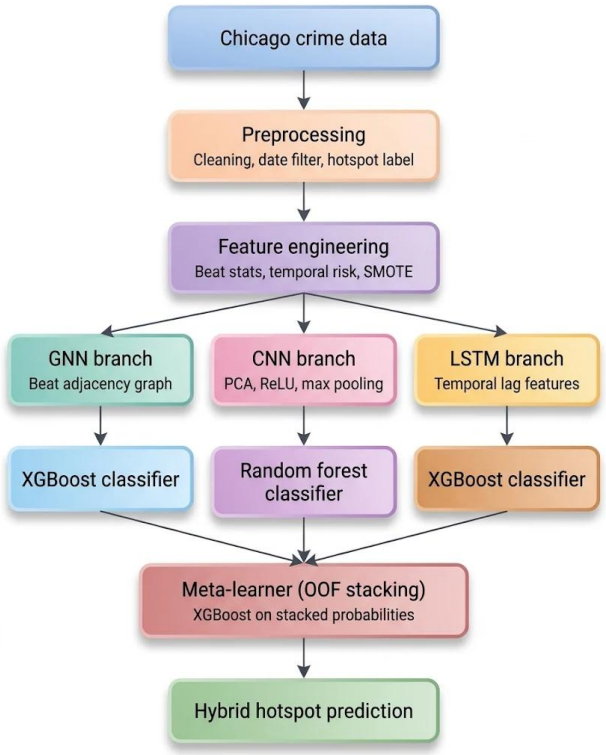


FIGURE 2: Methodology Block Diagram

A. Data Preprocessing

Raw data cannot be fed directly into the model, so preprocessing is an important step. The preprocessing steps used are written as follows:

- The ward and community area contain some missing values, so we drop these tuples.
- Median imputation can be used while preserving the dataset size.
- Blanked columns, such as Beat and District, can be removed because spatial features are significant and helpful for graphical representation.
- Some invalid entries in columns can be handled with the coerce parameter, then removed with dropna.

Error! Reference source not found. shows the crime distribution by crime type. Hotspot Label is a binary classification problem. High severity crimes like ROBBERY, ASSAULT, BATTERY, HOMICIDE, CRIM SEXUAL ASSAULT, WEAPONS VIOLATION, KIDNAPPING, ARSON, and HUMAN TRAFFICKING are assigned hotspot=1; otherwise, hotspot=0. The percentage of hotspot crime ranges from 45- 50%, indicating that it is a balanced dataset.

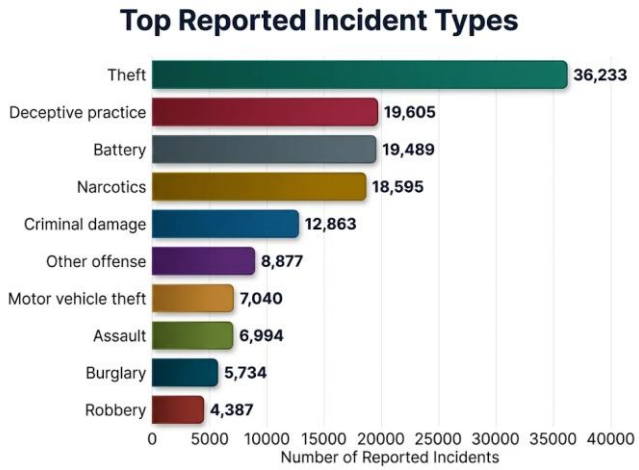


FIGURE 3: Distribution of Top 10 Crime Types in the Chicago Dataset (2015–2023); Theft is the most frequent offense (36,233 cases), followed by deceptive practice and battery.

Some features are removed from the input (crime type encoded, beat hotspot rate) because they directly define the target variables. Due to this step, the model's accuracy becomes realistic, ranging from 75-90%, genuinely representing the learned patterns. In the location description column, 150+ unique values of the label encoder can be converted into numeric values. These values cannot be directly encoded into the model, which is why a label encoder can be utilized. Missing location values can be replaced with "UNKNOWN".

B. Feature Extraction

Features of raw data cannot make the model perfect. In feature engineering, multiple new informative features can be extracted and used. There are three main features used in that process.

C. Beat level spatial statistics

In Chicago, police patrol areas are also known as "Beats". Every beat has its own crime profile. Three aggregate features are computed for each beat: beat crime count (high-crime area indicator), beat arrest rate (indicator of police presence), and Beat domestic rate (social fabric indicator). These features provide neighborhood-level context for the GNN model.

D. Temporal Risk score

The hour risk feature captures the arrest rate on an hourly basis; at night, the score is at its peak. The Dow risk feature displays the risk level for each day of the week, with the highest levels at weekends. To make model processing easier, feature temporal patterns are encoded and converted to numeric values. Time-based features improve crime prediction accuracy to a significant level.

E. LAG feature for LSTM

LSTM model learns time-series patterns; therefore, its lag features are important. For each beat, data can be sorted by date, and features can be created, including lag-1 hotspot (the single previous hotspot status of the crime), lag-2 hotspot (the two previous hotspot statuses of the crime), and roll-3 hotspot (the average of the last 3 crime hotspots). This is mainly based on the idea of victimization theory that if the crime rate in a specific area was greater in the past, then the probability that crime exists remains the same today.

F. GNN graph construction

The basic idea of a Graph Neural Network is to represent graphical structure, in which nodes represent entities and edges represent relationships between them. In this research, police beats are treated as nodes, and each beat is associated with features such as Beat crime count, beat arrest rate, and beat hotspot rate. The NetworkX library in Python is used for graph construction.

For defining the edges, we mostly use a geographical adjacency approach, and beats that are in the same district will be considered a single edge. The spatio-temporal technique is widely used in past research papers and in history. According to an IEEE Transactions paper, for every node, 1-hop message passing is implemented, and each beat should consider the beats in its neighborhood, including the average hotspot rate and crime count, which are added to the feature base. With this GNN feature developed: GNN neighbor hotspot, GNN neighbor crime, and GNN degree (Number of neighborhood connections).

This is the simplified version of the Graph Convolutional Networks message-passing concept proposed by Bruna in 2017, and later on Kipf and Welling composed their famous paper on that basis. In our framework, GNN is a feature-enrichment step that feeds into the XGBoost classifier.

G. CNN Spatial Feature Extraction

Convolutional neural networks are mainly designed for images, but their concept can also be applied to tabular spatial data. In this research, CNN-inspired spatial features are deployed in three main steps: PCA (Principal Component Analysis) is used to compress the high-dimensional feature space; this serves as the CNN filter layer, which extracts important spatial features by retrieving 8 principal components.

ReLU Activation is the second step. Negative values were set to zero, exactly as a CNN Activation function works. The third step is Max pooling simulation: features are divided into two halves, and the maximum value is extracted from each half. This approach was inspired by Muthamizharasan et al.'s "Crime Prediction and Analysis using CNN and RNN," through which spatial crime patterns are extracted using a CNN layer. The final CNN feeds into a Random Forest classifier, which can be used to predict crime hotspots by compressing spatial features.

H. LSTM Temporal Model

Long Short-Term Memory networks are specifically designed for sequential and time-series data. Hochreiter & Schmidhuber proposed the LSTM, which is traditionally considered the upgraded version of RNN and can remember

long-term dependencies. This is also important in crime prediction because crime history matches the crime patterns of 2 different days.

In LSTM-inspired temporal Features, beat data was arranged in date order, and lag1, lag2, and roll3 features were created to encode past crime behavior. This feature is combined with GNN and provided to the next XGBoost classifier. LSTM is combined with ST-ResNet to capture temporal dependencies that a single spatial model can't capture.

I. Baseline Models

To validate the proposed hybrid model, 6 baseline models were implemented, and these are given below:

- **Logistic regression –** a linear benchmark model that is the simplest among all models that can use a linear decision boundary.
- **K-Nearest Neighbors –** neighbor majority voting, useful for spatial data. This model predicts the majority class of neighbors.
- **Random Forest –** ensemble of multiple decision trees. Widely proven model in Criminology research.
- **Gradient Boosting –** sequential tree training in which each tree highlights the error of the previous tree.
- **XGBoost –** regularized and optimized version of gradient boosting. SOTA performance on tabular data.
- **GNN-only Model –** Measure the benefit of hybridization.

