TYPE Technology and Code
PUBLISHED 12 April 2023
DOI 10.3389/fenvs.2023.1085708

# A new approach to meteorological observations on remote polar glaciers using open-source internet of things technologies

Simon Filhol[1*†], Pierre-Marie Lefeuvre[2†], Juan David Ibañez[1], John Hulth[1], Stephen R. Hudson[2], Jean-Charles Gallet[2], Thomas Vikhamar Schuler[1] and John F. Burkhart[1,3]

[1]Department of Geoscience, University of Oslo, Oslo, Norway, [2]Norwegian Polar Institute, Tromsø, Norway, [3]Statkraft AS, Oslo, Norway

OPEN ACCESS

EDITED BY
Ramanathan Alagappan,
Jawaharlal Nehru University, India

REVIEWED BY
Dongqi Zhang,
Chinese Academy of Meteorological Sciences, China
Anlu Zhang,
Huazhong Agricultural University, China

*CORRESPONDENCE
Simon Filhol,
✉ simon.filhol@geo.uio.no

†These authors have contributed equally to this work and share first authorship

SPECIALTY SECTION
This article was submitted to Environmental Informatics and Remote Sensing,
a section of the journal
Frontiers in Environmental Science

RECEIVED 31 October 2022
ACCEPTED 29 March 2023
PUBLISHED 12 April 2023

CITATION
Filhol S, Lefeuvre P-M, Ibañez JD, Hulth J, Hudson SR, Gallet J-C, Schuler TV and Burkhart JF (2023), A new approach to meteorological observations on remote polar glaciers using open-source internet of things technologies.
*Front. Environ. Sci.* 11:1085708.
doi: 10.3389/fenvs.2023.1085708

Key regions of the world lack sufficient infrastructure to collect geophysical observations, often due to logistical challenges such as difficult accessibility and cost. With the advent of Internet-of-Things (IoT) technologies and low-cost electronics, it is possible today to build monitoring systems collecting spatially distributed, *in-situ* data with real-time connectivity to online servers for immediate and long-term usage at costs comparable to those of a single autonomous weather station. We present here a custom-built, modular system that collects quality data, and, that is, robust to adverse meteorological conditions and lack of energy. It integrates commercial and custom-built sensors connected to a node (main device) that manages power, data and radio communication. Data is sent to gateways and then to a server that parses, stores and quality controls the data. We deployed two networks in the vicinity of Ny-Ålesund in Svalbard, and operated from May 2021 to April 2022 to measure meteorological and glaciological variables. Our system collected reliable data and had sufficient power resources to survive 4–5 months of darkness during the polar night. Here, we present the design considerations and performance metrics, report our lessons learned from this challenging deployment, and suggest pathways for future improvements.

KEYWORDS

**Internet of Things, observation, meteorology, technology, polar science, wireless sensor network**

## 1 Introduction

Data collection for geophysical studies in remote regions, such as the Arctic, is logistically and practically challenging, and often lacks spatial coverage. Off-the-shelf technology, while of certified quality, may be costly, bulky, power consuming, or with few options for customization or integration into an automated data collection and processing pipeline. With the advent of open-source technology bridging hardware and software such as Arduino (Mellis et al., 2007), and the fast development of low-power microprocessor and low-cost sensors, geosciences are in reach of engineering solutions specific to our field of science, customized and specifically designed to tackle research questions or improve spatiotemporal representativeness using sensor networks. The IoT ecosystem is made of established technologies and many nascent ones (Wong and Kerkez, 2016; Baccelli et al., 2018; Joseph et al., 2018; Zhang et al., 2022). Today, there exist established and reliable open-source technologies (meaning no licensing fees, modifiable and scalable) for every step of a full-stack system. A full-stack system describes the entire set of software packages that performs data sampling, management, communication, storage, dissemination and visualization. Such technologies require a high level of computational literacy to identify, understand, combine and operate them. However, very much like Lego blocks, such technologies can be combined in creative ways to build specific and customized systems, pushing the *status quo* of collecting data with a better temporal and spatial coverage adapted to remote and polar regions.

Hart and Martinez (2006) explored the potential impact of a so-called "Environmental Sensor Network" on the study of fundamental processes, yet such systems remain relatively rare. Their development has been disparate. Pohl et al. (2014) established a network of standalone snow monitoring stations based on low-cost hardware. They deployed a total of 99 sensors collecting observations on snow, atmospheric conditions, and solar radiation. Their device is small and has low power consumption, in comparison to standard weather stations, but lacks telemetry capabilities. The low-cost sensors showed remarkable fidelity in comparison to state-of-the-art sensors. Chae et al. (2016) successfully deployed a network of 26 devices in Northern Alaska, collecting information about soil and air temperatures, and humidity. Their system is also low-power, equipped with a radio chip to push data to databases. Using this custom-built system, they have been able to capture spatially distributed data in a remote place in Alaska, and later added expert knowledge to estimate CO2 exchange between soil and atmosphere. Around the same time, two Californian watersheds have been equipped with wireless sensor networks (WSN) targeting the water cycle (Malek et al., 2017; Zhang et al., 2017). Both studies demonstrate the added value of their respective WSN to compute water balance more accurately in each watershed. Further details on the technology used were released later (Malek et al., 2019; Zhang et al., 2022). Cui et al. (2022) used the WSN data to improve precipitation estimates and orographic effect, two long standing problems in mountain hydrology. Ioannou et al. (2021) provide an extensive reflection on the diversity of technology and the complex set of choices one must make in designing a WSN system. These previous work highlight 1) the potential value of distributed and low-cost technology to geosciences, 2) the suitability and challenges associated with using these technologies in polar or cold environments, but also 3) highlight the diversity of approaches and choices of technology.

Several reasons can explain why sensor networks technology have not reached their full potential in natural sciences (Mao et al., 2019).

1. Geosciences is a niche market not necessarily attractive for major investments by private companies,
2. Forming a team combining expertises in software and electronic engineering is resource-demanding for a research group due to the high technical level and long term vision required for development,
3. Most attempted developments in sensor network systems are isolated and driven by few individuals therefore lacking transferability of knowledge,
4. Transforming a system from a proof of concept to a robust system is challenging, as for instance to fail-prove hardware/software issues encountered after deployment and in the choice of technology to demonstrate its feasibility and then standardization.
5. Research institutions in geosciences have few to no collaboration and coordination programs across their technical groups.

Despite these hindrances, we were able to develop for the past 5 years a proof of concept of a sensor network system to collect data in the remote regions of the Arctic mostly based on open source technologies. We describe here in detail the technological choices and combination, as well as evaluate performances, and share experience with research groups who may consider such endeavors.

The system presented here is custom-made to our needs of collecting meteorological and near-surface glaciological data on remote Arctic glaciers. It consists of a network of nodes, each corresponding to a small weather station, which samples and pushes data *via* a radio network to a gateway and finally to our local server. The system is energy-efficient, compact and autonomous. Each individual device measures air temperature, relative humidity, air pressure, as well as snow depth, wind speed, wind direction, longwave and shortwave outgoing radiation. These state variables can be directly measured at relatively low cost and inform geophysical models to better estimate other state variables like precipitation, that is, notoriously difficult to measure, especially in Arctic and Alpine environments (Lundquist et al., 2019). After collection, data are transmitted to a central database for storage and are dynamically accessible for post-processing and quality control. Finally, variables are documented and published along with metadata following the Findability, Accessibility, Interoperability, and Reuse (FAIR) principles (Wilkinson et al., 2016).

To test our system in Arctic conditions, two networks with a total of 15 nodes were deployed in the vicinity of the Ny-Ålesund Research Station in Svalbard on two glaciers: Kongsvegen (KNG) and Midtre Lovénbreen (MLB). These glaciers are particularly interesting due to their long time series of mass balance observations (Hagen and Liestøl, 1990) and dynamical behavior (Melvold and Hagen, 1998). They are located inland and in higher elevation than the main network of meteorological stations mainly placed along the coast (Figure 1). Moreover, Svalbard is one of the fastest warming regions on Earth (Isaksen et al., 2022). It is on the forefront of climate change and its glaciers experience considerable mass loss (Schuler et al., 2020). Therefore, it is of particular interest to collect *in-situ* observations with better representation of spatial and temporal variability of meteorological variables such as temperature or snow depth. Such data can be used to better validate, downscale and assimilate remote sensing products (e.g., Sentinel, Landsat and MODIS satellites), dataset from models (e.g., Schuler and Østby, 2020) or climate reanalysis products (e.g., Hersbach et al., 2020).

About 60% of the land surface of Svalbard (61,000 km$^2$) is glacier-covered and 50% of the land is higher than 350 m a.s.l. with the highest elevation being 1713 m a.s.l. However, most

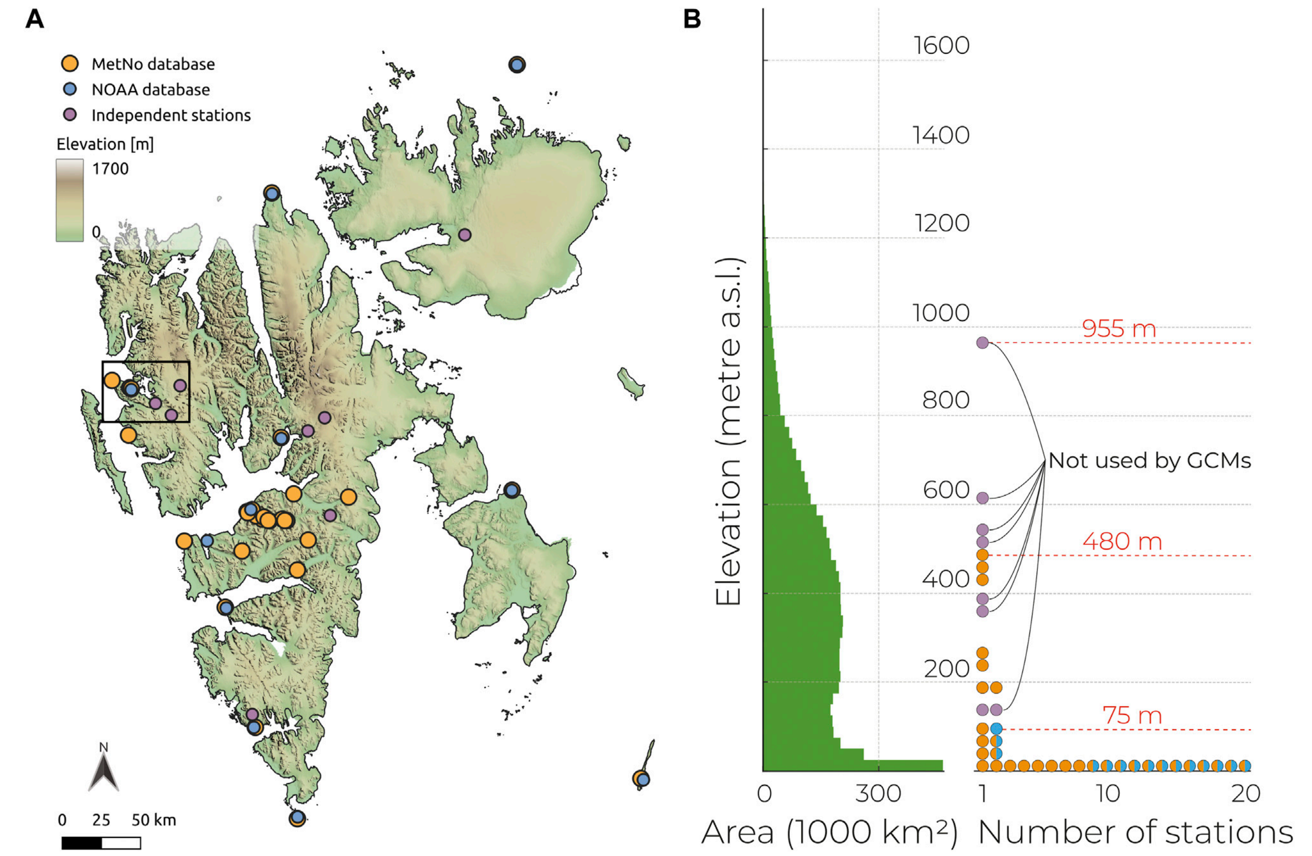

FIGURE 1
**(A)** Spatial distribution of weather stations in Svalbard from two publicly available data repositories: the Norwegian Meteorological Institute (Metno) database, the NOAA Hourly Observation Database, and stations maintained by independent research groups not available in the previous two databases. Some stations are present in both databases. **(B)** Elevation distribution of the weather stations by databases and the Svalbard land hypsometry. Elevation of the highest station in each database indicated explicitly in red.

meteorological observations are made in coastal areas and reach a maximum elevation of 480 m. These weather stations, available from the public repository of the Norwegian Meteorological Office (SIOS, COAT projects), are the only stations used in global and regional climate models (Figure 1). The only automated weather stations located on glaciers and at high elevations are maintained by independent groups for a variety of research projects (van Pelt et al., 2012; Schuler et al., 2014; Ignatiuk et al., 2022). Because of initial cost, bulkiness and high maintenance cost, those "research" stations are sparse and may be temporarily installed, with their objectives differing from meteorological monitoring programs. Data may not be systematically shared publicly and/or may have various methods of collection or degrees of quality control.

