Nowcasting Models to Predict Storm Movements in West Africa using ConvLSTM

Roshan Tariq.
201727213

Supervised by Douglas J. Parker

Submitted in accordance with the requirements for the
module MATH5872M: Dissertation in Data Science and Analytics
as part of the degree of

Master of Science in Data Science and Analytics

The University of Leeds, School of Mathematics

September 2024

Abstract

This research explores the use of Convolutional Long Short-Term Memory (ConvLSTM) networks for nowcasting storm movements in West Africa. Traditional Numerical Weather Prediction (NWP) models often find it difficult to accurately predict weather phenomena in this region due to the complex and dynamic environmental factors and the limited availability of observational data. To address these issues, this research proposes the use of a ConvLSTM-based approach, using satellite imagery to improve the accuracy of short-term weather forecasts. The ConvLSTM model is designed to handle spatiotemporal data, making it well-suited for predicting the movement of storms over time.

The dataset used comprises Convective Rainfall Rate (CRR) data collected from the Meteosat Second Generation satellites, processed and normalized to enhance the model's ability to learn from both high and low-intensity rainfall patterns. The performance of the ConvLSTM model is evaluated against traditional NWP methods and other deep learning models, using metrics like prediction accuracy and Fractional Skill Score (FSS). The results demonstrate that the ConvLSTM model outperforms existing NWP methods, and closely follows the existing deep learning methods. The ConvLSTM models are able to provide more timely and accurate storm predictions, which are crucial for effective disaster preparedness and response in West Africa.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Nowcasting refers to the prediction of storm movements on a very short-term basis. It is an important tool for weather forecasting due to its ability to give accurate and timely updates. This method involves the use of observational data with numerical models to make real-time weather information, making it useful for predicting severe weather phenomenons such as storms. The frequent and rapid formation of storms in the region of West Africa introduces a need for nowcasting models to provide early warnings and avoid potential damage. (Shi et al., 2015)

Severe storms frequently impact West African countries, causing extensive damage and disruption to human life and agricultural lands. Traditional Numerical Weather Prediction methods based on PySTEPS, have been previously used for predicting storm movements; however, these methods fall short in terms of accuracy due to complex and dynamic environmental factors. The limitations of these models could be due to the fact that they could not handle high variability and complexities in the weather systems.(Kumar and Hoerling, 1995)

Considering these limitations, there is a need for more timely and accurate predictions using satellite imagery based approaches. Convolutional Long Short-Term Memory (ConvLSTM) networks could be used as a good approach for improving the storm nowcasting by making use of satellite imagery to enhance weather prediction. The hypothesis behind this research is that a ConvLSTM model could outperform existing models such as CNN or U-Nets in terms of prediction accuracy. Recent developments in deep learning models have shown significant improvements in predicting weather phenomena, suggesting that ConvLSTM could be more effective than traditional methods (Shi et al., 2015).

1.2 Introduction to Nowcasting

The prediction of weather conditions over a very short period of time, typically from a few minutes up to six hours is referred to as **Nowcasting**. This method is especially useful for predicting extreme weather phenomena such as thunderstorms, heavy rainfall, and tornadoes. The main idea behind nowcasting is to produce real-time updates on weather conditions by making use of observational data from various data sources like radar, satellites and weather stations, in order to identify the current-state of the atmosphere as accurate as possible and then use it to predict short-term changes in the near future.

Nowcasting often relies on simplified versions of the equations used in traditional Numerical Weather Prediction (NWP), that include *Navier-Stokes* equation for fluid dynamics which

describes how air moves in the atmosphere, as well as thermodynamic equations that account for temperature and humidity changes. In nowcasting, these equations are solved over small spatial and temporal scales to provide detailed short-term forecasts.

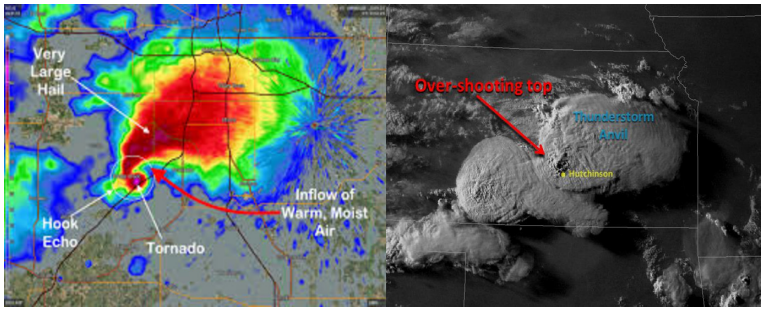

Figure 1.1: A side-by-side comparison of high-resolution Doppler radar and satellite imagery.

In Figure 1.1, on the left, radar imagery shows the details of a tornado-producing supercell, showing features such as a hook echo, inflow of warm, moist air, and areas of very large hail, which are important for immediate severe weather predictions, and on the right, the satellite image captures the same storm system, highlighting structural elements like the overshooting top and thunderstorm anvil, which indicates storm intensity and evolution. This observational data is then collected and integrated into numerical models to learn the behaviour of the storm, and using this assimilated data, the model tries to predict how the weather will evolve over the next few minutes to hours. Generally, these nowcasts will include details on precipitation, wind patterns, temperature changes, and other important weather variables. The usage of this method is therefore crucial for regions that are prone to sudden and severe weather fluctuations, such as West Africa, where early warnings can save lives and reduce damage to property.

1.3 Research Motivation and Objectives

The research is motivated by the need to overcome the limitations observed in the Numerical Weather Prediction models in West Africa in predicting weather phenomena. In spite of having several advancements in Weather Prediction Algortithms, the lack of reliable and accurate data from the weather stations in West Africa results in the region to be vulnerable to severe rainfall and storm movements. These can lead to improper disaster management and as a result, several lives could be endangered and agricultural land could be destoryed, leading to several socio-economic consequences. Therefore, there is an urgent need for a more accurate and timely weather prediction model that can further improve the early warning systems and prevent advent destruction from storms affecting West Africa.

With the recent successes of Machine Learning and Deep Learning in improving the Nowcasting accuracy, this research aims to make use of Convolutional Long Short-Term Memory (ConvLSTM) networks for storm prediction in West Africa. The primary aim of the research is to develop a ConvLSTM Nowcasting model, and to compare the performance of ConvLSTM model against traditional NWP models and other exisitng deep learning models, using metrics like prediciton accuracy, Fraction Skill Score (FSS) and ROC curves. The effectiveness of the model will also be checked by implementing various input-output configurations to find the combination that would give an adequate balance between accuracy and computational efficiency.

1.4 Literature Review

Recent advancements in weather nowcasting, particularly for regions like West Africa, have made use of a range of models and techniques to enhance prediction accuracy. The limitations of traditional Numerical Weather Prediction (NWP) models, which often struggle with the highly variable and dynamic atmospheric conditions of West Africa, have motivated researches to explore deep learning-based approaches. These methods aim to address the challenges posed by the lack of reliable observational data and the complex meteorological interactions in this region.

1.4.1 Traditional Nowcasting Techniques and Challenges

Nowcasting traditionally used to relied on Numerical Weather Prediction (NWP) models. This concept emerged in the mid-20th century alongside advancements in meteorology and radar technology. Kalnay (2003) and Warner (2010) highlight that traditional nowcasting techniques rely on the extrapolation of current weather patterns, using observations from radar, satellites, and surface stations to predict the immediate future. These models are fundamentally based on the principles of atmospheric dynamics and fluid mechanics, aiming to solve the set of differential equations that govern the behavior of the atmosphere. This approach, as documented by Kalnay (2003) and Warner (2010), has been crucial in providing weather forecasts for different storm phenomena. The earliest NWP models, developed in the 1950s, were relatively simplistic due to limited computational power. However, as computing technology advanced, so did the complexity and accuracy of these models, leading to more sophisticated and reliable weather predictions. Additionally, Bližňák et al. (2020) simulated the weather forecast processes using NWP methods such as COSMO to forecast convective storms in Czech Republic, where the models were used to try to forecast the weather for upto 5 days in advance. (Kalnay, 2003; Warner, 2010; Bližňák et al., 2017)

NWP models like the **European Centre for Medium-Range Weather Forecasts** (ECMWF) are among the more accurate forecasting models frameworks used globally. The ECMWF model operates by running multiple simulations, each with slightly different initial conditions, to account for the uncertainty that could occur in weather forecasting. This approach provides a range of possible outcomes which increases the reliability of forecasts. However, the model's performance heavily depends on the accuracy of the initial conditions. As Molteni et al. (1996) highlighted, the availability and quality of these initial conditions are often insufficient in regions with sparse observational data, such as West Africa, leading to less accurate predictions.

The accuracy of NWP models heavily depends on the quality and quantity of observational data used to initialize the model. In regions with dense observational networks, such as North America and Europe, these models can produce highly accurate forecasts. However, in data-sparse regions like West Africa, where observational stations are less in number, the initial conditions fed into the models are often incomplete or inaccurate, which can lead to significant errors in the predictions. Warner (2010) and Molteni et al. (1996) emphasize that the sparse observational network in West Africa exacerbates the inaccuracies of NWP models, particularly for nowcasting. (Warner, 2010; Molteni et al., 1996)

Another challenge in nowcasting models is the complexity of atmospheric processes, especially in regions with high variability in weather conditions such as West Africa, which experiences frequent Mesoscale Convective Systems (MCSs), which are more difficult to predict due to their rapid development and short lifespan. These systems are a major source of precipitation in the region, and their unpredictability can have severe impacts for agriculture, water manage-

ment, and disaster preparedness. Furthermore, NWP models are computationally demanding, requiring powerful supercomputers to run high-resolution simulations in real-time. This computational intensity limits the accessibility of these models in regions with limited technological infrastructure. Moreover, the time required to run these simulations can result in delays that reduce the usefulness of the forecasts for nowcasting purposes.

Therefore, to improve the accuracy of NWP models, especially in data-sparse regions, researches were conducted to make use of satellite observations. Satellites provide high-resolution spatial data that can be integrated into NWP models to improve the initial conditions. Liu et al. (2011) and Burton et al. (2022) highlight the value of using satellite data to capture the atmospheric phenomena that are not well represented by ground-based observations alone such as radars. However, these processes are computationally intensive and needs advanced techniques for prediction, which can prove to be difficult in regions with limited technological infrastructure, as noted by Warner (2010) and Liu et al. (2011).(Liu, 2011; Warner, 2010)

In West Africa, the limitations of traditional NWP models are more due to the region's climatic conditions and the scarcity of observational data. The ECMWF model, while being robust in a lot of aspects, find it difficult to accurately predict the rapid, localized weather changes that are common in this region. As Molteni et al. (1996) pointed out, the model's performance is highly dependent on the initial condition data, which is often inadequate in Africa due to the limited number of weather stations which reduces the availability of high-quality data for initializing NWP models. Burton et al. (2022) discusses how this scarcity of data can lead to significant forecast errors, particularly in the short term.(Molteni et al., 1996; Burton et al., 2022)

While traditional nowcasting techniques, particularly NWP models, have advanced weather prediction capabilities, they have limitations , especially in regions like West Africa. The technical and infrastructural challenges in the region instigate the need of alternative approaches that can overcome these limitations. Deep learning models, which learn directly from data and do not rely on complex physical equations, offers a good solution to this challenge.

1.4.2 Satellite based Nowcasting

Due to the limited availability of ground-based observational data in regions like West Africa, satellite-based nowcasting can help alleviate this problem. Satellites provide high-resolution, real-time data across vast areas, which helps in tracking the development and movement of weather systems. Satellites such as the Meteosat Second Generation (MSG) provide continuous coverage of the atmosphere, capturing essential data on cloud formation, precipitation, and atmospheric conditions. Burton et al.(2022) showed that satellite-based nowcasting could give out skillful predictions for Mesoscale Convective Systems with lead times of up to 4 hours in West Africa. By integrating satellite data with advanced nowcasting models such as PyCAST, researchers have been able to increase the accuracy and reliability of weather forecasts in West Africa. Roberts et al. (2022) discussed the progress made in nowcasting technologies for Africa, mostly about the integration of satellite data with other meteorological inputs, which is essential for producing accurate short-term forecasts in areas prone to rapid weather changes. The integration of satellite data into nowcasting has evolved significantly over the past few decades. Early methods primarily involved the visual interpretation of satellite images, which were limited by the subjective nature of human analysis and the inability to process large amounts of data quickly. One of the key advancements in satellite-based nowcasting has been the development of algorithms capable of automatically identifying and tracking convective cells. Burton et al. (2022) discuss the use of satellite-based techniques such as the Rapid Developing Thun-

derstorms (RDT) algorithm, which automatically detects, tracks, and forecasts the evolution of thunderstorms in real-time. This has been particularly valuable in West Africa, where thunderstorms can develop rapidly and with little warning. (Burton et al., 2022; Roberts et al., 2022)

1.4.3 The Use of AI in Sequential Prediction and Nowcasting

The rise of Deep Learning has revolutionised many fields, including meterology. Deep learning models, particularly Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, are able to capture complex spatio-temporal patterns in weather data that traditional methods often miss. These models can learn directly from raw data, bypassing the need for explicit feature extraction, which can be time consuming. Convolutional Neural Networks (CNNs) is a deep learning model that is very commonly used to deal with image data, including satellite images. Unlike traditional NWP models that rely on physical equations to predict weather phenomena, CNNs learn directly from data by capturing underlying patterns pixel-wise. One of the early examples of applying CNNs for frame-by-frame prediction, which is similar to the task of nowcasting, is the work by Tran et al. (2015) , which showed the use of 3D Convolutional Networks (3D-CNNs) in learning features from video data. CNNs have limitations in handling sequential data, as they are designed for spatial data processing, which is not helpful for nowcasting where both spatial and temporal patterns are important. While CNNs can capture spatial features effectively, they struggle to model the temporal dependencies in weather data. Moreover, the high computational cost of 3D-CNNs along with the challenges of training such deep networks on limited meteorological datasets gives further motivation to look into other alternatives such as ConvLSTM (Tran et al., 2015).

Several researches that compare the performance of deep learning models with traditional NWP model consistently show that deep learning approaches are better in nowcasting tasks. For example, Shi et al. (2015) applied ConvLSTM models to the task of precipitation nowcasting and showed a lot of improvements in accuracy compared to traditional NWP models. The ConvLSTM model's ability to effectively handle large volumes of high-dimensional meteorological data has been a key factor in its success. Similarly, Larerquist et al.(2021) focused on using deep learning models to nowcast convection by analyzing `Himawari-8` satellite data. They found that ConvLSTM models performed better than traditional nowcasting techniques, particularly in predicting the spatial coverage of convection. This aspect is very helpful for regions prone to severe storms, where timely and accurate nowcasting can help decrease the impacts of extreme weather events (Shi et al., 2015; Lagerquist et al., 2021).

The primary advantage of deep learning models like ConvLSTM in nowcasting lies in their ability to learn directly from data. Deep learning models can infer patterns directly from large datasets, unlike NWP models. This data-driven approach allows for the capture of complex, nonlinear relationships that might be overlooked or inadequately modeled by traditional approaches. While deep learning models like ConvLSTM have helped in improving nowcasting accuracy, it is mostly limited to regions with good observational networks and relatively stable weather patterns. As Shi et al. (2015) and Larerquist et al. (2021) showed how ConvLSTM models can outperform traditional NWP models in several aspects, their effectiveness in regions like West Africa which is prone to high variable storm patterns has not been explored yet. This gap points to the need for research focused on adapting these models to the West African weather systems, including their integration with satellite data and the development of region-specific training datasets. (Shi et al., 2015) (Lagerquist et al., 2021)

Chapter 2

Convective Rainfall Rate

The primary data source for this study is satellite imagery of **Convective Rainfall Rate** (CRR) over the West African region. Convective rainfall occurs due to the rapid upward movement of warm, moist air, leading to the formation of clouds with large vertical extents, and CRR quantifies the precipitation associated with Convective Clouds. These clouds often result in heavy, localized rainfall, which can cause severe weather events such as thunderstorms and flash floods. Unlike stratiform rainfall, which is more uniform and widespread, convective rainfall is highly variable and concentrated, making it more difficult to predict accurately.

The CRR makes use of data from the Spinning Enhanced Visible and Infrared Imager (SEVIRI) onboard the Meteosat Second Generation (MSG) satellites, to monitor cloud cover, type, and temperature. By making use of the infrared (IR) and visible (VIS) satellite data, the CRR algorithm determines rainfall intensity in millimeters per hour (mm/h), and using this data for Nowcasting models will help in tracking and forecasting the movement of storms with high precision and accuracy.

This dataset is obtained from the *Nowcasting Satellite Application Facility* (NWCSAF) in the European Organisation for the *Exploitation of Meteorological Satellites* (EUMETSAT). EUMETSAT provides high-resolution satellite data that is essential for capturing the dynamic and complex nature of storm movements. According to NWCSAF, the CRR product aims to estimate the precipitation rate for convective clouds. The algorithm used for finding the CRR product assumes that clouds being both high and with large vertical extent are more likely to produce rain.