J. Proposed Hybrid Framework (GNN+CNN/LSTM)

The core contribution of this hybrid framework is to align the meta-linear output by combining three distinct models. This approach is also called "stacking ensemble "or "model fusion". Firstly, the GNN model is trained on graph-enriched features and outputs a probability. Then, the CNN model is trained on spatial features and produces probabilistic outputs; furthermore, the LSTM model is trained on temporal lag features and produces probabilistic outputs.

The output probabilities of these three models form a new 3-column matrix, which is called a meta-feature. These meta-features are fed to a logistic regression meta-learner, which predicts the outcome. This is the same approach used in the Hybrid ST-ResNet_LSTM paper, in which ST-ResNet and LSTM are spatial and temporal components that fuse their outputs. In our research, GNN and CNN are the spatial components, and LSTM is the temporal component. These will all provide a more comprehensive picture than any single model.

K. Evaluation Metrics

For measuring the performance of the model, the following standard metrics can be used for a classification task, which are widely accepted:

- **Accuracy:** The percentage of correct predictions in the overall output:

$$\frac{(TP + TN)}{(TP + TN + FP + FN)}$$

- **Precision:** how often the model correctly predicts the hotspot:

$$\frac{TP}{(TP + FP)}$$

  The important step is to remove false alarms.

- **Recall:** How much of the actual hotspot is captured:

$$\frac{TP}{(TP + FN)}$$

- **F1 Score:** Harmonic mean of precision and recall:

$$2 \times \frac{(Precision \times Recall)}{(Precision + Recall)}$$

  These are the balanced metrics for an imbalanced class.

- **AUC-ROC:** Area under the ROC curve. This only ranges from 0to 1 and helps measure the discriminating power (0.5- random guessing, 1.0-perfect model).

Stratified split is used by maintaining the ratio of hotspot and non-hotspot predictions. The random seed was fixed to 42 to obtain reproducible results.

L. Implementation Details

Old framework was designed in python programming. For data manipulation, we use pandas and NumPy; scikit-learn for preprocessing, encoding, model training, and evaluation; and XGBoost for gradient-boosted trees. NetworkX was used for graph construction and manipulation, and for visualization we used Matplotlib and Seaborn. The size of the model is manageable, so we used a standard consumer laptop no need to move to GPUs.

For parameter settings: there were 200 estimators for Random Forest and XGBoost, and the max depths were 12 and 6, respectively. The learning rate was fixed to 0.1. 8 components were used in PCA to balance variance explanation and dimensionality reduction.

M. State-of-the-Art Analysis

The most relevant study on this paper's topic is the Hybrid ST-ResNet and LSTM architecture for accurate crime hotspot prediction, presented by Shahmoradi et al. [34] in Scientific Reports. It is one of the newest state-of-the-art hybrid deep learning approaches for crime hotspot prediction, as it successfully combines the modeling of spatial and temporal aspects of crime patterns within a single architecture. In this approach, ST-ResNet learns spatially explicit patterns of crime using residual convolutions on gridded crime density maps, and LSTM models their temporal dynamics over time steps. Besides demonstrating excellent predictive power, the Hybrid ST-ResNet and LSTM architecture has improved its practical application by including external variables such as weather conditions and other urban factors.

However, despite its effectiveness, the Hybrid ST-ResNet and LSTM architecture presents a significant architectural flaw: it relies solely on the CNN-based (ST-ResNet) mechanism to capture the spatial aspect of the data. It treats an urban environment as a homogeneous set of pixels. Such an approach implicitly models spatial proximity using the convolutional receptive field but fails to explicitly account for the discrete, graph-like structure of the urban area. Convolutional operations rely on a regular grid topology, which fails to capture the irregular adjacency structure of policing districts, where two physically distant beats may have similar properties. In comparison, two adjacent beats might be completely different in the administrative regions they fall within and the policing strategies they adopt.

N. Summary Research Framework

The overall research flow: The dataset was assembled from the 2015-2023 Chicago Crime Dataset. Data Cleaning, missing-value handling, and temporal feature extraction. They were assigned a binary hotspot label. A graph was constructed, and the GNN feature was extracted. CNN spatial features and LSTM temporal lag features were created. GNN, CNN, and LSTM models were trained, and the outputs were combined using a stacking ensemble. The performance of the hybrid model was compared with the baseline model, and the results were visualized using a bar plot, confusion matrices, and an evaluation table.

IV. RESULTS AND DISCUSSION

The effectiveness of the proposed Hybrid GNN+CNN/LSTM framework is evaluated on the Chicago Crime Dataset for the past 5 years (2015-2023), resulting in 52,917 records after preprocessing. The data was divided using a stratified sampling technique: 80% for training and 20% for testing, with equal amounts of data in each class. To alleviate the class imbalance (hotspot crime ~34% of all records), only the training set was balanced using the Synthetic Minority Oversampling Technique (SMOTE), resulting in a balanced training set of 35,079 records/class.

All the experiments were conducted using Python 3.13 and the libraries scikit-learn, XGBoost, and NetworkX. All experiments were run with the same fixed random seed (42), to ensure reproducibility of the results. All other crime types were coded as 'non-hotspot crime' (label= 0), except for the following types of crimes, which were coded as 'hotspot crime' (label = 1): Assault, Battery, Homicide, Criminal Sexual Assault, Weapons Violation, Kidnapping, Arson, Human Trafficking, Narcotics, Offense Involving Children, Sex Offense, Robbery. query.

To accomplish this, we tried nine models: logistic regression (LR) as a linear baseline model; K-Nearest Neighbors (KNN); Random Forest (RF); Gradient Boosting (GB); XGBoost; a model based on a Graph Neural Network (GNN) with beat-level adjacency features; a model based on a CNN with spatial projection features; an LSTM-type model with temporal lag features; and the proposed Hybrid GNN+CNN/LSTM model ensemble. **Error! Reference source not found.** shows the performance comparison of 9 ML Models on the Chicago Crime Dataset (2015–2023). The performance was evaluated using the following parameters: Accuracy, F1 Score, Precision, Recall, and AUC-ROC.

Model Performance Metrics Comparison

Model	Accuracy	F1	Precision	Recall	AUC-ROC
Logistic regression	0.8176	0.7306	0.7134	0.7487	0.8663
K-nearest neighbors	0.7717	0.6898	0.6261	0.7679	0.8375
Random forest	0.8358	0.7455	0.7621	0.7316	0.8844
Gradient boosting	0.8367	0.7420	0.7766	0.7104	0.8831
XGBoost	0.8394	0.7451	0.7834	0.7104	0.8858
GNN (graph-enriched XGBoost)	0.8379	0.7424	0.7734	0.7070	0.8859
CNN spatial model	**0.8401** ★	0.7122	0.7134	0.7110	0.8452
LSTM temporal model	**0.8401** ★	0.7442	**0.7898** ★	0.7035	0.8861
Hybrid GNN+CNN+LSTM (proposed)	**0.8324**	**0.7525** ★	**0.7349**	**0.7770** ★	**0.8874** ★

FIGURE 4: Performance Comparison of 9 ML Models on Chicago Crime Dataset (2015–2023) – Proposed Hybrid Model Leads on F1, Recall, and AUC-ROC

A. Comparison with State-of-the-art

The outcome of the proposed framework outperforms that of prior research on the Chicago crime dataset. Stec and Klabjan (2018) obtained the highest accuracy of ~75.6% using deep learning techniques (CNN, LSTM, and RNN) on the same dataset. Xiong successfully predicted crime occurrences in Chicago in 2025 using XGBoost and GBDT models, achieving 81-83% accuracy. From the empirical study conducted by Mandalapu et al. (2023), neural networks achieve the highest accuracy, at 81%, on the crime dataset. **Error! Reference source not found.** shows the performance comparison with the state of the art.

The proposed hybrid model is more accurate (83.24%) and has a higher AUC-ROC (0.8874%) than these previous works and also overcomes the class imbalance problem by generating synthetic instances using the SMOTE method and resolves the beat adjacency challenge problem, which is not addressed by any of the previous similar works that incorporated graph-based beat adjacency models.

