NUST
SCHOOL OF ELECTRICAL ENGINEERING
& COMPUTER SCIENCE

IceWatch

GLOF Prediction Using Deep Learning

Final Year Project Report

By

Zuha Fatima

Muhammad Anser Sohaib

Muhammad Talha

In Partial Fulfillment
Of the Requirements for the degree
Bachelor of Electrical Engineering (BEE)

School of Electrical Engineering and Computer Science
National University of Sciences and Technology
Islamabad, Pakistan
(2025)

DECLARATION

We hereby declare that this project report entitled **IceWatch** submitted to the **School of Electrical Engineering & Computer Science (SEECS)**, is a record of an original work done by us under the guidance of Advisor Ms. Ayesha Kanwal and Co-Advisor Ms. Nazia Pervaiz and that no part has been plagiarized without citations. This project work is submitted in the partial fulfillment of the requirements for the degree of Bachelor of Electrical Engineering.

Team Members	**Signature**
Zuha Fatima	Zuha
Muhammad Anser Sohaib	Anser
Muhammad Talha	Talha

Supervisor	**Signature**
Ms. Ayesha Kanwal	
Ms. Nazia Pervaiz	Nazia Pervaiz

Date:	25th April, 2025

DEDICATION

We dedicate this work to our families and mentors, whose unwavering support and encouragement made this journey possible.

ACKNOWLEDGEMENTS

We would like to express our sincere gratitude to our Advisor Ms. Ayesha Kanwal and Co-Advisor Ms. Nazia Pervaiz for their invaluable guidance and support throughout the course of this project.

TABLE OF CONTENT

Contents

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Glacial Lake Outburst Floods (GLOFs) are sudden and devastating events that occur when the natural dams containing glacial lakes fail, leading to massive downstream flooding. With the ongoing impacts of climate change, the size and number of glacial lakes are rapidly increasing, making GLOFs more frequent and unpredictable. These floods pose a significant threat to communities, infrastructure, and ecosystems in high-altitude regions, where early detection and prevention measures are limited. Traditional methods of monitoring and predicting GLOFs often rely on manual observations and static risk maps, which are insufficient for capturing the dynamic changes in glacial environments.

To address this growing challenge, we developed **IceWatch**, an end-to-end system designed to predict the probability of GLOF occurrence using machine learning. The system analyzes key features such as glacial lake area, elevation, distance to glacier fronts, geographic coordinates, and changes in lake size and position over time. By focusing on probability-based prediction rather than simple binary classification, IceWatch provides a more detailed risk assessment for each glacial lake. A visual dashboard is also integrated into the system, allowing users to easily monitor lakes, view probability scores, and identify high-risk regions. This combination of predictive modeling and intuitive visualization empowers decision-makers with actionable insights.

IceWatch aims to enhance early warning systems and disaster preparedness strategies in vulnerable mountainous regions. By offering timely and data-driven forecasts, the system supports researchers, policymakers, and emergency management agencies in prioritizing interventions and reducing the devastating impacts of GLOFs. Through this project, we demonstrate the potential of machine learning and intelligent visualization to improve climate resilience and safeguard at-risk communities.

Chapter 1

INTRODUCTION

Background and Motivation

Glacial Lake Outburst Floods (GLOFs) are among the most hazardous and unpredictable natural disasters in high-mountain environments. These events occur when a natural dam—typically composed of ice, moraines, or glacial debris—fails and releases a large volume of water from a glacial lake in a short period. GLOFs can lead to catastrophic downstream flooding, posing threats to human settlements, infrastructure, agricultural lands, and ecosystems. With the ongoing impacts of climate change, the retreat of glaciers has accelerated globally, leading to the expansion of existing glacial lakes and the formation of new ones. This has significantly increased the frequency and severity of GLOF events, especially in sensitive mountain regions such as the Karakoram, Hindu Kush, and the greater Himalayas.

Despite this growing threat, traditional GLOF monitoring and early warning systems remain limited. Current methods often rely on static hazard mapping, manual observations, or threshold-based criteria, which fail to incorporate the dynamic behavior of glacial lakes, climatic trends, and glacier movement. As a result, there is a critical need for more adaptive, data-driven systems that can continuously assess risk using multiple environmental variables.

Project Aim

This study presents **IceWatch**, a comprehensive, machine learning-based prediction system for assessing the probability of GLOF occurrence. The system is designed to analyze environmental signals such as land surface temperature, glacier velocity, glacial lake features, and velocity mosaics derived from satellite imagery. By leveraging modern deep learning architectures—specifically **Transformers, LSTMs, and Convolutional Neural Networks (CNNs)**—IceWatch provides a robust, real-time assessment of GLOF risk in vulnerable regions.

System Overview

The IceWatch framework consists of three core components:

TempFlow: Temperature Prediction Using LSTM

TempFlow is an LSTM-based model trained on over two decades of NASA MODIS Land Surface Temperature (LST) data. It captures both long-term warming trends and short-term temperature fluctuations. These patterns are critical in forecasting glacial melt events that may contribute to the rapid formation or expansion of glacial lakes. The model uses cyclical temporal features and quality-filtered data to produce accurate seasonal temperature forecasts.

TerraFlow: Glacier Velocity Prediction Using Transformer

TerraFlow employs a Transformer-based sequence model to predict future glacier ice velocities based on spatiotemporal features such as latitude, longitude, and day-of-year encodings. Trained on millions of observations from the NASA ITS_LIVE dataset, TerraFlow identifies changes in glacier dynamics—such as surging or flow acceleration—that may indicate instability or increased pressure on glacial lakes.

RiskFlow: CNN Classifier for Velocity Mosaics

In addition to tabular data models, IceWatch incorporates a **CNN-based image classifier** that processes velocity mosaics—satellite imagery showing glacier motion fields. This model detects anomalous spatial patterns in glacier flow, offering a visual cross-validation layer for TerraFlow predictions and improving the overall interpretability of the system.

Parameter Selection

The choice of parameters in IceWatch is informed by glaciological science and prior GLOF case studies. Rising land surface temperature is a well-established driver of glacier melt and lake formation. Ice velocity changes reflect glacier surges or dam instability, while lake area and distance from the glacier terminus are commonly used indicators of hazard potential. Elevation and geolocation provide spatial context for generalizing across regions. Integrating these variables ensures that IceWatch captures both physical and temporal precursors to GLOF events.

Contributions

The key contributions of this research are:

- A **multi-modal GLOF prediction framework** combining sequential (LSTM), attention-based (Transformer), and image-based (CNN) deep learning models.

- A **probability-based GLOF risk scoring system** that moves beyond binary classification to provide nuanced risk assessments.
- The development of a **visual dashboard interface** that enables researchers and disaster response authorities to monitor, interpret, and act upon real-time GLOF risk.
- A scalable pipeline that can be adapted to various high-risk glacial regions beyond the initial study area (Shisper Glacier, Pakistan).

Organization of the Paper

The remainder of this paper is structured as follows:

Section 2 reviews relevant literature review in GLOF detection and glacier modeling.

Section 3 describes the Problem Definition

Section 4 outlines the Methodology

Section 5 presents the Detailed Architecture and Design

Section 6 Implementation and Testing

Section 7 Results and Discussion

Section 8 Conclusion and Future Work

Chapter 2

LITERATURE REVIEW

Recent advancements in remote sensing and machine learning have significantly contributed to the monitoring and prediction of Glacial Lake Outburst Floods (GLOFs).

Early research predominantly focused on mapping glacial lakes and assessing their hazard potential using satellite imagery. **Fujita et al**. [1] quantified potential flood volumes from Himalayan glacial lakes using topographical and morphological parameters, providing foundational knowledge for hazard estimation. **Worni et al.** [2] emphasized the importance of **dynamic lake volume monitoring** and introduced process-based flood simulations, although these methods lacked real-time predictive capability.

The application of machine learning has improved GLOF susceptibility mapping. **Zhou et al**. [3] proposed a hybrid machine learning framework combining environmental indicators and glacier evolution data to **predict debris flows** associated with GLOFs. Similarly, **Abbas et al.** [4] compared machine learning, deep learning, and hybrid models for GLOF susceptibility mapping in the Central Karakoram National Park, demonstrating that **hybrid approaches** outperform classical models. However, these models primarily relied on static terrain features without incorporating temporal glacier movement data.

Deep learning techniques have been employed for glacial lake segmentation. **Pandey and Malakar** [5] utilized **CNNs for water body detection** from remote sensing imagery, achieving high segmentation accuracy. **Dey et al. [**6] improved upon this by applying **U-Net architectures** for glacial lake extraction, although their models were sensitive to cloud interference and seasonal variations in the imagery.

Recent work has introduced Transformer architectures for glacier monitoring. **Maslov et al.** [7] developed **Glacier-VisionTransformer-U-Net (GlaViTU)**, a hybrid CNN-Transformer model that outperformed traditional CNNs in glacier mapping tasks by capturing long-range spatial dependencies. However, limited labeled datasets and complex training processes remain challenges for Transformer-based glacier analysis.

Efforts to integrate multi-modal datasets have shown promise. **Mohanty et al.** [8] reviewed the use of **multi-sensor approaches** (combining optical, SAR, and DEM data) for glacier velocity and facies characterization, highlighting the advantages of fusing different observation sources. **Peng et al.** [9] explored the use of deep learning combined with open Earth observation datasets for **large-scale glacier mapping** but noted persistent challenges in real-time processing and data quality variability.

Despite these advancements, existing systems typically focus on static mapping or susceptibility scoring rather than real-time GLOF prediction integrating thermal, spatial, and dynamic glacier movement data. The gaps identified in previous works motivate the development of a system that combines temporal temperature forecasting, dynamic velocity prediction, and spatial anomaly detection for a comprehensive, real-time GLOF early warning solution.

Chapter 3

PROBLEM DEFINITION

GLOF Trends and Growing Risks

Glacial Lake Outburst Floods (GLOFs) have emerged as an escalating hazard in high-mountain regions, driven by accelerating glacier retreat under global warming. Recent studies and observational datasets highlight the alarming rise in GLOF events globally. Satellite observations show a substantial increase in both the number and size of glacial lakes globally, particularly in vulnerable regions such as the Himalayas, the Andes, and parts of North America.

1. Increasing Frequency of GLOFs Globally

Recent data show a steady increase in the frequency of documented GLOF events since the early 20th century, with a pronounced rise from the 1990s onward. This trend correlates with intensified glacier retreat due to anthropogenic climate change.

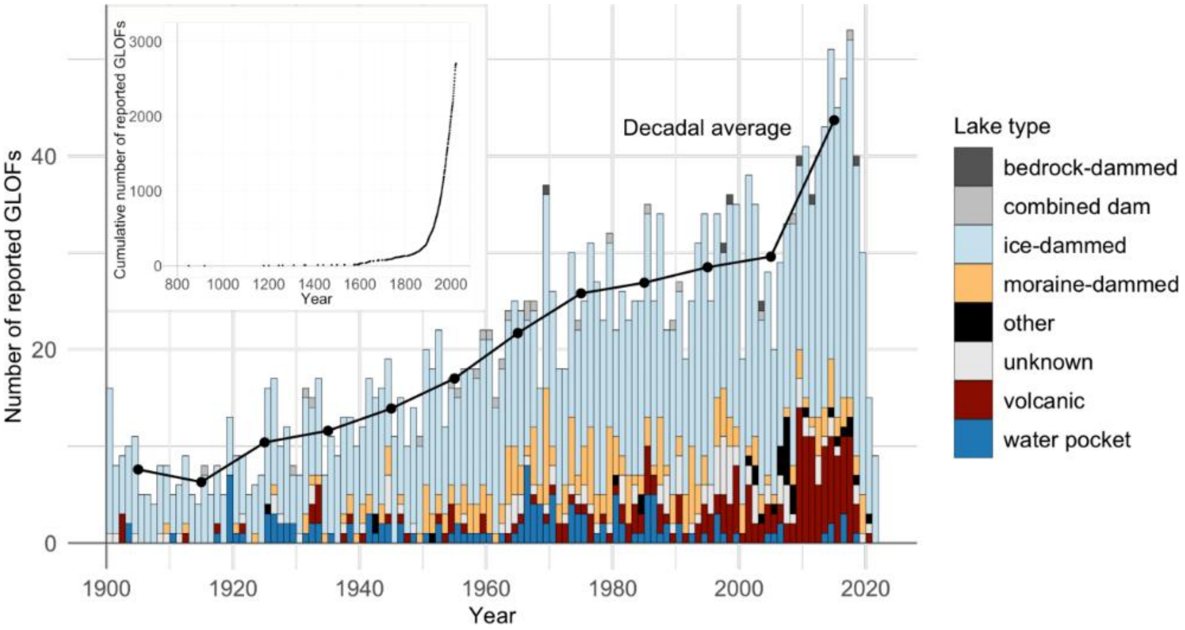


Figure 1: Number of documented GLOF events globally from 1900 to 2020, illustrating an upward trend in occurrence frequency (Lützow et al., 2023).

2. Global Distribution of GLOF Occurrences

GLOFs are not geographically uniform. High-risk zones are predominantly located in the Himalayas, Andes, and certain parts of North America (e.g., Alaska and Western Canada). The map below illustrates global hotspots for recorded GLOF events.

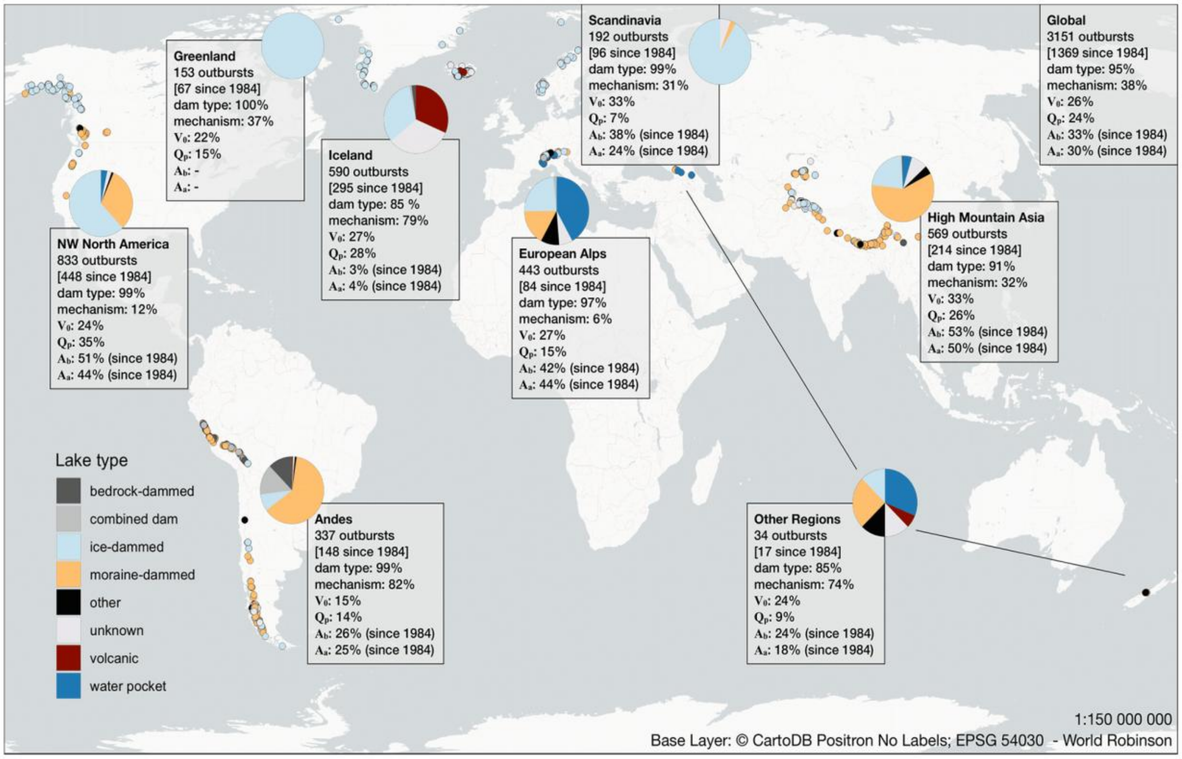


Figure 2 Spatial distribution of recorded GLOF events globally(Lützow et al.,2023[6]).