We propose here possible solutions to those issues and provide a perspective towards enhancing long term climate monitoring of remote high elevation areas with low-cost and open source weather stations. While in constant evolution, the system described here identifies key technologies our community could rely on when designing and initiating monitoring programs, or integrating into existing data infrastructures. Our work finds lots of similarities to what Horsburgh et al. (2019) designed but our application is more specifically tailored to meet requirements of the high Arctic and glaciology. It also encompasses simultaneous development in hardware and softwares.

## 2 System description

Our system consists of multiple parts: hardware of the nodes, software of the nodes, network architecture, data transmission, data management and quality-control, metadata and publication following the FAIR principles. This stack of technologies (Figure 2) covers sampling of physical variables, power management, communication and database infrastructure, followed by visualization, quality control and dissemination of data into public data repositories. The selection and development in software and hardware overcomes technical challenges while maintaining flexibility thanks to open-source technologies having lower cost of scalability and being highly customizable. A key part of our innovation is a reliable firmware for nodes with an emphasis on configuration flexibility, power efficiency and reliable data management. The hardware follows a version release (v1 and v2) whereas the software is a rolling release independent of the hardware release.

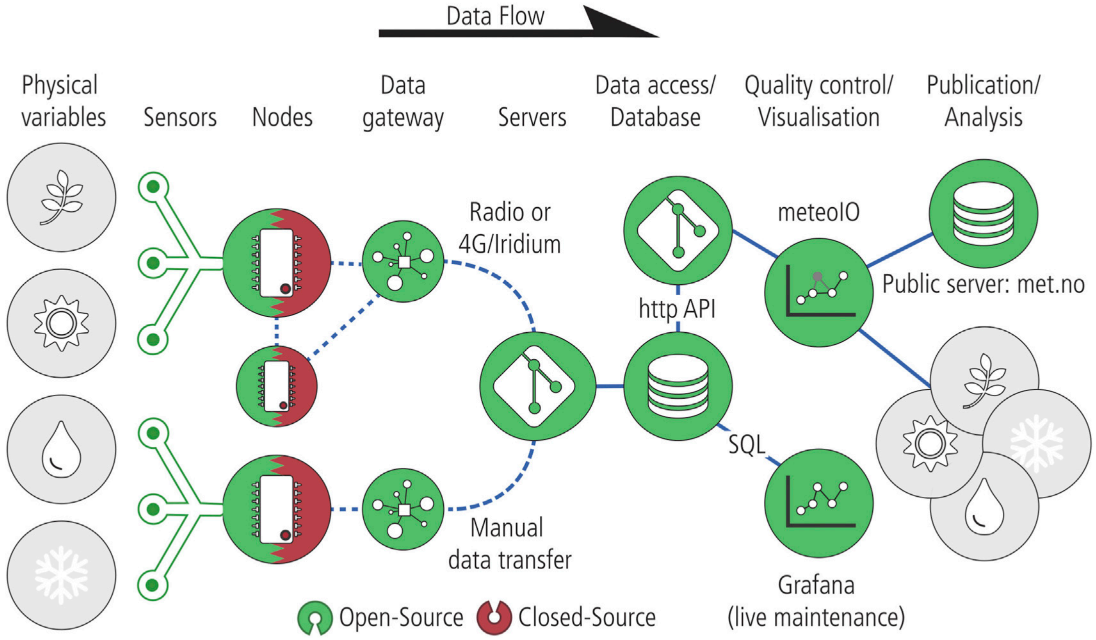

FIGURE 2
Overall architecture of the full-stack data pipeline designed and built to sample, transport, manage, store, visualize and disseminate data. This architecture is almost fully based on Open-Source technology (green), with an exception in this setup of a closed source microprocessor hardware motherboard (red).

## 2.1 Node hardware

The node hardware is essentially composed of a microcontroller, a power system, a set of sensors (commercial and custom made), communication modules, a clock, and a compact casing. A node (Figure 3) is designed around the main motherboard (Waspmote by Libelium) containing a microprocessor (ATmega1281), a Real-Time Clock (RTC), a power charge regulator for solar panel and battery, a communication port for data transfer, a microSD card reader for data storage and logging, and a number of digital and analog ports (Figure 3A). The waspmote is provided with an open-source Application Programming Interface (API) but the hardware itself is not under an open license. This choice was a legacy of initial investment for a platform providing significant hardware advantages in order to start this project. On top of the waspmote, we designed a custom-made shield to control power and facilitate communication with a variety of sensors. These sensors are both commercial as well as custom-made items, and the range of their applicability is not limited to the ones presented here.

The waspmote motherboard manages power using a charge regulator between a 6 W solar panel and a pair of 6.6 Ah 3.7 V lithium-ion batteries. The batteries directly supply power to the waspmote through 3.3 and 5 V regulators before redirecting power to all devices. This motherboard-controlled redistribution happens *via* a combination of software and hardware. Our added shield with bipolar NPN transistor switches can also control power supply to each sensor individually (Table 1). The LoRa radio requires 8 mA when used in full transmission power. All sensors and accessories are powered only when used. Using the RTC alarm system, our firmware maintains the microprocessor in deep sleep, sensors and other accessories depowered unless needed. This logic is built on top of the Libelium API and handled *via* the coroutine library (see Section 2.2). With a sampling rate of 10 min, this optimized system maintains power for an entire polar winter during 4 months of darkness at 79°N.

Each node can accommodate a variety of sensors using 3.3 or 5 V power supply, and is compatible with four digital communication protocols: SDI-12, I$^2$C, UART, and OneWire. With these possibilities, we selected a suite of sensors partly off-the-shelf sensors for geosciences, and partly small low-cost sensors combined into a custom-made PCB board (Table 1). We named this kit of sensors the Lagopus sensor kit, named after the emblematic birds of the tundra genus *Lagopus*. Two integrated parts compose the Lagopus, a strip fitting inside a OneSet RS3-B radiation shield, and a bottom plate facing the snow surface. The strip accommodates three sensors: TMP117, BME280, and SHT31 measuring air temperature, air pressure, and relative humidity. On the bottom plate, the MLX90614 measures skin surface temperature, the VEML7700 and the VCNL4040 measure ambient light level (i.e., outgoing integrated shortwave), and the AS7341 measures outgoing shortwave radiation over eleven bands. A BME280 also monitors temperature and relative humidity inside the node casing. Nodes can be completed with a 2D sonic anemometer ATMOS22 by Meter for measuring wind speed and wind direction and a sonic ranger Maxbotix MB7389 for measuring snow surface height. Other sensors can be added if the application differs (e.g., measuring water level).

There are four communication channels possible to push the data collected out: 2.4 GHz Xbee radio, 868 MHz LoRa radio, Global

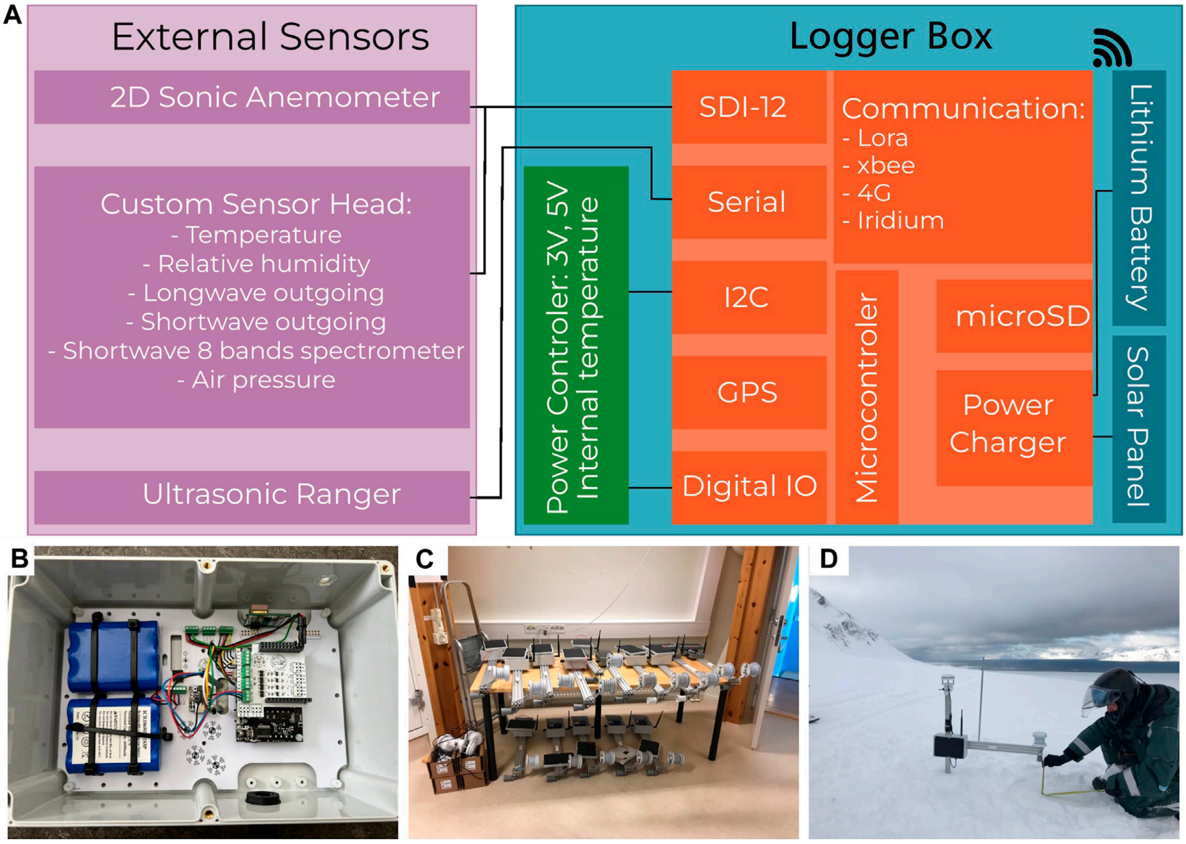

FIGURE 3
**(A)** Hardware diagram of one node. A node consists of a main logging box (blue) with inside a set of devices all managed by the main board (orange box), and a set of sensors either commercial or custom made (purple). Photos of the inside of the logger box **(B)**, preparation prior to deployment of the nodes **(C)**, and a node deployed **(D)**.

**TABLE 1 Sensors description and price estimates.**

| Sensor type | Model | Variables | Price[a] ($USD) | Power consumption |
|---|---|---|---|---|
| 2D sonic anemometer | ATMOS 22 | Wind speed, direction | NPA[b] | 8 mA |
| Ultrasonic ranger | MB7389 | Distance to surface | 95–126 | 2.3 mA |
| Lagopus sensor kit (custom-design) | BME280 | Air temperature, air pressure | 6–7 | 3.6 µA |
| | TMP117 | Air temperature | 5–7 | 3.5 µA |
| | SHT31 | Air temperature, relative humidity | 6–8 | 1.7 µA |
| | MLX90614 | Skin temperature | 19–20 | 25 mA |
| | VEML7700 | Ambient light | 1–2 | 8 µA |
| | VCNL4040 | Light and proximity sensor | 2–3 | 300 µA |
| | AS7341 | 11 bands spectrometer (405–905 µm) | 7–10 | 210 µA |
| **Optional** | | | | |
| Thermistor string | DS18B20 | temperature | 5–10 | 1 mA |
| CTD | CTD-10 | water head, water temperature, electrical conductivity | NPA[b] | 0.5 mA |

[a] As of 4 January 2023. Prices may vary according to purchase quantity and date, and are therefore simply indicative.
[b] Not Publicly Available.