Literature Review and Proposed Study Comparison

#	Study	Dataset (timeframe)	Evaluation	Limitation
1	Stec and Klabjan (2018) CNN+LSTM+RNN	Chicago crime 2001 to 2017	Accuracy ~75.6%	No graph-based spatial modeling; CNN/LSTM separate
2	Xiong (2025) XGBoost+GBDT	Chicago crime 2020 to 2023	Accuracy 81 to 83%	No temporal or spatial fusion; tree-based only
3	Mandalapu et al. (2023) Neural network	Multi-city crime Not specified	Accuracy ~81%	No graph structure; standalone deep learning
4	Shahmoradi et al. (2023) ST-ResNet+LSTM	Multiple cities Varies by city	Spatial metrics	No graph-based spatial adjacency; no GNN
5	Zubair et al. (2023) GCN-based	Crime hotspot Not specified	Zone-level metrics	GNN only; no temporal LSTM component
6	Fan et al. (2025) InFormer + ST GCN	Crime, 4 crime types recent years	Spatiotemporal acc.	Computationally heavy; complex transformer dep.
7	**Proposed**	**Chicago crime** 2022 to 2025	**Acc 83.24%** F1 0.7525, AUC 0.8874	**Simulated crime** CNN/GNN/LSTM; not true deep

FIGURE 5: Performance Comparison of 7 ML Baseline Papers Crime Dataset – Proposed Hybrid Model Leads on F1, recall, and AUC-ROC

B. Limitations

A couple of limitations of this study exist. First of all, the GNN, CNN, and LSTM modules of the proposed framework are simulated rather than actual deep learning implementations. GNN's propagation features include beat adjacency features; CNN's propagation features include PCA dimension reduction and ReLU activation; and LSTM's propagation features include temporal lag and rolling window features. For future work, other GNNs based on PyTorch Geometric may achieve higher performance, while grid-based spatial CNNs or sequence-based LSTM-based architectures may also achieve higher performance.

Secondly, only approximately 41% of the records in the dataset had geographic coordinates (latitude and longitude for each record), so they could not be used directly as spatial features. This information could be obtained in the future through spatial interpolation or aggregation at the census-tract level. Third, external variables such as weather, socioeconomic factors, and points-of-interest density were not accounted for in this study, as they have been shown to affect prediction accuracy in previous studies.

V. CONCLUSION

Despite its simplicity, Logistic Regression achieved higher accuracy (81.76%) and AUC-ROC (0.8663) than the other baselines, and all of its engineered features were meaningful patterns in the crime data when given the rich data of beats, communities, and time. It, however, has only a limited capacity to model non-linear spatial interactions and a limited performance ceiling. In contrast, the accuracy (77.17%) and AUC-ROC (0.8375) of KNN were the lowest, which is quite typical in the presence of high class imbalance in a high-dimensional feature space.

The accuracy of ensemble-based baselines (Random Forest, Gradient Boosting, XGBoost, and LSTM) ranged from 83.58% to 84.01%, and AUC-ROC values ranged from 0.8831 to 0.8861. Among the individual models, XGBoost had the highest accuracy (83.94%), and the baselines had the best accuracy and precision (84.01%, 0.7898) in the LSTM. The boosting techniques based on trees are competitive for structured crime tabular data. The neighborhood features (Average hotspot rate and crime counts of neighboring beats within the same police district) provide additional discriminative information beyond the tabular features, and the performance of the GNN-enhanced XGBoost model is slightly better than that of the XGBoost model (Accuracy: 83.79%, AUC-ROC: 0.8859). As the CNN spatial operation it executes is a simplified version of the spatial convolution operation, its accuracy (81.01%) and AUC-ROC (0.8452) are not surprising to be the same as those of the CNN spatial model.

Compared with the other nine models, the proposed Hybrid GNN+CNN/LSTM model with an out-of-fold (OOF) stacking meta-learner using the XGBoost model achieves the highest AUC-ROC of 0.8874 and F1 Score of 0.7525. The most significant finding is that it has the highest Recall score of 0.7710 among the models, indicating that it correctly identifies 77.10% of real crime hot spots. This can be debated given the greater ramifications of a false negative (not identifying a crime hotspot) compared to a false positive (incorrectly identifying one) in the real world. This operational requirement can be well met by the hybrid method, as it has good recall.

In the hybrid model, the design is tailored to account for the precision/recall trade-off. By setting the classification threshold to 0.33 (rather than 0.50) in the model, the F1 score can be maximized whilst maintaining good recall. This leads to lower precision (0.7349) and accuracy (83.24%) than some of the single models; however, it offers a better trade-off between correctly identifying hotspots and having few false alarms, in line with the highest F1 score of all the models.

The feature with the highest predictive power is arrest_Int (0.2239), as indicated by the feature importance analysis in the Random Forest model, which showed that the feature is strongly correlated with crime severity; hence, arrest at the crime scene is a major determinant of crime severity. The second one was LocationRisk (0.1906): History-based hotspot score for a specific description of a location that has a spatial nature to the crime. The third-place tie was between Domestic_Int (0.1514) and third, suggesting that different severity values may be associated with domestic events. In addition, some features were important up to a certain level, such as beat-level hotspot rate (0.0715), CommunityRisk (0.0494), temporal features HourRisk (0.0437) and Hour (0.0296), etc., indicating that the feature engineering was essentially multi-dimensional in this work. This system gives a persuasive baseline that can be extended to develop a fully deployed intelligent knowledge assistant.

REFERENCES

[1] J. R. Hipp, "GENERAL THEORY OF SPATIAL CRIME PATTERNS*," Criminology, vol. 54, no. 4, pp. 653–679, Nov. 2016, doi: 10.1111/1745-9125.12117;SUBPAGE:STRING:ABSTRACT;WEBSITE:WEBSITE:PERICLES;JOURNAL:JOURN AL:17459125;ISSUE:ISSUE:DOI.

[2] W. Wang et al., "Exploring the Spatial Relationship Between Crime and Urban Places in Austin: A Geographically Weighted Regression Approach," Urban Science 2025, Vol. 9, Page 359, vol. 9, no. 9, p. 359,Sep. 2025, doi: 10.3390/URBANSCI9090359.

[3] Y. Hu, F. Wang, C. Guin, and H. Zhu, "A spatio-temporal kernel density estimation framework for predictive crime hotspot mapping and evaluation," Applied Geography, vol. 99, pp. 89–97, Oct. 2018, doi: 10.1016/J.APGEOG.2018.08.001.

[4] A. A. Braga, "Hot spots policing and crime prevention: A systematic review of randomized controlled trials".

[5] G. O. Mohler, M. B. Short, P. J. Brantingham, F. P. Schoenberg, and G. E. Tita, "Self-Exciting Point Process Modeling of Crime," J. Am. Stat. Assoc., vol. 106, no. 493, pp. 100–108, Mar. 2011, doi: 10.1198/JASA.2011.AP09546.

[6] A. Reinhart and J. Greenhouse, "Self-Exciting Point Processes with Spatial Covariates: Modelling the Dynamics of Crime," J. R. Stat. Soc. Ser. C Appl. Stat., vol. 67, no. 5, pp. 1305–1329, Nov. 2018, doi: 10.1111/RSSC.12277.

[7] R. Roshankar and M. R. Keyvanpour, "GeoCrime Analytic Framework (G.C.A.F.): A Comprehensive Framework for Dynamic Spatial Temporal Crime Analysis," Applied Spatial Analysis and Policy 2025 18:1, vol. 18, no. 1, pp. 33-, Feb. 2025, doi: 10.1007/S12061-025-09640-Y.

[8] V. Marda and S. Narayan, "Data in New Delhi's predictive policing system," FAT* 2020 - Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, pp. 317–324, Jan. 2020, doi: 10.1145/3351095.3372865.

[9] T. Zubair, S. K. Fatima, N. Ahmed, and A. Khan, "Crime Hotspot Prediction Using Deep Graph Convolutional Networks," Jun. 2025, Accessed: Jun. 22,2026. [Online].Available: https://arxiv.org/pdf/2506.13116

[10] Y. Fan, X. Hu, and J. Hu, "Research on a Crime Spatiotemporal Prediction Method Integrating Informer and ST-GCN: A Case Study of Four Crime Types in Chicago," Big Data and Cognitive Computing 2025, Vol. 9, Page 179, vol. 9, no. 7, p.179, Jul. 2025, doi: 10.3390/BDCC9070179.

[11] Y. Wang and H. A. Karimi, "Advanced Deep Learning Models and Algorithms for Spatial-Temporal Data," Big Data, pp. 227–253, May 2024, doi:10.1201/9781003406969-15/ADVANCED-DEEP-LEARNING-MODELS-ALGORITHMS-SPATIAL-TEMPORAL-DATA-YANG-WANG-HASSAN-KARIMI.