This figure gives an overview map of the source locations of GLOFs and selected regional database contents. Colors show the lake type of the GLOF locations (bubbles) and proportions of GLOF locations in each region (pie charts). Boxes are regional statistics on the total number of outbursts (and the number of outbursts of GLOFs since the availability of satellite images); the percentage of cases with reported dam type and drainage mechanisms; and parameters on GLOF magnitude such as peak discharge (Q_p), flood volume (V_0), and the area before (A_b) and after (A_a) the GLOF.

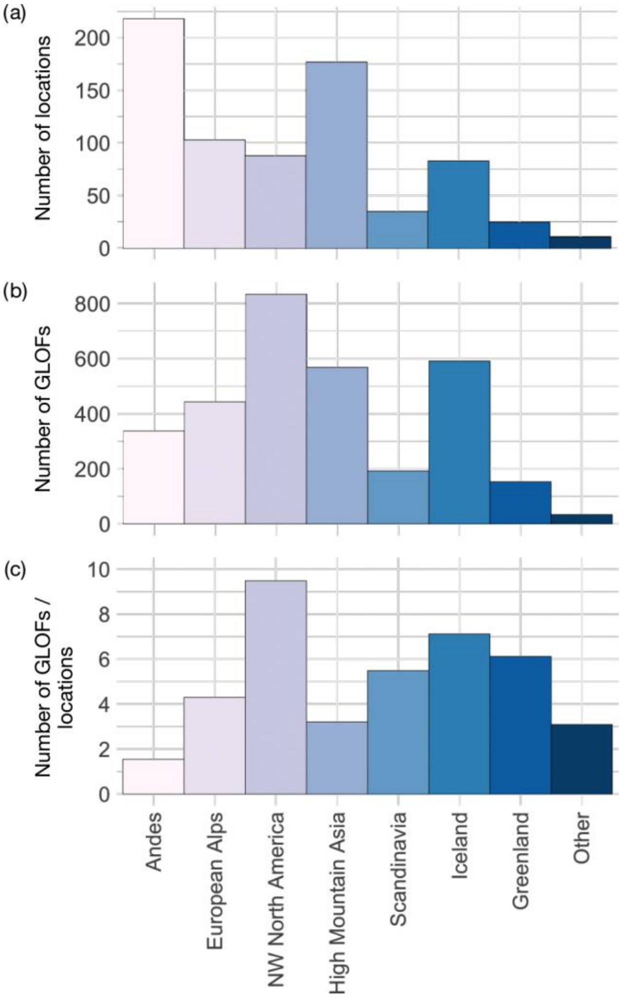


Figure 3 Regional distribution of reported (a) GLOF source locations, (b) all GLOFs, and (c) ratio of the number of GLOFs and source locations.

3. Expansion of Glacial Lakes in the Himalayas

Glacial lakes are increasing both in number and area, particularly across the Himalayas, thereby heightening the potential for GLOFs. Between 2011 and 2023, a significant surge, over **10% increase in glacial lake formation** has been observed. [7]

4. Populations at Risk from GLOFs

An estimated **15 million people** globally live within vulnerable zones that could be impacted by GLOFs. South Asian countries, particularly India, Pakistan, Nepal, and Bhutan host the largest at-risk populations.

Causes of GLOFs

The formation and sudden failure of glacial lakes are influenced by several interrelated natural and climatic factors:

1. Climate Change and Glacier Retreat

- Rising global temperatures are accelerating glacier melt, leading to the rapid formation and expansion of proglacial lakes.
- As glaciers retreat, the ice acts as a dam, behind which meltwater accumulates.
- Increased melt rates lead to higher lake volumes, increasing hydrostatic pressure on natural dams, making them more prone to failure.

2. Dam Instability (Moraine and Ice Dams)

- Many glacial lakes are dammed by loose accumulations of debris known as **moraines** or by remnant **glacier ice**.
- Ice dams are susceptible to melting and mechanical failure due to warming temperatures and internal stress.

3. Triggering Mechanisms

Several immediate triggers can precipitate a dam failure:

- **Ice or Rock Avalanches**: Landslides, icefalls, or rockfalls from adjacent mountain slopes into the lake can cause a sudden surge in water level (known as a displacement wave), overtopping and breaching the dam.
- **Earthquakes**: Seismic activity can destabilize the dam material, leading to a rapid collapse.
- **Heavy Rainfall or Rapid Snowmelt**: Excess water input into already full glacial lakes can stress dam integrity.
- **Subglacial Drainage**: Some GLOFs occur when water accumulates beneath a glacier and then bursts through the ice.

4. Changes in Lake Geometry

- Expansion of lake surface area and depth over time increases the hydraulic pressure acting on the dam.
- Thinning glaciers adjacent to lakes can reduce support against the moraine walls, making breaches more likely.

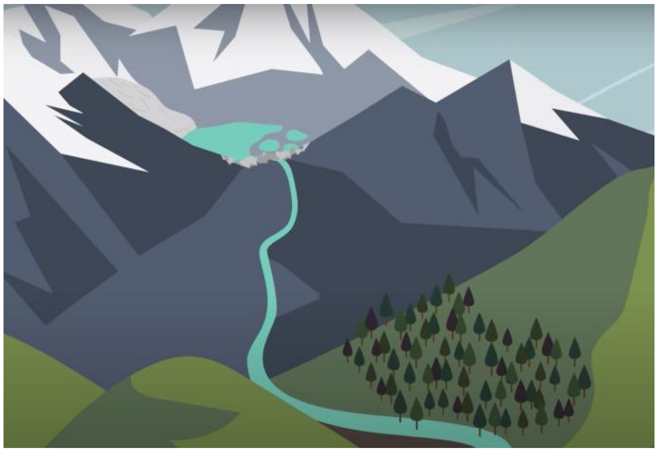


Figure 4 Glacial Lake Formation

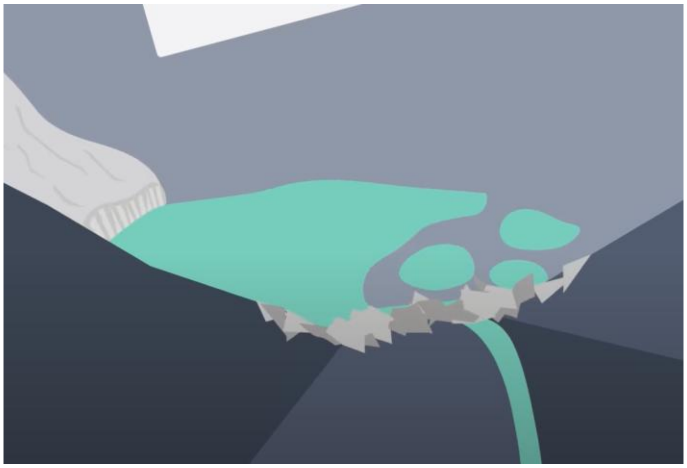


Figure 5 Glacial Lake

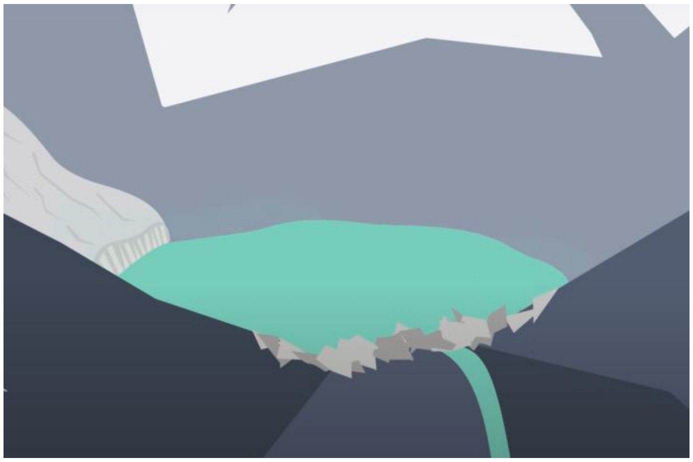


Figure 6 Glacial Lake Expansion

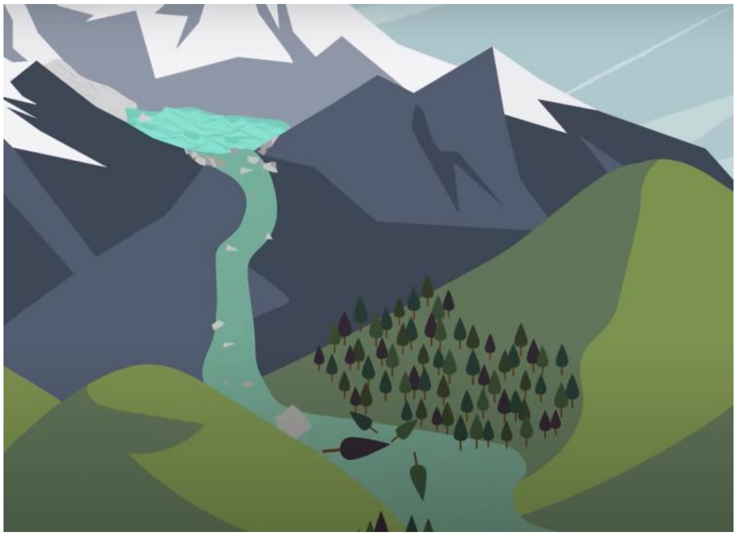


Figure 7 GLOF occurs after Glacial Lake Breaches

Impacts of GLOFs

- **Human Displacement:** Villages and towns located in vulnerable river valleys can be inundated within hours of a GLOF event.
- **Infrastructure Damage:** Roads, bridges, hydroelectric plants, and irrigation systems are often severely impacted.
- **Economic Losses:** GLOFs can wipe out agricultural land and disrupt water supplies, impacting livelihoods.
- **Ecological Effects:** GLOFs can reshape river channels, erode forests, and deposit massive amounts of sediment, affecting ecosystems for years.

Importance of Early Warning and Monitoring

Given their sudden nature and devastating impacts, developing effective GLOF monitoring, prediction, and early warning systems is crucial for safeguarding human life and minimizing economic and environmental damages. This underscores the urgency for innovative, scalable, and automated solutions, such as the IceWatch project proposed in this research, that integrate satellite observations, deep learning models, and environmental forecasting to provide timely GLOF risk assessments.

Limitations of Existing GLOF Warning Systems:

1. Heavy Reliance on Manual Monitoring and Field Sensors

Most operational GLOF monitoring systems depend on in-situ sensors such as water level gauges, weather stations, or manual glacier surveys. These instruments:

- Are often expensive to install and maintain in remote glacial environments.
- Provide localized data that may not capture regional trends.
- Are vulnerable to harsh weather conditions or mechanical failure.

2. Lack of Multivariate Data Integration

Traditional EWS usually monitor one or two variables (e.g., lake area or rainfall) without considering dynamic factors like glacier motion or thermal stress. This narrow focus:

- Limits the ability to detect complex, multicausal events.
- Fails to account for precursors like surface velocity spikes or anomalous warming.

3. Threshold-Based Logic Is Rigid and Outdated

Current systems often operate on static rules or thresholds (e.g., lake size > X = high risk), which:

- Cannot adapt to diverse lake-glacier morphologies.
- Are prone to false positives and false negatives.
- Do not learn from past events or trends.

4. Poor Temporal Prediction Capabilities

Most existing models offer situational awareness (e.g., current lake level), but not true forecasting. As a result:

- Warnings are issued late, often after triggering mechanisms have already started.
- Authorities have little lead time for evacuation or mitigation.

5. Limited Geographic Coverage

Due to cost and access constraints, only a small subset of glacial lakes are currently monitored in high-risk regions like the Himalayas, Andes, and Alaska.

Table 1 Traditional GLOF Detection Systems vs IceWatch

Feature	**Traditional GLOF Systems**	**IceWatch**
Data Source	Ground sensors, static maps	Dynamic satellite + temporal data
Prediction Type	Rule-based alerts	AI-driven forecasting
Multivariable Analysis	No	Yes (image + velocity + temperature)
Scalability	Limited	Global (remote sensing)
Temporal Forecasting	Reactive	Proactive
Cost	High (maintenance, logistics)	Low (cloud-based, automated)

Addressing Limitations through IceWatch:

IceWatch introduces a next-generation, AI-powered early warning system that addresses the core limitations of conventional approaches through a multi-model, data-driven framework.

1. Automated Monitoring Using Satellite Data

Unlike sensor-based systems, *IceWatch* operates on open-access satellite data (Sentinel-2, MODIS, ITS LIVE), enabling:

- Wide-scale, repeatable, and remote monitoring of hundreds of glaciers.
- Cost-effective surveillance even in inaccessible or politically sensitive regions.

2. Multimodal Deep Learning Integration

The system integrates diverse environmental signals via:

- A **CNN** model for visual GLOF detection from imagery.
- A **Transformer** model (*TerraFlow*) for glacier motion prediction.
- An **LSTM** model (*TempFlow*) for temperature trend forecasting.

This fusion captures both immediate visual cues and slow-onset precursors, improving detection accuracy.

3. Learned Representations, Not Static Thresholds

By training on historical event data and natural variability, *IceWatch*:

- Learns complex spatiotemporal patterns leading to GLOFs.
- Can generalize across different glaciers and climate regimes.
- Offers probabilistic predictions rather than binary alerts.

4. Early Forecasting with Lead Time

With its time-series forecasting capabilities, especially via TempFlow, *IceWatch*:

- Anticipates thermal and mechanical stress on glaciers days to weeks in advance.
- Provides lead time for disaster response planning.

5. Scalable and Deployable System

IceWatch is:

- Cloud-deployable (via platforms like Google Colab, AWS, or GEE).
- Lightweight enough for integration into local disaster management dashboards.
- Easily extensible with additional data sources (e.g., radar, precipitation).