2.1 CRR Algorithm and Processing

The CRR algorithm combines SEVIRI data with ground-based radar observations to estimate the rainfall rate associated with convective clouds. The radar data is used to calibrate the CRR product by matching satellite observations with radar-derived rainfall measurements. The calibration functions are generated by combining SEVIRI and Radar data. SEVIRI data is critical for detecting convective clouds and estimating their rainfall rates by analyzing the brightness temperatures and reflectances. Radar systems measure precipitation by emitting microwave signals and analyzing the returned signals that bounce off raindrops. This data gives values of rainfall intensity and is used to calibrate satellite-derived estimates. The combination of satellite and radar data is critical for generating reliable CRR estimates, which are crucial for accurate short-term weather predictions in the West African region. The CRR algorithm employs the following methods:

- **Calibration:** The calibration of the CRR product is critical for maintaining the accuracy of rainfall estimates. SEVIRI data, which includes IR brightness temperatures and VIS reflectances, is combined with radar-derived rainfall rates from the ground radar satellites, to create calibration functions. These functions are used to estimate the basic CRR value for each pixel. The calibration process also takes into account the relationship between cloud properties observed from space and the actual precipitation measured on the ground, providing a good dynamic framework for accurate rainfall estimation.
- **Filtering:** Filtering is used to differentiate between convective and stratiform rainfall. Stratiform rain is generally uniform and widespread, while convective rain is localized and intense. The filtering process eliminates data points associated with stratiform rain, focusing the CRR estimates on convective events. Making this distinction is important since convective rainfall is associated more often with severe weather such as thunderstorms and flash floods.
- **Correction Factors:** Several correction factors can be applied to account for temporal and spatial variability, moisture, cloud growth, cloud-top temperature gradients, parallax, orographic effects, and lightning activity.
  - **Temporal and Spatial Variability:** Adjustments made to account for changes in cloud characteristics and environmental conditions over time and across different locations.
  - **Moisture Correction:** Adjustments based on the moisture content in the atmosphere, which affects rainfall intensity.
  - **Cloud Growth Correction:** Factors that consider the growth or decay rates of clouds, indicating active convection associated with precipitation.
  - **Cloud-Top Temperature Gradients:** Corrections based on the temperature differences at the top of the clouds, indicating the presence of convective activity.
  - **Parallax Correction:** Adjustments to account for the apparent displacement of cloud positions due to the satellite's viewing angle.
  - **Orographic Effects:** Adjustments for the influence of terrain, such as mountains, on precipitation patterns.
  - **Lightning Activity:** Incorporating lightning data to enhance the accuracy of rainfall estimates, as lightning is often associated with intense convective activity.

The CRR product is structured such as to provide detailed precipitation rates by classifying them into twelve distinct categories, which are then used to generate hourly rainfall accumulations and rainfall intensity measurements.

2.2 Dataset Description

The dataset comprises a chunk of the satellite imagery of Convective Rainfall Rate (CRR) over West Africa, covering coordinates from 37.95, -19.95 (top left) to -34.95, 51.95 (bottom right), as seen in Figure 2.1. The dataset includes 3040 samples, each representing a day of data with 5 time frames, where each time frame corresponds to an hour.

The data has a spatial resolution of 256x256 pixels, which ensures that more features of storms patterns and movements are captured. Each sample provides data on rainfall intensity

Figure 2.1: Political Map of West Africa

over each region, which shows the region experiencing the most amount of rainfall or storm movements in that particular day. This is vital for understanding and predicting storm movements. This dataset is a subset of a larger dataset used in previous research (Burton et al.), showing its relevance and applicability to storm prediction. (Burton et al., 2022)

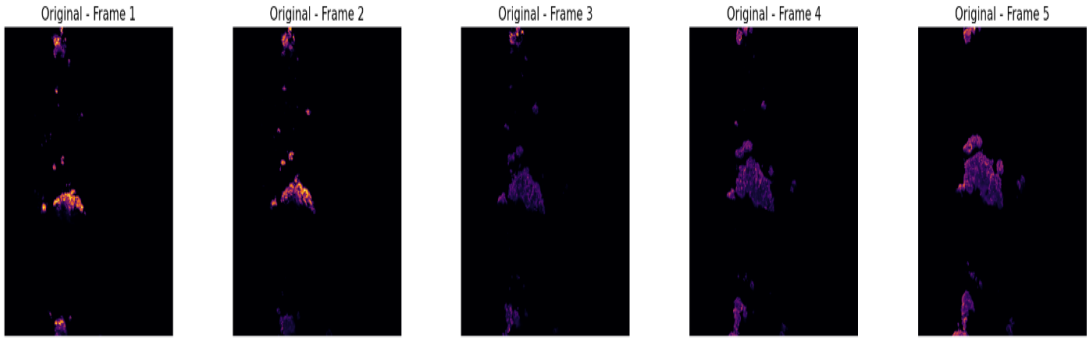

Figure 2.2: Sample frames from the dataset illustrating the convective rainfall rate over West Africa where each frame corresponds to an hourly interval.

In the visualized dataset sample in Figure 2.1, Frame 1 captures the initial observation of cloud cover and convective rainfall rate. In Frame 2, cloud patterns begin to evolve, indicating potential storm development. Frame 3 shows further development and movement of cloud formations. By Frame 4, there is continued evolution and movement of clouds. Finally, Frame 5 shows significant cloud movement, which is useful for predicting subsequent storm activity. This sequence highlights the dynamic changes in convective rainfall rate over time in storm prone regions such as West Africa, further stating the importance of temporal resolution in storm prediction.

2.2.1 Challenges of Analyzing Raw CRR Data

Due to the highly skewed nature of the CRR data, where most of the values are close to zero or very low, an analysis of this data would lead to the model missing out on important patterns in low intensity regions. This skewness can be observed in the histogram of the original CRR

data, as shown in Figure 2.3 (left). The distribution shows that a large number of data points are concentrated closer to zero. This could suggest that in many regions, especially during certain time frames, there is minimal or no convective activity, and that the low-intensity values could indicate that the cloud patterns are dissipating or that the region is experiencing minimal storm activity.

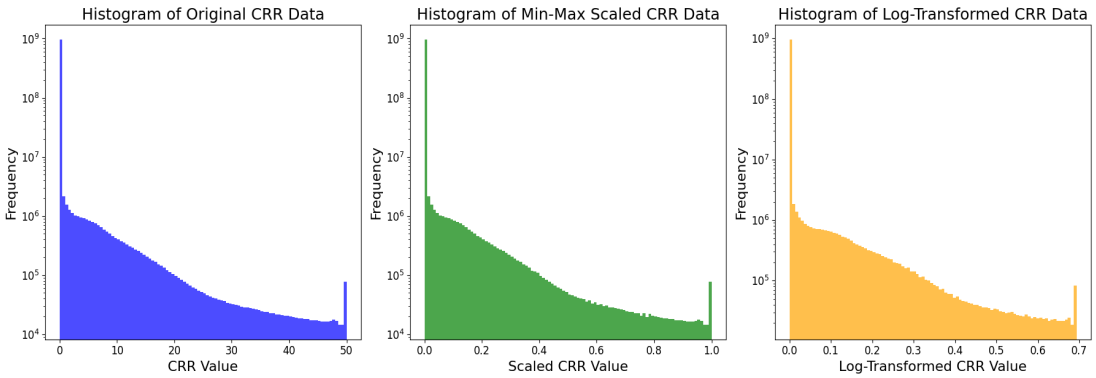

Figure 2.3: Histograms of the original Convective Rainfall Rate (CRR) data showing the skewed distribution towards lower intensity rainfall (left), Min-Maxed Scaled CRR data (middle) and Log-Transformed CRR data (right).

However, while training a model, it might overlook these low intensity regions and focus solely on high-intensity regions. This could lead to the model learning more on high-intensity storm movement patterns, and completely neglect the low-intensity regions, which could be crucial in predicting the early stages of storm development or movement patterns. To address the issues posed by a skewed distribution, a *Min-Max Scaling* was applied to try to normalize the range of the CRR values between 0 and 1, which helps to standardize the data for a more accurate analysis, which can be seen in Figure 2.3 (middle).

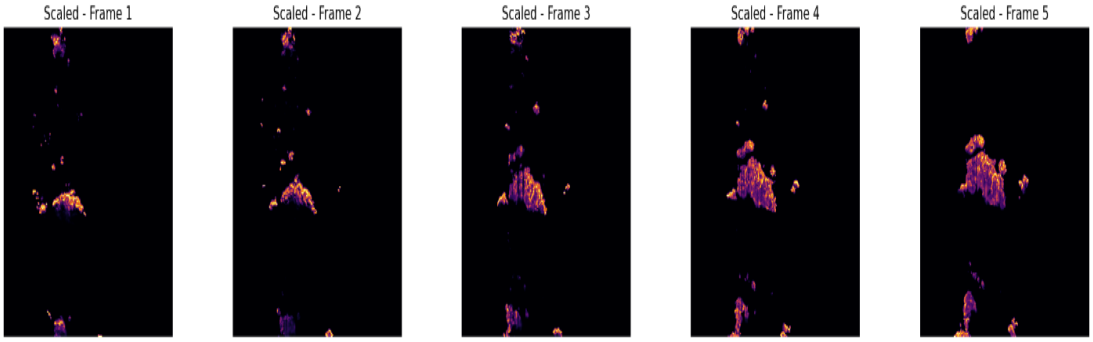

Figure 2.4: Sample frames from the dataset illustrating the convective rainfall rate over West Africa after applying Min-Max Scaling.

Finally, a log-transformation was applied to the scaled data to further improve the features visibility, making it easier for the model to learn more features from the storm movement patterns. This transformation will compress the range of high-intensity values while also expanding the range of low-intensity values, and this can be seen in Figure 2.3 (right), where the histogram of the log-transformed CRR data shows a more evenly set of distributed values. ConvLSTM models are designed to detect spatio-temporal dependancies, and small changes in low-intensity

regions may precede larger storm events. By applying log-transformation, these early changes can be made more pronounced, thereby helping the ConvLSTM model to learn the sequence of events leading to storm development and movement.

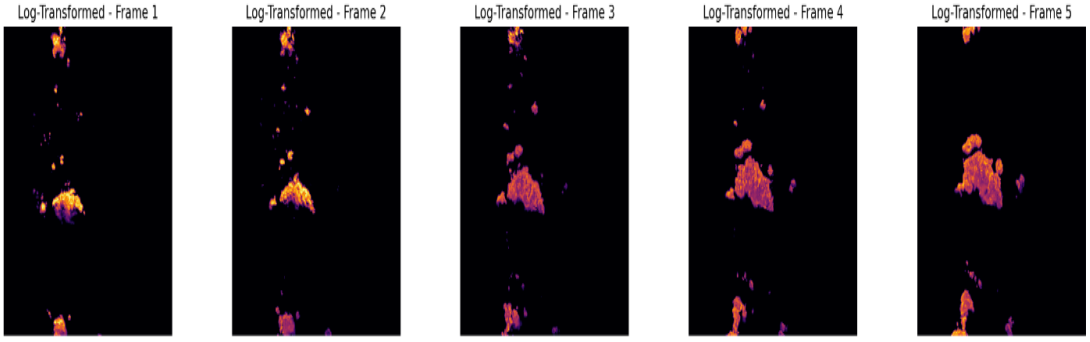

Figure 2.5: Sample frames from the dataset illustrating the convective rainfall rate over West Africa after applying Log-Transformed scaling.

To further justify the use of the log-transformed scaling, a visual comparison was performed on a randomly selected sample from the dataset across all five frames. In Figure 2.2, the plots mostly consist of low intensity rainfall rates, which has produced a dark, low-contrast image with only a few bright spots, making it difficult to identify regions with low Convective Clouds.

Next, in Figure 2.4, after applying Min-Max Scaling, the contrast has slightly improved as the CRR values are now normalized between 0 and 1, and some of the low intensity regions from the 'Original Sample' have now been plotted as bright spots. Finally, the Log-Transformed Scaling as seen in Figure 2.5 shows a more even distribution of pixel intensities, where low-intensity regions become significantly more distinguishable. Therefore, the usage of log-transforms is important if the model would be required to learn from both high-intensity and low-intensity rainfall patterns. By improving the contrast of the subtle changes in the low-intensity regions, the model could more accurately capture the minute details and changes in the storm movements, which would further help in predicting the development of severe storms that could stem from these low-intensity cloud patterns.

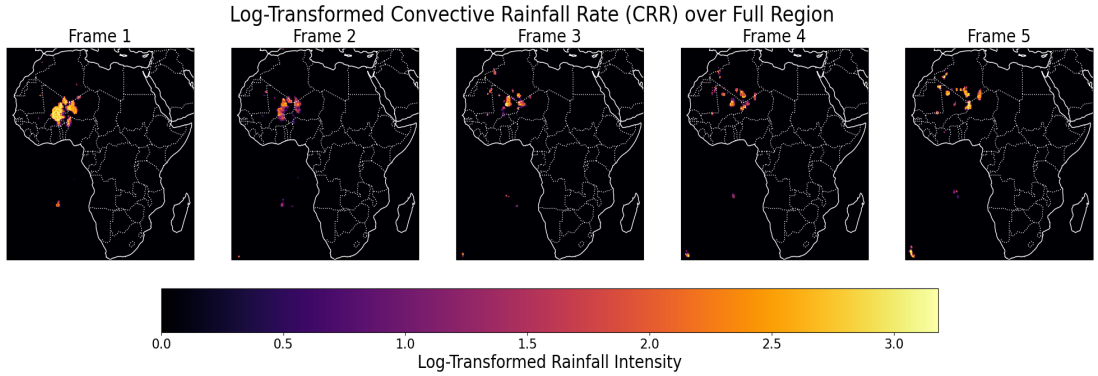

Figure 2.6: Visual representation of Sample 1 of convective rainfall patterns over West Africa.

Additionally, for further understanding of how each convective rainfall plot could affect regions in West Africa, the dataset was scaled to fit into the continent of Africa, which can be seen in Figures 2.6,2.7 and 2.8. Each figure represents a separate sample of convective rainfall

from the dataset, which shows how a storm can develop, intensify or dissipate across parts of the region of West Africa.

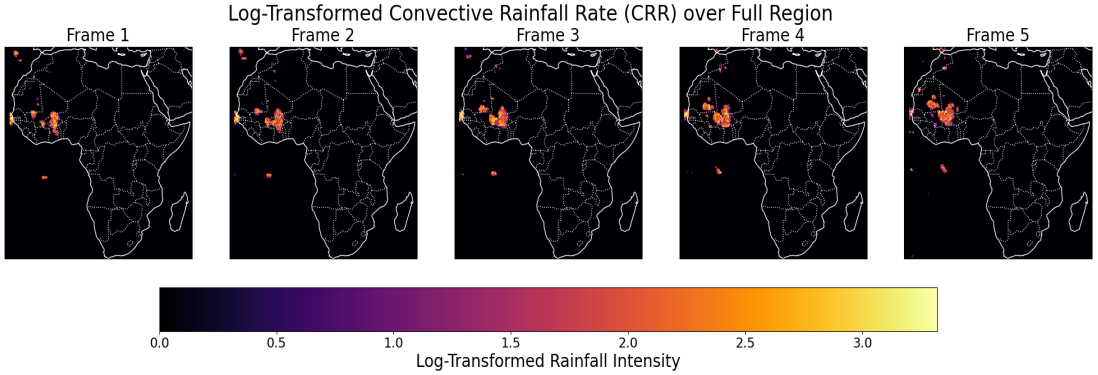

Figure 2.7: Visual representation of Sample 2 of convective rainfall patterns over West Africa.

By examining these random samples, we gain insights into the varying severity of convective rainfall events that occur in this region. In Figure 2.6, the storm patterns are more localized in nature, where the storms only affect a very small concentrated area in West Africa, while Figure 2.7 shows a more widespread but a less intense convective rainfall pattern that covers a broader region. Meanwhile figure 2.8 shows several smaller and isolated convective storm patterns across the entire region.

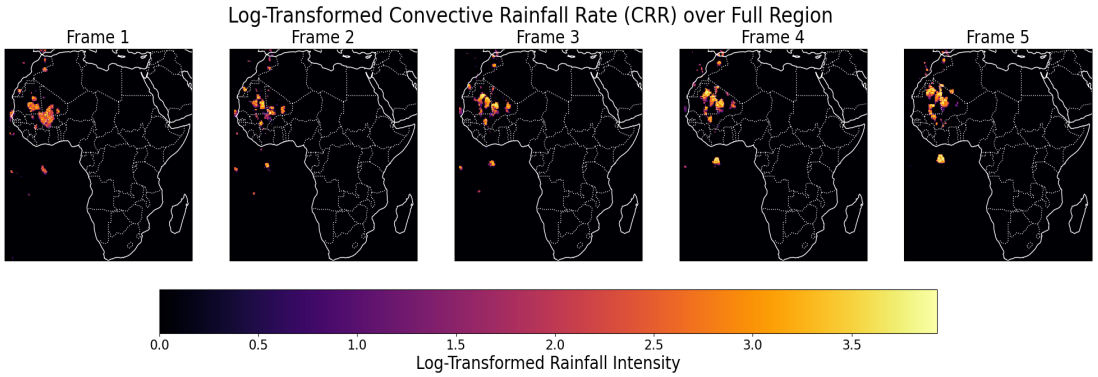

Figure 2.8: Visual representation of Sample 3 of convective rainfall patterns over West Africa.

These figures show the variability and unpredictability of convective storms, which further dictates a need for an efficient nowcasting predictive model to mitigate potential disasters. Analysing these frames on their own can help in understanding the movement patterns of convective storms, along with their development and evolution. These figures not only provide a visual representation of the data but also serve as a tool for analyzing and nowcasting convective rainfall events in West Africa. The sequential progression shown across the frames enables a deeper understanding of storm dynamics, essential for effective weather prediction and disaster preparedness.