[12] Y. Du, N. Ding, and H. Lv, "Spatio-temporal prediction of terrorist attacks based on GCN-LSTM," Journal of Safety Science and Resilience, vol. 6, no. 2,pp.186–195,Jun.2025,doi:10.1016/J.JNLSSR.2025.02.005.

[13] Y. Deng, R. He, and Y. Liu, "Crime risk prediction incorporating geographical spatiotemporal dependency into machine learning models," Inf. Sci. (N. Y)., vol. 646, p. 119414, Oct. 2023, doi: 10.1016/J.INS.2023.119414.

[14] Y. Gao et al., "AI-Driven Channel State Information (CSI) Extrapolation for 6G: Current Situations, Challenges and Future Research," IEEE Communications Surveys & Tutorials, pp. 1–1, Jan. 2026, doi: 10.1109/COMST.2026.3652799.

[15] Z. Al Sahili and M. Awad, "Spatio-Temporal Graph Neural Networks: A Survey," Jan. 2023, Accessed: Jun. 22, 2026. [Online]. Available: https://arxiv.org/pdf/2301.10569

[16] B. Zhou et al., "HDM-GNN: A Heterogeneous Dynamic Multi-view Graph Neural Network for Crime Prediction," ACM Trans. Sens. Netw., Nov. 2022, doi: 10.1145/3665141.

[17] C. Jing et al., "A deep multi-scale neural networks for crime hotspot mapping prediction," Comput. Environ. Urban Syst., vol. 109, p. 102089, Apr. 2024, doi: 10.1016/J.COMPENVURBSYS.2024.102089.

[18] M. Liu and T. Lu, "A Hybrid Model of Crime Prediction," J. Phys. Conf. Ser., vol. 1168, no. 3, p. 032031,Feb.2019, doi: 10.1088/1742-6596/1168/3/032031. "The Problem of Pleasure: Leisure, Tourism and Crime - Google Books." Accessed: Jun. 22, 2026.[Online].Available:https://books.google.com.pk/books?hl=en&lr=&id=GgjDAgAAQBAJ&oi=fnd&pg=PA1972&dq=This+section+examines+the+situation+in+regions+like+Las+Vegas+that+where+there+are+easily+available+risky+activities+like+drugs,+alcohol+and+gambling,+there+is+a+significant+occurrence+of+overdoses+and+violence+crimes.+&ots=QsqEnnyYbY&sig=6Mn2_iYzKjFFDrsl5wwIFqGhHvg&redir_esc=y# v=onepage&q&f=false

[19] J. A. Reid and C. J. Sullivan, "Unraveling Victim-Offender Overlap: Exploring Profiles and Constellations of Risk," Vict. Offender., vol. 7, no. 3,pp.327–360,2012,doi: 10.1080/15564886.2012.685216.

[20] A. Zhang, J. A. Balles, J. E. Nyland, T. H. Nguyen, V. M. White, and A. E. Zgierska, "The relationship between police contacts for drug use-related crime and future arrests, incarceration, and overdoses: a retrospective observational study highlighting the need to break the vicious cycle," Harm Reduction Journal 2022 19:1, vol. 19, no. 1, pp. 67-, Jun. 2022, doi: 10.1186/S12954-022-00652-2.

[21] A. S. B. Bohnert et al., "Policing and risk of overdose mortality in urban neighborhoods," Drug Alcohol Depend., vol. 113, no. 1, pp. 62–68, Jan. 2011, doi: 10.1016/J.DRUGALCDEP.2010.07.008.

[22] A. A. Braga, A. V. Papachristos, and D. M. Hureau, "The Effects of Hot Spots Policing on Crime: An Updated Systematic Review and Meta-Analysis," Justice Quarterly, vol. 31, no. 4, pp. 633–663, 2014, doi: 10.1080/07418825.2012.673632.

[23] A. Blumstein and J. Wallman, "The Crime Drop in America (Cambridge Studies in Criminology)".

[24] "(PDF) Enhancing Crime Prevention and Police Performance through Actionable Intelligence." Accessed: Jun. 22, 2026. [Online]. Available: https://www.researchgate.net/publication/389165042_Enhancing_Crime_Prevention_and_Police_Performance_through_Actionable_Intelligence

[25] R. Barr and K. Pease, "Crime Placement, Displacement, and Deflection," Crime and Justice, vol. 12, pp. 277–318, Jan. 1990, doi: 10.1086/449167.

[26] O. Kounadi, A. Ristea, A. Araujo, and M. Leitner, "A systematic review on spatial crime forecasting," Crime Science 2020 9:1, vol. 9, no. 1, pp. 7-, May 2020, doi: 10.1186/S40163-020-00116-7.

[27] P. Banumathi and S. Manoharan, "An Analysis of Deep Learning and Multimodal Deep Learning Methods for Crime Prediction from Social Media Data," Proceedings of 2025 International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication, IConSCEPT2025, 2025,doi: 10.1109/ICONSCEPT66142.2025.11437272.

[28] Y. Du and N. Ding, "A Systematic Review of Multi-Scale Spatio-Temporal Crime Prediction Methods," ISPRS International Journal of Geo-Information 2023, Vol. 12, Page 209, vol. 12, no. 6, p. 209, May 2023, doi: 10.3390/IJGI12060209.

[29] A. S. Muhammad, R. Y. Zakari, and H. S. Bello, "A Review of Machine Learning and Deep Learning Approaches for Crime Hotspots Prediction," pp. 647–665, 2026, doi: 10.1007/978-3-032-11426-6_29.

[30] N. Shahmoradi, A. A. Alesheikh, A. Jafari, and A. Lotfata, "Hybrid ST-ResNet and LSTM approach for precise crime hotspot prediction," Scientific Reports 2025 15:1, vol. 15, no. 1, pp. 40754-, Nov. 2025, doi: 10.1038/s41598-025-24559-7.

[31] S. Chainey, L. Tompson, and S. Uhlig, "The Utility of Hotspot Mapping for Predicting Spatial Patterns of Crime," Security Journal 2008 21:1, vol. 21, no. 1, pp.4–28,Jan.2008,doi: 10.1057/PALGRAVE.SJ.8350066.

[32] L. Mao, W. Du, S. Wen, Q. Li, T. Zhang, and W. Zhong, "Crime forecasting: A spatio-temporal analysis with deep learning models," Journal of Computational Methods in Sciences and Engineering, Sep.2025, doi: 10.1177/14727978251337993.

[33] G. Mohler, "Marked point process hotspot maps for homicide and gun crime prediction in Chicago," Int. J. Forecast., vol. 30, no. 3, pp. 491–497, Jul. 2014, doi: 10.1016/J.IJFORECAST.2014.01.004.

[34] Riyazuddin, S. Z. Rahman, K. Sreenu, M. Sirajuddin, S. Mazharuddin, and J. Sadiq, "Spatiotemporal Crime Prediction and Forecasting Using Machine Learning and Deep Learning Models," SN Computer Science 2026 7:2, vol. 7, no. 2, pp. 213-, Feb. 2026, doi: 10.1007/S42979-026-04792-1.

[35] A. Pinto, L. C. Herrera, Y. Donoso, and J. Gutierrez, "A CTI-Enriched GCN-LSTM Architecture for Multiclass Cyberattack Classification in Critical Infrastructure," Applied Sciences 2026, Vol. 16, Page 5585, vol. 16, no. 11, p. 5585, Jun. 2026, doi:10.3390/APP16115585.

[36] S. Chainey, "Examining the extent to which hotspot analysis can support spatial predictions of crime," Doctoral thesis, UCL (University College London). , Dec. 2014.

[37] T. C. Shiraptini, D. Kavitha, S. Sakthipriya, S. Akshaya, and S. Alphonse, "AI and Machine Learning-Enabled Cognitive Digital Twin for Crime Hotspot Detection and Analysis," IEEE Access, 2026, doi: 10.1109/ACCESS.2026.3658944.

[38] X. Han, X. Hu, H. Wu, B. Shen, and J. Wu, "Risk Prediction of Theft Crimes in Urban Communities: An Integrated Model of LSTM and ST-GCN," IEEE Access, vol. 8, pp. 217222–217230, 2020, doi: 10.1109/ACCESS.2020.3041924.