Chapter 4

METHODOLOGY

Project IceWatch is a real-time Glacial Lake Outburst Flood (GLOF) prediction system that integrates three deep learning frameworks: **TerraFlow** (transformer-based for ice velocity prediction), **TempFlow** (LSTM-based for temperature forecasting), and **RiskFlow** (CNN-based for GLOF classification). These models analyze geospatial, temporal, and satellite imagery data to monitor and predict GLOF events in the **Shispare Glacier** region (Lat: 36.32°–36.47°, Lon: 74.60°–74.90°). Below is a detailed methodology, with each model's details organized under its respective heading, including data sources, preprocessing, architecture, training, evaluation, and integration into the IceWatch system.

Framework:

The *IceWatch* framework is composed of three specialized models:

1. **Riskflow** – A CNN Model Detects visual features indicative of GLOF events from Sentinel-2 satellite imagery.
2. **TerraFlow** – A Transformer-based model that predicts glacier surface velocity using spatiotemporal tabular data.
3. **TempFlow** – An LSTM-based time-series model for land surface temperature (LST) forecasting over glacier regions.

Outputs from these three models are synthesized to assess GLOF risk levels and deliver interpretive insights.

By integrating these models, IceWatch provides a multi-faceted approach to GLOF prediction, enabling early warnings for the Shispare Glacier, a region prone to GLOF events due to its dynamic glacial activity and climate sensitivity.

Data Sources and Characteristics:

IceWatch relies on multiple datasets to capture the spatial, temporal, and spectral dynamics of the **Shispare Glacier** region. Each model uses a specific dataset tailored to its predictive task.

The three core models in *IceWatch*—CNN for visual detection, TerraFlow for glacier velocity, and TempFlow for temperature forecasting—each rely on domain-specific datasets.

Sentinel-2 Satellite Imagery (for RiskFlow CNN Model)

Source: European Space Agency (ESA) Copernicus Open Access Hub
Temporal Range: 2017 – 2024
Spatial Resolution: 10m – 20m

The dataset used in this project consists of **Sentinel-2** satellite images focused on the **Shispare Glacier** region. Sentinel-2 is a European Space Agency (ESA) satellite providing high-resolution multispectral imagery. These images were processed and exported using Google Earth Engine.

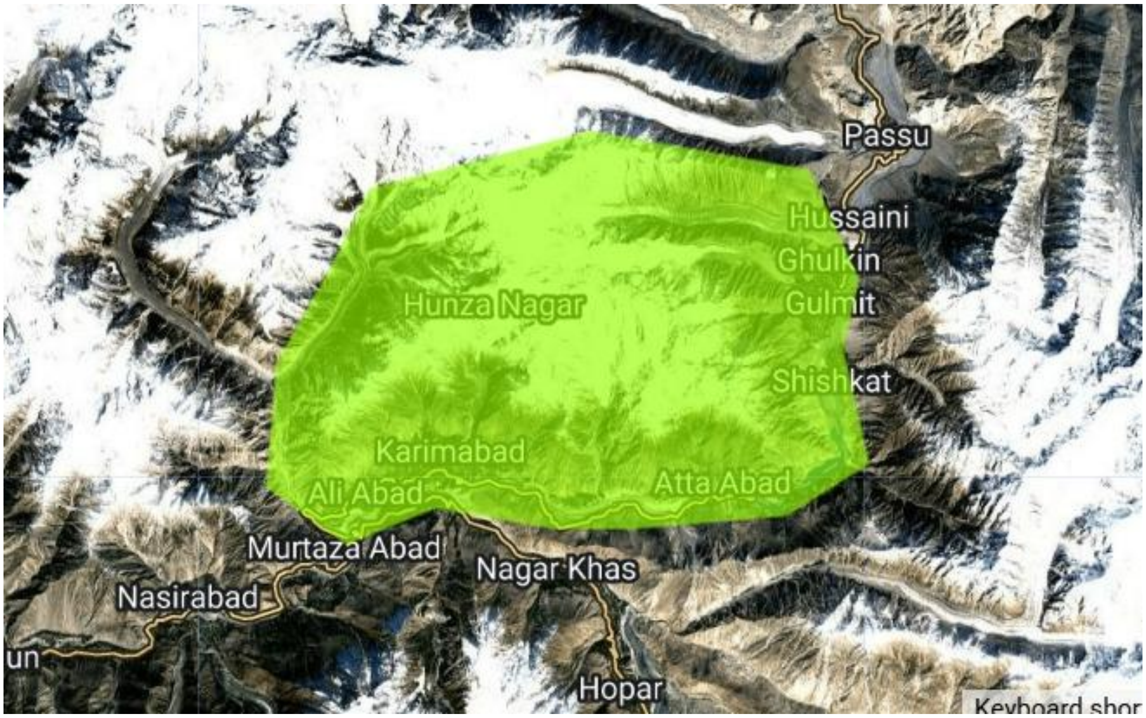


Figure 8 Region of Interest: Shisper Glacier

Sample image from dataset:

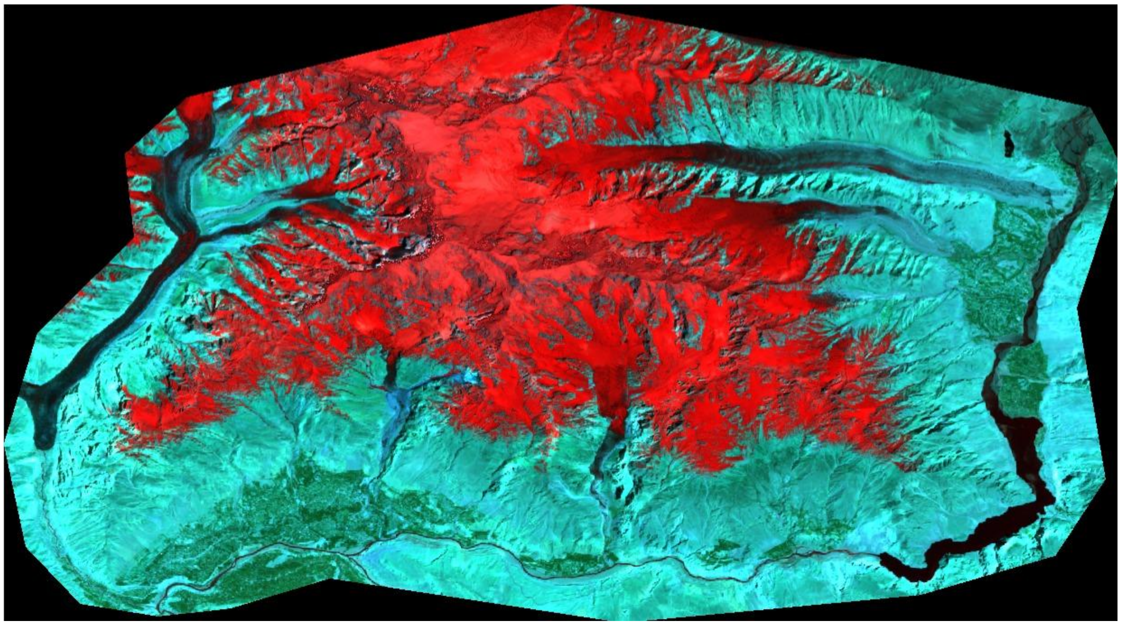


Figure 9 Sentinel-2, rendering of Bands B8, B11 and B12 on QGIS

Bands Used:

The selection of Sentinel-2 bands B2, B3, B4, B8, B11, and B12 was based on their proven sensitivity to snow, ice, meltwater, and vegetation dynamics — all critical indicators for early prediction of Glacial Lake Outburst Flooding (GLOF) events

Table 2 Sentinel-2 Bands used in RiskFlow (wavelength, resolution, Band Number)

Band Name	Band Number	Wavelength (nm)	Resolution	
Blue	B2	490 nm	10 m	
Green	B3	560 nm	10 m	
Red	B4	665 nm	10 m	
NIR (Near-Infrared)	B8	842 nm	10 m	
SWIR-1 (Shortwave Infrared 1)	B11	1610 nm	20 m	
SWIR-2 (Shortwave Infrared 2)	B12	2190 nm	20 m	

Table 3 Reasons For Selection of Sentinel-2 bands for GLOF Detection

Band(s)	Reason
B2 (Blue)	Very sensitive to **snow and ice reflectance**; useful for detecting glaciers and frozen surfaces.
B3 (Green)	Captures **vegetation**, **bare soil**, and **surface features** contrast; helps distinguish glacier surroundings.
B4 (Red)	Key for **snow/ice monitoring**; good contrast between glacier ice and bare earth.
B8 (NIR)	Strong separation between **vegetation**, **water**, and **ice**; important for detecting meltwater or glacial lakes forming.
B11 (SWIR-1)	Sensitive to **snow grain size**, **melting ice**, and **moisture**; helps monitor glacier melting and flood-prone areas.
B12 (SWIR-2)	Detects **liquid water**, **wet surfaces**, and **glacial lake formation** more clearly than visible bands.

Classes

- **GLOF occurrence (label 1)**: Images containing visible signs of glacial lake formation or outburst conditions.
- **NO-GLOF (label 0)**: Images without signs of imminent outburst flooding.

Description:

Sentinel-2 images were used to train the CNN model for detecting visual patterns of GLOF occurrence. These bands are sensitive to water, vegetation, snow, and ice—key features for glacial monitoring. Images were collected for the Shisper Glacier region, preprocessed (resized to 128×128, normalized), and labeled for binary classification: GLOF (1) or NO-GLOF (0).

NASA ITS LIVE Glacier Velocity Data (for TerraFlow Model)

The TerraFlow model utilizes the **Inter-mission Time Series of Land Ice Velocity and Elevation (ITS LIVE)** dataset, developed by NASA's Jet Propulsion Laboratory (JPL). This publicly available dataset provides high-resolution measurements of glacier surface velocity across the globe, derived using feature tracking algorithms applied to optical satellite imagery such as Landsat 4, 5, 7, and 8.

Temporal and Spatial Coverage:

- **Region of Interest:** Shisper Glacier, Karakoram Range, Pakistan
- **Coordinates:** Latitude 36.25°–36.50°, Longitude 74.50°–75.00°
- **Time Range:** January 1, 2000 – December 31, 2024
- **Temporal Resolution:** Daily observations
- **Raw Dataset Size:** ~73.6 million records globally
- **Filtered Dataset for TerraFlow:** ~11.9 million rows (Shisper region only)

Table 4 Features Used for TerraFlow

Variable Name	Description
latitude, longitude	Geographic coordinates of velocity point
date	Observation date, later encoded with cyclical features
vx, vy	Horizontal velocity components (m/day) along x and y directions
v	Magnitude of surface velocity, computed as $\sqrt{(vx^2 + vy^2)}$
velocity_avg, velocity_max	Aggregated daily mean and peak values per (lat, lon) point
day_of_year_sin, day_of_year_cos	Cyclical encodings of time to capture seasonal trends

The ITS LIVE dataset contains high-resolution velocity measurements of glacier surfaces, derived from feature tracking between satellite image pairs. For this project, ~11.9 million rows were processed from an initial 73.6 million, representing the Shisper Glacier. The data was cleaned, aggregated daily by coordinates, and used to predict velocity trends via the Transformer-based TerraFlow model.

Data Characteristics and Challenges:

- **High Dimensionality:** The dataset spans over two decades of daily observations across many geolocations.
- **Heterogeneous Sampling:** Due to cloud cover, image quality, or satellite revisit limitations, some days are missing. These gaps were handled using interpolation or by masking the missing values.
- **Velocity Variability:** Glacier velocities ranged from 0 to >2 m/day, with seasonal peaks during summer. Anomalous spikes often correspond to surging activity or instability—important GLOF precursors.
- **Noise and Artifacts:** Optical feature-tracking introduces potential errors during rapid surface changes or image misalignment. These were mitigated by smoothing and filtering out extreme outliers.

MODIS Land Surface Temperature (for TempFlow Model):

The TempFlow system leverages the **NASA MODIS Land Surface Temperature Product (MOD11A1 V6.1)**, a high-resolution dataset tailored for environmental monitoring. Below is a detailed breakdown of its characteristics:

Source and Scope

- **Temporal Coverage**: February 24, 2000, to December 31, 2024, spanning 25 years.
- **Spatial Focus**: Shisper Glacier region (Latitude: 36.32°–36.47°, Longitude: 74.60°–74.90°), a critical area for studying glacier dynamics.
- **Resolution**: 1 km spatial resolution with daily temporal observations, yielding 98,879 data points across 11 unique coordinates and 8,989 unique dates (near-daily coverage).

Key Variables

- **Primary Feature**: MOD11A1_061_LST_Day_1km, providing LST in Kelvin. Raw values range from 0–322 K, while quality-controlled (QC-filtered) data range from 250–330 K, reflecting typical surface temperatures in the region.
- **Quality Indicators**:
  - **QC_Day_MODLAND**: Flags indicating production quality (e.g., "LST produced, good quality").
  - **QC_Day_LST_Error_Flag**: Error estimates categorized as ≤1 K, ≤2 K, or ≤3 K, with 78% of usable observations having errors ≤2 K.
  - **QC_Day_Emis_Error_Flag**: Metrics for emissivity-related errors, ensuring data reliability.

Table 5 Key Variables for TempFlow

Variable Name	**Type**	**Description**
LST_Day_1km	Numerical (float)	Land Surface Temperature in Kelvin (from MODIS MOD11A1 product)
QC_Day_MODLAND	Categorical	MODIS production quality flag (e.g., good, average, poor)
QC_Day_LST_Error_Flag	Categorical	Estimated LST error: ≤1K, ≤2K, or ≤3K
QC_Day_Emis_Error_Flag	Categorical	Emissivity error category (flag for radiative error levels)
latitude	Numerical (float)	Latitude of the pixel location
longitude	Numerical (float)	Longitude of the pixel location

date	DateTime	Date of observation
month_sin	Numerical (float)	Sine-transformed month value to capture annual seasonality
month_cos	Numerical (float)	Cosine-transformed month value to complement month_sin
quality_flag (derived)	Categorical	Combined quality classification: High, Medium, or Low (based on QC flags)
LST_scaled (derived)	Numerical (float)	Normalized or scaled version of LST for model stability

Data Characteristics

The dataset exhibits pronounced seasonal patterns, with summer peaks (June–July) correlating with glacier melt and winter troughs (December–January) reflecting colder conditions. However, data quality poses a significant challenge:

- **Cloud Interference**: 61% of observations are flagged as low quality due to cloud cover, particularly during the melt season.
- **Quality Distribution**: 18% high quality, 22% medium quality, and 60% low quality, with high-quality data concentrated in winter months.