System for Mobile communication (GSM) 4G module, and RockBlock9603 for Iridium satellite communication. A node may have none of the above, or up to two communication modules simultaneously except for a Xbee-Lora pairing. The ranges of Xbee and LoRa modules are up to 7 and 20 km in line of sight, respectively. The GSM 4G module requires coverage by a national GSM provider. Iridium is available virtually everywhere but costly. The two radios (i.e., LoRa and Xbee) are designed to emit in license free frequencies available in many countries of the world. However countries often have limitations for maximum transmission power and duty cycle, typically 1% per hour.

A RTC DS1337C is available on the Waspmote operating at a 32.768 kHz frequency with a drifting of 26 s per month at a temperature range of −40°C to +85°C. Its integrated alarm system provides the microprocessor with interrupts exciting the system from the deep sleep state. Because timing is crucial for sampling and coordinating network activities, we added a Global Positioning System (GPS) module to update the RTC time at regular intervals (1–3 days).

Each node is packaged into a box (240 × 160 mm, rated IP65) mounted on a 800 mm long, horizontal aluminum profile. The opposite tip of the aluminum profile carries the sonic ranger and the Lagopus sensor kit (Figures 3B–D). This aluminum profile is equipped with a quick and simple fastening system to be mounted on vertical poles up to 65 mm in diameter. A single node has an approximate weight of 2.5 kg. All nodes can be flashed with the firmware indoor, closed and readily deployable in the field. The design minimizes the number of cables exposed to the elements to only the anemometer cable. The deployment of one node requires only four bolts to be tightened, making the whole process fast (~10 min) given that a mounting pole is already in place. In case of maintenance, the operator can either fix the node in the field, or unmount and bring the node back indoors for easier manipulation. This convenient packaging is key to handling a large number of nodes for an entire network in challenging weather conditions (cold, wind, precipitation).

The choice of sensors and hardware is a combination of fitness to the goal of collecting relevant data, and minimizing cost. The price of each sensor (prior to electronic integration) is indicated in Table 1. To this cost, one must add the manufacturing cost of electronic boards, the batteries, the waspmote main board, the radio module, the radiation shield, and miscellaneous hardware. Cost of hardware alone is dependent on the quantity built, as costs for electronic components and manufacturing decrease significantly with large production batches. The two most costly components of a node are the 2D sonic wind anemometer (commercialized by Meter Group), and the Waspmote kit. All software is open-source requiring no additional cost except the development of the main firmware itself.

This system has gone through two main iterations: an original 2019 version (v1) and an updated 2021 version (v2) built upon field experience since 2019. Improvements include a larger box, a larger solar panel, and an extra microprocessor. The Lagopus sensors are now connected to their own microprocessor (the Adafruit QtPy) that controls sensor communications and interacts with the Waspmote *via* the SDI-12 protocol. This new setup has allowed electrical isolation of sensors potentially exposed to circuit shorts during moist conditions. Both versions are currently collecting data in Ny Ålesund, presented in (Section 3).

## 2.2 Node software

The node software, so-called firmware, is tailor-made for polar and high mountain conditions based on the C++ API from the development environment for Waspmote provided by Libelium. The core of the firmware combines basic functionalities of the Waspmote API and C++ libraries with a collection of custom-made libraries suited to efficiently and reliably manage power, communication, multitasking and logging of activities (Figure 4). The primary purpose of the firmware is to behave as a logging device for a variety of independent sensors, by packaging, storing and sending data. We ported sensor drivers or communication protocol from Arduino to the Waspmote development environment when needed (e.g., SDI-12 protocol). These external libraries are typically developed by independent groups of engineers (e.g., EnviroDiY) or electronic manufacturer and retailers Adafruit and Sparkfun under Open Source (OS) licenses. The firmware design is flexible and extendable to new sensors given the power and communication protocol constraints described in the previous section. The firmware also integrates directly into the rest of the data pipeline (Section 2.3).

Regarding pre-deployment, the firmware is compiled and uploaded to the node's microprocessor *via* USB. Because of memory limitations of the microprocessor (128 kB of program memory and 8 kB of data memory), the user must select the relevant libraries to be compiled and specifically configure each node based on its components. After compilation, the firmware is uploaded to the microprocessor and the node starts running immediately. It starts by checking the presence and state of all necessary components (battery level, microSD card, GPS, and communication device). It is followed by an interactive menu prompt for finer firmware configurations including sensor activation, sampling rate, level of logging, microSD interaction, and verification of sensor performance and network communication. This startup menu is optional and if not actively triggered by the user, the node goes directly into operational mode.

When powered, the node boots, sets alarms and enters deep sleep. Each sensor, activated by the user, is added to a table of activities handled by the coroutine library. This library manages each sensor independently and instructs the RTC when to trigger the next alarm interrupting the default deep sleep state of the node. During activation, the node enters a loop and checks battery level, performs activities planned by the coroutine library, and sets the next alarm.

The frequency of the looping procedure depends on the battery level and is optimized for winter survival. When the battery level drops below 75%, the coroutine library doubles the deep sleep time period for each activity. When the battery falls below a critical 30%, the node stops all network activities, and decreases the loop frequency by four from the original setup. In case the battery recharges more than 30% or more than 70%, the node resumes to normal behavior autonomously. This behavior preserves the batteries, even for long periods of limited or unavailable solar power. The node may experience such situations during long polar nights, or burial by snow.

When sampling a sensor, data is packaged into frames and stored locally on the microSD. Each frame contains a timestamp, the node name, and data fields with their associated sensor name. The size of the frame is optimized by compressing floats into integers and

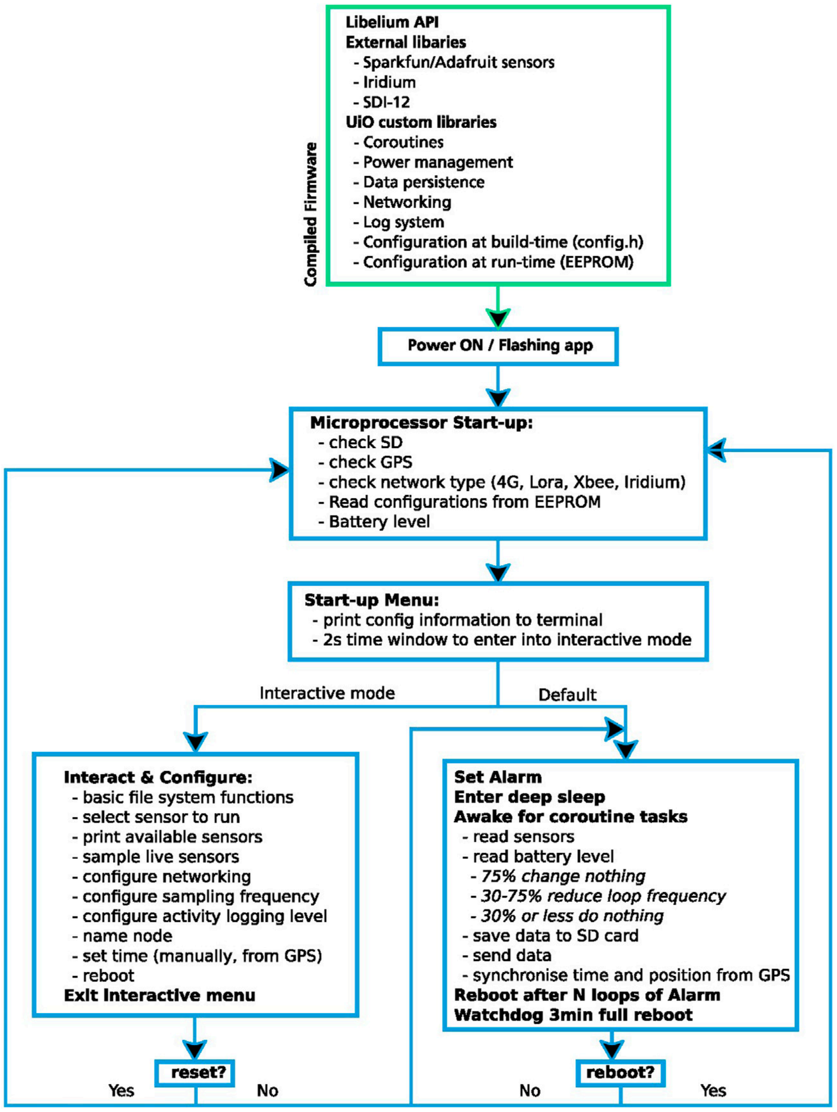
Compiled Firmware
Libelium API
External libaries
- Sparkfun/Adafruit sensors
- Iridium
- SDI-12
UiO custom libraries
- Coroutines
- Power management
- Data persistence
- Networking
- Log system
- Configuration at build-time (config.h)
- Configuration at run-time (EEPROM)
Power ON / Flashing app
Microprocessor Start-up:
- check SD
- check GPS
- check network type (4G, Lora, Xbee, Iridium)
- Read configurations from EEPROM
- Battery level
Start-up Menu:
- print config information to terminal
- 2s time window to enter into interactive mode
Interactive mode
Default
Interact & Configure:
- basic file system functions
- select sensor to run
- print available sensors
- sample live sensors
- configure networking
- configure sampling frequency
- configure activity logging level
- name node
- set time (manually, from GPS)
- reboot
Exit Interactive menu
Set Alarm
Enter deep sleep
Awake for coroutine tasks
- read sensors
- read battery level
- 75% change nothing
- 30-75% reduce loop frequency
- 30% or less do nothing
- save data to SD card
- send data
- synchronise time and position from GPS
Reboot after N loops of Alarm
Watchdog 3min full reboot
reset?
Yes
No
reboot?
No
Yes

FIGURE 4
Diagram summarizing the node firmware algorithm: 1) in green the libraries compiled into a firmware then flashed to the node's microprocessor, and 2) in blue, the on-board logic structure of the firmware.

packing data into binary arrays. Floats are multiplied by $10^n$, n being the significant number of decimals and then truncated to an integer. Arrays are compressed by differentiating each element to their predecessor with the first element as a reference. For instance (235, 234, 236, and 237) becomes (235, −1, 2, and 1). Importantly the size of the frame depends on the installed communication module: 73 bytes for Xbee and 255 bytes for LoRa. Thus, data from one timestamp may be stored in several frames, recombined by the server. This frame structure facilitates its incorporation into the server database using sampling time and node identity. To send frames, the firmware creates a list of frames to transfer following the First In First Out (FIFO) principle, except for Iridium communication which uses Last In First Out (LIFO). The whole data handling is designed to avoid plain loss of data, duplication or confusion with metadata (timestamp, origin). If no communication occurred, the user can still retrieve data from the microSD card. All activities of the node are logged into a main log file on the microSD card for troubleshooting and analyzing the node's performance. There are five levels of logging priority, from more information to fewer: TRACE, DEBUG, INFO, WARN, and ERROR. Logged activities remain at the node and are not sent through the network.