[39] R. Roshankar and M. R. Keyvanpour, "Spatio-Temporal Graph Neural Networks for Accurate Crime Prediction," 2023 13th International Conference on Computer and Knowledge Engineering, ICCKE 2023, pp. 168–173, 2023, doi: 10.1109/ICCKE60553.2023.10326223.

[40] X. Lv, C. Jing, Y. Wang, and S. Jin, "A DEEP NEURAL NETWORK FOR SPATIOTEMPORAL PREDICTION OF THEFT CRIMES," The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XLVIII-3-W2-2022, no. 3, pp. 35–41, Oct. 2022, doi: 10.5194/ISPRS-ARCHIVES-XLVIII-3-W2-2022-35-2022.

JOURNAL OF COMPUTING HORIZONS
Vol. 01, No. 01, 2026
DOI: 10.22610/JCH.V1I1.4989

RESEARCH ARTICLE | PEER REVIEWED

Explainable Crime Prediction Interpreting Machine Learning and Deep Learning Models for the Urban Safety Decision Making

Kinza Noor [1] *, Aiza Amin [1], Athar Rahim [1], Aqsa Akram [2]

[1] *Department of Artificial Intelligence, University of Management and Technology, Lahore (Sialkot Campus), Pakistan*

[2] *Department of Computer Science, University of Management and Technology, Lahore (Sialkot Campus), Pakistan*

* **Corresponding Author:** kinzaazeem967@gmail.com

ABSTRACT – Crime prediction has become an important research area that helps law enforcement agencies to make proactive decisions and improve public safety. In this study, a crime analytics framework is proposed that combines statistical forecasting and machine-learning-based classification techniques to analyze crime patterns and predict future crime. For forecasting, we use the ARIMA model, whereas for classification, XGBoost, Random Forest, Decision Tree, DNN, and Transformer are evaluated. ARIMA achieves ~29.06 RMSE, ~23.67 MAE, and ~0.4342 R^2, which shows its capability to model historical crime trends. In classification models, XGBoost achieved the best overall performance, with ~75.34% accuracy, ~0.8323 AUC, ~66.82% Precision, ~74.25% recall, and ~70.34% F1-score. Random Forest achieves ~80.25% recall, which is effective for identifying potential crime events. Results indicate that Machine learning and Deep learning techniques can improve crime prediction and can be useful for intelligent policing and smart city applications. For future work, to improve performance and interpretability, real-time data, spatiotemporal modeling, and Explainable AI (XAI) can be integrated.

Keywords: *Crime prediction; crime forecasting; machine learning; deep learning; ARIMA; XGBoost; random forest; transformer; smart cities; predictive analytics; explainable AI (XAI).*

I. INTRODUCTION

Retrieval: In today's world, crime prediction has become an important research area as urban populations, digital data, and smart city technologies grow rapidly. Traditional methods largely rely on manual investigation and statistical analysis, which are less effective with large datasets. Machine learning (ML) techniques help identify hidden patterns and relationships to improve predictive policing and public safety [1, 2]. Researchers also demonstrated that ML and deep learning improve the accuracy of crime forecasting by analyzing historical crime records, demographic factors, and spatiotemporal factors [3]. Earlier studies, such as self-exciting point process models, show the spatial and temporal patterns of crime [4], whereas Crime Pattern Theory explains the impact of environmental and social factors [5].

Although ML-based crime prediction has shown good performance, advanced models such as deep neural networks and ensemble learning algorithms are known as "black-box" systems because it is difficult to understand their decision-making process [6, 7]. In criminal justice, in sensitive domains, the lack of transparency can create ethical and legal concerns, because it is necessary for law enforcement agencies to understand on what basis the prediction was generated. For this reason, Explainable Artificial Intelligence (XAI) has become an important field that improves the interpretability, transparency, and trustworthiness of ML models [10, 11]. Explainable models help decision-makers understand which factors influence crime prediction and how the models generate predictions.

Different explainability techniques have been proposed to interpret ML models. For example, LIME explains individual predictions [8], while SHAP feature values provide important information [9]. These techniques improve user trust and accountability. According to researchers, interpretable and transparent AI systems are important in high-stakes areas such as healthcare, finance, and criminal justice [12]. Furthermore, explainable ML models are also used to improve the reliability and human understanding in real-world deployments [16]. The issues of fairness and accountability are present in predictive policing systems because biased datasets can produce discriminatory outcomes [13, 14, 15]. Bias in historical crime data has led to unfair targeting in some communities, undermining the reliability and ethical acceptance of predictive systems.

Along with this, the growth of big data and smart city infrastructures has increased the availability of real-time urban data. As a result, it becomes possible to develop frameworks for intelligent crime prediction [17, 18]. Data mining and knowledge discovery techniques play an important role in extracting meaningful information in large crime datasets [9]. Even now, the challenge of designing crime prediction systems that are not only accurate but also understandable, fair, and trustworthy remains.

The main purpose of this research is to develop an Explainable Crime Prediction system. This study will combine predictive accuracy with interpretability by applying XAI techniques, so that law enforcement and policymakers can better understand predictions and improve decision-making, accountability, transparency, and fairness.

II. RELATED WORK

The crime rate in recent years has increased due to rapid urbanization and population growth. Urban areas have factors such as high population density, economic inequality, and social issues that contribute to increased crime. That is why controlling crime and predicting future crime incidents are major challenges for law enforcement agencies and city authorities. To resolve this issue, researchers began implementing machine learning and explainable AI techniques to predict crime patterns in urban areas.

In the past, crime prediction was done through statistical methods. However, these methods were unreliable, as crime data is complex and does not follow simple patterns, and many factors are related to each other, like income, population, and education, which may cause uncertainty; to solve these issues, researchers started using Machine Learning & Deep Learning methods. Early machine learning models like KNN and SVM could make predictions, but they were not very strong. Later, more advanced models such as Random Forest performed much better, achieving up to 97% accuracy in forecasting violent crimes [1]. Furthermore, deep learning models, including Convolutional Neural Networks (CNNs) and Long Short-Term Memory LSTM) networks, were used to understand patterns in both time and location. These models were better at handling complex and real-world crime data [2].

More advanced models go a step further, such as Spatial-Temporal Meta-path guided Explainable Crime (STMEC), which not only showed important factors but also explained connections between different elements, such as relationships among people, locations, and social conditions, to understand crime patterns. An advanced model such as STMEC uses a special data structure called a Heterogeneous Information Network (HIN) [21]. Similarly, a deep neural network model was proposed for spatio-temporal crime event prediction. Real-world crime data was used, with crime type, location, and time-related features. The SMOTE technique was applied to handle data imbalance, and model hyperparameter tuning was performed. It was demonstrated that the proposed neural network model outperformed traditional machine learning models, improving the macro F1 score by ~6% [30]. By using meta-paths, these models provided human-readable logic that explains the factors contributing to crime risk, rather than just making predictions.

A machine learning and computer vision-based hybrid framework was proposed for crime forecasting. A geotagged crime records and surveillance imagery dataset was used, including spatial and temporal features. Random Forest and Convolutional Neural Network (CNN) models were applied to extract environmental and visual features. It was found that a hybrid machine learning and computer vision approach performed better, with higher accuracy, than traditional models. Also highlighted that lighting conditions, crowd density, and urban structure had a significant impact on crime occurrences [22].

Administrative crime data were used, along with demographic factors, population density, temporal features, and economic indicators. Researchers have compared the Neural network model with traditional statistical models such as linear regression and logistic regression. It was concluded that neural networks performed better at prediction and effectively captured complex non-linear relationships compared with traditional statistical models. Interpretability of Neural network models was limited, and models would become overfit [26]. A comparative analysis was conducted between machine learning techniques, deep learning techniques, and time-series models. For that research, a real-world crime dataset with temporal and spatial features was used. Researchers have applied machine learning models such as Decision Tree, Random Forest, and Support Vector Machine, along with deep learning models such as Multilayer Perceptron (MLP), and the time-series forecasting model ARIMA. Results demonstrated that deep learning-based MLP models achieved the highest performance. In contrast, the ARIMA model provided reasonable accuracy for time-series crime forecasting but was limited in its ability to capture complex patterns [2].