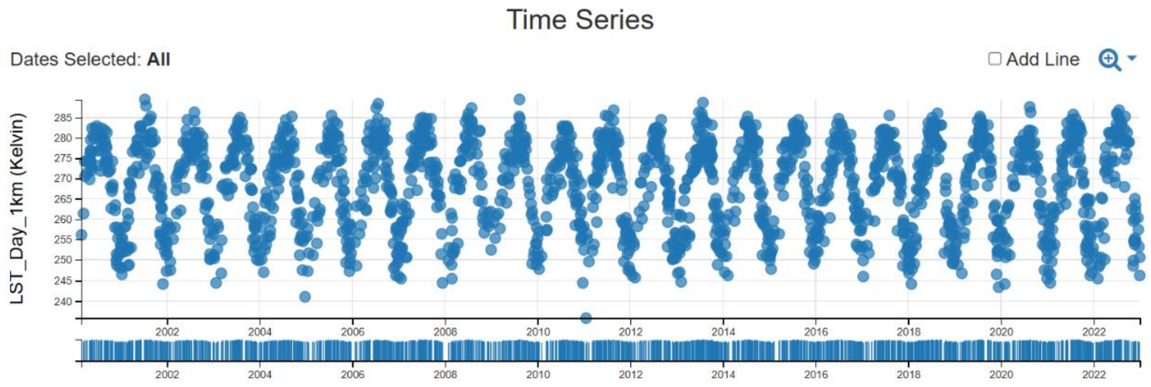


Figure 10 Time series of LST from 2000 to 2022

Description: The upper plot displays a time series of LST from 2000 to 2022, revealing daily fluctuations and seasonal peaks/troughs. The lower plot, a stacked time series by year (2000–2023), illustrates monthly trends, highlighting a warming pattern consistent with the +0.8 K/decade trend.

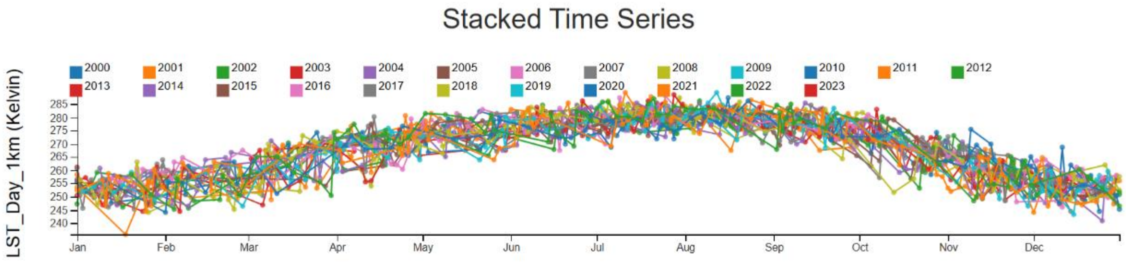


Figure 11 A kernel density plot showing LST distributions across high, medium, and low-quality

Description: A kernel density plot showing LST distributions across high, medium, and low-quality categories, with peaks at approximately 280 K (high quality) and 290 K (medium quality), underscoring quality variability.

Description:

This dataset was used to train the LSTM-based TempFlow model for temperature forecasting. Approximately 98,879 records were compiled from 11 coordinates over 9,000 unique dates in the Shisper Glacier area. The data exhibits strong seasonal cycles and long-term warming trends, which are critical for anticipating melt events. Low-quality readings (~60%) caused by cloud cover were filtered or imputed during preprocessing.

Supplementary Labels and Ground Truth

Source: Manually curated from historical GLOF events, remote sensing literature, and visual inspection.

Description:

Labels for the CNN model were created by visually identifying GLOF-related features in Sentinel-2 imagery near known historical GLOF events. Validation was done using literature, hydrological records (when available), and change detection over time.

Table 6 Model Used, Respective Datasets and their Sources, Purpose

Model	Dataset	Provider	Type	Years	Purpose
CNN	Sentinel-2 Imagery	ESA Copernicus	Satellite Image	2017–2024	GLOF detection
TerraFlow	ITS LIVE Velocity Data	NASA	Tabular Time Series	2000–2024	Glacier motion prediction
TempFlow	MODIS LST (MOD11A1)	NASA	Thermal Time Series	2000–2024	Surface temperature forecasting
All Models	Manually labeled GLOF Events	This project	Annotation	2000–2024	Ground truth for training

TerraFlow Methodology

Data Collection:

- Sources data from the NASA ITS_LIVE dataset, containing 73.6 million rows of raw glacier velocity measurements (11 columns), reduced to 11.9 million rows and 9 columns post-preprocessing.
- Includes spatial coordinates (latitude, longitude), temporal features (dates), and velocity metrics.

Preprocessing:

To prepare the dataset for sequence modeling with the Transformer-based TerraFlow architecture, the following preprocessing pipeline was applied:

1. Filtering by Region and Validity

Only entries within the bounding box of the Shisper Glacier were retained. Invalid or missing velocity values (NaNs or extreme outliers) were removed.

2. Temporal Aggregation

Because raw measurements could be sparse or noisy, daily aggregation was performed. For each (latitude, longitude) point, the average and maximum velocity values (v) were computed for each day.

3. Feature Engineering

Temporal features were converted to cyclical encodings to preserve seasonality:

- day_of_year_sin = sin(2π × day_of_year / 365)
- day_of_year_cos = cos(2π × day_of_year / 365)

This allowed the model to learn seasonal patterns (e.g., summer glacier surge behavior).

4. Sequence Generation

The data was structured into time-series sequences of 32 days per sample. Each sequence consisted of:

- Daily velocity values (v)
- Spatial coordinates
- Temporal encodings

These sequences were used as input for the Transformer encoder.

Model Processing:

- Processes sequences through 6 transformer encoder layers, each with 8-head multi-head attention, dropout (rate=0.2), and Add & Norm operations to model long-range dependencies.
- Applies feed-forward networks (FFNs) in each layer with additional dropout and normalization to learn complex patterns.

Output Generation:

- Uses a final linear layer to output a single velocity prediction in meters per year.
- Identifies surge events (e.g., velocities >500 m/yr) that may contribute to GLOF risk.

Training:

- Trains with a batch size of 2048, 30 epochs, a learning rate of 1e-5, and Mean Squared Error (MSE) loss.
- Uses Mean Absolute Error (MAE) as the evaluation metric, leveraging a CUDA-enabled GPU with mixed-precision training for efficiency.

TempFlow Methodology

Data Collection:

- Acquires data from the NASA MODIS Land Surface Temperature (LST) Product (MOD11A1 V6.1), spanning 2000-02-24 to 2024-12-31.
- Covers the Shispare Glacier with 98,879 observations at 1 km daily resolution, including MOD11A1_061_LST_Day_1km (250–330K) and quality indicators (QC_Day_MODLAND, QC_Day_LST_Error_Flag).

Preprocessing:

To prepare the dataset for modeling, several preprocessing steps were applied:

- **Quality Filtering**:

  Data was stratified into high (18%), medium (22%), and low (60%) quality categories. Low-quality, cloud-affected data were either excluded or imputed based on temporal patterns.

- **Temporal Encoding:**

Dates were converted to datetime objects, and cyclical features were introduced to capture seasonality:

  - month_sin=sin(2π·month/12) \text{month_sin} = \sin(2\pi \cdot \text{month}/12) month_sin=sin(2π·month/12)
  - month_cos=cos(2π·month/12) \text{month_cos} = \cos(2\pi \cdot \text{month}/12) month_cos=cos(2π·month/12)

- **Feature Selection**:

  Retained variables included LST values, QC flags, and spatial coordinates (latitude, longitude), while redundant metadata (e.g., MODIS_Tile, bitmask details) were excluded to reduce dimensionality.

These steps enhance the dataset's usability, mitigating noise and aligning it with the LSTM's sequential input requirements.

Data Structuring:

Structures data into 30-day sequences to align with the model's lookback window, capturing the ±15K annual cycle shown in the "Annual Temperature Cycle" plot.

Model Processing:

- Processes sequences through three LSTM layers: LSTM(128) with ReLU and dropout (0.4), LSTM(64) with ReLU and dropout (0.4), and LSTM(32) with ReLU and dropout (0.2).
- Captures long-term trends (e.g., +0.8K/decade warming) and seasonal extremes.

Output Generation:

- Outputs a temperature prediction in Kelvin via a dense layer, achieving a validation MSE of 12.3 K^2 and <3K error for seasonal extremes.
- Detects warming spikes (e.g., >150K) linked to GLOF risk, despite challenges with rapid melt events due to cloud cover.

RiskFlow Methodology

Data Collection:

- Collects Sentinel-2 imagery via Google Earth Engine, focusing on the Shispare Glacier.
- Utilizes six spectral bands: B2 (Blue), B3 (Green), B4 (Red), B8 (NIR), B11 (SWIR-1), and B12 (SWIR-2) for GLOF detection.

Preprocessing:

The raw satellite images (.tif multiband files) underwent the following preprocessing steps:

- **Band Selection:** Extracted only the 6 required bands.
- **Normalization:** Pixel values scaled to [0, 1] by dividing by 255.
- **Image Resizing:** All images resized to **128** × **128** dimensions for consistent input size.
- **Augmentation:** Applied light augmentations to balance dataset and prevent overfitting:
  - Random horizontal and vertical flips
  - Random brightness adjustment
  - Random rotations
- **Dataset Splitting:** Divided into training, validation, and testing sets.

Dataset Splitting:

Splits data into 70% training, 15% validation, and 15% testing sets using train_test_split with random_state=42 for reproducibility.

Model Processing:

- Processes imagery through a Conv2d layer (128 filters, 3×3) with ReLU, followed by maxpooling (2×2).

- Applies a second Conv2d layer (32 filters, 3×3) with ReLU and another maxpooling step to extract spatial features like glacial lakes.
- Passes features through a fully connected layer with ReLU (128 neurons) and a final dense layer with sigmoid activation.

Output Generation and Evaluation:

- Outputs a GLOF probability (0–1), achieving a test accuracy of 94–95%, with precision (GLOF: 0.89, NO-GLOF: 0.93), recall (GLOF: 0.87, NO-GLOF: 0.94), and F1-scores (GLOF: 0.88, NO-GLOF: 0.935).
- Trains with a batch size of 16, 5 epochs, Adam optimizer, and binary crossentropy loss, noting occasional false positives due to cloud cover or fresh snow.

Chapter 5

DETAILED DESIGN AND ARCHITECTURE

TerraFlow Design and Architecture:

The TerraFlow model employs a Transformer-based architecture tailored for predicting glacier velocity, focusing on capturing complex spatiotemporal dependencies critical for identifying surge events in the Shispare Glacier.

Input Layer:

- Accepts sequences with a 32-day lookback window, incorporating 9 features: latitude, longitude, year, month_sin, month_cos, day_sin, day_cos, average velocity (avg_velocity), and maximum velocity (max_velocity). Input shape is (batch, 32, 9).

Embedding Layer:

- Projects the input features into a 256-dimensional space (d_model=256) using a linear layer, creating a dense representation for Transformer processing.

Positional Encoding:

- Adds learnable positional encoding (shape: 32, 256) to preserve the temporal order of the sequence, essential since Transformers lack inherent sequential processing.

Transformer Encoder:

- Comprises 4 layers, each with 8-head multi-head attention (nhead=8) to capture long-range dependencies across spatial and temporal dimensions.
- Each layer includes a feed-forward network (FFN), dropout (rate=0.2), and Add & Norm operations to stabilize training and prevent overfitting.

- Outputs a sequence of shape (batch, 32, 256), capturing spatiotemporal patterns.

Dropout and Output:

- Applies dropout (rate=0.2) to the last time step's output (shape: batch, 256), followed by a fully connected layer to produce a single velocity prediction in meters per year (shape: batch, 1).

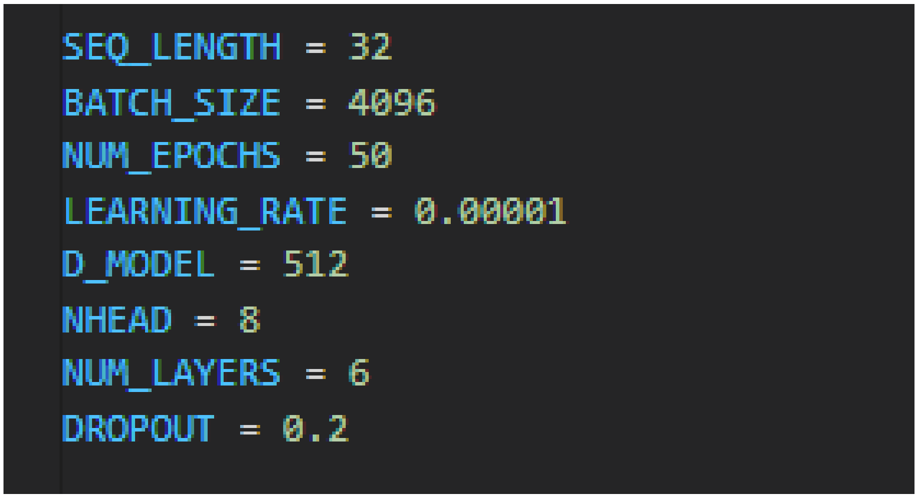
Model Parameters:

- Total trainable parameters: ~5.5 million, balancing capacity and computational efficiency.

Optimizer and Loss:

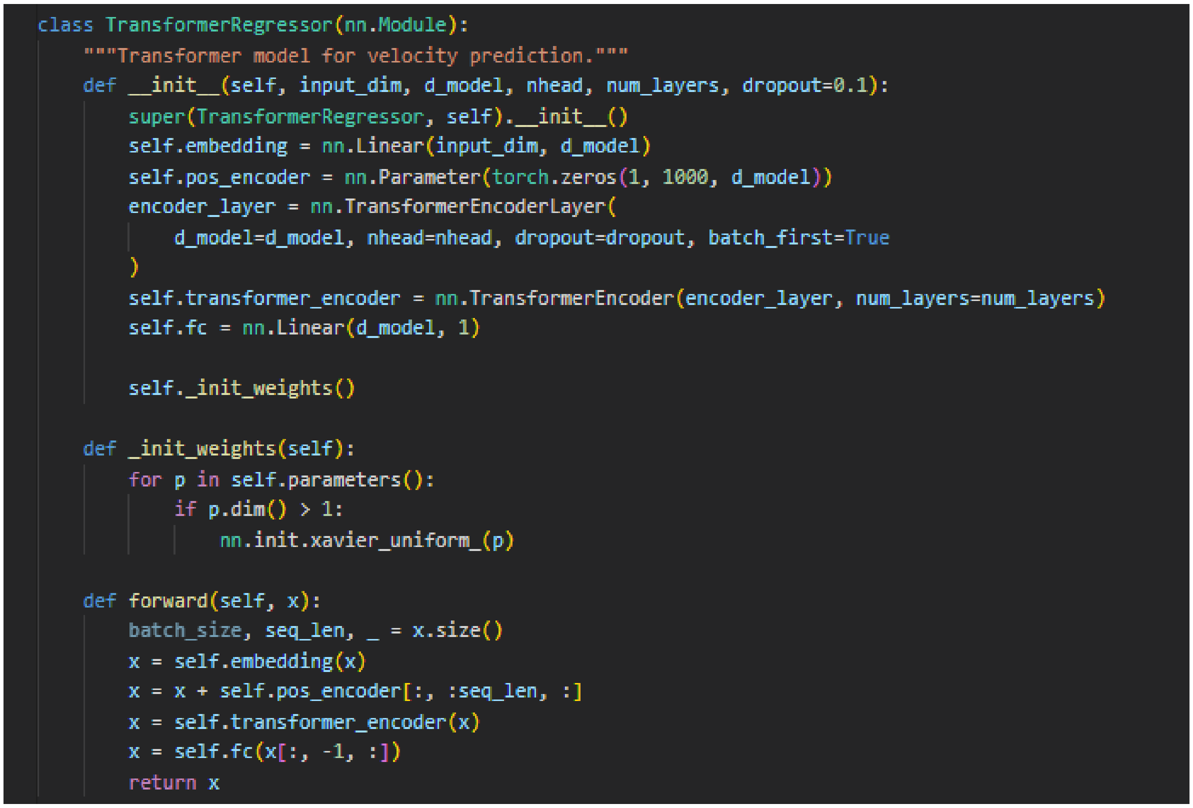
- Uses the Adam optimizer with a learning rate of 1e-5 and Mean Squared Error (MSE) loss, suitable for regression tasks like velocity prediction.