Chapter 3

Methodology

The methodology focuses on using ConvLSTM networks to enhance the accuracy of storm predictions by capturing both spatial and temporal dependencies in the satellite imagery data. This chapter details the steps involved in designing, training, and evaluating the ConvLSTM models, along with the use of Optical Flow methods for short-term visualization, and the evaluation metrics that would be used to evaluate the trained models.

3.1 Introduction to Optical Flow

Optical Flow methods are used to visualize the potential movement patterns of storms. These methods estimate the motion of pixels between two frames of images, in this case satellite imagery, and helps in giving information on the direction and speed of the storm, while also giving insights on the possible areas that the storm could affect in the next frame. It is the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer (e.g., a camera or satellite) and the scene (e.g., cloud formations in CRR data). The method represents the velocity field, which is a vector, of pixel intensities in consecutive frames and is used to estimate the motion of objects in the image plane. The fundamental assumption of optical flow, called the brightness constancy assumption, is that the intensity of a pixel remains constant between consecutive frames while the pixel moves, which can be expressed mathematically as:

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t) \tag{3.1}$$

where $I(x, y, t)$ is the intensity of the pixel at position (x, y) at time t, and $(\Delta x, \Delta y)$ represents the displacement of the pixel over time Δt. To derive the optical flow, the brightness constancy assumption is used along with the Taylor series expansion to approximate the change in intensity. This leads to the optical flow constraint equation:

$$\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} = 0 \tag{3.2}$$

where $\frac{\partial I}{\partial x}$ and $\frac{\partial I}{\partial y}$ are the spatial gradients, $\frac{\partial I}{\partial t}$ is the temporal gradient, and (u, v) are the horizontal and vertical components of the optical flow vector. This equation represents a single constraint for two unknowns (u and v). Additional constraints or assumptions are required to solve it, leading to different optical flow methods.

In meteorology, optical flow methods are used to analyze and predict the movement of storm systems. Applying these methods to satellite imagery allows meteorologists to visualize the flow

of clouds and precipitation, gaining understanding into the direction and speed of storm movement. These methods enhance the understanding of storm dynamics, improve the accuracy of weather predictions, and provide real-time visualizations for better decision-making. For example, the Farneback method can be used to visualize the continuous movement of clouds, while the Lucas-Kanade method can track significant cloud formations and analyze their direction and speed (Farneback, 2003; Baker and Matthews, 2004).

3.1.1 Farneback and Lucas-Kanade Method

The Farneback and Lucas-Kanade methods are two well-established optical flow techniques that give complementary insights into storm movement patterns, that could be useful for meteorological analysis. Both methods are unique in their concepts and are used to track and predict the movement of storms in weather nowcasting systems.

The Farneback method is a dense optical flow algorithm that estimates the motion of each pixel between two frames, making it useful to analyze the continuous motion of storm patterns. It provides a detailed motion field by calculating the optical flow for all pixels, thereby capturing fine details of the movement. The algorithm starts with a polynomial expansion of the pixel neighborhood to approximate the image's local structure. It then estimates the motion by solving the optical flow equations for each pixel based on this polynomial expansion. The initial estimates are iteratively done by warping the image and re-estimating the flow. Mathematically, the polynomial expansion can be expressed as:

$$f(x) \approx f(0) + \nabla f \cdot x + \frac{1}{2} x^T H_f x \tag{3.3}$$

where ∇f is the gradient and H_f is the Hessian matrix of f. This dense approach provides insights into the continuous motion of storm clouds, making it suitable for analyzing convective systems where storm movements are complex and fluid. In Figure 3.1 (middle), the Farneback method highlights the regions of intense activity with greater clarity, making it possible to track the flow of storms with high precision. The bright regions shows areas with significant motion that corresponds to the rapidly evolving storm patterns. (Farneback, 2003)

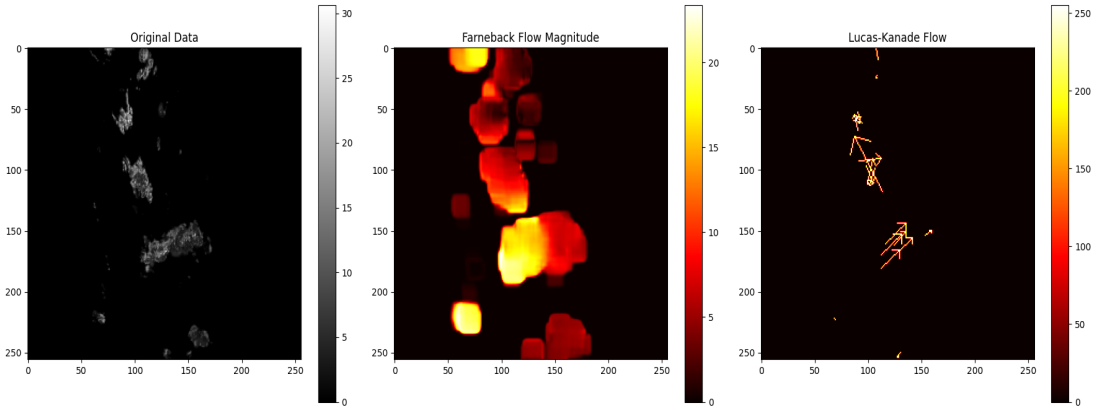

Figure 3.1: Optical flow visualization using the Farneback method and Lucas-Kanade method to estimate storm motion between two frames.

On the other hand, Lucas-Kanade method is a sparse optical flow algorithm that estimates the motion of a few set of feature points within the image. It operates under the assumption that

the flow is constant in a small subset or region around each pixel, making it more computationally efficient when compared to the Farneback technique. The method starts by detecting a set of feature points using corner detection techniques such as Shi-Tomasi, that identifies points of interests such as the corners or edges in the patterns shown in a frame. For example, in Figure 3.1 (right), around 15 points were chosen from frame 1 as points of interest. For each of these points, the optical flow equations are solved within small neighborhoods to estimate the motion. The estimates are iteratively found to improve accuracy. (Zhang and Chanson, 2018)

Mathematically, the optical flow in a local neighborhood is given by:

$$\sum_{i=1}^{n} w(i) \left(\frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v + \frac{\partial I}{\partial t} \right)^2 \tag{3.4}$$

where $w(i)$ is a weighting function for the neighborhood pixels. In Figure 3.1 (right), the arrows originating from these chosen points of interests indicate the direction and speed of the storm movement. Longer arrowheads indicate that the storm moves significantly faster between the two frames, stating that the storm could travel a large distance in a short period of time, and the arrowheads point towards the direction of storm movement. This allows for a more focused analysis of the storm development and movement.

Both these models offer good insights into the storm development and movement. By combining the dense motion analysis from the Farneback method with the precision tracking from the Lucas-Kanade method, this approach could be a good method to understand the movement patterns within storm systems. However, these methods fail to capture the temporal relations between the frames as they rely on analysing the motion of pixels between the two consecutive frames. This limits its ability to predict the future evolution of such storms for longer periods of time. Unlike optical flow methods, ConvLSTM's can learn how the storm would develop over time, since they excel in spatio-temporal form of learning, where they learn the spatial aspects of the strom patters as well as how they move with time. By learning from large amounts of historical data, the model will be able to accurately predict the different weather scenarios in complex regions such as West Africa.

3.2 ConvLSTM Network

The primary model used in this study is the Convolutional Long Short-Term Memory (ConvLSTM) network, which is specifically designed to handle spatiotemporal data. ConvLSTM combines the strengths of convolutional neural networks (CNNs) in handling spatial data and Long Short-Term Memory (LSTM) networks in handling temporal sequences, thereby enabling the model to learn spatial and temporal data simultaneously.

In a ConvLSTM cell, the input data, hidden state, and cell state are treated as 3D tensors. The convolutional operation is applied to these tensors, allowing the model to capture the spatial correlations in the input data and the temporal correlations across several time frames, making them a really good predictive model for nowcasting. Studies have shown that ConvLSTM networks outperform traditional NWP models in both accuracy and computational efficiency for short-term forecasts. This is particularly relevant in West Africa, where timely and accurate nowcasting can significantly impact disaster preparedness and response efforts. ConvLSTM networks also addresses the limitations of optical flow methods by using information across time steps, allowing for a better understanding of storm dynamics. While optical flow methods focus on detecting motion of pixels between consecutive frames, ConvLSTM networks can

learn the sequence of events leading to storm development and track these developments over time. (Shi et al., 2015)

The ConvLSTM unit updates its states using the following key equations:

Input Gate:

$$i_t = \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \circ C_{t-1} + b_i) \tag{3.5}$$

Forget Gate:

$$f_t = \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \circ C_{t-1} + b_f) \tag{3.6}$$

Cell State Update:

$$C_t = f_t \circ C_{t-1} + i_t \circ \tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c) \tag{3.7}$$

Output Gate:

$$o_t = \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} \circ C_t + b_o) \tag{3.8}$$

Hidden State:

$$H_t = o_t \circ \tanh(C_t) \tag{3.9}$$

In these equations, $W_{x\_}$, $W_{h\_}$, and $W_{c\_}$ represent the weights for the input, hidden, and cell states, respectively. The output at each time step is influenced not only by the current input but also by the information stored in the previous states, allowing the model to maintain context across time steps.

3.2.1 Model Architecture

The ConvLSTM model used in this research is designed to capture the storm movement patterns in a sequence of Convective Rainfall Rate (CRR) data frames. The model architecture as shown in Figure 3.2, consists of multiple layers optimized for handling the spatiotemporal nature of the data, with 3 ConvLSTM2D layers, each followed by Batch Normalisation and Dropout layers to improve generalization and prevent overfitting.

The input is a sequence of satellite imagery frames, each of size 256x256 pixels, representing the CRR data over a specific period. The input data is of the format - `(sequences, time frames, height, width)`, where sequences amount to 3040 in total with 5 time frames each, and every frame contains spatial information (i.e., the location and intensity of rainfall) that is crucial for accurate storm prediction. Since ConvLSTM models expects 5 dimensions in the input data, with the 5^{th} dimension being 'channel' (indicates the number of RGB layers), the input data dimensions were expanded to include this, making its data shape format as `(sequences, time frames, height, width, channel)`, with the 5^{th} dimension being equal to 1 as the images are gray-scaled.

The dataset is then split into training, validation, and test sets. The training set comprises 70% of the data, the validation set includes 15%, and the test set includes the remaining 15%. This split ensures that the model is trained on a diverse set of samples and validated and tested on unseen data to evaluate its performance accurately. After the split, the training dataset had a shape of $(2432, 5, 256, 256, 1)$, while the validation and test dataset each had a shape of $(304, 5, 256, 256, 1)$. The training set was used to fit the model, the validation set was used

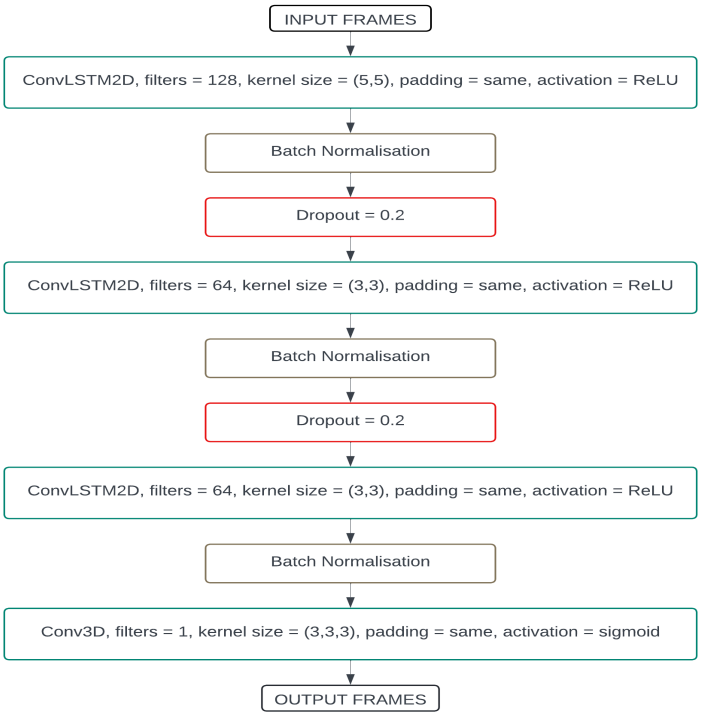

Figure 3.2: Sample architecture of the ConvLSTM model

to tune the model's hyperparameters, and the test set was used to assess the final model's performance.

The first ConvLSTM2D layer uses a larger kernel size of $(5, 5)$ to capture more spatial features across multiple frames. This could help the model to learn the larger storm movement patterns much better. This layer also makes use of 128 filters, where each filter slides over the input frames to capture these larger scale features at each time step. The padding of the resultant output of the first layer was set to be same as the input to in order to maintain the same spatial dimensions as the input. This also ensures that the spatial features that are at the corners or borders of the image is not lost after convolution.

Mathematically, the output at each location or pixel of the input image is calculated by performing convolution operation between the filter values and the pixel value at that point. The output of this convolution process is then sent through an activation function *ReLU* (Rectified Linear Unit), which sets all negative values to zero and returns the input directly as x if it is positive (3.10). It helps in introducing non-linearity to the model, that helps it to learn complex patterns in the data.

$$ReLU = max(0, x) \tag{3.10}$$

ReLU also helps in avoiding possible vanishing gradient issues that could occur during backpropagation. Vanishing gradient issues are when the gradients become so small after a certian number of epochs that it would become harder for the model and could eventually stop the training altogether. This could also be fixed by using techniques like *Batch Normalisation*. It is a technique that helps the learning process to be more faster and stable. It standardizes the output from the first layer so that the data is centered around zero with a consistent scale. It

could also lead to faster convergence and improves learning efficiency across epochs, which will further improve the computational efficiency of the model. This output of this layer is then fed to a dropout layer, which is a regularisation technique that helps in preventing overfitting, which occurs when a model becomes too familiar with the traning data that it fails to perform well on unseen data. During training, this layer randomly 'drops out' a certain percentage of neurons, which forces the model to learn more robust features, thereby increasing generalisation to new and unseen data. A standard dropout rate of 20% was chosen as a balance between retaining enough neurons for training while also having sufficient regularisation.

The second ConvLSTM2D layer is similar to the first one, but uses a smaller kernel size of (3,3) to focus more on localized features, such as smaller cloud formations or smaller areas of intense rainfall. This could help the model to learn the more finer details in the input data and thereby increase precision. This layer also used only 64 filters since it is quite common to decrease the spatial dimensions and parameters with each layer. The output of this layer is again fed into a Batch Normalisation layer and a Dropout layer to standardise the outputs and stabilise the training process. The third ConvLSTM2D layer now makes use of 32 filters to capture the spatio-temporal patterns from the images, and this output of this layer is then fed to a Batch Normalisation layer.

The final layer is the Conv3D layer that integrates both spatial and temporal features of the data to produce the final predictions of the convective rainfall rates. The layer makes use of three dimensions from the input data - height, width and time frames. The final output are frames that show the nowcasted rainfall intensity for the next timesteps, depending on the input-output configurations used. In this layer, a *Sigmoid* activation function is used rather than ReLU since the output values range between 0 and 1, which align with the normalised CRR values. The Sigmoid function is mathematically defined as:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3.11}$$

where $\sigma(x)$ maps the input x to a value between 0 and 1, making it ideal for binary classification problems or normalized output ranges. However, it is to be noted that the final Conv3D layer was replaced by Conv2D layer for model configurations that were only predicting 1 temporal frame, thereby focusing entirely on the spatial aspect of prediction. This was done by setting the `return_sequences` parameter (which returns only the last time step's output and reducing the tensor's dimensionality to 4D) to **False** in the final ConvLSTM2D layer, since the Conv2D layer expects a 4D input tensor (batch_size, height, width, channels). When this 4D output is passed to a Conv2D layer, the layer treats it as a standard convolution operation over spatial dimensions. It does not process it across a temporal sequence, since there is not an apparent need to make use of the temporal information to predict further frames in this case.

3.2.2 Model Configurations

The computational efficiency and predictive accuracy of the model can be evaluated by using different input - output configuration tested in the ConvLSTM model mentioned in Figure 3.2. The different setups allow for an exploration on how changing the input frames and the prediction metrics impacts the models performance.

Model 1 : Predicting Frames 4 and 5 from Frames 1 to 3 (Nowcast of t+2 hours)

In this configuration, the model takes the first three consecutive frames (Frames 1, 2 and 3) as the input, having an input shape of $(None, 3, 256, 256, 1)$ and predicts the next two frames (Frames

4 and 5). Here, *None* indicates that the model will take the number of available sequences in the training dataset (for the input) and validation dataset (for the output) respectively as the total input sequences by default. The model processes each input frame through three ConvLSTM layers in parallel. This results in a total of nine layers being used, as each frame has a dedicated path of 3 full layers. The ConvLSTM layers are arranged in pairs with a total of nine boxes as seen in Figure 3.3. The three layers per input frame captures both the short-term and long-term dependancies within the data, which transforms the data shape to $(None, 3, 256, 256, 128)$ due to the use of 128 filters, thereby allowing the model to extract spatio-temporal features which focuses on both the current spatial features and the temporal changes across frames. This shape of data stays consistent across the first ConvLSTM2D, batch normalisation layer and the dropout layers, and then changes to $(None, 3, 256, 256, 64)$ after passing through the second ConvLSTM layer. These sequences are further passed through the third layer which changes the data shape to $(None, 3, 256, 256, 32)$ and finally, the extracted features are passed through the Conv3D layer, which aggregates the outputs and then produces an output sequence of shape $(None, 2, 256, 256, 1)$ with the help of a lambda function, which constitutes of the predicted two frames. Using consecutive frames will help the model to learn the immediate temporal changes that takes place in storm developments across several regions, making this configuration an effective model for predicting storm developments in the near future.