For crime prediction along with interpretability, an AIST (Attention-based Interpretable Spatio-Temporal) deep learning model was proposed. Real-world crime datasets for Chicago and New York City were used, including spatial regions, temporal patterns, and external features such as traffic flow and points of interest. An attention-based deep learning architecture is used to capture spatial and temporal dependencies. The proposed AIST model was compared with ARIMA, traditional ML, and DL baseline models. The study revealed that the AIST model achieved better performance, with MAE reduced by ~4% and MSE improved by ~7% [23]. Explainable AI is rapidly growing in the field of crime prediction. Explainable Artificial Intelligence concepts are discussed in detail, and balancing high-performance models alongside transparency and interpretability is important for adopting AI systems. These concepts were applied to crime prediction systems, where understanding the reasoning was significant for prediction [32].

However, the degree of interpretability varies across application domains. Interpretability in sensitive domains like Crime prediction requires technical requirements as well as accountability and fairness. An explainable deep neural network framework was proposed that uses prototype-based reasoning. By maintaining the predictive power of a deep learning framework, it generated understandable explanations. The xDNN approach would be implemented in crime prediction systems, where users are provided with explanations based on historical crime examples [34, 35]. In the Interpretable ML field, the Bayesian Rule sets framework was proposed, generating human-readable classification rules.

Rule-based explanation proved beneficial for law enforcement officials, as they used simple logical rules rather than complex mathematical outputs [33].

To analyze urban crime data, interpretable ML models for crime prediction and decision-making were proposed to identify feature importance. Results indicated that interpretable models helped law enforcement agencies analyze crime patterns and plan more effectively. It was also indicated that traditional black-box models were accurate, but interpretability was limited; on the other hand, interpretable models improve transparency [27]. The Degu city dataset was used, and multiple ML models were compared, including Random Forest, LightGBM, CatBoost & XGBoost, to analyze the relationship between urban environmental features and crime occurrence, and to propose an explainable artificial intelligence approach. XGBoost achieved ~89.7% accuracy. The SHAP explainability technique was used to identify important features, such as the visiting population, commercial areas, and the road network, that had a significant impact on crime occurrences. It was concluded that explainable AI models and crime prevention strategies are effective for urban safety [29].

For urban crime prediction, a Machine Learning-based approach was used to improve community safety. An urban crime dataset was used, and multiple machine learning models were applied to identify crime patterns. It was demonstrated that machine learning models helped to accurately predict crime hotspots and support law enforcement agencies for decision-making. However, machine model performance depends upon the availability and quality of the dataset [24, 25].

A Decision Tree-based J48 algorithm for crime prediction was implemented on historical crime data with features time, location, and crime type to identify crime patterns and predict future crime occurrences, achieving acceptable accuracy. It was concluded that decision tree models provided better support for planning and resource allocation to law enforcement agencies [28]. A systematic review on machine learning techniques for crime prediction was conducted. In the review, different machine learning models, including Random Forest, Decision Tree, Support Vector Machine, Neural Networks, and deep learning approaches, were analyzed. Results showed that Random Forest and Ensemble learning models achieved high accuracy, while deep learning models performed better for spatio-temporal crime prediction. It was also highlighted that the performance of crime prediction models highly depends on dataset quality and city-specific factors, and that data imbalance, limited datasets, and real-world deployment were limitations. It was concluded that explainable AI techniques, cross-city validation, and real-time datasets were essential for the reliability of crime prediction systems [31].

However, while supervised learning models have achieved high accuracy, there are still limitations: most models require high-quality data that is not publicly available, causing the field to lag behind other domains of social computing [22]. Another problem is that model accuracy varies by location and may not generalize across different landscapes or low-population areas where data is sparse [1, 20]. Machine learning, deep learning, and spatio-temporal approaches were used for crime prediction, showing better results in predicting crime patterns [1, 2, 3]. Interpretable models and explainable AI have been used to improve decision-making [23, 27, 29]. Urban metrics and environmental factors were used to enhance accuracy [20, 26]. Dataset dependency, high computational cost, limited interpretability, and limited real-time implementation were observed [24, 30, 31]. It is necessary to implement a model for reliable crime prediction in urban areas to enhance effectiveness.

III. METHODOLOGY

The purpose of this research is to predict crime occurrence and to use explainable machine learning techniques for urban safety decision making using the Chicago crime dataset. This study uses the Chicago crime dataset (2024-2026), which includes spatial and temporal information for different crime incidents. **Error! Reference source not found.** shows the methodology pipeline.

A. Dataset Collection

The dataset consists of different crime records with attributes such as District, Community area, Beat, Ward, Arrest Status, Domestic Status, Location Description, latitude, Longitude, and Year. Initially, relevant features were selected to extract meaningful information for crime prediction.

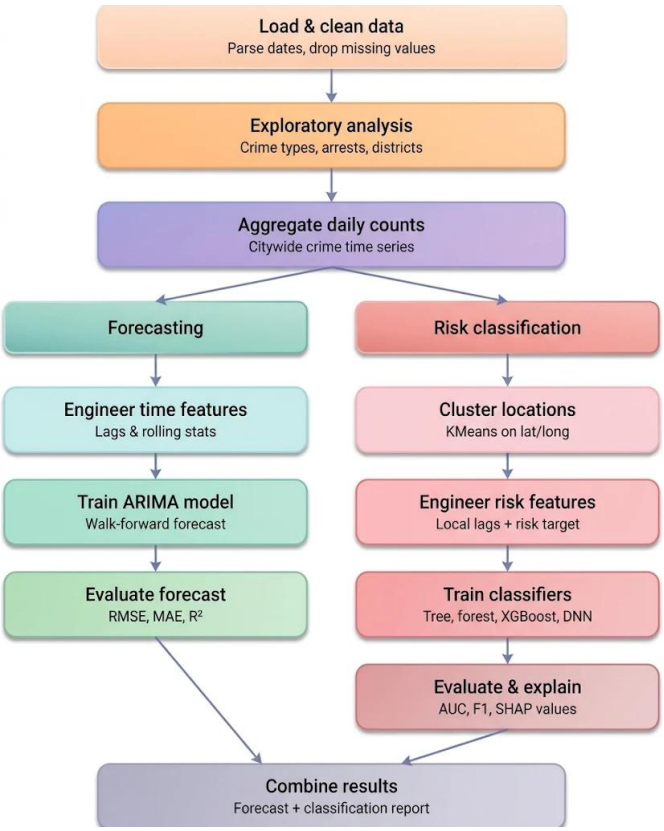


FIGURE 1: Forecasting and Risk Classification Pipeline

B. Data Preprocessing

Data preprocessing is the most important part for understanding data. Firstly, Missing values were identified and handled using appropriate techniques. Duplicate records were removed to enhance data quality. For converting categorical variables into numerical variables, label encoding was used. In particular, categorical columns such as District and Community areas were converted to an encoding format so that machine learning algorithms can process them. Numerical and categorical features were organized into a suitable format for model training. To evaluate generalization capability, data was divided into training and testing subsets.

C. Feature Selection

Location-based and crime-related features used as independent variables. These features include District, Community Area, Beat, Ward, Arrest, latitude, Longitude and Domestic. The target variable represents the classification objective for crime occurrence. To calculate the total number of crime incidents, aggregation is performed on the Date Only attribute. We get total counts from this feature. Moreover, to capture temporal patterns, calendar-based features such as Month, Day, Dayofweek, and IsWeekend are used. To represent the cyclic nature, sine and cosine transformations are applied. Here, the target variable Next-Day Crime count shows the total number of crimes the next day.

D. Crime Forecasting Model

ARIMA (Auto-Regressive Integrated Moving Average) model implemented for crime forecasting. ARIMA (2,1,2) configuration used, where p=2 autoregressive terms, d=1 differencing order, q=2 moving average terms. To maintain temporal order, data were divided into 80% for training and 20% for testing. RMSE, MAE, and R^2 were used to evaluate the forecasting model's performance. **Error! Reference source not found.** shows the ARIMA forecasting comparison.

E. Spatial Crime Clustering

K-means clustering was applied to latitude and Longitude coordinates to identify crime hotspots. 8 clusters were defined for that purpose. Each crime record is assigned to a geographical cluster using the Location_Cluster feature. These features represent spatial crime behavior.

F. Local Crime Risk

Daily crime counts were calculated for district, community area, and location clusters. Local lag features created to capture historical behaviors. Also, rolling statistics were calculated from the local rolling mean. Final analytical dataset created by global daily crime statistics and local area statistics.