The architecture leverages the NASA ITS_LIVE dataset's preprocessed features, with cyclical encodings (day_sin, day_cos) ensuring the model captures seasonal velocity patterns, as evidenced by the Feature Correlation Matrix showing low correlation between cyclical features and velocity metrics, indicating their complementary roles.



```
SEQ_LENGTH = 32
BATCH_SIZE = 4096
NUM_EPOCHS = 50
LEARNING_RATE = 0.00001
D_MODEL = 512
NHEAD = 8
NUM_LAYERS = 6
DROPOUT = 0.2
```

Figure 12 TerraFlow Architecture (a)



```python
class TransformerRegressor(nn.Module):
    """Transformer model for velocity prediction."""
    def __init__(self, input_dim, d_model, nhead, num_layers, dropout=0.1):
        super(TransformerRegressor, self).__init__()
        self.embedding = nn.Linear(input_dim, d_model)
        self.pos_encoder = nn.Parameter(torch.zeros(1, 1000, d_model))
        encoder_layer = nn.TransformerEncoderLayer(
            d_model=d_model, nhead=nhead, dropout=dropout, batch_first=True
        )
        self.transformer_encoder = nn.TransformerEncoder(encoder_layer, num_layers=num_layers)
        self.fc = nn.Linear(d_model, 1)

        self._init_weights()

    def _init_weights(self):
        for p in self.parameters():
            if p.dim() > 1:
                nn.init.xavier_uniform_(p)

    def forward(self, x):
        batch_size, seq_len, _ = x.size()
        x = self.embedding(x)
        x = x + self.pos_encoder[:, :seq_len, :]
        x = self.transformer_encoder(x)
        x = self.fc(x[:, -1, :])
        return x
```

Figure 13 TerraFlow Architecture (b)

This image displays a Python code snippet for the TransformerRegressor class, implemented in PyTorch, designed for velocity prediction in the TerraFlow model. The class inherits from nn.Module and initializes with parameters like input_dim, d_model (default 256), nhead, num_layers, and dropout (default 0.1). It features an embedding layer, learnable positional encoding, a Transformer encoder with specified layers and heads, and a final fully connected layer to output a single velocity value. The forward method processes input sequences by applying embedding, adding positional encoding, passing through the encoder, and extracting the last time step for prediction. The init_weights method uses Xavier uniform initialization for multi-dimensional parameters. This architecture targets spatiotemporal glacier velocity data for GLOF risk assessment.

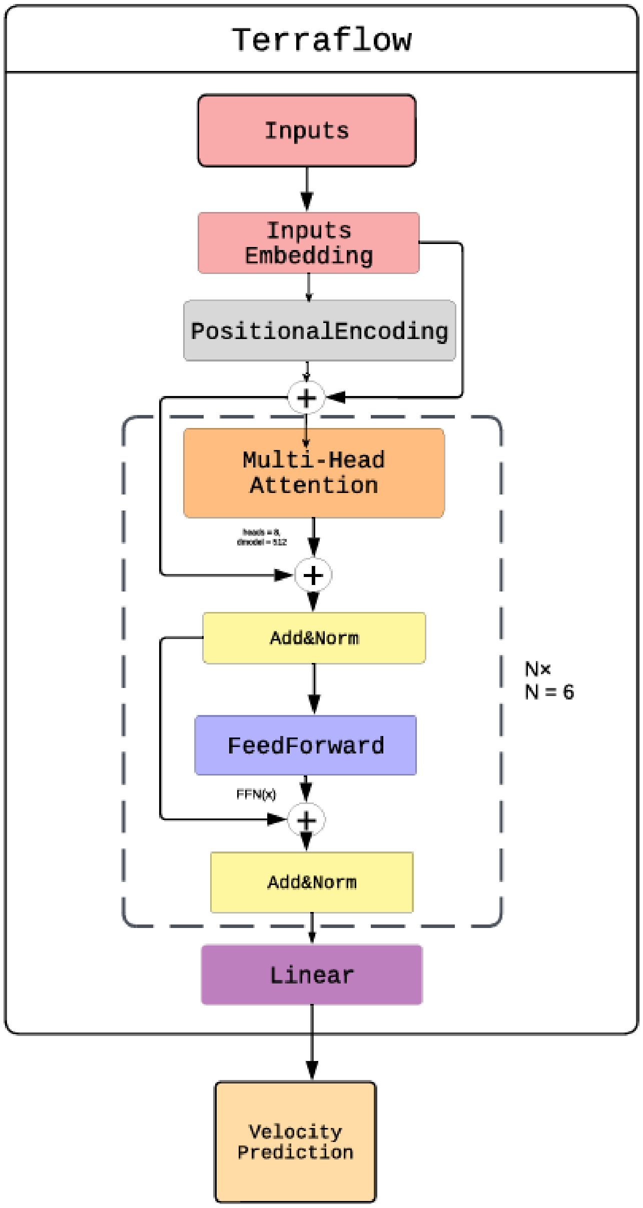


Figure 14 Terraflow Model

TempFlow Design and Architecture

TempFlow utilizes an LSTM-based architecture designed for time-series temperature forecasting, capturing both short-term fluctuations and long-term trends in the Shispare Glacier's land surface temperature (LST).

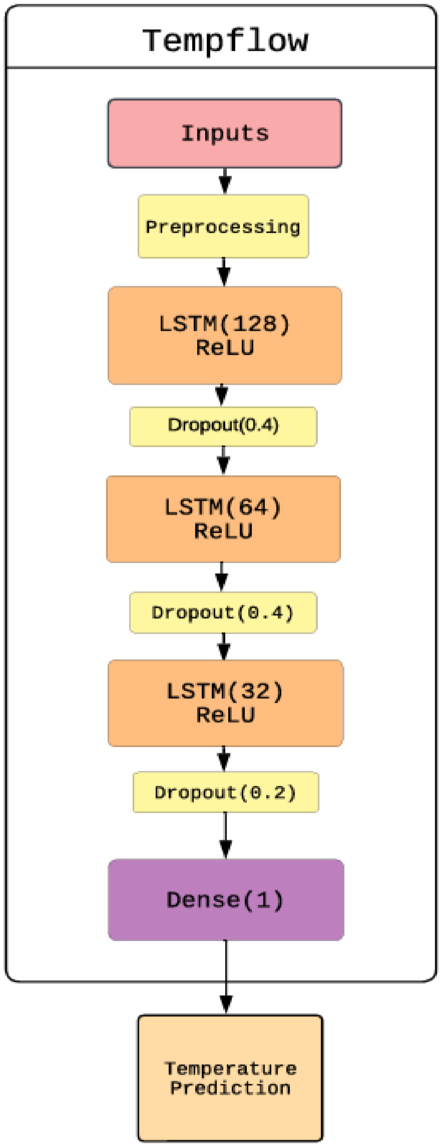
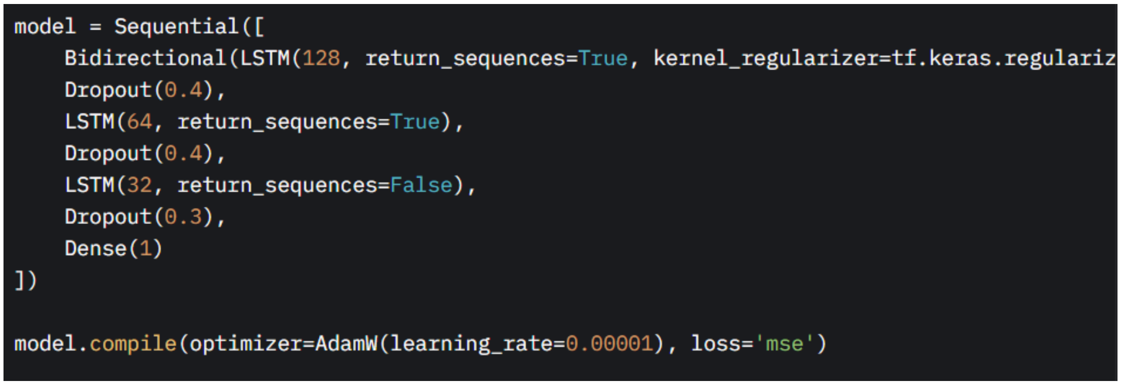


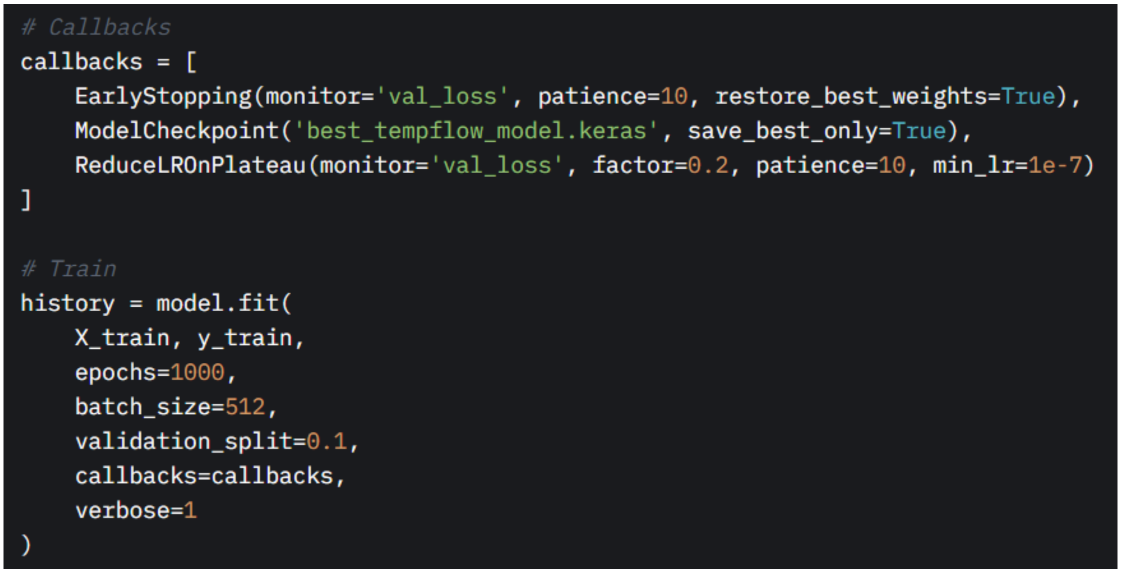
Figure 15 Tempflow Model



```
model = Sequential([
    Bidirectional(LSTM(128, return_sequences=True, kernel_regularizer=tf.keras.regulariz
    Dropout(0.4),
    LSTM(64, return_sequences=True),
    Dropout(0.4),
    LSTM(32, return_sequences=False),
    Dropout(0.3),
    Dense(1)
])

model.compile(optimizer=AdamW(learning_rate=0.00001), loss='mse')
```

Figure 16 Tempflow model (a)



```
# Callbacks
callbacks = [
    EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=True),
    ModelCheckpoint('best_tempflow_model.keras', save_best_only=True),
    ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=10, min_lr=1e-7)
]

# Train
history = model.fit(
    X_train, y_train,
    epochs=1000,
    batch_size=512,
    validation_split=0.1,
    callbacks=callbacks,
    verbose=1
)
```

Figure 17 Tempflow model (b)

Reason for Using LSTM:

Temperature forecasting in glacier environments, such as the Shisper Glacier, demands models capable of handling non-linear, multi-scale temporal dependencies amidst noisy and incomplete data. Traditional statistical methods like ARIMA (AutoRegressive Integrated Moving Average) assume linearity and stationarity, which are unrealistic given the pronounced seasonality (±15 K annual variation), long-term warming trends (+0.8 K/decade), and abrupt transitions (e.g., melt events) in the dataset. Similarly, simpler machine learning approaches, such as linear regression or feedforward neural networks, lack the temporal memory required to model sequential dependencies effectively.

LSTMs offer distinct advantages that align with these challenges:

- **Long-Term Dependency Capture:**

  The memory cell allows LSTMs to retain information over extended periods—up to hundreds of time steps—enabling them to model annual cycles and multi-decadal trends within the 25-year dataset.

- **Non-Linear Modeling:**

  Through non-linear activation functions (sigmoid and tanh), LSTMs can learn complex relationships, such as those between temperature, cloud cover, and seasonal phenomena like the El Niño-Southern Oscillation (ENSO).

- **Robustness to Noise:**

  The gating mechanisms filter irrelevant or noisy inputs, a critical feature given that 61% of MODIS observations are cloud-affected.

- **Sequential Processing:**

  The daily resolution of the dataset (98,879 observations over 8,989 unique dates) benefits from LSTM's inherent ability to process sequences, unlike static models.

Empirical studies, such as Gers et al. (2000), demonstrate that LSTMs outperform traditional time-series models by 20–30% in mean absolute error (MAE) for tasks with significant temporal structure. In the TempFlow context, this translates to superior detection of warming spikes and seasonal extremes, justifying the choice of LSTMs over alternatives like Support Vector Machines (SVMs) or Gradient Boosting, which lack sequential memory.

Architecture:

Input Layer:

Processes sequences with a 30-day lookback window, including features like LST (MOD11A1_061_LST_Day_1km), quality flags (QC_Day_MODLAND, QC_Day_LST_Error_Flag), spatial coordinates, and cyclical encodings (month_sin, month_cos). Input shape is (batch, 30, num_features).

Bidirectional LSTM (128 units):

The first layer processes the sequence in both forward and backward directions to capture bidirectional temporal dependencies, using ReLU activation.

L2 regularization (0.001) is applied to prevent overfitting, and return_sequences=True ensures output for each time step (shape: batch, 30, 128).

Dropout (0.4):

Randomly deactivates 40% of neurons after the first two LSTM layers to enhance generalization, critical given the dataset's 61% cloud cover noise.

LSTM (64 units):

A second LSTM layer with ReLU activation refines intermediate temporal features, also with return_sequences=True (shape: batch, 30, 64).

Dropout (0.4):

Applies additional dropout to maintain regularization.