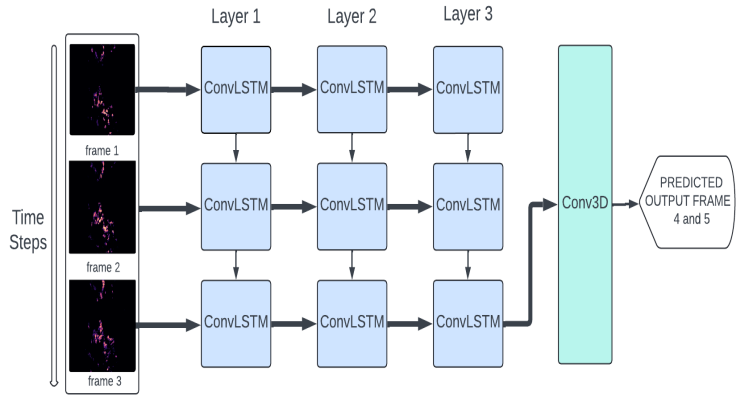

Figure 3.3: Model 1 configuration: Predicting Frames 4 and 5 from Frames 1 to 3.

Model 2 : Predicting Frame 5 from Frames 1 to 4 (Nowcast of t+1 hours)

In **Model 2**, the first four sequential frames (Frames 1-4) will be given as the input frames to the model, having a data shape of $(None, 4, 256, 256, 1)$, which provides more temporal context to the model for the next frame prediction which are then processed in parallel through a series of 3 ConvLSTM layers (Figure 3.4), where the data shape is converted to $(None, 4, 256, 256, 128)$. The operation taking place is similar to the Model 1 configuration, with the only difference being in the number of frames used as the input to predict the a different number of frames as output, as well as the conversion of the 5D tensor to 4D tensor, resulting in the output of the final ConvLSTM2D layer to maintain a shape of $(None, 256, 256, 64)$. As this output tensor is passed into the Conv2D layer, which predicts a single output frame (Frame 5), with an output data-shape of $(None, 256, 256, 1)$. This model configuration is ideal for cases where accurate long-term prediction from a larger set of inputs may be required, trading off the number of

predicted output frames for potentially greater accuracy and the addition of another input frame could help the model to capture the long-term dependancies and the gradual storm evolution patterns more effectively.

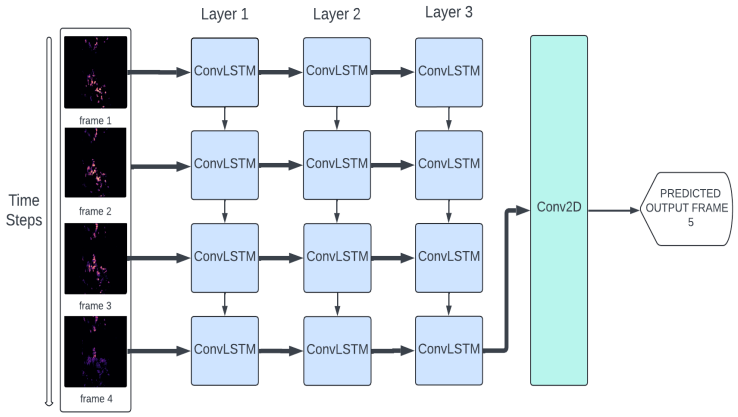

Figure 3.4: Model 2 Configuration: Predicting Frame 5 from Frames 1 to 4.

Model 3 : Predicting Frames 4 and 5 from Frames 1 and 3 (Nowcast of t+2 hours)

Model 3 takes a different approach by using non-consecutive frames (Frames 1 and 3), having an input datashape of $(None, 2, 256, 256, 1)$ which are then passed through three ConvLSTM layers as shown in Figure 3.5. Here, the data is shaped into the form $(None, 2, 256, 256, 128)$, which remains consistent throughout regularisation layers, and then changes to $(None, 2, 256, 256, 64)$ after the second layer, and further into $(None, 3, 256, 256, 32)$ after the third layer. The major difference here is the focus on capturing the significant temporal gaps between the input frames, which could help in predicting storms that show less frequent but more distinct changes. The Conv3D layer combines the outputs from these layers to predict the next two frames (Frames 4 and 5), with the data shape of $(None, 2, 256, 256, 1)$. This setup could be crucial in trying to identify or understand the sudden large temporal shifts that could occur in storm developments, which could prove to be more fatal if these convective storms are detected much later by the radar systems.

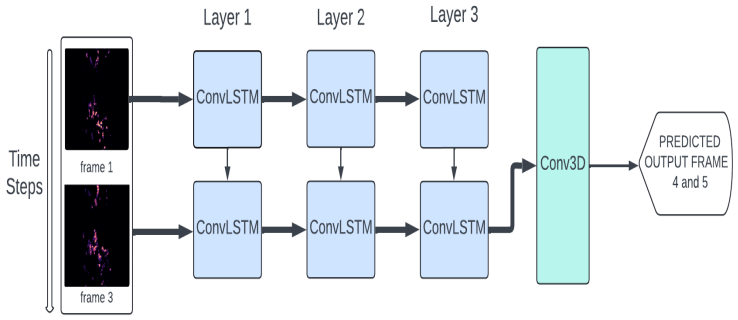

Figure 3.5: Model 3 Configuration : Predicting Frames 4 and 5 from Frames 1 and 3.

Model 4 : Predict Frame 5 from Frames 1 and 4 (Nowcast of t+1 hours)

Finally, in **Model 4**, the model takes input from two frames with a much larger temporal gap than the ones chosen in **Model 3** (Frames 1 and 4). Similar to the previous models, each input frame is passed through 3 ConvLSTM layers in parallel with the same data shapes, with the final ConvLSTM2D layers giving an output tensor of shape $(None, 256, 256, 64)$, similar to Model 2, and the extracted spatio-temporal features are fed into the Conv2D layer which will predict the next immediate frame (Frame 5) as the output (Figure 3.6). This model configuration tries to understand even longer temporal gaps between frames, which may not focus on gradual evolution of storms but rather the sudden distinct changes in the storm patterns.

These model configurations explore different ways for balancing the capture of temporal and spatial information across all input frames. By adjusting the number of input frames and the temporal spacings, each model could learn to predict and handle different types of severity of storms, from rapidly changing storms with sudden distinct changes to gradually evolving storm patterns. Each of these model configurations will have different computational requirements, which will be determined by the number of input frames given to the model and the number of frames being predicted by the model, and each can be tailored to specific needs such as short-term nowcasting (around 4-6 hours) or long-term forecasting (around 6-10 hours).

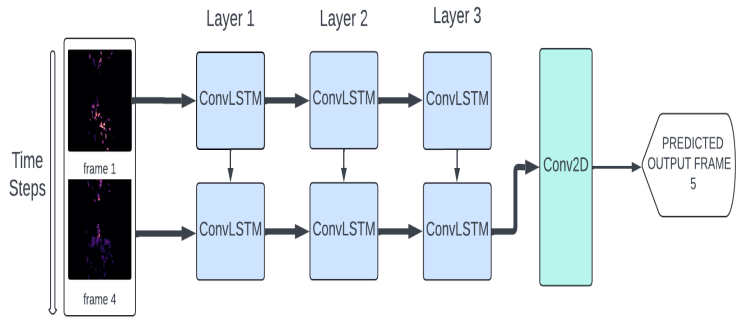

Figure 3.6: Model 4 Configuration: Predict Frame 5 from Frames 1 and 4

3.2.3 Training Parameters and Callbacks

The **ADAM** (Adaptive Moment Estimation) Optimizer is used in this model since it is able to adapt the learning rate for each parameter during training. The optimizer was designed to start with a learning rate of 0.0001 and is then further adjusted based on both the first moment (mean) and the second moment (variance) of the gradients, making it particularly effective for handling sparse gradients and noisy data. Larger learning rates could make the model to converge quickly, which could result in the models not being able to learn a variety of the underlying patterns due to the availability of scarce data. Adam's robustness ensures that the model converges efficiently even in the presence of such irregularities. The next parameter is the batch size which determines the number of batch samples each epoch will use to train the model for, and a batch size of 16 samples was chosen as a balance between computational and training efficiency. However, smaller batch sizes could be chosen if the model training process proves to be computationally intensive. Smaller batches give more frequent updates to the model which could lead to better generalisation to unseen data, but it could also cause noisier gradient estimates.

The next parameter that is crucial for the model training process is the learning rate, which determines how much the model's parameters are updated with respect to the gradient during each iteration. The *ReduceLROnPlateau* callback is designed to adjust the learning rate dynamically during training. As training progresses, models often reach a point where improvements in performance become minimal, known as a plateau. When the callback detects this plateau—usually by monitoring the validation loss—it reduces the learning rate by a predefined factor. A lower learning rate allows the model to make smaller, more precise adjustments to its weights, which can lead to improved performance and potentially help the model escape local minima. This is especially useful in the later stages of training when significant improvements are harder to achieve. If validation loss has not improved over the last patience epochs, the learning rate is reduced as follows:

$$\text{new\_learning\_rate} = \text{current\_learning\_rate} * \text{factor} \tag{3.12}$$

Another training callback that was used in this model was the *Early Stopping*, which is a technique used to prevent overfitting. During training, the model's performance is evaluated on both training and validation sets, and early stopping monitors the validation loss, which is a parameter that shows the model's performance to unseen data. If the validation loss stops improving after a certain number of epochs (patience), then the training is halted. This ensures that the model does not continue learning noise and irrelevant patterns from the training data which could make it too familiar with it, thus preventing overfitting. If `val_loss(epoch)` is the validation loss at a given epoch, early stopping checks if equation 3.13 is satisfied or not. The total number of epochs were set to 100 for this model, and if this condition holds true, then the training stops and the model can be used to evaluate on test data. Furthermore, the batch size for the training models were set to 8 for Models 1 and 2 (to avoid 'Out of Memory' issues with the GPU due to restrictions in the GPU units being used), and 16 for Models 3 and 4.

$$\text{val\_loss(epoch)} > \text{val\_loss(epoch} - \text{patience)} \tag{3.13}$$

Finally, *Mean Squared Error* (MSE) was chosen as the loss metric during the model training. MSE is commonly used for regression and prediction tasks, and it is suitable in this case since continuous pixel values of rainfall intensities are being predicted by the model. MSE is defined by equation 3.14.

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{3.14}$$

where y_i represents the actual value, $\hat{y}_i$ is the predicted value, and N is the number of pixels. As a loss function, MSE calculates the average of the squared differences between the actual values and the predicted values and their mean gives out a single value that represents the overall loss of that model for that particular epoch. The goal of the tranining process is to minimize loss, so the backpropagation algorithm makes use of the gradient of the MSE loss with respect to the models weights and biasses to update the parameters, which helps in reducing loss. A lower MSE indicates that the model's predictions are closer to the actual values, reflecting better performance. The training and validation losses are monitored during the training process to identify overfitting and ensure that the model generalizes well to unseen data. The training loss is the loss calculated on the training dataset during each epoch, while validation loss is calculated on the validation dataset during each epoch. It gives an estimate of the model's performance on unseen data and indicates how well the model is learning from the training data.

While Binary Cross Entropy was also considered as a metric that could be used for this model, it was not used since it is primarily used for binary classification tasks, such as distinguishing regions as "storm" or "no-storm", which does not align with the research motivations, since the major task is to predict the continuous values that would be produced in the next frames. The model could be accustomed to be a binary classification task, however, that would require the data to contain values that are either 0's or 1's, and this could work against the nature of the prediction task where the intensity of the storm patterns is a factor that need to be predicted as well. Considering these points, MSE was chosen as the better loss metric for the model.

3.3 Evaluation Metrics

3.3.1 Accuracy

Accuracy is one of the most basic and commonly used evaluation metrics in machine learning and predictive modelling. It calculates the percentage of correct predictions made relative to the total number of predictions (3.15), and it correlates how often a model is making correct predictions across the dataset.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \tag{3.15}$$

For the ConvLSTM model, accuracy was measured across the training and validation phases of the model. During training, accuracy helps in assessing how well the model is learning the patterns present in the training data, while validation accuracy provides insight into how well the model generalizes to unseen data, which is crucial for preventing overfitting. A gradual increase in accuracy indicates that the model is effectively learning from the data. Conversely, a stagnant or decreasing accuracy, particularly on the validation set, could be an early sign of overfitting. If both training and validation accuracy continue to improve, it suggests that the model is steadily converging towards an optimal solution. If training accuracy continues to increase while validation accuracy stagnates or decreases, the model might be overfitting. In such cases, regularization techniques like dropout or early stopping could be applied to prevent further overfitting. If both training and validation accuracy are low, the model may be underfitting, meaning it is too simple to capture the underlying patterns in the data. This might indicate a need for a more complex model or additional training epochs.

In the case of weather nowcasting, the accuracy metric might not fully capture the model's performance in predicting events such as severe storms. For this reason, accuracy should be complemented by other metrics that consider the spatial and temporal distribution of predictions, such as the Fractional Skill Score (FSS), which is more suited to evaluating the spatial structure of predictions in weather forecasting.

3.3.2 Fraction Skill Score

The Fraction Skill Score (FSS) is a specialized evaluation metric used to assess the performance of spatial predictive models, particularly useful in weather forecasting and nowcasting where accurately predicting the spatial distribution and intensity of weather phenomena like rainfall is important. Unlike traditional accuracy metrics, which focus on the overall correctness of predictions, FSS evaluates the skill of a model by comparing the predicted and observed fields over various spatial scales. This makes it a good tool for assessing how well a model predicts localized weather events, such as convective rainfall. In this study, FSS was used to evaluate the

model's predictive performance to predict convective rainfall in West Africa, which is a region characterized by complex weather patterns. By using FSS, the study was able to quantify the spatial accuracy of the model's predictions at various scales, providing a more comprehensive evaluation than accuracy alone. Evaluation methods like accuracy is able to predict how often a model gives out correct predictions, but FSS is a method that checks how well the spatial predictions are when compared to the true input frames.

The calculation of FSS involves several steps that use spatial averaging and other comparisons at various scales. Firstly, for both the observed and predicted frames, fractions are computed for a defined neighbourhood over various thresholds, where the number of pixels exceeding the threshold is noted down. For this model, a threshold of 0.5 was chosen, meaning that regions with thresholds greater than 0.5 would be considered as "rainfall" regions.

The FSS calculates the similarity over a specified window size and counts where both the predicted and actual values exceed the threshold. A higher threshold generally means that only (higher values) are considered, which reduces the number of false positives (where the prediction incorrectly surpasses the threshold) and false negatives (where the prediction fails to reach the threshold but should have), thereby leading to a higher score because it focuses on the most significant features of the data which the model might predict more accurately. So by setting a higher threshold, the FSS algorithm filters out the low-values that could be noise, which increases the emphasis on strong signals. However, in order to maintain a balanced midpoint, it is necessary to not have a really high threshold as this could lead the metric to over saturate the results, therefore, a threshold of 0.5 offers a balanced midpoint between "Non-Convective Zones" (values less than or equal to 0.5) and "Convective Zones" (values greater than 0.5).

Once these fractions are computed, the next step would be to use MSE from equation 3.14 between the observed and predicted fractions over all regions. This was iterated over different window sizes (3x3, 5x5 and 10x10), that allows the assessment of spatial accuracy over different scales, which helps quantify the difference between the spatial patterns of rainfall between the observed and predicted data. This value is then normalized against a "worst-case MSE" which is typically a random or null forecast). Normalisation ensures that the evaluation method offers a relative scoring, where FSS values ranges between 0 and 1. The normalised MSE is then used to compute FSS using equation 3.16, where a value of 1 indicates a perfect forecast and a value of 0 indicates the worst forecast.

$$\text{FSS} = 1 - \frac{\text{MSE of worst forecast}}{\text{MSE of fractions}} \tag{3.16}$$

FSS offers several advantages when it comes to evaluating weather nowcasts. This metric accounts for the spatial distribution of the convective clouds in the predicted frames, while also assessing it over different spatial scales. This helps in understanding how the model is able to predict both localised short-term storms as well as large-scale storms. Moreover, unlike pixel-wise metrics like accuracy and MSE, FSS is more robust to slight spatial changes. For example, if the model predicts the storm to be slightly off from its observed location, then FSS would penalize this less severely since the model has captured the overall event correctly. Overall, FSS is a critial evaluation metric that when used in complement with other metrics such as accuracy, provides adequate information about the model performance and its spatial predictive capabilities, which is essential for nowcasting predictions.

3.3.3 ROC curve

The Receiver Operating Characteristic (ROC) curve is a graphical plot used to evaluate the performance of a binary classification model. Since the model being used here is a continuous value prediction model that is used to predict the next frames, this metric can be used to evaluate the model by defining a threshold and assigning points having values greater than the threshold as 1 and the rest as 0. Here, the mean of the prediction pixel intensities was chosen as the threshold.