Two compatible types of network are used to receive frames at the network gateway: Local Area Network (LAN) and Wide Area Network (WAN). LAN includes radio communication such as the Xbee and LoRa protocols, whereas WAN refers to GSM 4G and Iridium communications. LAN radio networks can have different topology. The Xbee protocol designed by Digi uses a mesh topology where data bounces from one node to another all the way to the gateway automatically determined by the radio protocol. Such topology is resilient in case some nodes fail. The LoRa protocol is often associated with the LoRaWAN solution using a star

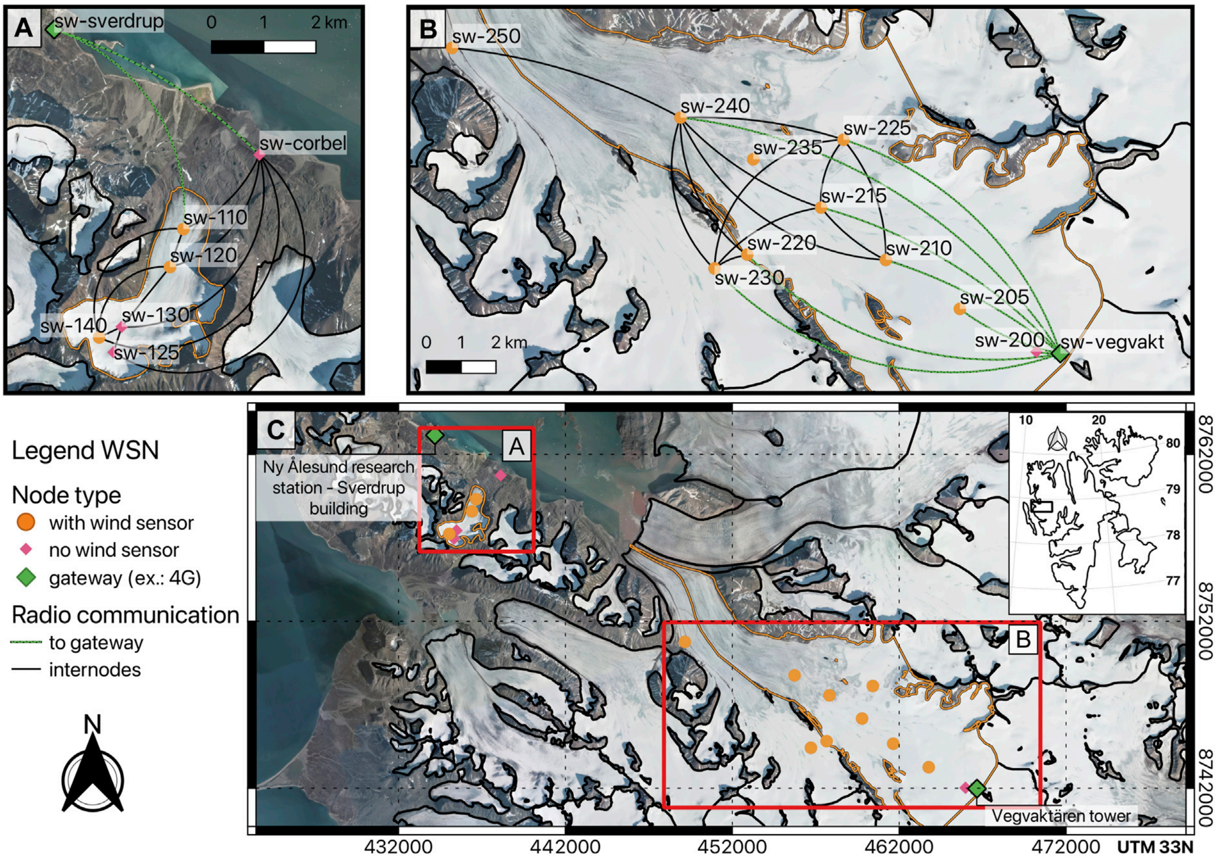

FIGURE 5
Map of the two networks deployed on the glaciers Midtre Lovénbreen (MLB) and Kongsvegen (KNG). MLB nodes send data either to the relay sw-corbel, or directly to an ethernet gateway installed at the Sverdrup research station **(A)**. KNG nodes send data to a 4G gateway located at the top of the glacier connecting the GSM network from Longyearbyen **(B)**. Lines in green indicate direct links node-gateway, and blue lines indicate node-node links. View of the two networks **(C)**.

topology. This solution being not suited to our use case (i.e., glaciers with elongated shape greater than 20 km, and lack of line of sight in some cases), we built our own topology where each node can send data to a defined gateway with the possibility to bounce off intermediate nodes. Figure 5 shows the usage of this multihop topology at our study site.

When using LoRa, every node is assigned an address from 1 to 255. Nodes can be configured to send to another node with a specific address, or the packages can be routed dynamically, by setting the destination address to zero. In the dynamic case, the node will first send a broadcast message. The first node that replies will be chosen as the destination. But only nodes with a lower address will reply, thus defining a multihop star topology, where packages flow from nodes with a higher address to those with a lower address, until they reach the gateway, which usually has the address number 1. Each receiving node will confirm reception with an acknowledgement (ACK) message to the emitting node, which removes the frame from its FIFO list. If the emitting node does not get an ACK, then it will later retry sending the same frame. If no ACK message is received three times in a row, then the emitting node will again search for a neighbor to send its data to. A frame is considered received when the receiver has stored it persistently, sending an Ack to the emitter. Frames are not immediately forwarded to the next node, instead they are first stored on the microSD card, in case a subsequent hop fails. Communication can resume from where it stopped and does not depend again from the original node. Lastly, the gateway forwards frames to the server parsing them into the database as described below.

## 2.3 Data management

The process of data management consists of the layer of technology built on top of the node network in order to manage, store, visualize and access the data. Collectively, we refer to this as the "upper stack", as it is, in itself, a unique and critical component of the network's functionality. The upper stack runs on a Debian Linux server. The entire application is developed with the Django web framework (Django Software Foundation, 2019), written in the Python programming language.

Data is sent from the networks to the upper stack server in three different ways.

**TABLE 2 Summary of data quality-control rules for nodes deployed in the region of Ny Ålesund.**

| Variable | min | Max | Other filters |
|---|---|---|---|
| Air temperature | 230 K | 290 K | |
| Surface temperature | 200 K | 310 K | |
| Air pressure | 910 hPa | 1070 hPa | Max rate limit: 0.1 hPa/s |
| Relative humidity | 1% | 120% | Soft min and max: 5% and 100% |
| Snow depth | 0 m | - | Max rate limit: 5.55e-6 m/s ~ 2 cm/h |
| Wind speed | −2 m/s | 70 m/s | Soft min and max: −0.2 m/s and 50 m/s |
| Wind direction | 0 deg | 360 deg | |

- Local node networks (Xbee or LoRa) transmit their data to a Raspberry Pi gateway, with internet connectivity. Data is forwarded to the server using a custom made API endpoint.
- Local network nodes send the data to another node equipped with a 4G module, which serves as a gateway, and forwards the data to the server using the 4G provider API.
- In special situations the gateway will be a node equipped with a satellite (Iridium) module sending data to the Iridium service, which will then forward it to the upper stack. In this case we only send a small part of the data, as the Iridium network is expensive.

Data is received, parsed and added to a PostgreSQL database. Data can also be added manually from a microSD card to the database using a custom Python script. This is particularly useful in case data fails to be pushed *via* the local network or for nodes equipped with Iridium pushing only sparse data updates.

Then data can be accessed in two ways.

1. The data can be visualized with Grafana (Chakraborty and Kundan, 2021), directly from the database *via* SQL query. Grafana allows the creation of live dashboards for daily monitoring, quick search of the data over the entire database, and assessment of potential issues prior to heading in the field for maintenance.
2. Data can be queried from the database using an API developed for this purpose and assisted with a custom made Python package (https://github.com/spectraphilic/wsn_client).

Django and Grafana are accessible with a user account and run on a Debian Linux virtual machine that includes other support software (i.e., OS solutions) such as the Nginx web server, the Monit monitoring software (https://mmonit.com/monit/), and the RabbitMQ message broker. Django also stores quality-controlled data in a separate database table.

## 2.4 Quality control, metadata and publication

Our quality control is constructed around MeteoIO, a C++ library designed to transform and filter weather data in order to make them easily readable and usable for meteorological models (Bavay and Egger, 2014). The quality control thus consists of 1) selecting relevant data, and 2) applying range, rate and deviation filters that are summarized in Table 2. MeteoIO was chosen for the automation prospect it offers, its open-source licensing, its proven experience in Switzerland, and its compatibility with repository requirements of the Norwegian Meteorological Institute system.

The quality control starts with selecting relevant sensors based on the node's version. Certain variables, like temperature, are measured independently by multiple sensors (BME280, TMP117, and SHT31 inside the radiation shield, and ATMOS22 and MLX90614 internal signal correction) (Table 1). At the moment, we select data from the most trusted sensor in terms of quality and performance. Snow measurements require field observation of a reference surface for computing snow depth from the sonic ranger. On a glacier, the reference surface is either the ice or snow surface at the end of the hydrological year also defined as the last summer surface. The selected data are then processed for quality control.

The first range filter eliminates outliers excessing a pre-defined plausible range (Table 2), while the second filter reassigns out-of-range values to the closest min/max values (Bavay and Egger, 2014). A third filter, applied only to air pressure and snow depth, removes points that exhibit a rate of change greater than a defined threshold. The min, max and rate values are determined based on visual inspection of the raw data and the filtered output. After the filters, the data are resampled at a 10-min interval, interpolated for gaps shorter than 1 hour. Metadata are added to the final netcdf file following the Attribute Convention for Data Discovery (ESIP, 2022). All these steps and parameters are defined using an input template file, that is, populated with node-specific metadata before sending data to the MeteoIO pipeline. Finally, the quality controlled metadata-rich data are sent to the repository at the Norwegian Meteorological Institute for publication and assigned a Digital Object Identifier (DOI).

# 3 Deployment on two arctic glaciers

## 3.1 Study site

Our system is installed on two glaciers representative of high-latitude climate conditions, in the vicinity of the Ny-Ålesund research station in Svalbard. The first glacier is Midtre

**TABLE 3 Summary of nodes deployed in the region of Ny Ålesund. MLB corresponds to Midtre Lovénbreen glacier and KNG to Kongsvegen glacier. Two average air temperatures are included, when available. First MJJ includes data from May, June, and July, followed by the average for May only.**

| Node name | Glacier | Vers | Wind sensor | Elev. M a.s.l. | Start date | End date | Air temp. MJJ/May mean 2021 (deg C) | Snow depth May mean 2021 (cm) |
|---|---|---|---|---|---|---|---|---|
| sw-110 | MLB | v2 | yes | 143 | 2021-04-26 | - | 0.68/−4.06 | 120 |
| sw-120 | MLB | v2 | yes | 219 | 2021-04-26 | 2021-12-12 | 0.27/−4.53 | - |
| sw-125 | MLB | v1 | no | 436 | 2021-04-27 | na | −/−−5.31 | - |
| sw-130 | MLB | v2 | no | 361 | 2021-04-26 | - | −0.11/−5.14 | 147 |
| sw-140 | MLB | v2 | yes | 422 | 2021-04-26 | 2021-08-23 | −0.42/−5.52 | 175 |
| sw-200 | KNG | v2 | no | 727 | 2019-04-06 | - | −2.70/−8.16 | 209 |
| sw-205 | KNG | v1 | yes | 659 | 2021-05-01 | 2021-08-12 | −2.50/−8.03 | 201 |
| sw-210 | KNG | v2 | yes | 590 | 2021-04-29 | 2021-12-03 | −2.24/−7.65 | 159 |
| sw-215 | KNG | v2 | yes | 530 | 2019-04-06 | - | −2.13/−7.66 | - |
| sw-220 | KNG | v2 | yes | 540 | 2019-04-06 | 2022-01-01 | −1.66/−6.71 | 141 |
| sw-225 | KNG | v2 | yes | 564 | 2021-05-01 | 2021-07-06 | −/−7.39 | 150 |
| sw-230 | KNG | v2 | yes | 505 | 2019-04-06 | - | −1.80/−7.03 | 177 |
| sw-235 | KNG | v1 | yes | 464 | 2021-04-29 | 2021-06-17 | −/−6.87 | - |
| sw-240 | KNG | v2 | yes | 394 | 2019-04-07 | - | −1.46/−6.66 | 149 |
| sw-250 | KNG | v1 | yes | 90 | 2021-05-01 | 2021-08-12 | 0.31/−4.16 | 92 |
| Corbel (relay) | MLB | v1 | no | 17 | 2021-04-26 | - | - | |
| Sverdrup (gateway) | MLB | - | no | 12 | 2021-05-01 | - | - | |
| sw-vegvakt (gateway) | KNG | - | no | 842 | 2021-04-29 | - | - | |

Lovénbreen, a valley glacier located 4.5 km South-East of Ny-Ålesund with an elevation ranging from 72 to 725 m a.s.l. and an area of 5.2 km$^2$. Its accumulation pattern is characterized by deposition of snow in the lee of its bowls and the southern mountain ridge line going from southeast to northwest. Since 1995, Midtre Lovénbreen has had a general negative mass balance trend, more pronounced in the last decade (Välisuo et al., 2017). The second glacier, Kongsvegen, is a marine-terminating glacier, which has an area of 108.3 km$^2$ reaching 1015 m a.s.l. at its top. The prevailing local katabatic wind flows from the southeast towards the fjord Kongsfjorden affecting snow redistribution on the glacier, whereas the main synoptic wind pattern over Svalbard is easterly (Beine et al., 2001).

The mass balance of both glaciers has been monitored by the Norwegian Polar Institute since 1968 for Midtre Lovénbreen (Hagen and Liestøl, 1990) and since 1987 for Kongsvegen (Melvold and Hagen, 1998), and provides valuable snow accumulation data to validate precipitation patterns in regional climate models (Pelt and Kohler, 2015; Østby et al., 2017). Annual snow depth on glaciers is the only measurement available to estimate orographic effects on precipitation since few precipitation gauges are available at high elevations (Figure 1).