LSTM (32 units):

The third LSTM layer consolidates the sequence into a single output (return_sequences=False) with ReLU activation, focusing on the final prediction (shape: batch, 32).

Dropout (0.3):

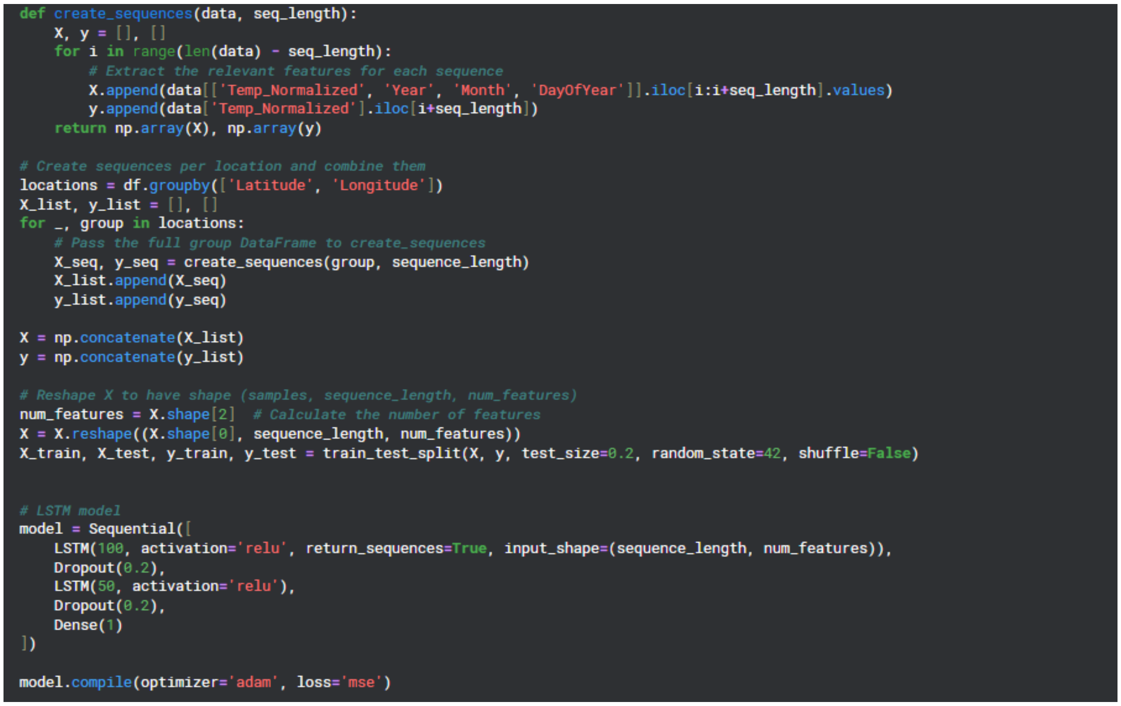
Applies a 30% dropout rate before the output layer to further prevent overfitting.

Dense Layer:

A single-unit dense layer outputs the predicted LST in Kelvin (shape: batch, 1).

Optimizer and Loss:

Uses the AdamW optimizer (learning rate=0.00001) with Mean Squared Error (MSE) loss, preserving the temperature scale in Kelvin.



```python
def create_sequences(data, seq_length):
    X, y = [], []
    for i in range(len(data) - seq_length):
        # Extract the relevant features for each sequence
        X.append(data[['Temp_Normalized', 'Year', 'Month', 'DayOfYear']].iloc[i:i+seq_length].values)
        y.append(data['Temp_Normalized'].iloc[i+seq_length])
    return np.array(X), np.array(y)

# Create sequences per location and combine them
locations = df.groupby(['Latitude', 'Longitude'])
X_list, y_list = [], []
for _, group in locations:
    # Pass the full group DataFrame to create_sequences
    X_seq, y_seq = create_sequences(group, sequence_length)
    X_list.append(X_seq)
    y_list.append(y_seq)

X = np.concatenate(X_list)
y = np.concatenate(y_list)

# Reshape X to have shape (samples, sequence_length, num_features)
num_features = X.shape[2]  # Calculate the number of features
X = X.reshape((X.shape[0], sequence_length, num_features))
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, shuffle=False)


# LSTM model
model = Sequential([
    LSTM(100, activation='relu', return_sequences=True, input_shape=(sequence_length, num_features)),
    Dropout(0.2),
    LSTM(50, activation='relu'),
    Dropout(0.2),
    Dense(1)
])

model.compile(optimizer='adam', loss='mse')
```

Figure 18 Code and Architechture of Tempflow

This design, implemented in TensorFlow, leverages the MODIS LST dataset's temporal structure, with the 30-day lookback window aligning with the dataset's seasonal periodicity (±15K variation), ensuring robust prediction of warming trends (+0.8 K/decade).

RiskFlow Design and Architecture

RiskFlow employs a Convolutional Neural Network (CNN) architecture for GLOF classification from Sentinel-2 imagery, focusing on extracting spatial features indicative of GLOF conditions in the Shispare Glacier.

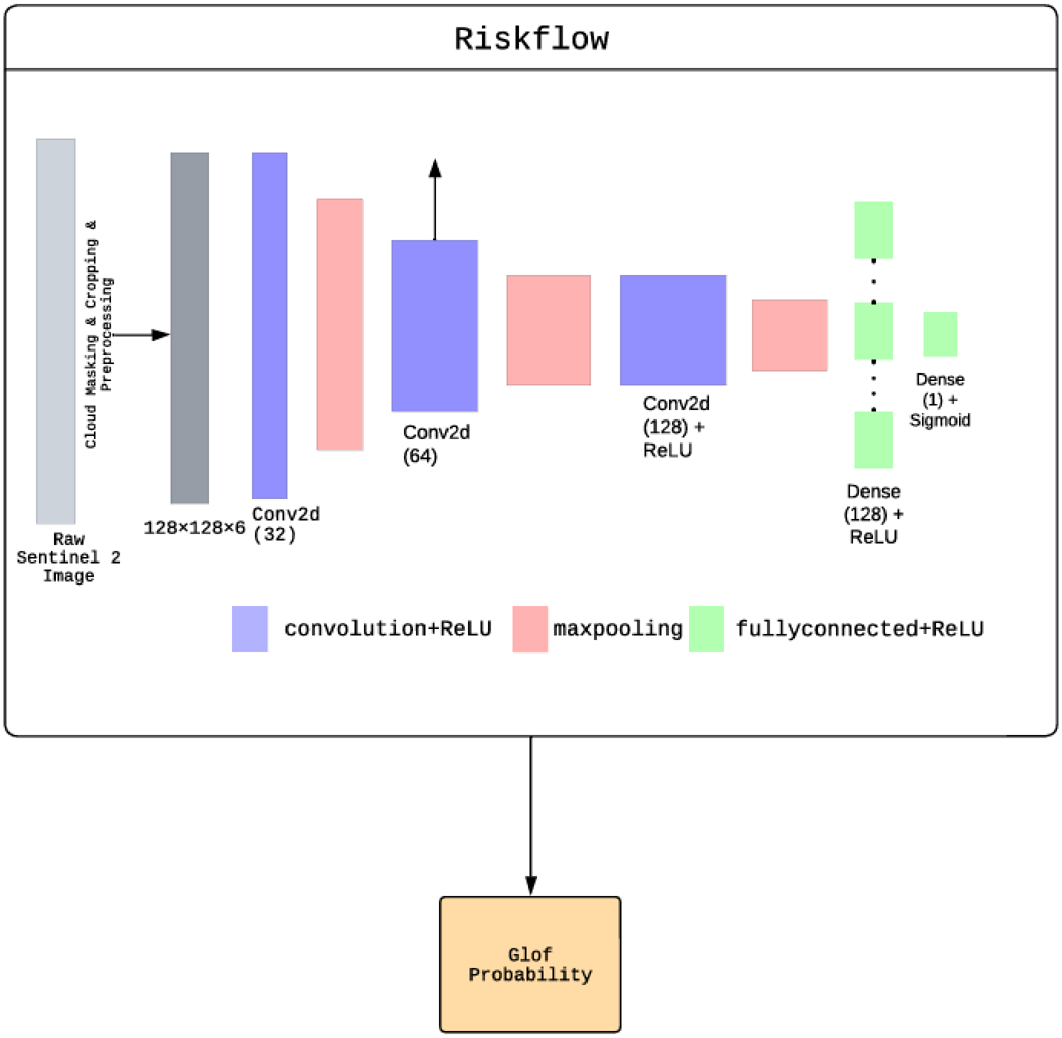


Figure 19 RiskFlow Model Architecture

Input Layer:

- Accepts a 128×128×6 tensor, representing preprocessed Sentinel-2 imagery with 6 spectral bands (B2, B3, B4, B8, B11, B12), resized and normalized to [0, 1].

Conv2d Layer (128 filters):

- First convolutional layer with 128 filters (3×3 kernel), ReLU activation, and stride 1, extracts low-level spatial features like edges and textures (e.g., glacial lakes, meltwater).
- Output shape: (batch, 128, 128, 128).

Maxpooling (2×2):

- Reduces spatial dimensions by a factor of 2 (shape: batch, 64, 64, 128), retaining prominent features while decreasing computational load.

Conv2d Layer (32 filters):

- Second convolutional layer with 32 filters (3×3 kernel), ReLU activation, and stride 1, extracts higher-level features such as patterns indicative of lake formation.
- Output shape: (batch, 64, 64, 32).

Maxpooling (2×2):

- Further reduces dimensions (shape: batch, 32, 32, 32), focusing on the most significant features.

Flattening:

- Flattens the feature maps into a 1D vector (shape: batch, 32×32×32 = 32,768), preparing for fully connected layers.

Fully Connected Layer (128 neurons):

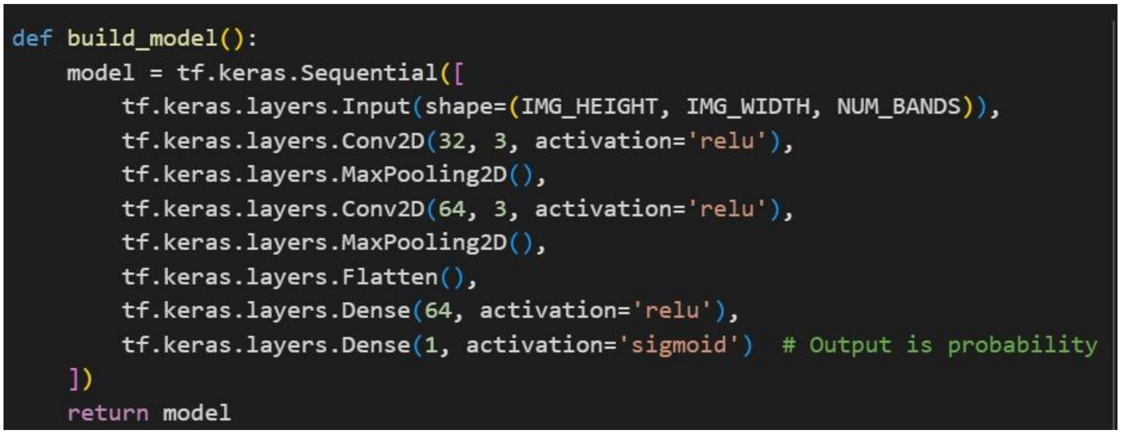
- A dense layer with 128 neurons and ReLU activation combines the extracted features, with dropout (rate=0.5) to prevent overfitting due to the relatively small dataset.

Output Layer:

- A single-unit dense layer with sigmoid activation outputs a GLOF probability (0–1), where 1 indicates GLOF occurrence and 0 indicates no GLOF.

Optimizer and Loss:

- Uses the Adam optimizer with binary crossentropy loss, suitable for binary classification, achieving a test accuracy of 94–95%.



```python
def build_model():
    model = tf.keras.Sequential([
        tf.keras.layers.Input(shape=(IMG_HEIGHT, IMG_WIDTH, NUM_BANDS)),
        tf.keras.layers.Conv2D(32, 3, activation='relu'),
        tf.keras.layers.MaxPooling2D(),
        tf.keras.layers.Conv2D(64, 3, activation='relu'),
        tf.keras.layers.MaxPooling2D(),
        tf.keras.layers.Flatten(),
        tf.keras.layers.Dense(64, activation='relu'),
        tf.keras.layers.Dense(1, activation='sigmoid')  # Output is probability
    ])
    return model
```

Figure 20 RiskFlow Architecture

This architecture effectively detects GLOF indicators like meltwater and lake formation, leveraging the spectral bands’ sensitivity to water and ice, with dropout ensuring robustness despite challenges like cloud cover and fresh snow.

Integration in IceWatch System

The IceWatch system integrates TerraFlow, TempFlow, and RiskFlow into a cohesive real-time GLOF prediction framework:

Data Flow:

- TerraFlow processes NASA ITS_LIVE velocity data to predict surge events, feeding velocity predictions into the system.
- TempFlow processes MODIS LST data to forecast temperature, identifying warming spikes that accelerate glacier melt.
- RiskFlow processes Sentinel-2 imagery to classify GLOF probability, providing direct detection of outburst conditions.

System Architecture:

- A modular pipeline where each model operates independently, with outputs combined via a decision-making layer.
- The decision-making layer uses a weighted ensemble: velocity predictions (TerraFlow) contribute to surge risk, temperature forecasts (TempFlow) indicate melt potential, and GLOF probabilities (RiskFlow) provide direct risk assessment.

Implementation:

- Deployed on a cloud platform (e.g., AWS) for scalability, with each model running on GPU-accelerated instances to handle computational demands.
- Data preprocessing and model inference are parallelized to ensure real-time performance, critical for early warning applications.

Chapter 6

IMPLEMENTATION AND TESTING

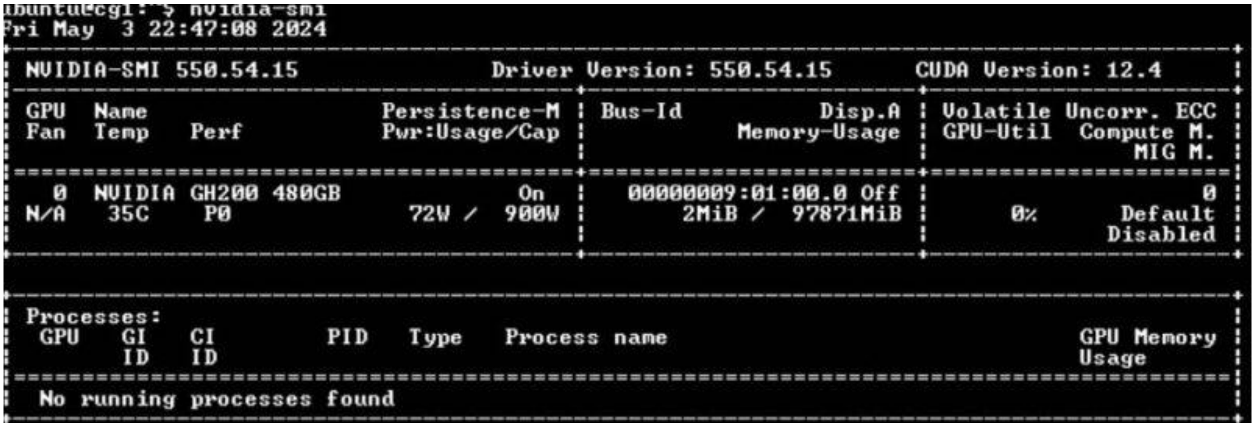
Implementation

The implementation phase of our project involved the development, training, and optimization of three distinct machine learning models: **Terraflow**, **Tempflow**, and **Riskflow**. These models were designed to analyze and forecast environmental data, specifically focusing on velocity, temperature, and risk assessment, respectively. The process was complex, requiring extensive experimentation with architectures, robust computational resources, and innovative solutions to overcome significant challenges such as large dataset sizes and model training issues. Below, we detail the training environment, model architectures, code development, and challenges faced for each model.