The ROC curve shows the trade-off between the True Positive Rate (TPR) and the False Positive Rate (FPR) across different threshold values. The True Positive Rate, also known as sensitivity or recall, represents the proportion of actual positive cases that the model correctly identifies, while the False Positive Rate indicates the proportion of negative cases incorrectly classified as positive. Here, positive cases are considered as "Convective Zones" and negative cases are considered as "Non-Convective Zones". Mathematically, the True Positive Rate and False Positive Rate are defined as:

$$\text{TPR} = \frac{TP}{TP + FN} \tag{3.17}$$

$$\text{FPR} = \frac{FP}{FP + TN} \tag{3.18}$$

where TP is True Positives, FN is False, Negatives FP is False Positives and TN is True Negatives. The Area Under the ROC Curve (AUC-ROC) is a single scalar value that summarizes the overall performance of the model. An AUC-ROC score of 1 indicates perfect classification, while a score of 0.5 suggests that the model's predictions are no better than random chance.

Chapter 4

Results and Discussion

In this chapter, the performance of the four ConvLSTM models developed in this study for nowcasting Convective Rainfall Rates (CRR) over West Africa will be evaluated and discussed. The chapter is organized as follows: Each section will discuss the performance of the models, beginning with Model 1. For each model, the training and validation loss and accuracy will be discussed to see how well the model has learned from the data and generalized to unseen samples. The prediction plots will provide visual insights into the model's predictive capabilities, while the testing accuracy and loss metrics are methods that show the model's performance on new data. Finally, the FSS will be used to evaluate the spatial accuracy of the model's predictions, which is critical in weather forecasting.

The results presented in this chapter will help in the comparative analysis in upcoming sections, where the performance of all four models will be compared, both against each other and against existing models in the literature (CNN and U-Net). This analysis will help identify the most effective model architecture and configuration for nowcasting in the context of West African weather systems.

4.1 Model 1 Results

Model 1 was designed to predict the convective rainfall rates using three input frames to predict two output frames for the nowcasting prediciton in the region of West Africa. The model was trained over a total of 40 epochs for a total runtime of 6 hours, 18 minutes, and 33 seconds.

4.1.1 Training and Validation Loss

During the training process, Model 1 showed a significant decline in loss over the intial epochs, which demonstrated the model's increasing ability to learn the underlying patterns in the dataset. The final training loss was 0.000765, which indicates a model that fits the training data accurately. The loss curve stabilized as the training progressed, which is a typical behavior for models trained with the Adam optimizer. The validation loss also closely followed the training loss, indicating minimal overfitting. The final validation loss was 0.000780, slightly higher than the training loss, but still low enough to suggest that the model generalized well to the validation data. This close alignment between training and validation losses indicates that the model has learned relevant features without memorizing the training data, which could lead to poor generalization.

The training and validation loss curves, as shown in Figure 4.1, demonstrate the model's convergence. The model was trained for 40 epochs, and the curves are smooth with only one

Figure 4.1: Plot showing the Training and Validation Loss for Model 1.

significant spikes at around epoch 6 or 7. The sharp decline in loss values suggests that the model was effective in minimizing the error between the predicted and actual values during the early stages of training. However, after the significant spike, the training loss curve quickly converges with the validation loss curve over the next few epochs, which further supports the stability of the training process. The small difference between the training and validation losses at convergence highlights the model's robustness.

4.1.2 Training and Validation Accuracy

Model 1 achieved a final training accuracy of 0.969864, which means that approximately 96.98% of the training data points were correctly predicted by the model, i.e., the model was able to predict the pixel intensity and its location almost perfectly. The accuracy improved steadily over the epochs, which aligns with the reduction in training loss, and this indicates the model's capability in learning the patterns in the dataset. The final validation accuracy was recorded at 0.968092, which is very close to the training accuracy. This suggests that the model has generalized well to unseen data. The small gap between the training and validation accuracy further indicates that the model is not overfitting, which is essential for ensuring that the model's predictions remain reliable when applied to new data.

Figure 4.2 shows the training and validation accuracy curves obtained by training Model 1, and we can see that the training accuracy increases greatly during the first 5 epochs and then stagnates, meaning it has learned efficiently from the data, however, the validation accuracy has stagnated quite early with minimal improvement over the epochs, which could be due to the validation set not being challenging enough due to the limited availability of input data frames.

4.1.3 Intepretation of Loss and Accuracy

The decline and convergence in both the loss metrics along with the high accuracy rates throughout the training process show the model's strong performance. The final metrics suggest that the model was able to minimize the Mean Squared Error (MSE) across the training and validation datasets, which is important to reduce for predicting continuous values like rainfall intensities. The consistency between the training and validation metrics shows that the model's is able to generalize with the learned storm patterns, and also reducing the risk of overfitting. However,

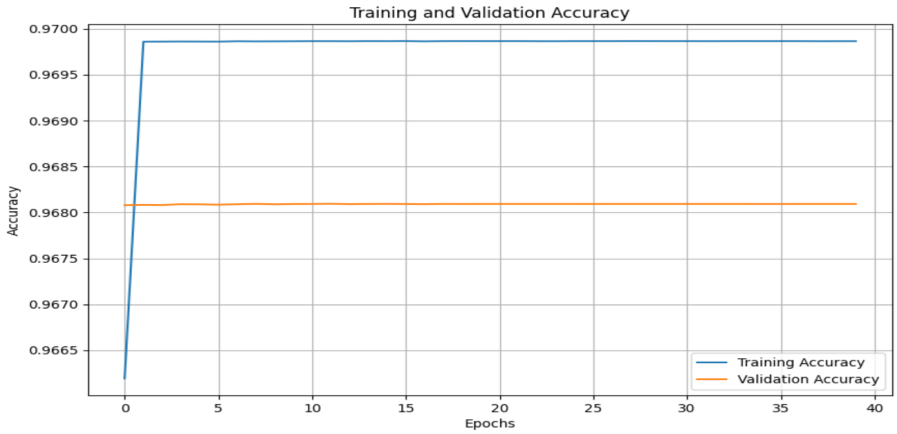

Figure 4.2: Plot showing the Training and Validation Accuracy for Model 1.

the very low loss and very high accuracy could be due to the large amount of 0's in the dataset, as was seen in 2.3. The model was able to almost perfectly capture the pixels with 0 intensity, which showed areas of close to zero storm activity, which is a good sign. Moreover, an accuracy value greater than 90% could also mean the model is able to accurately predict the regions with possible storm activity, which was the expected outcome from training the ConvLSTM models.

The initial high value of the validation accuracy could be due to the initial batches of the validation dataset being similar to the training dataset, thus resulting in the model performing well on these samples. As a result, the model might make simpler predictions initially, and as the model complexity increases during training (as it learns more complex patterns), the validation accuracy might stabilize or even decrease slightly before improving again. Another reason could be the use of 3 Batch Normalisation layers. When a model starts training, the weights are initialized randomly, and this randomness can cause variations in the activations of neurons, leading to unstable and inconsistent gradients during backpropagation. Batch normalization helps mitigate this by normalizing the activations (or outputs) of a layer to a standard distribution (usually with mean 0 and variance 1) before passing them to the next layer, increasing the performance on the validation set as a result.

The choice of the Adam optimizer could have been the reason for the quick convergence of the curves. The learning rate adjustments by the ReduceLROnPlateau callback, allowed the model to tune its learning process as the validation loss plateaued or stagnated past a point, helping to avoid overshooting the minimum loss and ensuring that the model did not get stuck in suboptimal minima, rather than the global minima, which is the optimal point that a model is expected to reach after backpropagating through the epochs. Moreover, the batch size of 8 was able to give a balanced between computational efficiency and the ability to capture diverse patterns in the data, however, it was chosen due to GPU restrictions that led to OOM errors. A larger batch size could have been more effective in reducing the noise in the updation process of the ConvLSTM model.

4.1.4 Testing Accuracy and Loss

Model 1 was tested on the unseen test data of shape $(304, 5, 256, 256, 1)$ to evaluate its generalisation performance. The model produced an accuracy of 0.9730, which is very close to the training and validation accuracies. This shows that the model is able to learn adequately with the

training data as well as predict almost accurately with unseen data as well. Moreover, the final testing loss was found to be 0.007, which was lower than the validation losses calculated during training. This means that the model has performed slightly better on the test dataset compared to the validation dataset, which could stem from the randomness used in splitting the input data into validation and test datasets.

4.2 Model 2 Results

Model 2 was designed similarly to Model 1 using a continuous sequence of input frames, with the difference being the use of 4 input frames to predict the final frame 5 for nowcasting convective rainfall rates in West Africa. The model ran for a total run time of 6 hours, 21 minutes, and 1 second over 30 epochs.

4.2.1 Training and Validation Loss

During the training of Model 2, the loss decreased significantly within the initial epochs, which indicates the model's good learning progression. The final training loss was 0.001345, showing the model's good fit to the training data, though not as low as Model 1. The validation loss stabilized at 0.001426, slightly higher than the training loss, suggesting that the model has been able to generalize well to the validation dataset, but with a slight possibility of a minimal overfitting, which is quite common while training deep learning models.

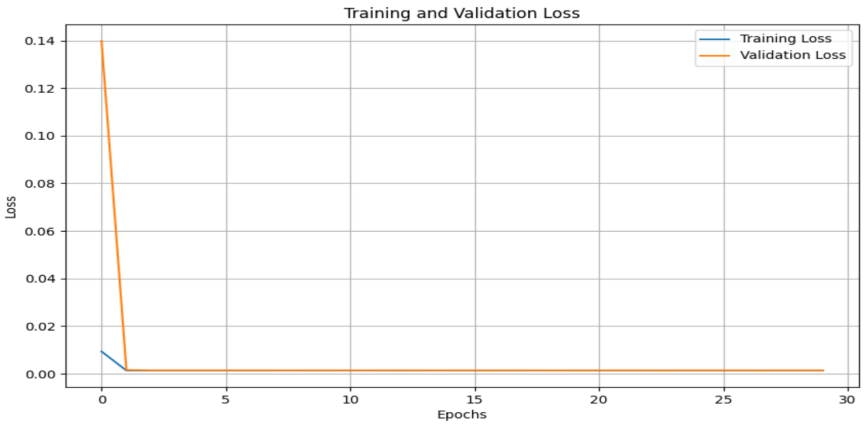

Figure 4.3: Plot showing the Training and Validation Loss for Model 2.

As seen in Figure 4.3, the training and validation loss curves demonstrate the model's convergence. The sharp decline in the early epochs around the first 5 epochs shows the model has learned effectively. The loss plateaued around epoch 20, which suggests that the model had learned most of the significant features by that point, and further epochs did not bring about any significant changes to the loss metric, indicating that the model was able to reach the optimal minima as needed.

4.2.2 Training and Validation Accuracy

Model 2 achieved a final training accuracy of 0.969865, meaning approximately 96.99% of the training data points were correctly predicted. The validation accuracy at the end of the training

was slightly lower at 0.967788, but this minimal gap between training and validation accuracy indicates good generalization. Since the difference between the training and validation accuracy is just a factor of 0.0022, it is not large enough to be indicative of overfitting when compared to Model 1 which had a difference of 0.0018.

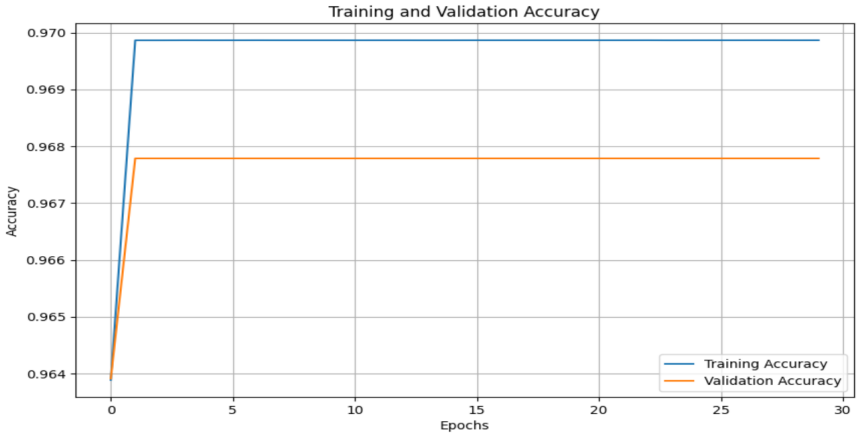

Figure 4.4: Plot showing Training and Validation Accuracy for Model 2.

In Figure 4.4, the training and validation accuracy curves are shown. The training accuracy quickly increased within the first few epochs, similar to Model 1, and then stabilizes. The validation accuracy also increases but does so more gradually, plateauing after around 4 or 6 epochs. This shows that Model 2 found the validation dataset to be more challenging when compared to Model 1, which could be due to the use of more input frames, which in turn led the model to generalise much slower to the validation dataset. However, the model could maybe benefit from the use of more Regularisation techniques.

4.2.3 Intepretation of Loss and Accuracy

The performance of Model 2, as shown by its loss and accuracy metrics, suggests that the model was effective in learning from the training data. However, the slight increase in validation loss compared to the training loss could be due to the complexity of the ConvLSTM layers in capturing the temporal and spatial features from the data. Moreover, similar to Model 1, the high accuracy and low loss could be due to the way the satellite-image based dataset is being presented with majority of the values being close to 0, which could be an expected pattern from the remaining two model configurations as well. Additionally, it can be noted that the validation accuracy starts from 0 for Model 2, which is significantly different from the performance of Model 1. This could be due to the use of 4 input frames which could have caused the model to make poor initial predicitons, or the combination of a small batch size of 8 and a learning rate of 0.0001 might have been too low for the model to be able to learn the complex patterns from four input frames. However, the validation accuracy was able to improve which indicates the model's predictive performance.

Furthermore, the use of the Adam optimizer and learning rate callback helped the model to avoid local minima and achieve near-optimal convergence. The slightly higher validation loss might suggest that Model 2 could use more regularization techniques, such as dropout or batch normalization, to further enhance its generalization and predictive abilities.

4.2.4 Testing Accuracy and Loss

When tested on unseen data, Model 2 achieved a test accuracy of 0.9726 and a test loss of 0.0012. These results are close to the training and validation metrics, showing that Model 2 has generalized well to new data. The small difference between the validation and test losses indicates that the model performed consistently across different datasets, further confirming its robustness and reliability in predicting rainfall intensities. Furthermore, the slight difference in test loss compared to validation loss could be attributed to the inherent variability in the test set or the model's sensitivity to certain features not present in the training or validation data. Overall, Model 2 has shown to be a competent model, though slightly less effective than Model 1, with potential areas for improvement in its handling of validation data.

4.3 Model 3 Results

Model 3 was designed to predict convective rainfall rates in West Africa using frames 1 and 3 as the input to predict frames 4 and 5. The expected predicted frames are similar to that of Model 1, with a difference being in the use of a larger time gap between the input, allowing the model to learn more sudden changes. The model was trained over a total of 63 epochs with a batch size of 16 for a total runtime of 5 hours, 20 minutes, and 33 seconds.

4.3.1 Training and Validation Loss

During the training of Model 3, the loss metrics showed a sharp decline during the initial epochs, indicating fast learning by the model. The final training loss was 0.000968, which indicates that the model fits the training data accurately. The validation loss, similarly, exhibited a smooth decline, stabilizing at 0.000945 by the end of the training. This close values between training and validation losses suggests that Model 3 generalized well to the validation data without overfitting.

Figure 4.5: Plot showing Training and Validation Loss for Model 3.

The training and validation loss curves in Figure 4.5 show a smooth convergence with no significant spikes after the initial few epochs. This suggests that the learning rate adjustments that was made through the ReduceLROnPlateau callback and the Adam optimizer were effective

in tuning the model's learning process. The lack of unnecessary fluctuations in the loss curves shows that the model did not encounter any major issues like vanishing or exploding gradients.

4.3.2 Training and Validation Accuracy

Model 3 achieved a final training accuracy of 0.969856, which means approximately 96.98% of the training data points were predicted correctly by the model. The validation accuracy plateaued at 0.968084, which was very close to the training accuracy. This showed that the model did not overfit and performed well on unseen data.

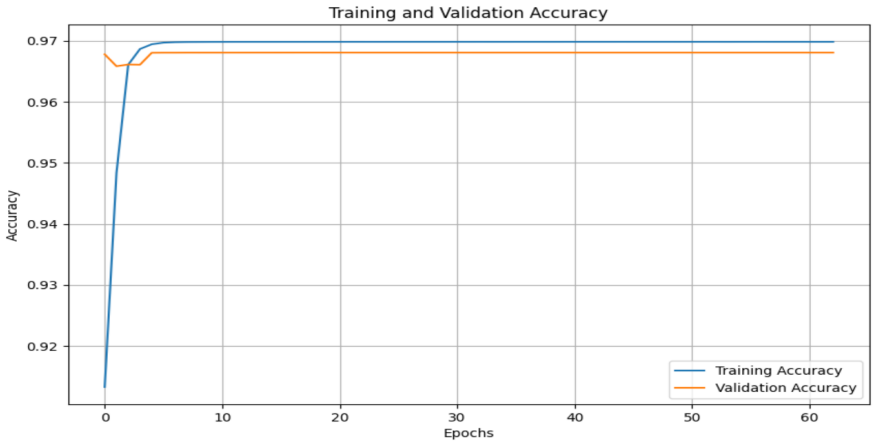

Figure 4.6: Plot showing Training and Validation Accuracy for Model 3.

As shown in Figure 4.6, the training accuracy increased significantly during the initial epochs before stabilizing. The validation accuracy closely followed the training accuracy which shows the model's ability to generalize from the training data. The early fluctuation in the validation accuracy, after about 10 epochs, suggests that the model might have quickly learned the underlying patterns of the data after an initial decline, but had limited further improvement, possibly due to the nature of the validation set or the model's architecture limitations.