The nodes we deployed are spread over 3 km and an elevation range of 300 m on Midtre Lovénbreen, and over 20 km and 650 m on Kongsvegen. Table 3 provides complementary information on the hardware version, presence of a wind sensor, first installation date of the node and two estimates of air temperature and snow conditions in spring 2021 for each of the deployed nodes.

To evaluate the performance of our sensors, we compared one node (sw-115) to an Automatic Weather Station (AWS) operated by the Norwegian Polar Institute, based on sensors commonly used in geosciences (Section 4.3.3). This AWS was approximately 30 m from the node and is composed of a 10-m mast and a classical meteorological suite of sensors. In detail, air temperature and relative humidity are measured with a Campbell Scientific HC2S3 probe mounted in an Apogee TS-100 ventilated shield. Wind speed and direction are measured with a Wind Monitor HD Model 05,108 by RM Young. Both of these instruments are installed at the ends of a cross arm approximately 2 m above the ice surface. Snow thickness is derived from a Campbell SR50 A sonic ranger placed on a separate mount, ~12 m down glacier from the main mast. On the same mount, two heated-ventilated Kipp and Zonen CMP22 and two Kipp and Zonen CGR4 radiometers measure incoming/outgoing shortwave and longwave radiation, respectively. The AWS is located near the mass balance stake 6 on Kongsvegen, at 530 m a.s.l., along the glacier centerline and near the long-term equilibrium line (78.774°N, 13.19°E). The sampling rate of the AWS (1 min) is higher than the 10 min of the sw-215 node. The comparison period lasted in 2021 until the AWS stopped recording on 17 January 2022 due to a power failure.

## 3.2 Node deployment

In 2021, we deployed a total of 15 nodes, two gateways and one relay over the glaciers Kongsvegen and Midtre Lovénbreen. Two hardware versions of nodes are currently deployed (*i.e.*, v1 and v2) and twelve of these nodes are equipped with a 2D sonic anemometer (Table 2). All nodes were deployed in late April and early May 2021 with the same version of the firmware. All nodes are equipped with a LoRa radio sending either directly to the *sw-sverdrup* gateway connected to an online Raspberry Pi for Midtre Lovénbreen or for Kongsvegen to the *sw-vegvakt* gateway pushing data on the 4G signal available from the distant town of Longyearbyen. Nodes that are not in direct reach of these two gateways are autonomously searching for the closest node to use as a relay. The nodes on both glaciers were placed to match the current mass balance monitoring program, with most nodes located around the equilibrium line of the glaciers (Figure 5). Nodes are attached to a 6 m long and 40 mm diameter aluminum stake drilled into the ice down to 3–4 m depth. The drilling depth is adjusted to the expected net annual mass balance at each specific location. The nodes are oriented to have their solar panel facing south, and the anemometer towards the geographical North.

## 3.3 Results

### 3.3.1 Node behaviors

Out of the 15 nodes, seven maintained activity all year around, whereas eight stopped prematurely because of hardware issues, software bugs, or instabilities yet to be identified. Among those that stopped, four were v1 nodes (out of four) and five were v2 nodes (out of eleven). All v1 nodes experienced the same problem associated with an older design flaw of the sensor kit prone to shorts on the data line of the I$^2$C communication. This problem interferes with the node RTC clock, also using the I$^2$C protocol. In this situation the node keeps running without any coherent timestamp. The cause of halting of the five v2 nodes remains unknown. All five nodes stopped all activities at different times throughout the year (Table 3) and never rebooted until maintenance in spring 2022. The logged activities do not demonstrate a problem with our firmware but rather suggest instabilities with the alarm, watchdog or reboot systems inherent to the waspmote board.

Despite these nodes' activity issues, all of the nodes retained healthy batteries (>30%) that recovered quickly once the Sun came back in March. For those that ran long enough to experience polar night, we observe the three-levels of power management. Figure 6D shows the battery level for five nodes. When the battery levels remain around 100%, the batteries are continuously recharged by the Sun. Then, in mid-October, the polar night settles and the battery levels drop to 75%. Under this behavior, the node consumes 0.8 mA on average. Below that level, a power-saving mode kicks in and the battery drains 0.2 mA as the number of activities is reduced in all nodes by the firmware. With the exception of *sw-130*, none of the batteries dropped to the next power threshold at 30% where all network activities are stopped and the sampling rate is divided by four. Even for *sw-130*, batteries never discharged fully. Once the Sun was back, all nodes recharged fully within 3 weeks.

This 1 year experience shows that monitoring Arctic glaciers with wireless network of low power radio solutions, e.g., LoRa is feasible despite adverse conditions like rimming, fog, and mainly limited supply of power for 4 months of the year. Over the entire year, 64% of the v2 nodes survived and the five compromised nodes still collected for 176 days on average. The number and large spatial coverage of the nodes enabled the study of a full set of meteorological data over the two studied Arctic glaciers.

### 3.3.2 Network behavior

Figure 5 shows the communication topology of the LoRa network for the nodes both on Kongsvegen and Midtre Lovénbreen glaciers. Data was sent from node to node (black) and from node to gateway (green). Estimates of communication fluxes extracted from the logs of the nodes and gateway show temporal variations in communication activity and submission success rate between nodes and to the gateway (Figure 6A). Nodes *sw-220, sw-215, sw-210, sw-205,* and *sw-200* were programmed to directly send data to *sw-vegvakt* whereas for the other nodes, data transfer cascaded, as intended, through the other nodes until they reach their endpoint, the gateway (Figure 6B). Between May 2021 and May 2022, the rate of successfully sent and acknowledged data is 62%–89%. All the nodes successfully connected to the gateway *sw-vegvakt* but the volume of submitted data decreases with distance, roughly 1 Mb per 3 km for the entire year. The path length and an increased number of multi-hops are the limiting factors for data transfer as shown in (Figures 6B, C). Periods with high air humidity or snowfall attenuated the signal, as revealed by the Received Signal Strength Indicator (RSSI). A second factor is related to battery levels that are used for power management, limiting the node radio activities to

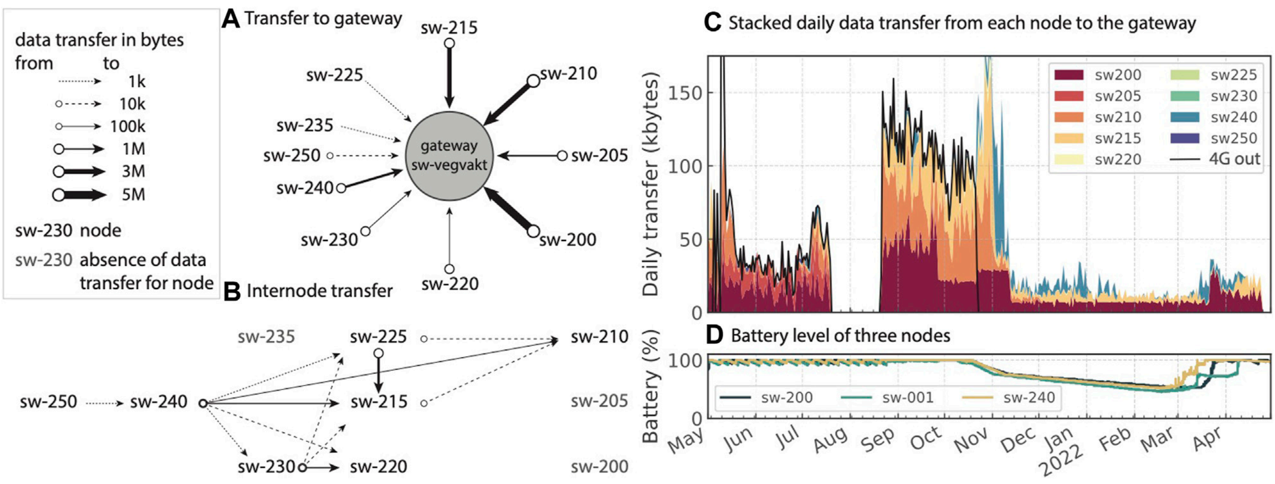

FIGURE 6
Variations in data volume transferred to the gateway *sw-vegvak* and between nodes at Kongsvegen for the period May 2021- May 2022. Schematic summary of transfers **(A)** from all nodes to the gateway and **(B)** internodal transfer between nodes. **(C)** Temporal variations in data transfer from each node to the gateway sw-vegvak and **(D)** in battery level for three nodes. The volume of data sent from the gateway to the server by 4G (black line) adapts to the volume of received data until its failure on 22nd October.

maximize the survival span, by reducing communication attempts to two or zero, depending on battery level. The behavior is seen in mid-November on Figure 6C. Although difficult to quantify, competition in terms of connections from one node to the next or to the 4G for the gateway may explain some transfer variations. On October 22, the 4G module of the *sw-vegvakt* gateway stopped to push out data. As less time was spent on 4G communication, the gateway could increase its listening time, leading to a 30% increase in data received. From mid-November, the battery level of the nodes dropped, which in turn intentionally limited communication. In regards to the competition between receiving or submitting data, the node firmware will privilege submission. The network on Midtre Lovénbreen, smaller in size, worked in similar ways, but the gateway stopped completely on 9 November 2021 for unknown reasons. In the meantime, nodes continued collecting data stored locally.

### 3.3.3 Variability of meteorological data

For the six nodes that collected data over the entire winter 2021-2022, we can compare the variables that exhibit spatial variability: air temperature, wind speed, wind direction, and snow depth (Figure 7). The selected nodes show a temperature gradient with elevation on both glaciers, warmer at the ablation area at low elevation (*sw-110*), than at the equilibrium line (*sw-240, sw-215,* and *sw-130*), or at the accumulation area (*sw-200*) further up-glacier. The nodes on Midtre Lovénbreen reveal a mean lapse-rate of 0.47°C per 100 m in between the glacier tongue (*sw-110*) and the highest nodes (*sw-130*). On Kongsvegen, we observe a lapse-rate of 0.37°C per 100 m in between sw-240 and sw-200. We can also observe more frequent wind events on Kongsvegen glacier than at the Ny-Ålesund station, with the equivalent of 62 days of wind speed greater than 6 m s$^{-1}$ *versus* 35 days for Ny-Ålesund. 6 m s$^{-1}$ is a typical wind speed which initiate snow redistribution. Midtre Lovénbreen, sheltered by mountain ridges, shows much less wind events above 6 m s$^{-1}$, only 11.5 days. Additionally, Figure 7 shows the data collected at the operational weather station in Ny-Ålesund (black line), that are assimilated into global and regional reanalysis products (*e.g.*, ERA5, CARRA) (Pelt and Kohler, 2015).

Figure 7A shows the weekly mean air temperatures for all nodes that ran continuously until maintenance in April 2022, and for the operational weather station in Ny Ålesund. As expected, temperatures are highest in Ny-Ålesund, located close to sea level. For the other nodes, temperatures decrease with both distance from the fjord and elevation. Figure 7B shows snow depth for the same nodes, revealing overall synchronicity of snow falls but with a great variability in snow amount. Ny-Ålesund receives less snow than the nearby glaciers mainly due to the orographic effect. For instance, the upper part of Midtre Lovénbreen (*sw-130*) receives the most snow followed by the upper part of Kongsvegen (*sw-200*). The upper part of MLB is surrounded by steep slopes in a bowl-shape area, where the snowpack is less affected by wind redistribution and erosion than the lower parts of glacier. In contrast, Ny-Ålesund experiences rapid snow erosion following snowfalls. Strong winds swept away the fresh snow not yet compacted (*e.g.*, mid-November). The melting of snow between July and September is symptomatic of the glacier ablation and shows an expected spatial pattern: melting at the top of Kongsvegen (*sw-200*) is already interrupted mid-august by an early snow accumulation, whereas the rest of the nodes indicate a first snow accumulation in early October, if not November. The wind speeds (Figure 7C) and directions (Figure 7D) are helpful to understand snow redistribution events in intensity (proportional to wind speed). We see that the node on Midtre Lovénbreen experiences less wind than elsewhere due to its sheltered location. Local wind directions are directly aligned with the local topography. Kongsvegen is dominated by strong winds from the East-Southeast, along the glacier axis.