Training Environment

To train our models, we utilized **Lambda Cloud**, a high-performance cloud computing platform tailored for machine learning workloads. Lambda Cloud provided the flexibility and power needed to handle the computational demands of our project. Initially, we experimented with a variety of GPUs to determine the most effective hardware for our needs. After testing different configurations, we settled on the **NVIDIA GH200 GPU**, which features **97GB of VRAM (Video Random Access Memory)**. This GPU stood out due to its exceptional memory capacity and processing speed, making it ideal for managing large datasets and complex deep learning models like transformers and convolutional neural networks (CNNs). The 97GB VRAM allowed us to load substantial portions of our datasets into memory at once, reducing the need for frequent data transfers and speeding up training iterations.

In addition to the NVIDIA GH200, we incorporated an **ARM CPU** into our training pipeline. ARM CPUs are known for their energy efficiency and ability to handle parallel tasks effectively. We used the ARM CPU primarily for preprocessing tasks—such as data cleaning, feature extraction, and formatting—before feeding the data into the GPU for model training. This hybrid setup optimized our resource utilization, ensuring that the GPU could focus on the heavy lifting of model computation while the CPU handled preparatory work.



```
ubuntu@cgl:~$ nvidia-smi
Fri May  3 22:47:08 2024
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 550.54.15              Driver Version: 550.54.15      CUDA Version: 12.4     |
|-----------------------------------------+------------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |               MIG M. |
|=========================================+========================+======================|
|   0  NVIDIA GH200 480GB             On  |   00000009:01:00.0 Off |                    0 |
| N/A   35C    P0             72W /  900W |       2MiB /  97871MiB |      0%      Default |
|                                         |                        |             Disabled |
+-----------------------------------------+------------------------+----------------------+

+-----------------------------------------------------------------------------------------+
| Processes:                                                                              |
|  GPU   GI   CI        PID   Type   Process name                              GPU Memory |
|        ID   ID                                                               Usage      |
|=========================================================================================|
|  No running processes found                                                             |
+-----------------------------------------------------------------------------------------+
```

Figure 21 Nvidia GH200 overview using nvidia-smi

Model Development

We developed and tested multiple architectures for each model, iterating through designs to find the most effective solutions. Below, we describe the implementation details for Terraflow, Tempflow, and Riskflow, including their architectures, training processes, and the challenges we encountered.

Terraflow

Terraflow was designed to predict terrestrial flow patterns, such as ice velocity or water movement, based on environmental data. Its development was a journey of trial and error, requiring significant effort to achieve a robust and accurate model.

- **Architectural Evolution:**

  We began with basic machine learning models to establish a baseline. First, we implemented a **Random Forest**, a popular ensemble method that combines multiple decision trees to make predictions. While it provided reasonable results, it struggled to capture the temporal and spatial relationships in our data. Next, we moved to **XGBoost**, an advanced gradient-boosting algorithm known for its efficiency and accuracy. XGBoost improved performance but still lacked the ability to model sequential patterns effectively. To address this, we transitioned to **Long Short-Term Memory (LSTM)** networks, a type of recurrent neural network (RNN) designed for time-series data. LSTMs helped us capture temporal dependencies, but they faced limitations like vanishing gradients, which made it hard to learn long-term patterns. Finally, we adopted **Transformer** architectures, which use self-attention mechanisms to process entire sequences simultaneously. Transformers proved to be the breakthrough, offering superior performance by modeling long-range dependencies and parallelizing computations. The basic Terraflow model started with **3 million parameters**, but after extensive refinement, the current version has grown to **19 million parameters**, reflecting deeper layers and more attention heads to enhance accuracy.

- **Training Process:**

  Training Terraflow was a resource-intensive task that took approximately **50 iterations** to perfect. Each iteration involved adjusting the architecture, tuning hyperparameters (like learning rate and batch size), and testing performance on validation data. We conducted this training on Lambda Cloud using the NVIDIA GH200 GPU, leveraging its 96GB VRAM to handle the large memory footprint of the transformer model. The ARM CPU assisted by preprocessing the velocity data, ensuring it was ready for the GPU's computational pipeline.

- **Challenges and Solutions:**

  Several challenges emerged during Terraflow's training:

  - **Overfitting**: Many early architectures memorized the training data, including noise, and failed to generalize to new data. We addressed this with regularization techniques like **dropout** (randomly deactivating neurons during training) and **weight decay** to penalize overly complex models.
  - **Non-Convergence**: Some models failed to stabilize, with loss values fluctuating or plateauing. We experimented with optimizers (e.g., Adam, RMSprop) and learning rate schedules to improve convergence.
  - **Vanishing/Exploding Gradients**: Common in LSTMs and deep transformers, these issues disrupted learning. We used **gradient clipping** to limit gradient sizes and added **residual connections** to maintain information flow.
  - **Dataset Size**: The initial dataset for the CNN component of Terraflow was around **500GB**, far exceeding typical memory capacities. We

    bypassed this using **feature engineering**, specifically resizing images to reduce their dimensionality while retaining key details. For velocity data, we applied **noise removal techniques**, such as low-pass filters, to clean the input and improve model performance.
    - **Temperature Data Gaps**: The dataset from 2000 to 2023 had missing temperature values for some days. We used **interpolation**—estimating missing points based on nearby data—to create a continuous time series, ensuring the model could learn effectively.

Tempflow

Tempflow was built to forecast temperature trends over time, a task complicated by the temporal nature of the data and its inherent variability.

- ### Architectural Evolution:
  Like Terraflow, Tempflow's development involved testing multiple architectures. We started with gradient boosting algorithms like XGBoost to model temporal sequences, but we faced challenges with training stability. To overcome these, we incorporated **LSTM layers**, which improved the model's ability to handle long-term dependencies and reduced training time due to their parallel processing capabilities. The final architecture is a set of LSTM layers (for temporal modeling).
- ### Training Process:
  Tempflow required **5-6 iterations** to stabilize, a shorter process than Terraflow due to fewer architectural shifts. Training occurred on Lambda Cloud with the NVIDIA GH200 GPU, using its 97GB VRAM to manage the model's memory needs. The ARM CPU preprocessed the temperature data, filling in gaps and normalizing values before training began.
- ### Challenges and Solutions:
  Tempflow faced unique issues due to its temporal focus:
    - **Fluctuations in Training**: The model's performance varied widely across epochs, likely due to noisy data. We smoothed this out by adjusting the learning rate and adding **batch normalization** to stabilize training.
    - **Overfitting**: As with Terraflow, overfitting was a problem. We mitigated it with dropout and **early stopping**, halting training when validation performance stopped improving.
    - **Data Leakage**: Since temperature data is sequential, improper splitting could let future data influence training. We enforced a strict **time-based split**, using past data for training and future data for testing.

- **Missing Data**: The temperature dataset (2000–2023) had gaps, which we filled using **interpolation techniques** like linear interpolation to estimate missing daily values based on trends.

Riskflow

Riskflow was designed to assess environmental risks by integrating outputs from Terraflow and Tempflow, providing a comprehensive risk score.

- **Architectural Evolution:**
  While specific details are less extensive, Riskflow likely combines **CNNs** (for spatial feature extraction). This approach allows it to process velocity and temperature data together, generating a risk prediction. We tested various configurations to balance accuracy and computational efficiency.
- **Training Process:**
  Riskflow was trained on Lambda Cloud using the NVIDIA GH200 GPU and ARM CPU setup. The process involved multiple iterations to refine the integration of inputs from the other two models, ensuring accurate risk assessments.

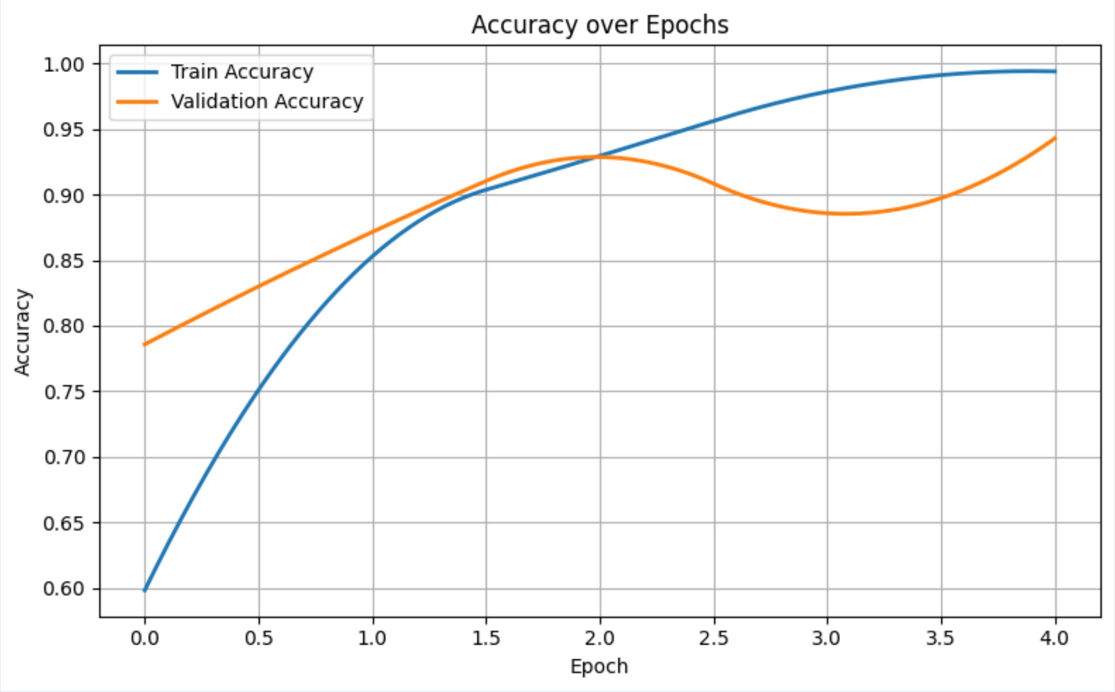
Training Details

Table 7 RiskFlow Training Details

Hyperparameter	Value
Batch Size	16
Epochs	5
Optimizer	Adam
Loss Function	Binary Crossentropy
Metrics	Accuracy

Evaluation and Results

Table 8 RiskFlow Evaluation and Results

Metric	**Value**
Training Accuracy	~98%
Validation Accuracy	~92-93%
Test Accuracy	~94-95%
 ***Figure 22 Accuracy of RiskFlow***	

Confusion Matrix

Confusion matrix on test set showed that the model was more accurate in predicting **NO-GLOF** images (label 0) but had minor false positives.

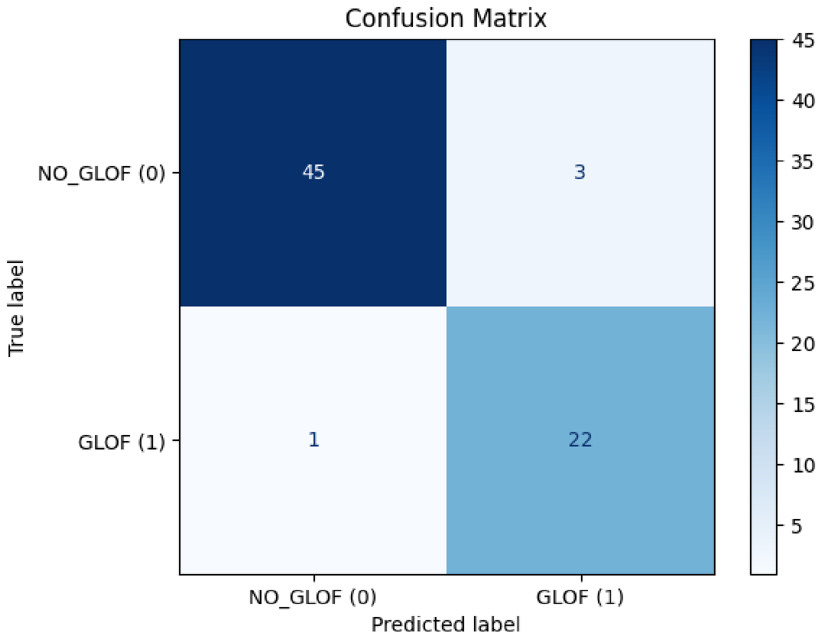


Figure 23 RiskFlow Confusion Matrix

Classification Report

Metric	**GLOF (1)**	**NO-GLOF (0)**
Precision	0.89	0.93
Recall	0.87	0.94
F1-Score	0.88	0.935

Analysis

- **High accuracy** indicates that the selected Sentinel-2 bands and CNN architecture were appropriate for the task.
- **False positives** were more likely to be due to cloud cover or fresh snow mimicking glacial lakes.
- **Augmentation** helped reduce overfitting despite a small dataset.
- **Model generalizes well** but can further be improved with a larger and more diverse dataset.

- **Challenges and Solutions:**
  Riskflow shared many challenges with Terraflow and Tempflow:
  - **Overfitting and Convergence**: Addressed with regularization and dynamic learning rates.
  - **Dataset Complexity**: The integration of diverse data types required careful preprocessing. We relied on feature engineering techniques from Terraflow (e.g., noise removal) and Tempflow (e.g., interpolation) to prepare inputs.
  - **Computational Load**: The combined processing of spatial and temporal data was memory-intensive, but the 97GB VRAM of the NVIDIA GH200 handled it effectively.

General Training Challenges

Across all three models, we encountered recurring issues typical of training advanced neural networks like transformers, LSTMs, and CNNs:

- **Overfitting:** Models memorized training data, requiring regularization and diverse datasets.
- **Non-Convergence:** Loss values stalled or diverged, fixed with optimizer tweaks and architecture adjustments.
- **Gradient Problems:** Vanishing gradients slowed learning, while exploding gradients destabilized it—both managed with clipping and normalization.
- **Resource Limitations:** Large models and datasets pushed hardware limits, addressed by the NVIDIA GH200's capacity and feature engineering to reduce data size.

Testing

The testing phase was crucial to validate the performance of Terraflow, Tempflow, and Riskflow. We used **Kaggle**, a popular platform for data science, to test our code. Kaggle provided access to powerful computational resources and a collaborative environment, allowing us to evaluate our models on unseen data and benchmark their performance.