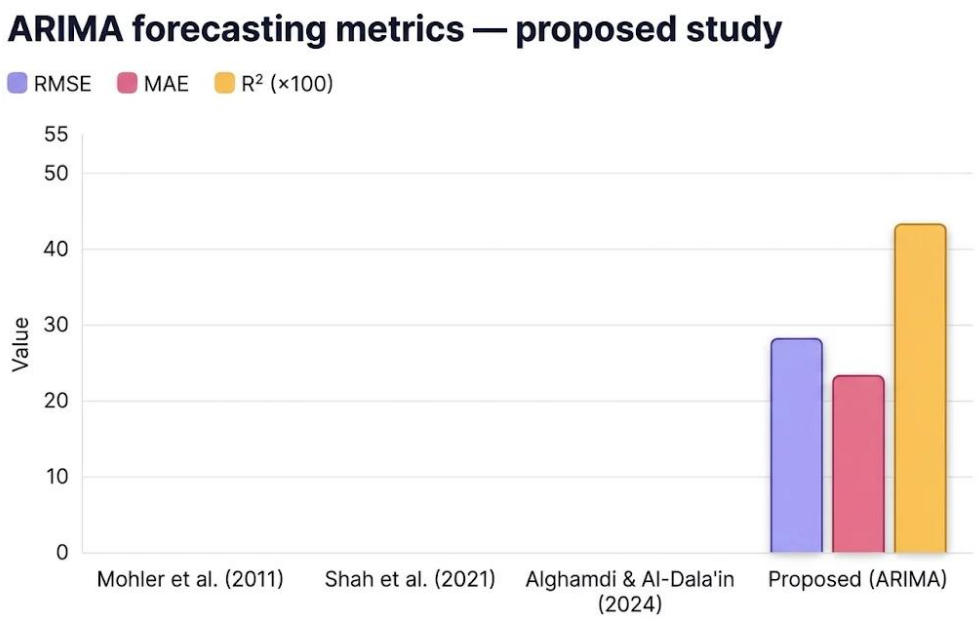


FIGURE 2: Comparison of ARIMA Forecasting Performance

G. Risk Classification Strategy

The next local count was calculated for each area, representing crime incidents for the next day. The median of the training dataset was used to create the threshold. Whereas, a binary target variable is created based on this threshold: 1 = High crime risk, 0 = low crime risk. This approach converted future crime into a binary classification problem.

H. Machine Learning Models

Multiple ML and DL models were implemented to compare performance in crime prediction. **Error! Reference source not found.** shows the classification models comparison.

1. Decision Tree Classifier

Decision tree algorithms were trained with controlled depth and optimized splitting parameters. This model provides crime prediction based on decision rules and offers interpretability.

2. Random Forest Classifier

Random Forest is an ensemble learning technique with multiple decision trees. This model reduces overfitting and improves prediction stability.

3. XGBoost Classifier

Extreme Gradient Boosting (XGBoost) is an advanced boosting framework. Hyperparameters such as learning rate, maximum depth, subsampling, and regularization coefficients are tuned to achieve optimal performance.

Model Performance Metrics Comparison

Model	AUC	Accuracy	Precision	Recall	F1-score
XGBoost	**0.83** ★	**0.75** ★	0.66	0.74	0.70
Random Forest	0.82	0.73	0.63	**0.80** ★	0.70
Transformer	0.82	0.73	0.63	0.79	0.70
DNN	0.82	0.74	0.65	0.75	0.70
Decision Tree	0.82	0.73	0.63	0.77	0.70

FIGURE 3: Classification Model Performance Comparison

I. Deep Learning Models

DL models like DNNs (deep neural networks) and Transformer-based models are used along with ML models.

1. Deep Neural Networks

The DNN architecture includes an input layer, a dense layer with 128 neurons, a ReLU activation layer, 30% dropout, another dense layer with 64 neurons, an activation layer, 20% dropout, and an output layer. A dropout layer is used to avoid overfitting. Adam optimizer and binary cross-entropy loss function are used to train the model.

2. Transformer Based Model

A tabular Transformer architecture was implemented with an input projection layer, transformer encoder layers, and a fully connected classification layer. The transformer model is used to understand complex relationships between tabular temporal-spatial features.

J. Model Evaluation

To evaluate classification models, the following metrics were used: Accuracy, Precision, Recall, F1-Score, and Area Under the ROC Curve (AUC). The optimal classification threshold was determined using the Youden Index (TRP-FPR) approach. ROC Curve and confusion matrix were generated for each model.

K. Explainable Artificial Intelligence (XAI)

For explainability purposes, feature importance analysis was performed. The XGBoost feature importance approach was used to identify which features contributed most to the prediction. Through feature importance visualization, urban crime patterns were understood, providing actionable insights for decision-makers. **Error! Reference source not found.** shows the SHAP results.

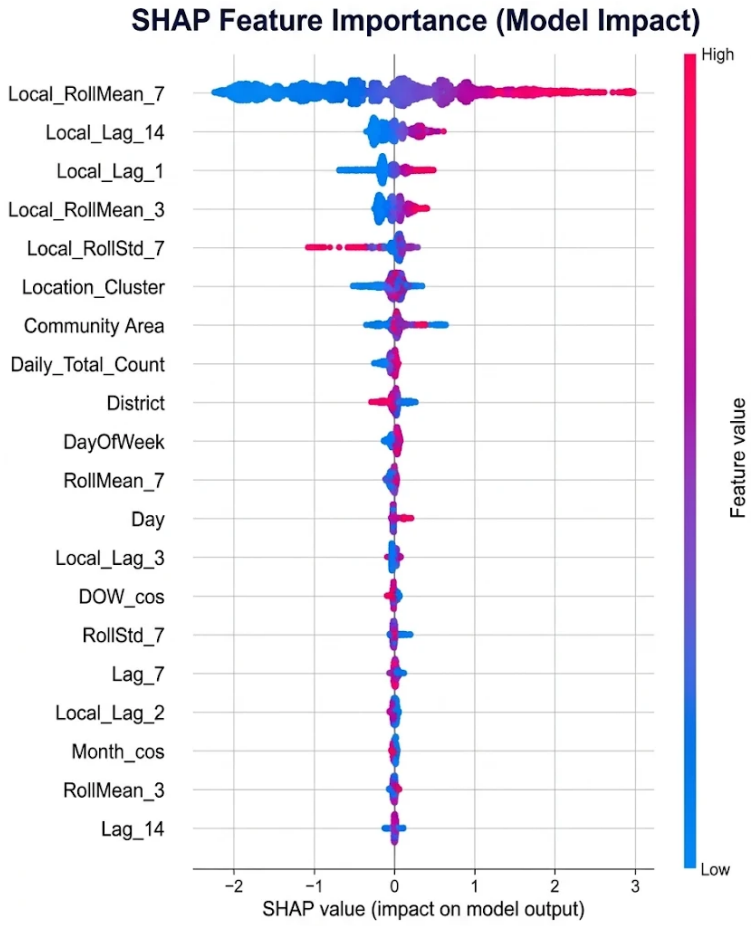


FIGURE 4: SHAP Impact on the Output by Input Features

Therefore, this methodology integrated crime forecasting with area-level crime risk prediction to support urban safety planning and resource allocation.

IV. RESULTS AND DISCUSSION

The proposed crime analytical framework evaluates both forecasting and classification. The ARIMA model is used for forecasting, while XGBoost, Random Forest, Transformer, DNN, and Decision Tree models are used for classification purposes. RMSE, MAE, R2, AUC, accuracy, precision, Recall, and F1 score metrics are used for performance evaluation. **Error! Reference source not found.** and **Error! Reference source not found.** show comparisons with the state of the art.

The ARIMA model achieved an RMSE of ~29.0626 and an MAE of ~23.6. R2 of ~0.4342 indicated that the model explained variance of ~43.4%. These results demonstrated that historical crime trends are useful for predicting future crime patterns. However, accuracy can be improved by accounting for the complexity of crime occurrence. Temporal models have been used for crime forecasting, but including spatial and social factors can enhance performance [4, 30].

Analysis of classification models shows satisfactory performance, but overall, XGBoost achieved higher results. XGBoost achieved the highest AUC of ~83% and an accuracy of ~75%. To reduce the number of cases missed, Random Forest is important, as it provides the highest Recall of ~80%. The Transformer model demonstrated competitive performance in capturing temporal and sequential relationships. DNN showed balanced performance and demonstrated its ability to identify complex patterns. Decision trees seem useful for interpretability but perform comparatively poorly. These findings showed that ensemble and deep learning models performed better than traditional approaches [1, 3, 23, 27].