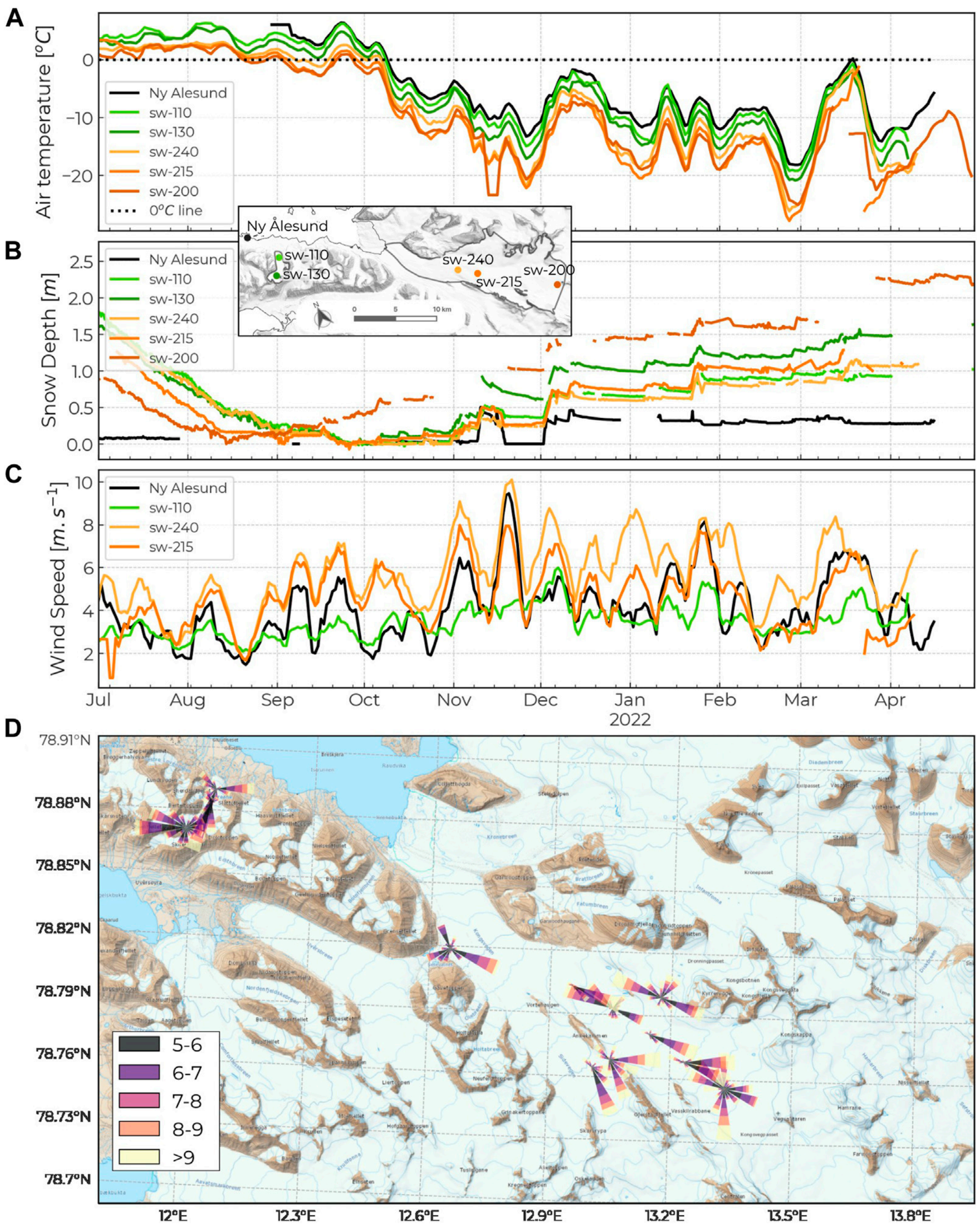

FIGURE 7
Measurements timeseries from the WSN in comparison to the operational weather station operated by the Norwegian meteorological institute in Ny Ålesund for **(A)** weekly mean temperature, **(B)** snow depth, and **(C)** weekly mean wind speed. The map inset shows the location of each node. **(D)** Normalized windroses for wind speed of 5 m s$^{-1}$ or more. 5 m s$^{-1}$ is a typical wind speed at which snow may start drifting.

### 3.3.4 Sensor performance

To evaluate the performance of the sensors, meteorological data from *sw-215* are compared to those of the nearby AWS at stake 6, located in about 30 m distance (Figure 8). During the overlapping data period from 1 September 2021 to 17 January 2022, there are 11,690 values recorded for each variable on the same time interval.

The regression fit of air temperature between the precision sensor TMP117 on the node and the HC2S3 probe on the AWS is excellent, reaching a correlation coefficient of 1, with a bias of −0.09 °C (Figure 8A). Discrepancies may occur in winter if the radiation shield of one or the other weather station experiences riming or because the AWS radiation shield is heated and ventilated, maintaining constant temperature of the sensor whereas sensors from the nodes are not. Despite this known radiation shield difference, the node's sensor performs as good as the AWS's sensor over the observed temperature range of −30°C–5°C.

The relative humidity measured by the node exhibits a systematic bias of 13.95% and a scaling of 1.2 relative to the AWS, although the time series have a high correlation with a

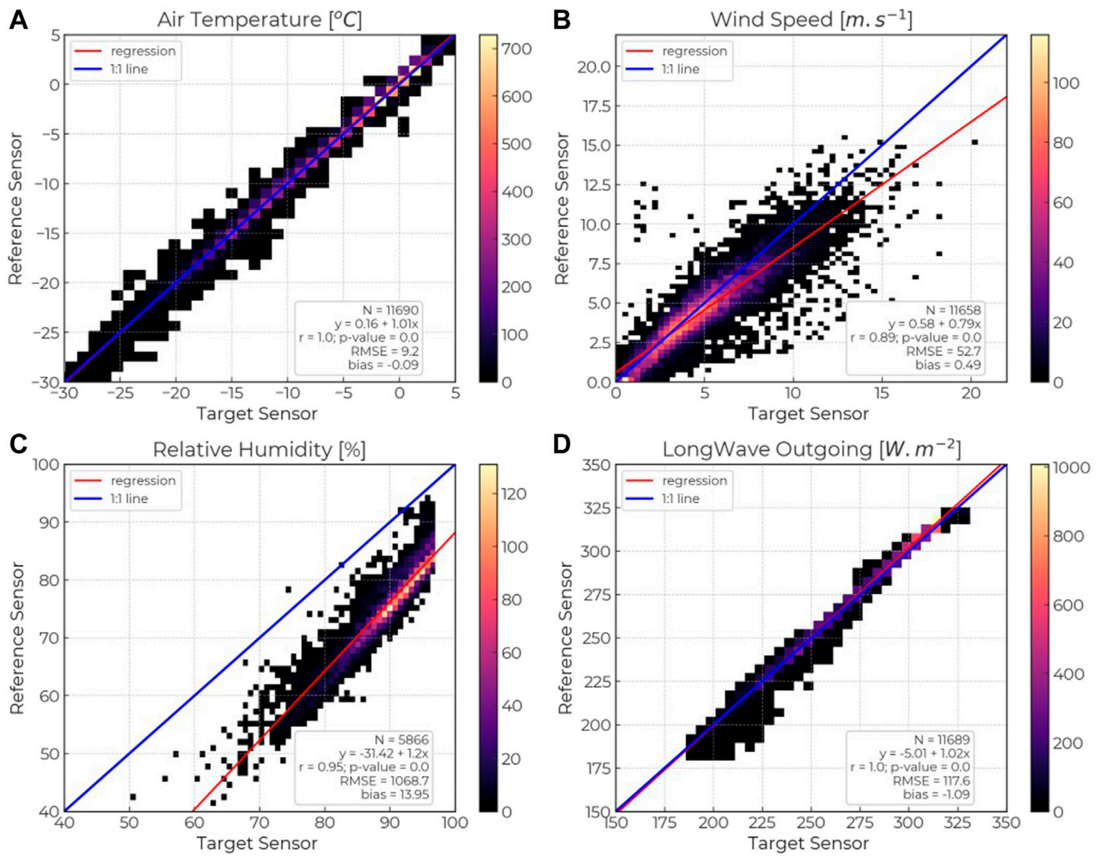

FIGURE 8
Bi-variate histograms of the reference sensor (classical weather mast) *vs.* the corresponding sensor from the adjacent node *sw-215*. No processing is applied except discarding outliers.1:1 line (blue) and a linear regression fit (red) are overlaid onto the histogram. The colorscales indicate the count of common values.

Pearson correlation coefficient (r-value) of 0.95 (all values with Relative Humidity (RH) > 97% were discarded) (Figure 8C). A consequence of the observed bias is that the SHT31 sensor at the node saturates whereas the HC2S3 of the AWS reaches 85% relative humidity in many instances. A possible explanation for the more frequent saturation is condensation on the sensor. The SHT31 sensor is equipped with an internal heating resistor, which was not used. The sensor performance of the node after the bias correction is satisfactory within the recorded range 50%–85% of relative humidity but has apparent limitations in high humidity conditions.

A sensor comparison between the node's 2D sonic anemometer ATMOS22 and the AWS's propeller anemometer (RM YOUNG Wind Monitor HD Model 05,108) needs to account for the differences in sensor concept, associated drawbacks, sensor height and sampling rate. The former uses ultrasound time delay to determine the wind speed, whereas the other measures the frequency of rotations induced by wind drag on the propeller. The sampling rate of the sonic anemometer is shorter (i.e., averaging over 10 s) than the AWS's anemometer (1s sampling averaged over 1 min). Figure 8B still shows a good fit with a 0.8 scaling difference. Differences may be due to sensitivity to riming on either of the sensors (Wyngaard, 1981; Bowen, 2007), to the 1 m difference in height, or different impact of gusts and turbulence on measurements.

Radiation is measured by four independent sensors (AS7341, VCNL4040, VEML7700, and MLX90614). MLX90614 measures outgoing skin temperature and is converted to longwave radiation according to the Stefan-Boltzman law, assuming perfect emissivity. When compared to the Kipp and Zonen CGR4 at the AWS, we find a good fit between the two sets of measurements with a r-value of 1 ($p$-value<<0.05) and a bias of −1 W m$^{-2}$ (Figure 8D).

Shortwave radiation (SW) is measured by three independent sensors: AS7341, VCNL4040, and VEML7700. AS7341 is an eight narrow band spectrometer with two shortwave broadband in the visible and one Near-Infrared band. Unfortunately, the gain setting of the sensor did not allow us to collect useful measurements during summertime. Therefore this sensor is not evaluated here. We report our experience to improve future efforts as we have indications that this sensor can be of great value. In May 2022, we adjusted the gain parameter (from 2 to 8) of this sensor to collect useful measurements in all light conditions. Data from the sensor VCNL4040 are not useful as the sensor is saturated for conditions reflecting 80 W m$^{-2}$ or more.