4.3.3 Intepretation of Loss and Accuracy

The smooth convergence of both the training and validation loss curves, along with the high accuracy metrics, suggests that Model 3 is robust and effective in minimizing prediction errors. The final metrics show that the model was able to accurately predict the continuous values of rainfall intensities with precision. As mentioned earlier, the high accuracy could be due to the nature of the dataset, which further instates the need for other evaluation metrics such as FSS, which would be discussed in the upcoming sections. In theory, Model 3's methodology is very similar to that of Model 1, since both the models try to predict 2 output frames. This could be one of the reasons why their performances are almost similar, with Model 1 performing slightly better, which could be due to the use of continuous time frames as opposed to the gap in the temporal aspect of the input frames used in Model 3. Additionally, it can be noted that the model was able to achieve a very high initial validation accuracy, which could be due to the model being able to learn adequately with the learning rate of 0.0001 and the batch size of 16

Furthermore, Model 3's performance is likely due to the balanced complexity and efficiency of the model architecture, as well as the Adam optimizer, which helped in adjusting learning

rates dynamically. The learning rate schedule provided by ReduceLROnPlateau further helped in fine-tuning the model's parameters, preventing overfitting while ensuring optimal convergence. The batch size, along with these optimizations, helped in the model's ability to capture complex spatio-temporal patterns in the data without overfitting.

4.3.4 Testing Accuracy and Loss

Model 3 was tested on unseen test data, giving out a test accuracy of 0.9729 and a test loss of 0.0008. These results are consistent with the validation metrics, indicating that the model performed well on new data. The slight improvement in the test loss compared to the validation loss could mean that the model may have benefitted from the random distribution of data during testing, which could have included easier-to-predict samples or avoided outliers present in the validation set. Overall, Model 3 has a strong ability to generalize and almost accurately predict rainfall intensities.

4.4 Model 4 Results

Model 4 was designed to predict convective rainfall rates by using an even larger temporal gap than the one used in Model 3. Here, frames 1 and 4 were used to predict the final frame 5, by making use of the same ConvLSTM model architectures as used in the previous models. The training process for this model was fairly quick, completing in 4 hours, 19 minutes, and 16 seconds. The model was trained over a total of 50 epochs and showed a good performance across various metrics.

4.4.1 Training and Validation Loss

During training, Model 4 showed consistent learning progress with a steady decline in loss values. The final training loss was 0.001358, which indicates the model's good fit to the training data. The final validation loss was at 0.001437, closely aligning with the training loss, which means the model was able to generalize well to the unseen data. This slight difference between training and validation loss points to a model does not overfit, while also being able to generalize across unseen data.

Figure 4.7: Plot showing Training and Validation Loss for Model 4.

Figure 4.7 illustrates the training and validation loss curves, which show a rapid convergence early in the training process at around epoch 7 or 8, after which the loss values stabilized and showed little improvement. This stability across epochs suggests that the model was able to learn effectively from the data, minimizing the error in predictions with increasing epochs. The lack of fluctuations in the validation loss curve further supports the model not overfitting to the training data.

4.4.2 Training and Validation Accuracy

Model 4 achieved a final training accuracy of 0.969865, indicating that the model almost accurately predicted the outcomes for nearly 97% of the training data points. The final validation accuracy was slightly lower, at 0.967788, but this close alignment with the training accuracy shows that the model has strong generalization capability. The high accuracy in both training and validation datasets also highlights the effectiveness of the model in capturing the underlying features needed for accurate convective rainfall nowcasting.

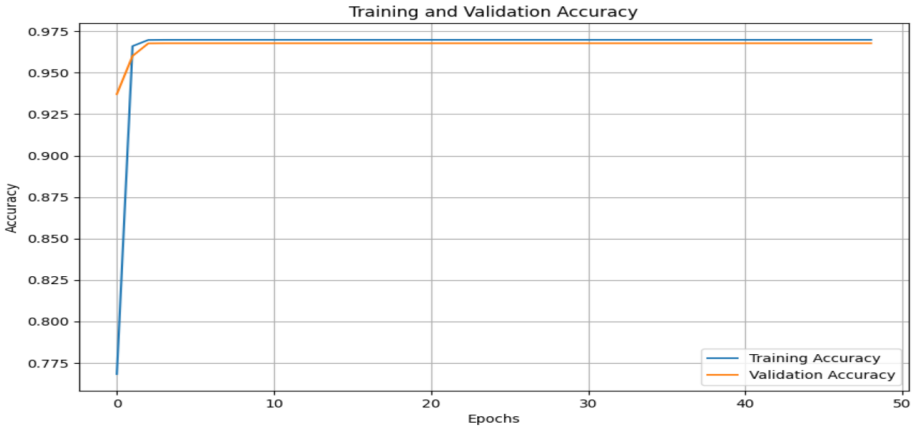

Figure 4.8: Plot showing Training and Validation Accuracy for Model 4.

As shown in Figure 4.8, the accuracy curves display an initial rapid increase, followed by a plateau as the model reaches its optimal performance. The validation accuracy curve closely follows the training accuracy, which further shows the model's ability to produce high accuracy predictions without overfitting to the training data.

4.4.3 Interpretation of Loss and Accuracy

The quick convergence in both loss and accuracy metrics throughout the training process indicates that Model 4 was able to effectively minimize prediction errors and generalize to unseen data. The consistent low validation loss suggests that the model was capable of handling the complexities of the data without overfitting. The use of regularisation techniques like dropouts and batch normalisation, paired with the tuned learning rate could have ensured that the model could adjust its learning process dynamically.

4.4.4 Testing and Accuracy Loss

Model 4 was evaluated on an unseen test dataset, resulting in a test accuracy of 0.9726 and a test loss of 0.0018. These metrics are consistent with the validation results, suggesting that the

model was able to generalize well beyond the training data. The test loss being slightly higher than the validation loss is within an acceptable range, indicating that the model maintained its predictive accuracy on new data. Overall, Model 4 showed strong performance metrics, making it a reliable choice for operational nowcasting tasks. The stable training process, coupled with high accuracy and low loss across training, validation, and test datasets, underscores the robustness of the model architecture and its suitability for deployment in real-world weather prediction scenarios.

4.5 Prediction Plots

Prediction plots give a visual representation of the model's performance by comparing the predicted CRR maps with the true maps. The ground truth samples that were randomly chosen from the unseen test data are shown in 4.9 and 4.10, and these provide a reference against which the predicted frames will be compared. The following analysis will discuss the model's accuracy in predicting these frames, focusing on the spatial distribution and intensity of the rainfall as captured in the prediction plots.

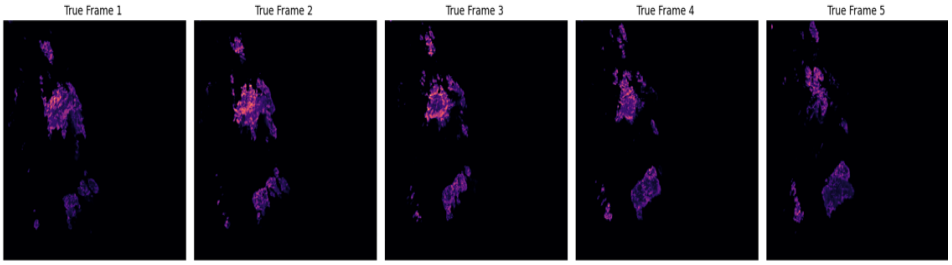

Figure 4.9: True Frames of the actual Convective Rainfall Maps used for comparing with predicted frames (Sample 1).

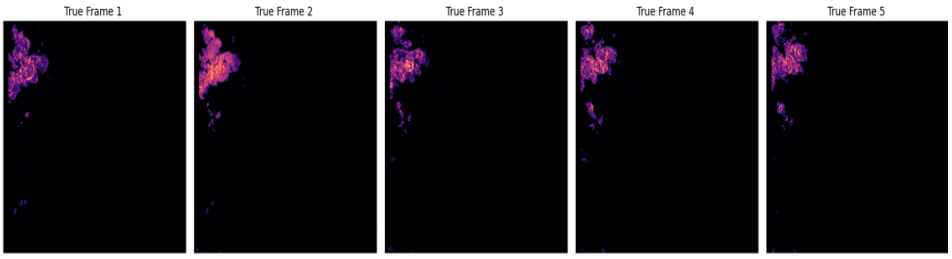

Figure 4.10: True Frames of the actual Convective Rainfall Maps used for comparing with predicted frames (Sample 2).

4.5.1 Model 1 Prediction Plots

The predictive performance of Model 1 in predicting the convective rainfall patterns in West Africa could be visually compared by comparing the predicted samples from Figures 4.11 and 4.12 with the true sample frames from Figures 4.9 and 4.10 respectively. In the given plots, it can be observed that Model 1 has been able to have a relatively good spatial accuracy, almost correctly predicting the location of the rainfall structures. However, in Figure 4.11, it can be

seen that some regions are not being covered, which could be due to the sudden increase in the rainfall areas from frame 3 to frame 4 in the true sample in Figure 4.9. Overall, the predicted frames have successfully identified general regions where rainfall is expected, indicating that the model is effective at locating the primary areas of rainfall. The smoothing effect of the predicted frames could also be due to the use of Mean Squared Error (MSE) loss function, which tends to average out extreme values, leading to a slightly less sharp representation of high-intensity rainfall regions, where the edges are smoothened out, and the intensity is slightly higher than the expected pixel intensities.

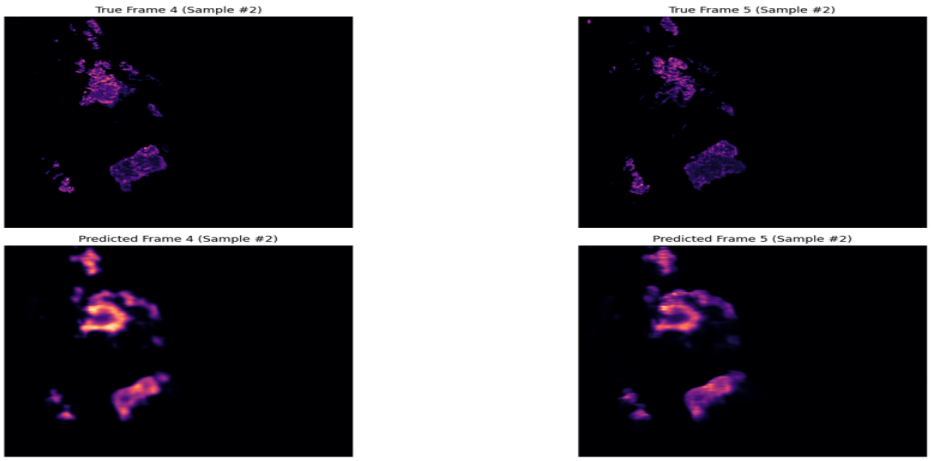

Figure 4.11: Predicted vs. actual convective rainfall maps for Model 1 (Sample 1)

In terms of the predicted intensity of the pixels, which corresponds to rainfall rate in the region, the predicted frames show some variability when compared to the ground truth samples, with some being dispersed or somewhat smoothed intensities, particularly noticeable in Figure 4.12. However, the model was almost able to pick up on the general regions where more concentrated rainfall is to be expected, which indicates that the model captures the overall distribution of rainfall, but may struggle with accurately predicting high-intensity rainfall events.

Additionally, it can be observed that the predicted frames show some temporal consistency in terms of the movement and development of rainfall structures, which is important for nowcasting applications. However, small discrepancies in the rate of movement or development of these structures might be present. The limitations as mentioned above could be avoided by possibly using a larger dataset with more temporal frames, allowing the model to learn more complex, subtle and sudden changes in the convective rainfall patterns, which could better represent extreme weather conditions. Overall, the model performs well in identifying the general area of convective rainfall, but further refinements may be necessary to enhance its accuracy in predicting high-intensity events.

4.5.2 Model 2 Prediction Plots

The predictive performance for Model 2 has been visually represented using the Figures 4.13 and 4.14. The true and predicted frames show that Model 2 had almost accurately predicted the convective storm pattern on Frame 5, and was able to capture the increase in the affected rainfall regions with minimal spatial misalignment, particularly in Figure 4.13. The usage of four input frames had clearly helped the model in trying to learn more underlying patterns, hence producing a more spatially accurate result. Moreover, in Figure 4.14, it can be observed that the

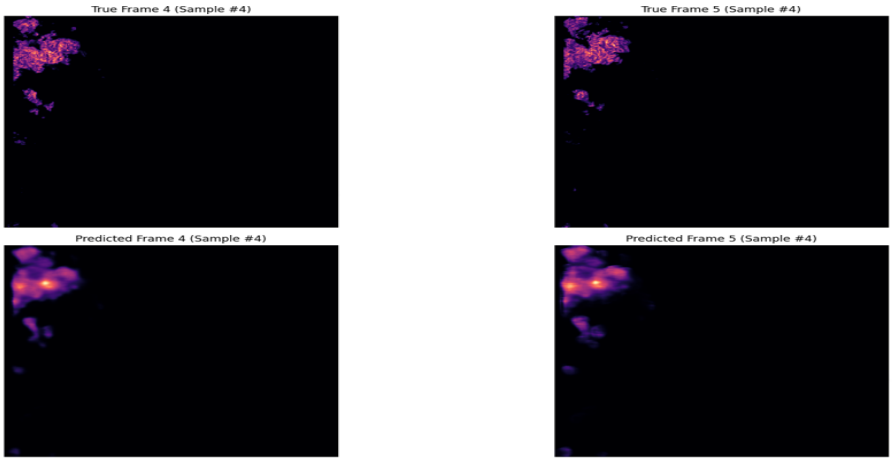

Figure 4.12: Predicted vs. actual convective rainfall maps for Model 1 (Sample 2)

model was able to capture the smaller affected regions around the bottom of the frame, thereby predicting the convective cells' general locations but with slight shifts that led to deviations from the actual storm's position. Although compared to Model 1, which had close to zero shift in the for the same sample in the predicted frames, Model 2 was able to predict the gradual changes in the storm patterns more efficiently.

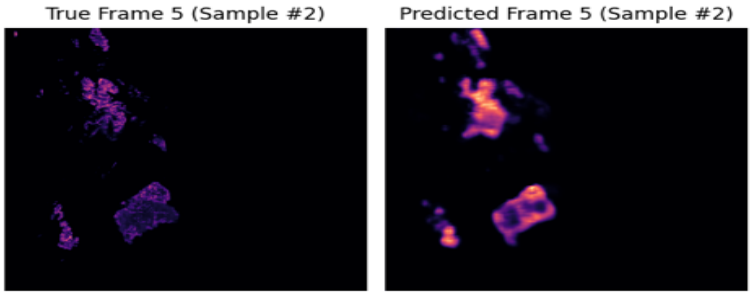

Figure 4.13: Predicted vs. actual convective rainfall maps for Model 2 (Sample 1)

Furthermore, while the model had captured the general shape and structure of these convective systems, it however overestimated the intensity of the convective storms in several regions, which was a characteristic similar to Model 1. In Figure 4.13, the model had clearly expected the storm to evolve in a more intense manner, however the ground truth frames indicate the storms diminishing over time, which could mean the model may find it challenging to predict de-evolving storm patterns over temporal frames.

Overall, Model 2 shows a strong ability to predict convective rainfall, with success in capturing large-scale patterns. However, the observed spatial inaccuracies and intensity variations could suggest the further architecture adjustments to the model, which could help improve its precision. The model could benefit and possibly learn more about the evolution of storm patterns through the use of several temporal frames. Moreover, similar to Model 1, the boundaries and internal structures appear less distinct and more smoothened in the predictions, which could indicate the model struggles to capture the very fine details in the dataset.

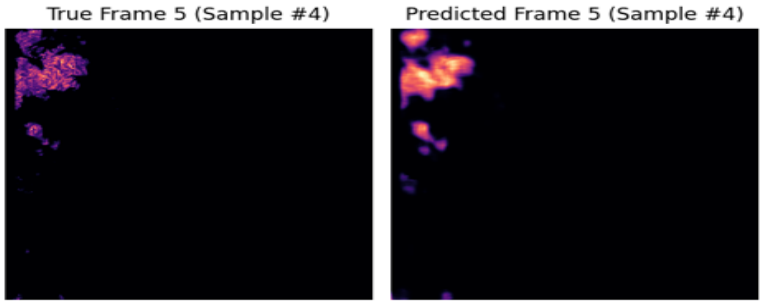

Figure 4.14: Predicted vs. actual convective rainfall maps for Model 2 (Sample 2).

4.5.3 Model 3 Prediction Plots

Model 3's predictive performance was analysed using the plots shown in Figures 4.15 and 4.16, that show us how the model was able to capture the spatio-temporal patterns from the input data. At first glance, it can be observed that the predictions for model 3 closely resemble that from model 1, with some minor changes in the predicted pixel intensities. In both the predictive plots, the model was able to identify the major regions of convective rainfall, where the general structure of the storms were captured. Furthermore, the model had also smoothened out high-intensity areas, which resulted in a loss of sharpness and precision in predicting the exact locations of high-intensity rainfall. These differences in the finer details, where the model either overestimated or slightly misplaced certain areas. This could indicate that while the model has a good learning of large-scale patterns, it struggles with accurately predicting the smaller-scale features, similar to Model 1.