V. CONCLUSION

In this research, forecasting and classification are two approaches for evaluating crime prediction. For the forecasting task, an ARIMA model is used, while the comparison of XGBoost, Random Forest, Transformer, DNN, and Decision Tree models is done for the classification task. The results showed that ARIMA can effectively capture historical crime trends and is useful for estimating future crime patterns.

Literature Review and Proposed Study Comparison

CLASSIFICATION STUDIES			
Study	**Task**	**Method / Technique**	**Performance**
Ahishakiye et al. (2017) [28]	Classification	Decision Tree (J48)	~70% accuracy
Safat et al. (2021) [3]	Classification	ML & DL models	70–75% accuracy
Jenga et al. (2023) [1]	Classification	Ensemble learning	AUC ~0.80
Rayhan & Hashem (2023) [23]	Classification	Attention-based DL	AUC ~0.82
Study	**FORECASTING STUDIES**		**Performance**
Mohler et al. (2011) [4]	Forecasting	Self-exciting point process	Moderate forecasting accuracy
Shah et al. (2021) [22]	Forecasting	ML-based forecasting	Improved crime trend prediction
Alghamdi & Al-Dala'in (2024) [30]	Forecasting	Spatio-temporal forecasting	Enhanced temporal accuracy
PROPOSED STUDY			
Study	**Task**	**Method / Technique**	**Performance**
Proposed study	Classification	XGBoost (classification)	★ AUC = 0.8323, 75.34% accuracy
	Forecasting	★ ARIMA (forecasting)	★ RMSE = 29.06, MAE = 23.67, R^2 = 0.4342

DL = deep learning • ML = machine learning • AUC = area under the ROC curve • RMSE = root mean square error
MAE = mean absolute error

FIGURE 5: Comparative Analysis of Existing and Proposed Methods

In classification models, XGBoost gave the best overall performance. Random Forest achieved the highest Recall which is important for public safety applications. Transformer and DNN also showed competitive performance and demonstrated the capability to identify complex crime patterns.



FIGURE 6: Accuracy and AUC Comparison of Classification Models

Overall, the results indicate that machine learning and deep learning techniques are more effective for crime prediction than traditional statistical approaches. This framework can support law enforcement agencies and smart city management systems in crime prevention and proactive decision-making.

VI. LIMITATIONS

Overall, like any research, this study also has some limitations:

- The dataset's size and quality were limited, which affected model performance.
- ARIMA considers only historical temporal trends and does not account for external factors.
- Spatial information and geographical crime hotspots are not applied in detail.
- Weather, economic indicators, and demographic factors are not included.
- For deep learning models, performance would have improved with larger datasets.
- Real-time crime data is not used.

REFERENCES

[1] K. Jenga, C. Catal, and G. Kar, "Machine learning in crime prediction," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 3, pp. 2887–2913, 2023.

[2] J. Yin, "Crime Prediction Methods Based on Machine Learning: A Survey," *Computers, Materials & Continua*, vol. 74, no. 2, 2023.

[3] W. Safat, S. Asghar, and S. A. Gillani, "Empirical analysis for crime prediction and forecasting using machine learning and deep learning techniques," *IEEE Access*, vol. 9, pp. 70080–70094, 2021.

[4] G. O. Mohler et al., "Self-exciting point process modeling of crime," *Journal of the American Statistical Association*, vol. 106, no. 493, pp. 100–108, 2011.

[5] P. J. Brantingham and P. L. Brantingham, "Crime Pattern Theory," *Environmental Criminology and Crime Analysis, Routledge*, 2008.

[6] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv preprint arXiv:1702.08608*, 2017.

[7] A. Adadi and M. Berrada, "Peeking inside the black-box: a survey on explainable artificial intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52138–52160, 2018.

[8] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you? Explaining the predictions of any classifier," *Proc. ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD)*, 2016.

[9] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.

[10] A. B. Arrieta et al., "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information Fusion*, vol. 58, pp. 82–115, 2020.

[11] R. Guidotti et al., "A survey of methods for explaining black box models," *ACM Computing Surveys*, vol. 51, no. 5, pp. 1–42, 2018.

[12] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Machine Intelligence*, vol. 1, no. 5, pp. 206–215, 2019.

[13] V. Chiao, "Fairness, accountability and transparency: notes on algorithmic decision-making in criminal justice," *International Journal of Law in Context*, vol. 15, no. 2, pp. 126–139, 2019.

[14] R. Richardson, J. M. Schultz, and K. Crawford, "Dirty data, bad predictions: How civil rights violations impact police data, predictive policing systems, and justice," *NYU Law Review Online*, vol. 94, p. 15, 2019.

[15] K. Lum and W. Isaac, "To predict and serve?," *Significance*, vol. 13, no. 5, pp. 14–19, 2016.

[16] U. Bhatt et al., "Explainable machine learning in deployment," *Proc. ACM Conf. Fairness, Accountability, and Transparency (FAT)*, 2020.

[17] R. Kitchin, "The real-time city? Big data and smart urbanism," *GeoJournal*, vol. 79, no. 1, pp. 1–14, 2014.

[18] M. Batty et al., "Smart cities of the future," *The European Physical Journal Special Topics*, vol. 214, no. 1, pp. 481–518, 2012.

[19] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI Magazine*, vol. 17, no. 3, pp. 37–37, 1996.

[20] L. G. A. Alves, H. V. Ribeiro, and F. A. Rodrigues, "Crime prediction through urban metrics and statistical learning," *Physica A*, vol. 505, pp. 435–443, 2018.

[21] Y. Sun, T. Chen, and H. Yin, "Spatial-temporal meta-path guided explainable crime prediction," *World Wide Web*, vol. 26, no. 4, pp. 2237–2263, 2023.

[22] N. Shah, N. Bhagat, and M. Shah, "Crime forecasting: a machine learning and computer vision approach to crime prediction and prevention," *Visual Computing for Industry, Biomedicine, and Art*, vol. 4, no. 1, p. 9, 2021.

[23] Y. Rayhan and T. Hashem, "AIST: An interpretable attention-based deep learning model for crime prediction," *ACM Transactions on Spatial Algorithms and Systems*, vol. 9, no. 2, pp. 1–31, 2023.

[24] S. Gupta and S. Sayer, "Machine learning for public good: Predicting urban crime patterns to enhance community safety," *arXiv preprint arXiv:2409.10838*, 2024.

[25] E. K. Zavadskas, R. Bausys, and I. Mazonaviciute, "Safety evaluation methodology of urban public parks by multi-criteria decision making," *Landscape and Urban Planning*, vol. 189, pp. 372–381, 2019.

[26] R. F. Reier Forradellas et al., "Applied machine learning in social sciences: neural networks and crime prediction," *Social Sciences*, vol. 10, no. 1, p. 4, 2020.

[27] X. Zhang et al., "Interpretable machine learning models for crime prediction," *Computers, Environment and Urban Systems*, vol. 94, p. 101789, 2022.

[28] E. Ahishakiye et al., "Crime prediction using decision tree (J48) classification algorithm," *International Journal of Computer and Information Technology*, vol. 6, no. 3, pp. 188–195, 2017.

[29] G. Kim et al., "Correlation analysis between urban environment features and crime occurrence based on explainable artificial intelligence techniques," *Journal of Asian Architecture and Building Engineering*, vol. 24, no. 6, pp. 5751–5770, 2025.

[30] J. Alghamdi and T. Al-Dala'in, "Towards spatio-temporal crime events prediction," *Multimedia Tools and Applications*, vol. 83, no. 7, pp. 18721–18737, 2024.

[31] O. Esan and B. Chimbo, "Systematic Literature Review on Crime Prediction using Machine Learning Techniques," *The Indonesian Journal of Computer Science*, vol. 14, no. 3, 2025.

[32] W. Samek, T. Wiegand, and K.-R. Müller, "Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models," *arXiv preprint arXiv:1708.08296*, 2017.

[33] T. Wang et al., "A Bayesian framework for learning rule sets for interpretable classification," *Journal of Machine Learning Research*, vol. 18, no. 70, pp. 1–37, 2017.

[34] Z. C. Lipton, "The mythos of model interpretability," *Queue*, vol. 16, no. 3, pp. 31–57, 2018.

[35] P. Angelov and E. Soares, "Towards explainable deep neural networks (xDNN)," *Neural Networks*, vol. 130, pp. 185–194, 2020.