The sensor VEML7700 measures luminance. Luminance weighs different spectral bands to match the sensitivity of a

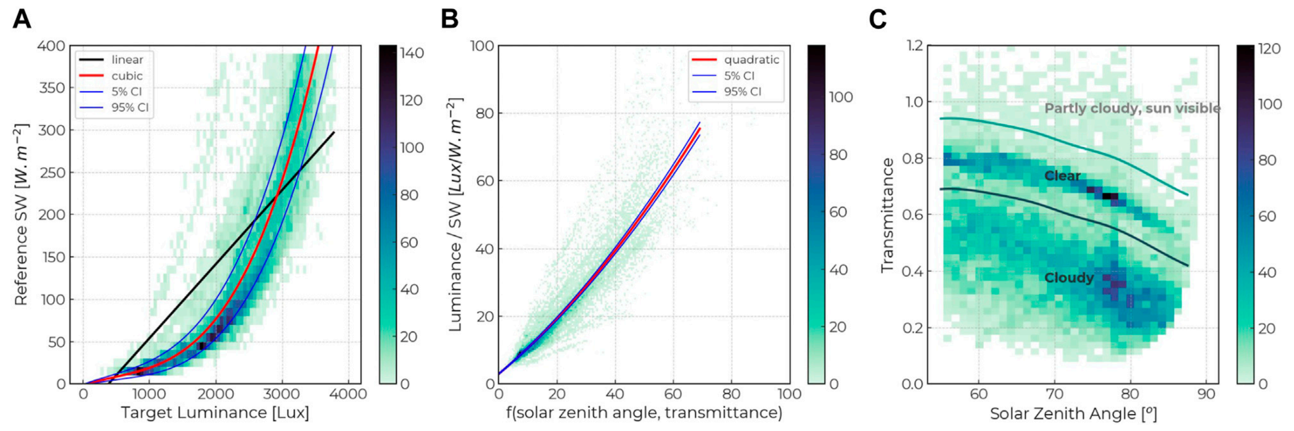

FIGURE 9
Performance for the target light sensor VEML7700 against the reference shortwave radiation sensor Kipp and Zonen CMP22. **(A)** Histogram showing the direct relationship between white luminance measured by VEML7700 and outgoing shortwave radiation from CMP22. **(B)** Histogram showing the fitted function of the solar zenith angle and transmittance to luminance against the ratio of luminance over outgoing shortwaves. **(C)** Histogram showing the solar zenith angle *vs.* transmittance measure, highlighting the sky conditions (cloudy, clear sky, thin cloud with sun).

human eye, with no straightforward conversion to a radiation flux (Michael et al., 2020). Luminance is expressed in Lux. Figure 9A shows the direct correlation between the outgoing luminance and the outgoing shortwave radiation from the CMP22 sensor, revealing considerable differences. However, we can derive a useful relationship between the measured luminance and the measured outgoing flux (CMP22), based on solar geometry and atmospheric transmissivity (transmittance). Using incoming shortwave radiation from the CMP22 (AWS), we can compute the transmittance T ($SW_{measured}/SW_{top\ of\ the\ atmosphere}$), and derive the solar zenith angle $\theta$ from astronomical relations. We can then fit a function *f(T, θ)* defined as follows:

$$f(T,\theta) = C + a.e^{b.\cos(\theta)} + d.T^{h}$$

with *C, a, b, d, h* the fitting parameters, against the measured luminance. Figure 9B shows the improvement of this fitted function against the ratio of luminance over outgoing shortwave radiation. We observe a tight regression. Together, T and $\theta$ can be used to assess the sky conditions (Figure 9C) and therefore provide a proxy to decompose the spectral signal used in computing luminance by the VEML7700. So without information on transmittance, it is hard to make use of a single downward-looking light sensor. Ideally we would need a second one looking upward. This was not included in this version because of technical obstacles, but should be pursued as an obvious improvement to the sensor kit.

The snow depth sensor, MB7389, is calibrated by the manufacturer. Its output accounts for variations in temperature and humidity. We confirmed the correctness of the sensor readings in a calibration experiment at room temperature. The sensor facing a wall is incrementally moved from 500 to 4500 mm every 500 mm. We found a mean bias of 3 mm between the actual distances and the sensor readings (12 per position). This value is well within the experimental error and beyond the accuracy needed for measuring snow depth.

## 4 Discussion

The wireless system presented here and developed to collect geophysical data in remote and cold regions is proven useful and robust to a variety of weather conditions and energy availability circumstances. This system was incrementally adjusted and improved over 5 years to a final test realized on glaciers in the vicinity of Ny-Ålesund in Svalbard. The status of the project reached a level of maturity sufficient to be used by and shared to the broader geoscientific community. While this system was developed and tailored to our needs, technological choices and return of experience should be of interest to any group wishing to pursue the development of similar systems. The project is portable in its entirety as well as for selected parts.

The network we deployed in Ny Ålesund successfully collected data from May 2021 to April 2022. Most nodes remained active throughout the dark season of the polar winter. The data collected capture the spatial patterns of atmospheric and snowpack conditions across two glaciers from their ablation to accumulation areas. Records of the two networks reveal the temperature lapse-rates, the occurrence of local inversions, the spatial heterogeneity in snow precipitation and redistribution, and the patterns of summer melt, demonstrating the relevance of collecting spatially distributed observations on glaciers. We show that even on short distances, meteorological conditions can vary significantly. Local deposition of snow or wind patterns can be resolved more finely over glaciers being among the most impacted by climate change in the world. Moreover, our two networks collected valuable observations in areas scarce of operational weather observations (Figure 1). The collected records are now publicly available after quality control.

Our system experienced successes but also a number of failures, some that can be fixed, others not. The gateways experienced independent troubles that were identified and fixed. The server-database system experienced no issues. The LoRa network showed robustness to the dark season when no solar energy was available to recharge batteries. The nodes that survived behaved as expected, reducing network and sampling activities while the battery had low capacity, and increasing again activities when solar energy returned. From the nodes that failed, we identified 1) a hardware design problem on the version V1 nodes, 2) an unknown instability leading the node to stop all activities. Out of 15 nodes, 6 collected data for the whole year. Failure 1) was resolved with version 2021 of the nodes but failure 2) is still prone to occur with no obvious clues on how to fix it. We suspect a low level issue potentially linked to hardware.

While we showed the network transferred data from the nodes to the remote server, it did not carry the full amount of data collected. The gateway on Kongsvegen ran the whole period receiving data from the LoRa network but stopped pushing data *via* 4G. Because the gateway is placed in a remote location, no fix was possible until April 2021, when snow conditions enabled access by snowmobile. Receiving data *via* the network serves two main purposes, 1) transfer of the data itself, and 2) provides insights into what problems a particular node may experience, allowing for targeted annual maintenance. Maintenance may include hardware fix or replacement, or simply update of firmware. The latter requires physical access to the nodes. Over-The-Air programming is theoretically possible but its implementation and realization from a practical point of view in remote regions through intermediate relay is not yet in reach. It would require a long period of open and continuous wireless connection to carry the entire new program (~100 kb) to each node. Moreover, in its current design, certain configurations require physical interactions with the node. For better assessment of local conditions, we realized that a LIFO management of the frame would provide the latest information of a given node at the expense of not sending all data.

Over the years of development, we resolved many failures either due to poor or ill design choices, introduced bugs, incompatibilities between various software layers or software dead-ends. Our lack of experience in both software and hardware development combined with our ambition to simultaneously develop a wireless network with custom-made sensors and a new hardware kit, led us to encounter a large number of problems occurring in parallel. Rapidly, we were in need of more advanced and abstract programing, taking us away from the basic scripting/sketch logic of the Arduino Do-It-Yourself approach. The Arduino ecosystem is rich and diverse offering easy access to programming electronics. However the more technologies are combined together the less suited is the simplistic approach of Arduino sketches. Moreover, the limited memory of the ATmega1281 chip requires an efficient usage of code and program compilation. While the original intent pulled inspiration and promises from the ease of use of the Arduino ecosystem, our neophyte and naive approach was rapidly challenged. Those tempted to follow a similar endeavor may either 1) keep the Arduino/Do-It-Yourself approach by constraining themself to a limited number of tasks, or 2) seek the usage of advanced programming ecosystems *via* operating systems for embedded devices (e.g., RTOS, RIOT-OS).

The system presented here has been designed in order to be flexible to a wide range of situations and applications. The suite of sensors is expandable and adaptable to the needs of projects using established digital communication protocols (I2C, SDI-12, and UART). Its efficient energy management allows the system to be equally deployed in Arctic regions as well as in mid latitudes, in low and high elevations. Battery preservation is the highest priority activity followed by data sampling activities by the nodes over pushing data to the servers. The node's highest priority is to survive, then collect data, and if possible upload the data. Data is stored at every step of the transfer so that we minimize data loss.

While we met some successes, we still encountered a number of challenges, the reasons for which we have not been able to identify, but are suspectedly associated with hardware choice. Future development should focus on microprocessors with larger memory such as the ARM Cortex M architecture series, combined with multi-threading embedded device operating systems. We also learned that compartmenting design and development is key to identify and separate issues encountered allowing faster development. For this purpose, we isolated our sensors in v2 with their own microprocessor accomplishing a simple and limited number of tasks. The main board (waspmote) was then focused on triggering the sampling, managing the data and controlling network activity.

Improvements of the system described here could mainly focus on designing a new node version, both in terms of hardware and software. This node could be based on an ARM microprocessor, with a LoRa radio. The firmware could be entirely written in a general embedded device operating system such as RIOT-OS or RTOS. Such OS supports a broad number of hardware, and is maintained by a wider community than a development environment developed by a single company. OS for embedded devices includes a well tested set of technologies for networking, data brokerage, and time and power management. This new firmware could include functionality to receive configuration commands *via* the network, and generalize the usage of LIFO list for sending data through the network. Logging could also be improved to be machine readable and therefore be parsed and archived in the database. The configuration and interactive menu could be simplified further, minimizing the currently error prone menu. Finally, there exists disk filesystem format alternative options to FAT, such as littleFS, that is, more adapted to heavy usage of SD cards and resilient to power failure. In terms of hardware, the first priority is to decrease the number of parts to assemble, potentially taking advantage of 3D printing for replacing expensive parts (e.g., radiation shields). In terms of electronics, the different modules (RTC, SD card reader, power management, GPS, radio, and MCU) could all be integrated in one large PCB board reducing the number of wires and connections. With the advent of satellite constellations such as Starlink, and consequently the drop in the cost of satellite communication, specific low-power autonomous gateways could be designed in the future to reach remote places common to the Arctic. Overall, the two main principles to apply are 1) using robust open-source software for embedded devices, and 2) reducing everywhere possible the number of parts.

Following our experience, the widespread and popular premise of low-cost electronics seems a lure. The basic building blocks are cheap but combining them into reliable devices for a wide range of tasks demand extensive knowledge, and time for development quickly overcoming the low-cost premise. While it took years of development, we still see some advantages to the approach presented here over the classical data collection in geosciences using off-the-shelf sensors and loggers.

1. Ability to build custom and targeted applications,
2. Ability to lower the cost when scaling up the project,
3. Gaining technical knowledge required for future creative designs,
4. Increasing independence of research from commercial solutions therefore lowering cost and increasing sustainability on the long term,
5. Building smaller devices therefore reducing footprint in the landscape,
6. Integrating data autonomously into a data management and database system
7. Allowing targeted maintenance,

Within the scientific community these arguments are rarely brought forward at the expense of the apparent low-cost of the equipment with disregard to the costs associated with development. In the short term, the approach we followed to develop our own system and integrate a number of technologies has limited benefit. But if considered as a long term investment into knowledge, know-how, and infrastructure building, such an approach will start showing its advantages. Cohen. (2015) identifies a number of obstacles to be aware of and provides a broad overview of what development requires from a hands-on point of view. Furthermore, the combination of open-source license for software and hardware design with low-cost hardware parts is also an advantage to transfer the technologies in regions having limited financial resources which are also often plagued with low data coverage. Scaling up is the key in reducing cost. It can only be achieved after thorough prototyping, itself expensive and rarely suited to an academic setting. So scaling up may be achieved as in the example of the Mayfly (by EnviroDIY) in which an entire community commits to a set of technologies.

Most attempts available in the literature (Hart and Martinez, 2006; Pohl et al., 2014; Zhang et al., 2017; Mao et al., 2019) are presented as proof of concept, or tend to be experimental. The fast pace and myriad of technological developments, the technical literacy required and the lack of consolidated development across institutions lead to a fragmented community of developers. However, we show, similarly to previous studies, that our system produced data of quality and of greater spatial representativeness in a remote region at a relatively low cost in comparison to traditional existing solutions. As a roadmap to consolidate efforts of development across groups, we identify 6 domain of focus.

1. Appropriation of embedded Operating System technology for application in geosciences, with for instance wider support of sensors, and jargon-free documentation including tutorials,
2. Sharing of hardware design and packaging on open platforms (e.g., Github, Gitlab) with associated Bill-Of-Material or 3D print drawings. Such approach will improve robustness and quality, minimizing incompatibilities and redundancies (Chan et al., 2021),
3. Better integration of network development and deployment with the modeling community, what Varadharajan et al. (2019) refer as the co-design approach,
4. Development of sensors based on solid-state technology such as light sensors, lidar, radars, GPS, that include redundancy, and more advanced and adaptable sampling scheme than a single measurement at a single point,
5. Converge metadata collection system and quality control routines (Bavay and Egger, 2014; Horsburgh et al., 2019).
6. Plan for data dissemination either through institutional data repositories (as presented here) or *via* data sharing portal as in (Horsburgh et al., 2019)

This, we hope, can nurture further coordination and support for using IoT technologies in geoscience and lower development cost for individual groups.

## 5 Conclusion

We present here a custom-made system to collect meteorological data in the high-Arctic for spatially distributed observations over two glaciers. This system is made of a combination of hardware and software technologies that work together in collecting, managing, communicating, storing, and accessing data. We used as much open-source licensed technologies as we could and released our work under MIT License. This system was tested on two glaciers in the vicinity of Ny-Ålesund over a 1 year period. We installed two networks that survived the four-month-long polar night at 79° North. Our experience shows that low-cost sensors can perform significantly well in comparison to traditional sensors, and provide insights into spatial variability of environmental conditions over large glaciers. It also demonstrates the feasibility to design and deploy IoT technologies in the high Arctic (78°N) environment with 4 months long polar night.