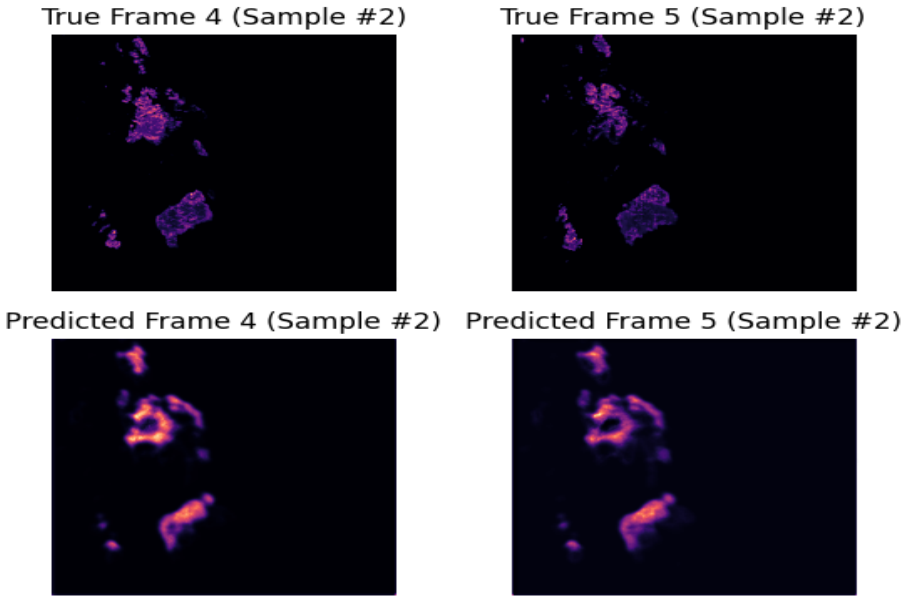

Figure 4.15: Predicted vs. actual convective rainfall maps for Model 3 (Sample 1).

In 4.15, it can also be seen that the model has severly underestimated some regions, where the model predicted some regions of moderate intensity as 0. This could be due to the temporal gap in the input frames. Although it was less computationally intensive than Models 1 and 2

due to the use of lesser frames, the model could be struggling to learn some of the less intense rainfall affected regions. Overall, the model was capable of learning the majority of the patterns associated with convective storms from the dataset, and this technique of using larger temporal gaps within the input frames could benefit from the use of several more frames to help the model learn the underlying patterns of storm evolution more efficiently.

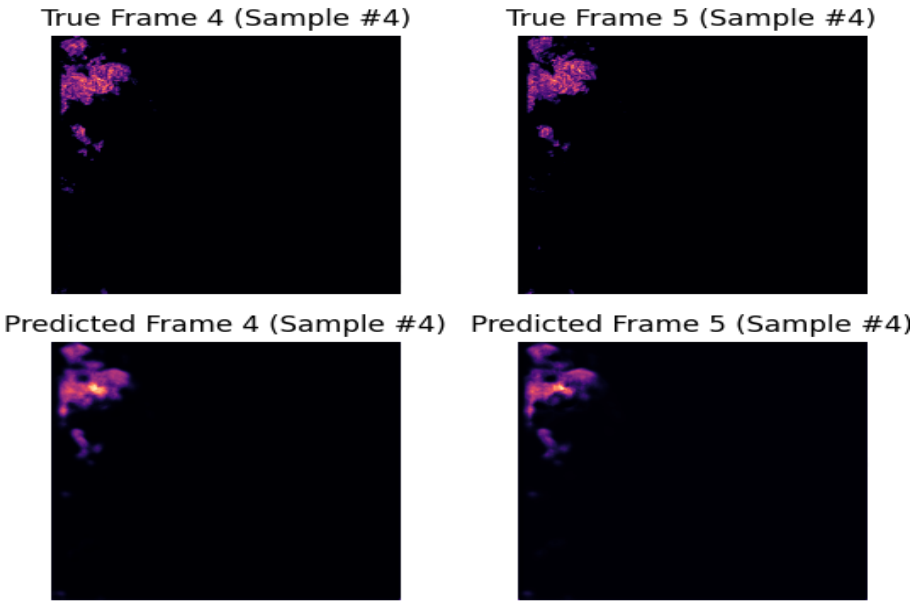

Figure 4.16: Predicted vs. actual convective rainfall maps for Model 3 (Sample 2)

4.5.4 Model 4 Prediction Plots

The predictive plots for Model 4 were visualised in Figures 4.17 and 4.18, where it can be seen that the model was successful in capturing the spatial characteristics of the storm patterns accurately, where the predicted frames were closely aligned with ground truth frames with respect to the general shape and location of the convective storms. The predicted frames were also able to capture the pixel intensities almost accurately, which has shown improvements when compared to the predictive plots of Model 2 in 4.13 and 4.14. Although, the pixel intensities have been overestimated in a lot of regions, particularly in the corners or the boundaries, the model was able to almost accurately match the spatial extent of these regions.

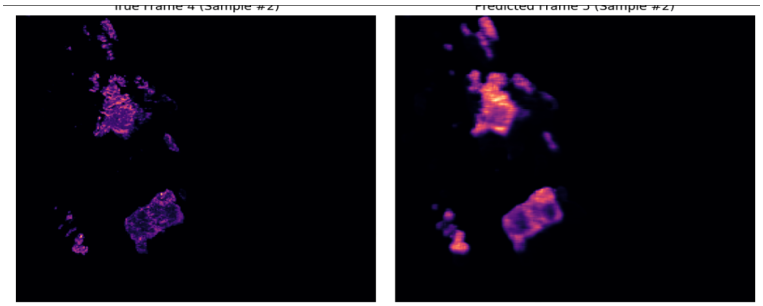

Figure 4.17: Predicted vs. actual convective rainfall maps for Model 4 (Sample 1).

The overall performance of Model 4, as indicated by these prediction plots, suggests that while the model can capture the spatial distribution and intensity of convective rainfall events, it is not able to fully replicate the fine details of the patterns. The tendency to over-predict rainfall intensity shows potential areas for model refinement, such as using additional training data, applying regularization techniques, or adjusting the model architecture to better capture subtle variations in rainfall patterns.

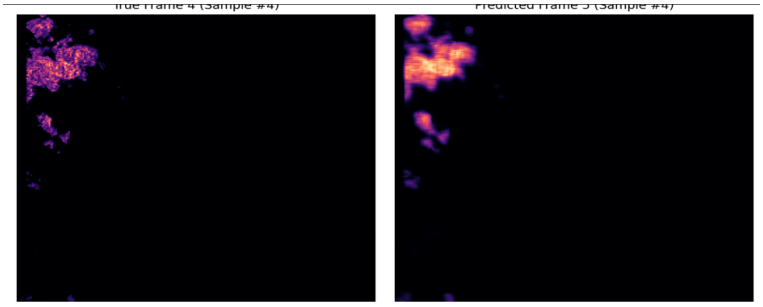

Figure 4.18: Predicted vs. actual convective rainfall maps for Model 4 (Sample 2).

4.6 Discussion of Comparitive Analysis

4.6.1 Comparison of FSS scores amongst each Model

The Fractional Skill Score (FSS) results for each model across various window sizes of 3, 5 and 10, comparing their ability to predict CRR has been summarised in 4.1. These scores reflect the models' performance in predicting frames 4 and 5 for Models 1 and 3, and predicting frame 5 for Models 2 and 4.

Model	Frame	Window Size 3	Window Size 5	Window Size 10
Model 1	4	0.5119	0.5234	0.5524
	5	0.3466	0.3485	0.3581
Model 2	5	0.5216	0.5401	0.5782
Model 3	4	0.3604	0.4729	0.5732
	5	0.3494	0.4610	0.5556
Model 4	5	0.4921	0.5180	0.5608

Table 4.1: Performance of different models with varying window sizes.

Model 1 showed moderate FSS values, with a some improvement as the window size increased. Model 1 generally performs better for Frame 4 compared to Frame 5. It can be noted that the FSS increases with larger window sizes, which suggests that Model 1 is more efficient at predicting broader spatial features rather than fine-grained details, which is an expected behaviour at larger scales. The relatively low FSS for Frame 5 indicates that the model struggles with predicting the next time step accurately, which could be due to the cumulative error propagation in sequential prediction. Model 2 on the other hand showed higher FSS values compared to Model 1 across all window sizes, which could be due to the use of 4 continuous input frames that could have allowed the model to learn more patterns from the data. Also, it can be observed that the FSS values increased as the window size increased as well. These scores indicate that Model 2 was more effective in capturing both finer-detailed patterns and broader-scale features

than Model 1. The higher scores could be due to more effective training that allowed the model to learn underlying patterns.

Next, Model 3 showed a balanced result, with FSS values that were consistent between both predicted frames 4 and 5 across different window sizes, which was a characteristic Model 1 failed to achieve, since it can be noticed that Model 1 produced very low scores for the predicted frame 5. The consistency of FSS values between the predicted frames suggests that Model 3 is able to both capture broader-scale details, as well as maintain accuracy at finer details as well. The constancy of FSS scores between frames show Model 3 to be a well-generalized model, where the architecture and training process were able to effectively manage the complexity of CRR prediction without overfitting to small-scale patterns.

Model 4 achieved the high FSS scores overall across all window sized, however, the model was not able to capture the small-scale details as effectively as Model 2. Model 4 was able to capture both fine-scale details and broader spatial features more effectively than Model 1 and was better at predicting frame 5 than Model 3, and the steady increase in FSS values with increasing window sizes suggest that Model 4 maintains a good balance between detailed and generalized predictions.

The FSS scores for all models generally increase as the window size increases. This trend is expected because smaller window sizes are more sensitive to fine spatial details, whereas larger window sizes concentrate on broader patterns and can negate or look over the errors in smaller details. At window size 3, the models are evaluated based on their ability to predict very localized rainfall patterns. Models 1 and 2 show strong performance here, which suggests they are effective at capturing detailed spatial structures. The next window size of 5 evaluates a model's ability to balance fine detail with some level of spatial generalization. Model 2 shows the best performance here, suggesting it can generalize predictions better than the other models while still capturing important spatial details. Finally, at window size 10, the FSS assesses how well models can predict more generalized rainfall patterns over larger regions. Here, Model 2 produced the best results than the other 3 models. Overall, it can be observed that Model 2 was able to produce the best FSS scores, indicating that it was able to successfully predict the storm affected regions with good accuracy, and was able to do so consistently across various window sizes.

The graph above in Figure 4.19 shows the comparison of FSS scores for Frame 5 across different models and window sizes. From the graph, it's clear that Model 2 consistently performs the best across all window sizes, with Model 4 closely following it. Model 3, although better than Model 1, still shows has some room for improvement, especially at smaller window sizes. Model 1 and Model 4 may be superior to the other 2 models due to the models design to predict only one output frame. Since Models 1 and 3 depend on the predicted frame 4 to predict the output frame 5, it is expected for these models to perform poorly compared to the rest. However, for real-life nowcasting scenarios where multiple frames are expected to be predicted for subsequent hours (t+1, t+2 or t+3), Model 3 would be able to give the best possible predictions since it has shown great improvement in the FSS scores over broader scales, indicating it is capable of both predicting the general spatial structure of the storms as well as the temporal shifts in the storm patterns that occur across each input frames. Although Model 1 made use of continuous input sequences, it surprisingly produced poor FSS scores for the final predictive frame 5, making it both computationally intensive as well as less accurate in its predictive capabilities.

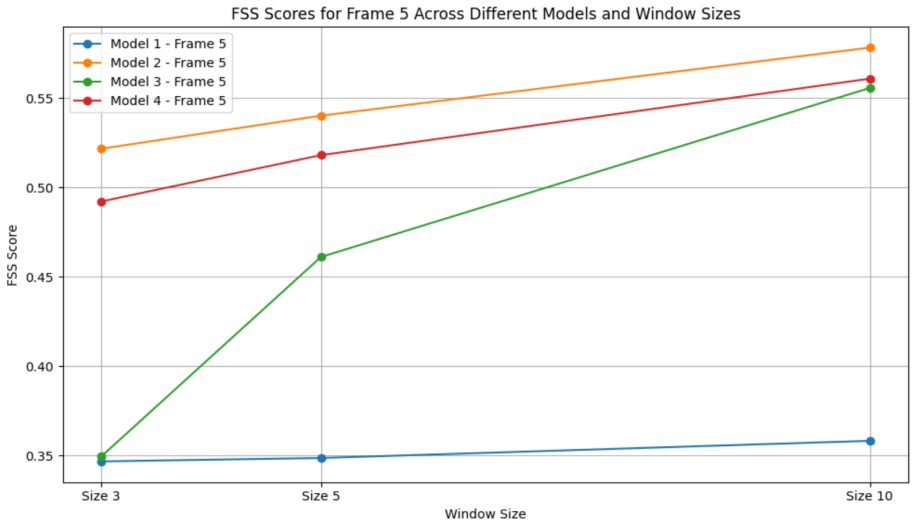

Figure 4.19: Comparative Fraction Skill Scores (FSS) across different models and window sizes for predicted Frame 5.

4.6.2 ROC Curves and Model Comparison

In addition to evaluating model performance using the Fractions Skill Score (FSS), Receiver Operating Characteristic (ROC) curves were generated for all four models to evaluate their ability to distinguish between the presence and absence of convective rainfall. The ROC curves for Models 1 through 4 are shown in Figure 4.20. As seen from the ROC curves, all models performed well, with Model 4 achieving the highest AUC-ROC value of 0.97, followed closely by Model 1 with an AUC-ROC of 0.96. This indicates that these models have a strong ability to correctly identify convective rainfall events while minimizing false positives in the process.

The ROC curve for Model 1 shows a near-perfect AUC of 0.96, indicating that this model effectively balances sensitivity and specificity. This performance suggests that the model is capable of detecting true events with high accuracy. Next, despite achieving the highest score in the FSS evaluation among the four models, the ROC curve for Model 2 reflects a lower AUC of 0.80, indicating that this model has a higher tendency for false positives compared to Models 1, 3, and 4. This could be due to the model's configuration or the choice of input frames, which may not have been as effective in capturing the temporal patterns of rainfall events.

The ROC curve for Model 3 shows a significant improvement over Model 2, with an AUC of 0.92. This shows good overall performance in trying to identify positive and negative cases, making it a good choice for nowcasting, particularly in scenarios where the false positive rate needs to be minimized (cases where it is important for the model to not classify Non-Convective Zones as Convective Zones). Finally, The ROC curve for Model 4 shows the highest AUC of 0.97 among all models, which shows it is the better model amongst the four in terms of tasks related to classifying areas as Convective and Non-Convective Zones. The high AUC value suggests that this model is good for tasks that require high precision and recall, such as predicting severe weather events.

Model 4's high AUC and FSS scores across all configurations suggest it is the best model overall. Its ability to have a high classification accuracy while also producing predictions high in spatial accuracy, indicates that it effectively detects convective rainfall events and accurately predicting their spatial distribution. While Model 1 has a high AUC, its lower FSS scores

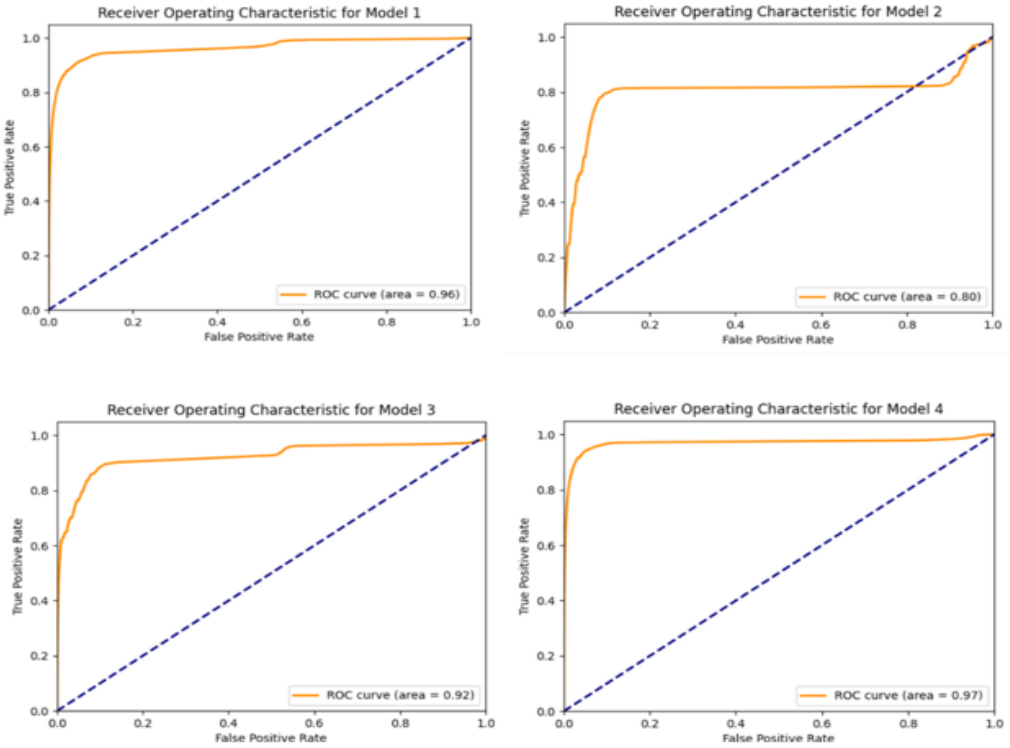

Figure 4.20: ROC Curve for Model 1, 2, 3 and 4

suggest that its strength lies more in its classification ability rather than in spatial accuracy. This could suggest that while Model 1 is effective at identifying events, it may struggle with precisely locating them in regions. Model 2 shows a relatively lower AUC but high FSS scores, suggesting that while Model 2 may not be as effective in convective event detection (as shown by AUC), it is more reliable in capturing the spatial details once an event is detected, which is the basis of nowcasting. From these results, it can be concluded that Model 1 offers a good balance between classifying zones as Convective events, and then predicting the spatial patterns followed by these convective storms, making it useful as an ideal model for Nowcasting tasks.