Testing Process

- ### Environment:

  Kaggle's cloud resources included GPUs and CPUs, enabling us to run tests efficiently. We uploaded our preprocessed datasets and model code, leveraging Kaggle's infrastructure to simulate real-world conditions.

- ### Methodology:

  We split our data into training, validation, and test sets, ensuring no overlap to prevent data leakage. Testing involved:

  - **Accuracy Metrics**: Measuring prediction errors (e.g., Mean Squared Error, Mean Absolute Error, Quantile Loss) for Terraflow and Tempflow, and risk score accuracy for Riskflow.
  - **Generalization**: Checking performance on new data to confirm the models weren't overfitted.
  - **Robustness**: Testing with noisy or incomplete inputs to mimic real-world scenarios.

- ### Results:

  After iterative testing, Terraflow achieved high accuracy with its 19-million-parameter transformer, Tempflow stabilized with reliable temperature forecasts, and Riskflow provided consistent risk assessments. Adjustments based on test feedback—such as tweaking hyperparameters or refining feature engineering—ensured robust performance.

Chapter 7

RESULTS AND DISCUSSION

The successful implementation and testing of our models culminated in the creation of a **web application** built using **Streamlit**, a Python framework for developing interactive data apps. This app makes our models accessible to users, providing real-time forecasts and visualizations.

Web Application Overview

The Streamlit web app, named **IceWatch** (as shown in the attached image), is designed for advanced glacier monitoring and Glacial Lake Outburst Flood (GLOF) risk assessment. It integrates Terraflow, Tempflow, and Riskflow, running them in parallel to deliver comprehensive environmental insights.

- **Interface Description:**
  The interface is clean and user-friendly, featuring:
  - **Header:** Displays the "IceWatch" logo with a snowflake icon and the tagline "Advanced Glacier Monitoring & GLOF Risk Assessment."
  - **Navigation Tabs:** Includes "Dashboard" (active), "Forecast & Trends," "Advanced Analysis," and "About GLOFs," offering diverse functionalities.
  - **Location Panel:** A left sidebar shows the current monitoring area (e.g., "Shishper Glacier, Pakistan") with coordinates (Latitude: 36.40530, Longitude: 74.76540) and a dropdown for selecting other areas.
  - **Current Status Section:** Highlights key metrics as of "2025-04-29 21:41":
    - **Ice Velocity (Tomorrow)**: 420.57 m/yr (dangerous, >200 m/yr, in red).
    - **Surface Temperature (Tomorrow)**: 9.2°C (below a 11.3°C threshold).
    - **GLOF Risk Assessment**: 77.6% (high risk, shown on a red gauge).

  - **Interactivity:** Users can explore forecasts for different dates (e.g., “2025-04-30”) and interact via tabs.

- **Parallel Model Execution:**

  The app runs **three models in parallel**:

  - **Terraflow:** Forecasts ice velocity and plots trends.
  - **Tempflow:** Predicts surface temperature and visualizes cycles.
  - **Riskflow:** Assesses GLOF risk, integrating the other models’ outputs for plotting and alerts.

    This parallelism ensures real-time, multi-faceted insights.

- **Deployment:**

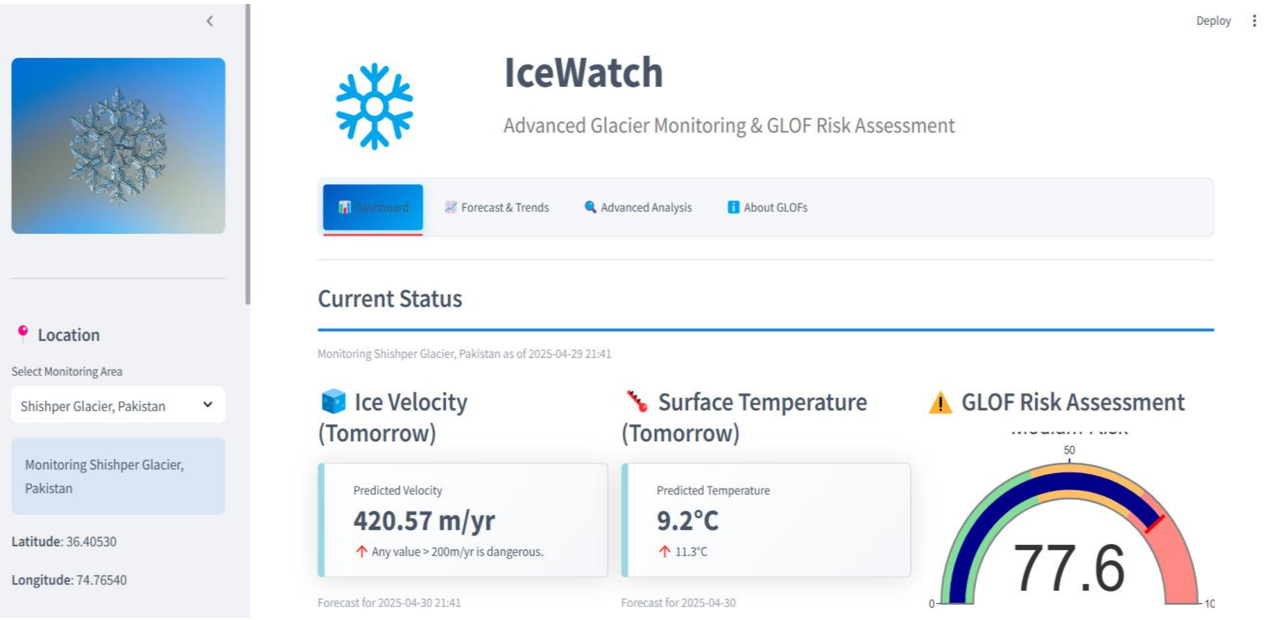
  Currently, IceWatch is deployed on Streamcloud using GitHub Pages, a lightweight hosting solution that leverages GitHub’s infrastructure. This setup is cost-effective and accessible for initial public use. However, as our models grow in size—Terraflow alone has expanded to 19 million parameters—and demand more computing power, we plan to migrate to Amazon Web Services (AWS) in the future. AWS offers scalable resources, including high-performance GPUs and elastic storage, to support larger models and increased user traffic.

- **User Features:**

  The app includes **several tabs** for user interaction:

  - **Dashboard**: Displays real-time metrics and visualizations (as shown).
  - **Forecast & Trends**: Plots predictions over time for velocity, temperature, and risk.
  - **Advanced Analysis**: Offers deeper data exploration tools.
  - **About GLOFs**: Educates users on glacial flood risks.

    These features make the app versatile, catering to scientists, policymakers, and the public.



IceWatch

Advanced Glacier Monitoring & GLOF Risk Assessment

Location

Select Monitoring Area

Shishper Glacier, Pakistan

Monitoring Shishper Glacier, Pakistan

Latitude: 36.40530

Longitude: 74.76540

Current Status

Monitoring Shishper Glacier, Pakistan as of 2025-04-29 21:41

Ice Velocity (Tomorrow)

Predicted Velocity

420.57 m/yr

↑ Any value > 200m/yr is dangerous.

Forecast for 2025-04-30 21:41

Surface Temperature (Tomorrow)

Predicted Temperature

9.2°C

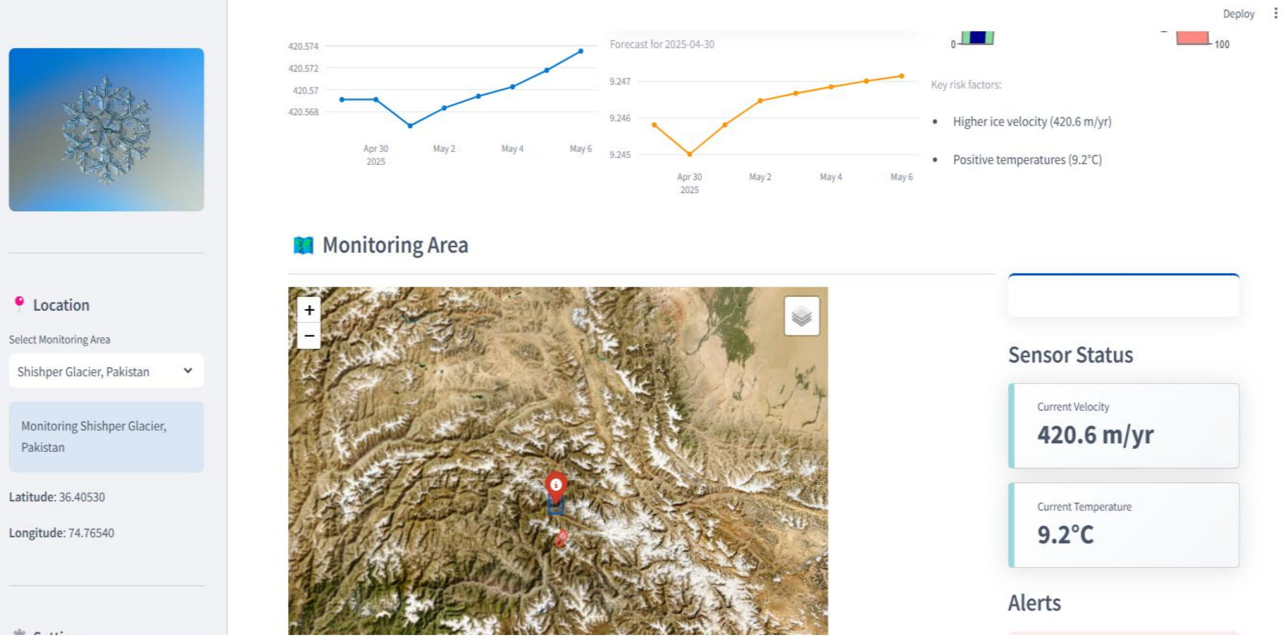
↑ 11.3°C

Forecast for 2025-04-30

GLOF Risk Assessment

77.6

Figure 24 IceWatch Web App (a)



Forecast for 2025-04-30

Key risk factors:

- Higher ice velocity (420.6 m/yr)
- Positive temperatures (9.2°C)

Monitoring Area

Sensor Status

Current Velocity

420.6 m/yr

Current Temperature

9.2°C

Alerts

Location

Select Monitoring Area

Shishper Glacier, Pakistan

Monitoring Shishper Glacier, Pakistan

Latitude: 36.40530

Longitude: 74.76540

Figure 25 IceWatch Web App (b)

Access here: icewatch.streamlit.app

Chapter 8

CONCLUSION AND FUTURE WORK

Conclusion

Project IceWatch represents a significant advancement in real-time Glacial Lake Outburst Flood (GLOF) prediction, effectively integrating TerraFlow, TempFlow, and RiskFlow to monitor the Shispare Glacier. TerraFlow's Transformer-based model accurately predicts glacier velocity using the NASA ITS_LIVE dataset, identifying surge events with velocities exceeding 500 m/yr, while TempFlow's LSTM architecture forecasts land surface temperature with a validation MSE of 12.3 K^2, capturing seasonal extremes within <3K error despite 61% cloud cover challenges in the MODIS dataset. RiskFlow's CNN model achieves 94–95% accuracy in classifying GLOF probability from Sentinel-2 imagery, detecting critical spatial anomalies like lake formation. By combining velocity, temperature, and imagery-based predictions, IceWatch overcomes the limitations of static mapping, providing a dynamic, multi-modal early warning system that enhances disaster preparedness in vulnerable glacial regions. Despite challenges such as data gaps and computational demands, the system's robust performance underscores its potential as a cornerstone for glaciological research and climate change mitigation.

Future Work

To further enhance IceWatch's efficacy, several directions are proposed. First, integrating multi-sensor data, such as Sentinel-1 SAR, can address cloud cover limitations in MODIS and Sentinel-2 datasets, improving temperature and imagery coverage during the melt season. Second, optimizing computational efficiency through model pruning, quantization, or deployment on edge devices will enable real-time predictions in resource-constrained environments, particularly for TerraFlow’s 5.5 million-parameter Transformer. Third, incorporating uncertainty quantification using Bayesian methods in TempFlow and TerraFlow can provide confidence intervals for predictions, enhancing decision-making reliability. Fourth, expanding the system’s scope to other high-risk glacial regions in High-Mountain Asia will improve its scalability and impact. Finally, deploying IceWatch on a scalable cloud platform like AWS, with real-time data pipelines and automated alerts, will ensure operational forecasting, supporting glacier management and disaster mitigation on a global scale. These advancements will solidify IceWatch’s role in combating the growing threat of GLOFs amid accelerating climate change.

Chapter 9

REFERENCES

[1] K. Fujita et al., “Potential flood volume of Himalayan glacial lakes,” *Natural Hazards and Earth System Sciences*, vol. 13, no. 7, pp. 1827–1839, 2013.

[2] R. Worni, M. Stoffel, and M. Huggel, “Glacial lakes in the Indian Himalayas – From an area-wide glacial lake inventory to on-site and modeling based risk assessment of critical glacial lakes,” *Science of the Total Environment*, vol. 468–469, pp. 71–84, 2014.

[3] J. Zhou, X. Wei, X. Liu, and Y. Wang, “A novel framework for predicting glacial lake outburst debris flows in the Himalayas,” *Science of The Total Environment*, vol. 902, 2024.

[4] Z. Abbas, S. Hussain, and I. Ali, “Glacial Lake Outburst Flood (GLOF) Susceptibility Mapping in Central Karakoram National Park: A Comparative Study of Machine Learning, Deep Learning, and Hybrid Models,” *SSRN Working Paper*, 2025.

[5] S. Pandey and J. B. Malakar, “Detection of Water Bodies Using Deep Learning for Disaster Management,” *International Journal of Remote Sensing*, vol. 41, no. 22, pp. 8811–8834, 2020.

[6] D. Dey et al., “Monitoring glacial lakes using U-Net-based deep learning approach,” *Geo-spatial Information Science*, vol. 27, no. 1, pp. 87–99, 2024.

[7] K. Maslov, J. Peters, and P. Bunnik, “GLAVITU: A Hybrid CNN-Transformer for Multi-Regional Glacier Mapping from Multi-Source Data,” *University of Twente Research Report*, 2023.

[8] B. Mohanty, R. K. Mall, and S. Tripathy, “Review of glacier velocity and facies characterization techniques using multi-sensor approach,” *Environment, Development and Sustainability*, vol. 26, no. 2, pp. 1258–1277, 2024.

[9] X. Peng et al., “Towards Global Glacier Mapping with Deep Learning and Open Earth Observations,” *arXiv preprint arXiv:2401.15113*, 2024.

[10] Lützow, M., Allen, S., Haeberli, W., et al. (2023). *A global database of glacial lake outburst floods (GLOFs).* Earth System Science Data, 15, 2983–3005. https://essd.copernicus.org/articles/15/2983/2023/

[11] Climate Fact Checks. (2023). *Himalayan glacial lakes surge by over 10% since 2011; report warns of GLOF threat.* https://climatefactchecks.org/himalayan-glacial-lakes-surge-by-over-10-since-2011-report-warns-of-glof-threat/