4.6.3 Comparison of ConvLSTM Models with existing Nowcasting Models

This research was inspired by the study conducted by Burton et al. (2022), which primarily uses the Fraction Skill Score (FSS) as the evaluation metric. It is particularly useful in nowcasting because it accounts for the spatial distribution of precipitation, allowing for a more nuanced evaluation than simple point-to-point comparison metrics like Mean Squared Error (MSE). The Burton et al. (2022) study emphasized the importance of FSS in nowcasting, particularly for assessing forecast skill over different spatial scales and lead times.The decision to focus on FSS is well-founded given that FSS is particularly effective in assessing the spatial accuracy of forecasts, which is important in nowcasting applications, especially in regions with complex weather patterns like West Africa. However, it's essential to compare the findings with those reported in similar studies by Burton et al. (2022), Shi et al. (2015), and Lagerquist et al. (2021), which also applied advanced deep learning techniques for nowcasting (Burton et al., 2022; Shi et al., 2015; Lagerquist et al., 2021).

Comparison with Burton et al. (2022)

Burton et al. (2022) focused on the satellite-based nowcasting of West African mesoscale storms. Their research showed that useful forecasts typically have FSS values greater than 0.5 across varying spatial scales and lead times. This can be used to analyse the FSS scores achieved in the models used in this research, where it was found that Model 2 demonstrated FSS scores of 0.5782 for frame 5 with a window size of 10, which is within the range considered useful in the Burton et al. study. The consistency in FSS scores above 0.5 across different models and window sizes suggests that the models captures the spatial distribution of rainfall events effectively (Burton et al., 2022).

Furthermore, the AUC metric was not specifically reported in Burton's study; however, the use of AUC in this research provides additional insight into the model's classification ability. For instance, Model 4, with an AUC of 0.97, shows a strong capability in distinguishing between true positive and false positive events. This combination of high FSS and AUC suggests that the ConvLSTM models not only is good in spatial accuracy but also in the overall prediction quality.

Comparison with Shi et al. (2015)

Shi et al. (2015) used ConvLSTM networks, focusing on the ability to capture spatio-temporal patterns in weather forecasting. Although their study primarily used metrics such as Root Mean Squared Error (RMSE) and Critical Success Index (CSI), the underlying objective was to improve spatial-temporal pattern recognition in precipitation nowcasting. While direct comparisons between FSS and RMSE are not possible, the ConvLSTM models developed in this study, particularly Model 2 and Model 4, demonstrated strong FSS scores that suggest comparable performance in capturing spatial patterns (Shi et al., 2015)

In terms of AUC, the high values observed in this research, expecially for Model 4, show a strong ability to correctly classify precipitation events. This aligns with Shi et al.'s objective of enhancing the accuracy of precipitation nowcasting through better spatio-temporal pattern recognition. Although FSS was not used by Shi et al., the high AUC scores in the current research show that the ConvLSTM models are effective not only in spatial accuracy but also in their overall prediction reliability, which also aligns with Burton's (2022) objectives and conclusions.

Comparison with Lagerquist et al. (2021)

Lagerquist et al. (2021) made use of the U-Net architecture for nowcasting, using FSS as both a loss function during training and a performance metric during evaluation. Their model achieved high FSS scores, particularly at shorter lead times (0 to 60 minutes, which is equivalent to 1 frame prediction), with FSS gradually declining as lead time increased to 120 minutes. This is similar to the results obtained in this research, where Model 2 and 4 which focused on predicting 1 temporal frame had higher FSS score than Models 1 and 3 which needed to predict 2 temporal frames. Despite this decline, their U-Net model consistently outperformed the persistence model, especially during peak convective activity periods in the afternoon and summer months (Lagerquist et al., 2021)

By comparing the Model 2's FSS results to Lagerquist's findings, it can be said that the ConvLSTM models perform robustly at the shorter lead times, and this performance declines when trying to predict for longer lead times (120 minutes or 2 time frames). Moreover, the

model's FSS scores and high AUC scores, suggest that it captures spatial patterns effectively, similar to Lagerquist's U-Net but possibly, particularly at shorter lead times.

Comparison with NWP models

NWP models, like those examined in the study by Bližňák et al. (2020), simulate weather processes using physical equations, making them particularly strong in medium- to long-term forecasting, but more computationally intensive. It was observed that their performance in short-term nowcasting, particularly within the first few hours, was less accurate compared to other deep learning models. The study found that, in the early hours of forecasting, simple extrapolation methods outperformed NWP models in terms of Root Mean Square Error (RMSE), Correlation Coefficient (CORR), and Fractions Skill Score (FSS). Specifically, the extrapolation method yielded a 25% lower RMSE, a 20% higher CORR, and a 15% higher FSS compared to NWP models during the initial hours (Bližňák et al., 2017).

In contrast, the ConvLSTM models showed consistently good predictive performance across short-term predictions, with good FSS values that suggest high spatial accuracy in predicting rainfall events. For instance, Model 2 achieved an FSS of 0.5782 for Frame 5 with a window size of 10, which is a better performance when compared to the performance metrics of NWP models at similar lead times. Unlike NWP models, which improve as the forecast period or lead time increases, the ConvLSTM model is able to maintain a consistently high accuracy even in the early stages, making it suitable for nowcasting applications where immediate and accurate predictions are important.

4.7 Summary

The results presented in this chapter show the effectiveness of the ConvLSTM models in the task of nowcasting convective rainfall in West Africa, particularly when evaluated using the Fraction Skill Score (FSS) and the Area Under the Receiver Operating Characteristic Curve (AUC), while also showing the prediction plots of each model configuration.

Across all models, the training and validation metrics showed consistent patterns, where each model showed a significant reduction in loss during early epochs, with a gradual or sometimes quick convergence of training and validation losses. This shows the effective learning and minimal overfitting of the models. Moreover, the high accuracy values achieved by all models (96%) further confirmed their robustness in learning the spatial and temporal patterns associated with convective rainfall. However, this was achieved due to the majority of the values in the input data being 0 or close to 0, thereby stressing the need to make use of other evaluation metrics for gaining better insights on the model performance. The models also had runtimes greater than 5 hours, which could be due to the very low learning rate of 0.0001, that allowed the model to the learn the more patterns from the input frames. A higher learning rate was tested which gave shorter runtimes, however it was not able to capture the patterns as efficiently.

Next, the prediction plots showed that all models had varying success in predicting the spatial distribution and intensity of rainfall. Model 1 struggled with finer details, particularly in predicting the evolution of rainfall events across subsequent frames. Model 2, which made use of more input frames, showed improved spatial accuracy of predicting rainfall patterns over time. However, the intensity predictions were often overestimated. Models 3 and 4 displayed a balance between spatial accuracy and the ability to generalize predictions over larger temporal gaps, and the increased temporal gap in Model 4's input frames led to better overall spatial predictions, stating its effectiveness in nowcasting tasks.

Then, the FSS scores provided evaluation of the models' spatial accuracy across varying window sizes. The results showed that Model 2 consistently outperformed other models, especially at larger window sizes. Model 4 followed closely, showing good performance in both fine-scale details and large-scale features. Model 1, while showing moderate FSS values, was more effective at predicting broader spatial features rather than fine details at smaller window size. Model 3 displayed consistent FSS values across both predicted frames, suggesting it was well-generalized and capable of maintaining accuracy in both broad and fine details. The trend of increasing FSS with larger window sizes was expected, as larger windows are less sensitive to small-scale errors and more focused on capturing broader spatial patterns.

This was followed by the use of ROC curves, which provided more insights into the models' classification abilities. Model 4, with the highest AUC of 0.97, was the most effective in distinguishing between convective and non-convective events. Despite Model 2 achieving the highest FSS scores, its AUC of 0.80 indicated a higher rate of false positives, suggesting a trade-off between spatial accuracy and classification precision. Models 1 and 3, with AUCs of 0.96 and 0.92 respectively, balanced both spatial and classification accuracy, making them good choices for scenarios where both features are important.

Finally, the results obtained were compared with the already existing models, particularly the ones used in Burton et al. (2022), Shi et al. (2015) and Langerquist et al. (2021), which showed comparable results between the ConvLSTM models used in this research and the models used in the previous researches. The comparison with existing models further highlights the strengths of ConvLSTM models, particularly in nowcasting tasks that require both spatial and temporal accuracy. These findings lay a solid foundation for further improvement and application of ConvLSTM models in nowcasting in regions with complex weather patterns like West Africa (Burton et al., 2022; Shi et al., 2015; Lagerquist et al., 2021).

Chapter 5

Conclusion and Future Work

The research discusses the application of Convolutional Long Short-Term Memory (ConvLSTM) networks for nowcasting convective storm movements in West Africa. The primary objective was to predict the convective storm patterns for upto 2 hour lead time (2 temporal frames), in a region that frequently experiences severe weather phenomena like West Africa. The ConvLSTM model was chosen due to its ability to handle spatiotemporal data effectively, making it suitable for nowcasting the dynamic and complex weather patterns experienced in this region.

The research findings showed that the ConvLSTM-based nowcasting model outperforms traditional Numerical Weather Prediction (NWP) models and other deep learning architectures like CNN and U-Net. The model's superior performance was evidenced through higher Fractions Skill Scores (FSS) across various spatial scales and lead times, particularly in capturing the spatial distribution and intensity of convective rainfall. The successful implementation of the ConvLSTM model shows promise for its integration into operational nowcasting systems, providing more accurate and timely storm predictions, which are crucial for disaster preparedness and mitigation efforts in West Africa.

5.1 Key Findings

The ConvLSTM models consistently outperformed traditional NWP models in short-term nowcasting tasks, particularly in predicting convective rainfall events with higher accuracy and better spatial coverage. The high FSS values across different models indicate the model's robustness in capturing the spatial distribution of rainfall. Moreover, within the ConvLSTM models used, the model configuration that was designed to predict 1 output frame generally performed better such as Models 2 and 4, however, in order to succesfully nowcast for the subsequent temporal frames, it would be better to make use of Models 1 and 3 since it was able to learn more from the temporal aspect of the storm evolution.

Furthermore, similar to other deep learning models like U-Nets, the ConvLSTM models demonstrated good understanding of temporal dynamics, which is critical in nowcasting applications. A more profound conclusion could have been made in the comparison of the ConvLSTM model with the existing models, however, the lack of spatio-temporal data might have diminished the evaluation results. Also, the results suggest that integrating convolutional operations with LSTM networks provides a more comprehensive approach to modeling complex weather phenomena.

The results contribute to the use of deep learning in meteorology, particularly in regions

with challenging weather patterns like West Africa. The success of ConvLSTM models in this study highlights their potential to replace or complement traditional NWP models in specific contexts, especially for short-term forecasting. Accurate nowcasting is crucial for disaster management and preparedness. The improved performance of ConvLSTM models in predicting rainfall intensities can lead to better early warning systems, potentially reducing the impact of severe weather events on vulnerable communities in West Africa.

5.2 Limitations

While the study achieved significant improvements in nowcasting accuracy, several limitations were encountered:

- **Computational Constraints**: The ConvLSTM models, particularly with larger input-output configurations, required more computational resources, which was not available at the time this research was conducted. This limited the exploration of more complex model architectures and longer training times, which could have produced even better results.
- **Temporal Resolution**: The research made use of input data that had only 5 temporal frames, and the ConvLSTM model could have benefitted from the use of more frames, which could have improved the models capability to capture the spatio-temporal patterns more efficiently.
- **Model Generalisation**: The model was trained and tested on a specific dataset derived from the Meteosat Second Generation satellites. The generalization of the model to other regions or different types of weather phenomena may require additional training and adjustments. Moreover, the model could benefit from the use of ground-based radar data along with the satellite images to provide more context to the model for learning the underlying patterns.

5.3 Future Work

- **Model Complexity and Architecture**: Exploring more advanced deep learning architectures, such as Transformer-based models or hybrid architectures that combine ConvLSTM with other techniques or using bootstrapping techniques that could progressively try to use the predicted frames to predict the next frame, could further improve the accuracy and efficiency of nowcasting models.
- **Extended Lead Times**: Expanding the model to predict weather conditions over longer lead times, beyond the 2-hour window, would significantly increase its use for both nowcasting and longer-term weather forecasting.
- **Cross-Regional Studies**: Testing the model in different geographic regions with varying climate conditions apart from West Africa would assess the model's adaptability and robustness, potentially leading to more generalized models applicable across different climatic zones.

Bibliography

Burton, R.R., Blyth, A.M., Cui, Z., Groves, J., Lamptey, B.L., Fletcher, J.K., Marsham, J.H., Parker, D.J., and Roberts, A., 2022. Satellite-based nowcasting of West African mesoscale storms has skill at up to 4-h lead time. *Weather and Forecasting*, 37(4), pp.445-455.

Roberts, A.J., Fletcher, J.K., Groves, J., Marsham, J.H., Parker, D.J., Blyth, A.M., Adefisan, E.A., Ajayi, V.O., Barrette, R., de Coning, E., Dione, C., Diop, A., Foamouhoue, A.K., Gijben, M., Hill, P.G., Lawal, K.A., Mutemi, J., Padi, M., Popoola, T.I., Rípodas, P., Stein, T.H.M., and Woodhams, B.J., 2022. Nowcasting for Africa: advances, potential and value. *Weather*, 77(7), pp.250-256. doi: https://doi.org/10.1002/wea.3936.

Conway, J., Tisdale, J., Smith, M., and Thompson, B., 2021. Frame-by-frame annotation of video recordings using deep neural networks. *Ecosphere*, 12(1), p.e03316.

Donahue, J., Hendricks, L.A., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K. and Darrell, T., 2014. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2625-2634).

Farneback, G., 2003. Two-frame motion estimation based on polynomial expansion. In *Scandinavian conference on Image analysis* (pp. 363-370). Springer, Berlin, Heidelberg.

Kumar, A. and Hoerling, M.P., 1995. Prospects and limitations of seasonal atmospheric GCM predictions. *Bulletin of the American Meteorological Society*, 76(3), pp.335-345.

Lim, S.L. and Lo, P.Y., 2024. Atmospheric Cloud Image Detection with Convolutional Neural Network (CNN). *International Journal of Integrated Engineering*, 16(3), pp.145-156.

Lucas, B.D. and Kanade, T., 1981. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence* (Vol. 2, pp. 674-679).

Ramani, K., Rajesh, M., Yasaswini, T., Shaharun, V.A., Chandu, V.D. and Duraiswamy, Y., 2024. Advanced Rainfall Classification and Pattern Analysis using Neural Networks. In *Proceedings of the International Conference on Computational Innovations and Emerging Trends (ICCIET- 2024)* (pp. 343-353).

Chatfield, C., 1996. *The Analysis of Time Series: An Introduction*. CRC press.

Shi, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K. and Woo, W.C., 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems* (pp. 802-810).

Tran, D., Bourdev, L., Fergus, R., Torresani, L. and Paluri, M., 2015. Learning spatiotemporal features with 3D convolutional networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 4489-4497).

Zheng, J., Ling, Q., Li, J. and Feng, Y., 2024. Improving the Short-Range Precipitation Forecast of Numerical Weather Prediction through a Deep Learning-Based Mask Approach. *Advances in Atmospheric Sciences*, 41(8), pp.1601-1613.

Liu, Y., 2011. Advanced Extreme Rainfall Simulation Using a Numerical Weather Prediction Model. *Atmospheric Chemistry and Physics*, 11(16), pp. 8791-8808.

Ronneberger, O., Fischer, P., and Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pp.234-241. Springer.

Bližňák, V., Sokol, Z., and Zacharov, P., 2017. Nowcasting of deep convective clouds and heavy precipitation: Comparison study between NWP model simulation and extrapolation. *Atmospheric Research*, 184, pp.24-34. Elsevier.

Kalnay, E., 2003. *Atmospheric modeling, data assimilation and predictability*. Cambridge University Press.

Warner, T.T., 2010. *Numerical weather and climate prediction*. Cambridge University Press.

Molteni, F., Buizza, R., Palmer, T.N., and Petroliagis, T., 1996. The ECMWF ensemble prediction system: Methodology and validation. *Quarterly journal of the royal meteorological society*, 122(529), pp.73-119.

Zhang, Y., Long, M., Chen, K., Xing, L., Jin, R., Jordan, M. I., and Wang, J., 2023. Skillful Nowcasting of extreme precipitation with NowcastNet. *Nature*, 619(20), pp.526-532. DOI: 10.1038-41586-023-06184-4.

Zhang, G., and Chanson, H., 2018. Application of local optical flow methods to high-velocity free-surface flows: Validation and application to stepped chutes. *Experimental Thermal and Fluid Science*, 90, pp.186-199. Elsevier.

Lagerquist, R., Stewart, J.Q., Ebert-Uphoff, I., and Kumler, C., 2021. Using deep learning to nowcast the spatial coverage of convection from Himawari-8 satellite data. *Monthly Weather Review*, 149(12), pp.3897-3921.

Baker, S., and Matthews, I., 2004. Lucas-Kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56, pp.221-255. Springer.

Barron, J.L., Fleet, D.J., and Beauchemin, S.S., 1994. Performance of optical flow techniques. *International Journal of Computer Vision*, 12, pp.43-77. Springer.