The system is built to be flexible and adaptable to a broad range of context and conditions. Because it incorporates a number of digital protocols, or wireless communication systems, it may be used for purposes different from what it was tested for. It is fully replicable. Though we encountered a number of failures, with some not attributable to specific issues. These led us to look for alternative hardware solutions in the near future, in combination with more advanced embedded operating systems. Such operating systems are capable of multi-threading (multitasking), but mainly consist of a myriad of sound and established technological solutions. Coming from the simplistic scripting approach provided by the Arduino ecosystem, it took 5 years of incremental development and testing for the system to reach sufficient maturity to collect data for a full year, and to be currently running.

We identified major obstacles at the interface of hardware and software development that are often misunderstood by the scientific community. We present here key technology we identified for hardware, embedded software and for the rest of the data pipeline. We show that technologies developed for the Internet of Things industry can be brought into rough and extreme environments and can actually prove very effective. What requires special attention is how to combine and accommodate these technologies for a robust, reliable and consistent behavior in a broad range of adverse conditions.

We hope by sharing our experiences and choice of technology, other groups may not struggle as long as we may have had, or do not overestimate the task they may undertake. Developing a full stack solution from sensor to data quality demands a number of problem solving. We believe that more communication, collaboration and if possible coordination is needed across groups of developers that new teams can benefit from past mistakes. Communities of the open-source movement build large collaborative projects which proved to be beneficial for the greater good. In this regard, experiences accumulated in previous studies point to the need of converging efforts of development. Our contribution has been to identify what may be improved by the broader community for IoT technologies to be more prevalent in geosciences. They have a great potential to study and monitor processes heterogenous in space and time, improve transparency in collection methods, and significantly decrease cost in comparison to proprietary solutions. We show that an in depth appropriation of IoT technology may also be beneficial to innovation.

Lastly, the development of technology should be informed by scientific, monitoring or modeling needs. We see many avenues where technologies explored here may be relevant to other research groups or institutions across the geoscience fields accompanied with the large possibilities provided by the field of data science, numerical modeling, and remote sensing.

## Data availability statement

All the code required to replicate the system described here is available in open-source on Github: The node firmware program: https://github.com/spectraphilic/wasp_sketches. The server and database system: https://github.com/spectraphilic/wsn_server. The Raspberry Pi gateway: https://github.com/spectraphilic/wsn_pi. The Python API client: https://github.com/spectraphilic/wsn_client. The sensor firmware: https://github.com/spectraphilic/wsn_arduino. Installation procedures are available in the respective repositories and are mainly adapted for Linux platforms. All QC data are available from the Arctic Data Centre operated by the Norwegian Meteorological: https://doi.org/10.21343/1r1z-ae07 (Kongsvegen), https:/https://doi.org/10.21343/5ht4-bj50/doi.org/10.21343/5ht4-bj50 (Midtre Lovénbreen).

## Author contributions

The manuscript was written by SF and P-ML and edited by all other co-authors. SF, P-ML, and SH analyzed the data. JB, SF, P-ML, J-CG, and TS participated in collecting the required funding to support development and deployment. SF, JI, and JH are the main contributors for the development in software and hardware. SF, P-ML, and J-CG deployed and recovered the nodes deployed on the two glaciers. SH provided the AWS data. P-ML performed the quality control of the data.

## Funding

This work was possible thanks to the funding support, in alphabetic order, from the Norwegian Research Council–Enhancing Snow Competency of Models and Operators (ESCYMO) project (NFR #244024), the University of Oslo eInfrastructure Competence Hub–Geohive, and the Norwegian Research Council—project “SvalbardIntegrated Arctic Earth Observing System Infrastructure development of the Norwegian Node SIOS-InfraNOR” (NFR # 269927).

## Acknowledgments

We, first, would like to acknowledge the support team at the research station of Ny-Ålesund, in particular the two chefs Marin and Thea, for making our life productive and comfortable in the field at 79° North. We would like to thank the Glacier team from the Norwegian Polar Institute led by Jack Kohler for smoothed and sustained collaboration and coordination in the field during deployment. Most of the needy-greedy testing took place at the research station of Finse where we would like to thank Erika Leslie. Finally, we would like to thank Mathias Bavay and Joel Fiddes for introducing us to MeteoIO, as well as Øystein Godøy and Lara Ferrighi at the Norwegian Meteorological Institute for support in releasing the QC data with a DOI.

## Conflict of interest

Author JB was employed by the company Statkraft AS.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher’s note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## References

Baccelli, E., Gundogan, C., Hahm, O., Kietzmann, P., Lenders, M. S., Petersen, H., et al. (2018). Riot: An open source operating system for low-end embedded devices in the IoT. *IEEE Internet Things J.* 5, 4428–4440. doi:10.1109/jiot.2018.2815038

Bavay, M., and Egger, T. (2014). MeteoIO 2.4.2: A preprocessing library for meteorological data. *Geosci. Model. Dev.* 7, 3135–3151. doi:10.5194/gmd-7-3135-2014

Beine, H. J., Argentini, S., Maurizi, A., Mastrantonio, G., and Viola, A. (2001). The local wind field at Ny–lesund and the Zeppelin mountain at Svalbard. *Atmos. Phys.* 78, 107–113. doi:10.1007/s007030170009

Bowen, B. (2007). *Improved wind and turbulence measurements using a low-cost 3-D sonic anemometer at a low-wind site. Lawrence livermore national lab.* Livermore, CA (United States): LLNL. doi:10.2172/926051

Chae, N., Yang, H., Lee, B. Y., and Lee, C. (2016). Measurement of environmental parameters in polar regions based on a ubiquitous sensor network. *Cold Reg. Sci. Technol.* 123, 22–31. doi:10.1016/j.coldregions.2015.11.003

Chakraborty, M., and Kundan, A. (2021). “Grafana,” in *Monitoring cloud-native applications* (Berkeley, CA: Apress. doi:10.1007/978-1-4842-6888-9_6

Chan, K., Schillereff, D. N., Baas, A. C., Chadwick, M. A., Main, B., Mulligan, M., et al. (2021). Low-cost electronic sensors for environmental research: Pitfalls and opportunities. *Prog. Phys. Geogr. Earth Environ.* 45, 305–338. doi:10.1177/0309133320956567

Cohen, A. (2015). *Prototype to product: A practical guide for getting to market.* Sebastopol, CA: O’Reilly Media, Inc.

Cui, G., Rice, R., Avanzi, F., Hartsough, P., Guo, W., Anderson, M., et al. (2022). Precipitation estimates and orographic gradients using snow, temperature, and humidity measurements from a wireless-sensor network. *Water Resour. Res.* 58, e2021WR029954. doi:10.1029/2021wr029954

Django Software Foundation (2019). Django. Available at: https://djangoproject.com.

ESIP (2022). Attribute convention for data Discovery 1-3 earth sci. Inf. Partn. ESIP. Available at: https://wiki.esipfed.org/Attribute_Convention_for_Data_Discovery_1-3.

Hagen, J. O., and Liestøl, O. (1990). Long-term glacier mass-balance investigations in svalbard, 1950–88. *Ann. Glaciol.* 14, 102–106. doi:10.3189/s0260305500008351

Hart, J. K., and Martinez, K. (2006). Environmental sensor networks: A revolution in the Earth system science? *Earth-Sci. Rev.* 78, 177–191. doi:10.1016/j.earscirev.2006.05.001

Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., et al. (2020). The ERA5 global reanalysis. *Q. J. R. Meteorol. Soc.* 146, 1999–2049. doi:10.1002/qj.3803

Horsburgh, J. S., Caraballo, J., Ramírez, M., Aufdenkampe, A. K., Arscott, D. B., and Damiano, S. G. (2019). Low-cost, open-source, and low-power: But what to do with the data? *Front. Earth Sci.* 7, 67. doi:10.3389/feart.2019.00067

Ignatiuk, D., Błaszczyk, M., Budzik, T., Grabiec, M., Jania, J. A., Kondracka, M., et al. (2022). A decade of glaciological and meteorological observations in the Arctic (Werenskioldbreen, Svalbard). *Earth Syst. Sci. Data* 14, 2487–2500. doi:10.5194/essd-14-2487-2022

Ioannou, K., Karampatzakis, D., Amanatidis, P., Aggelopoulos, V., and Karmiris, I. (2021). Low-cost automatic weather stations in the internet of Things. *Information* 12, 146. doi:10.3390/info12040146

Isaksen, K., Nordli, Ø., Ivanov, B., Køltzow, M. A., Øaaboe, S., Gjelten, H. M., et al. (2022). Exceptional warming over the Barents area. *Sci. Rep.* 12, 9371. doi:10.1038/s41598-022-13568-5

Joseph, K. M., Watteyne, T., and Kerkez, B. (2018). Awa: Using water distribution systems to transmit data. *Technol* 29, e3219. doi:10.1002/ett.3219

Lundquist, J., Hughes, M., Gutmann, E., and Kapnick, S. (2019). Our skill in modeling mountain rain and snow is bypassing the skill of our observational networks. *Bull. Am. Meteorol. Soc.* 100, 2473–2490. doi:10.1175/bams-d-19-0001.1

Malek, S. A., Avanzi, F., Brun-Laguna, K., Maurer, T., Oroza, C. A., Hartsough, P. C., et al. (2017). Real-time alpine measurement system using wireless sensor networks. *Sensors* 17, 2583. doi:10.3390/s17112583

Malek, S. A., Glaser, S. D., and Bales, R. C. (2019). Wireless sensor networks for improved snow water equivalent and runoff estimates. *IEEE Access* 7, 18420–18436. doi:10.1109/access.2019.2895397

Mao, F., Khamis, K., Krause, S., Clark, J., and Hannah, D. M. (2019). Low-cost environmental sensor networks: Recent advances and future directions. *Front. Earth Sci.* 7, 221. doi:10.3389/feart.2019.00221

Mellis, D. A., Banzi, M., Cuartielles, D., and Igoe, T. (2007). "Arduino: An open electronics prototyping platform,"Proceedings of 2007 Computer/Human Interaction Conference (CHI), San Jose, CA (MIT).

Melvold, K., and Hagen, J. O. (1998). Evolution of a surge-type glacier in its quiescent phase: Kongsvegen, spitsbergen, 1964–95. *J. Glaciol.* 44, 394–404. doi:10.3189/s0022143000002720

Michael, P. R., Johnston, D. E., and Moreno, W. (2020). A conversion guide: Solar irradiance and lux illuminance. *J. Meas. Eng.* 8, 153–166. doi:10.21595/jme.2020.21667

Østby, T. I., Schuler, T. V., Hagen, J. O., Hock, R., Kohler, J., and Reijmer, C. H. (2017). Diagnosing the decline in climatic mass balance of glaciers in Svalbard over 1957-2014. *Cryosphere* 11, 191–215. doi:10.5194/tc-11-191-2017

Pelt, W. V., and Kohler, J. (2015). Modelling the long-term mass balance and firn evolution of glaciers around Kongsfjorden, Svalbard. *Svalbard. J. Glaciol.* 61, 731–744. doi:10.3189/2015jog14j223

Pohl, S., Garvelmann, J., Wawerla, J., and Weiler, M. (2014). Potential of a low-cost sensor network to understand the spatial and temporal dynamics of a mountain snow cover. *Water Resour. Res.* 50, 2533–2550. doi:10.1002/2013wr014594

Schuler, T. V., Dunse, T., Østby, T. I., and Hagen, J. O. (2014). Meteorological conditions on an Arctic ice cap-8 years of automatic weather station data from Austfonna, Svalbard. *Svalbard. Int. J. Climatol.* 34, 2047–2058. doi:10.1002/joc.3821

Schuler, T. V., Kohler, J., Elagina, N., Hagen, J. O. M., Hodson, A. J., Jania, J. A., et al. (2020). Reconciling svalbard glacier mass balance. *Front. Earth Sci.* 8, 156. doi:10.3389/feart.2020.00156

Schuler, T. V., and Østby, T. I. (2020). Sval_Imp: A gridded forcing dataset for climate change impact research on svalbard. *Earth Syst. Sci. Data* 12, 875–885. doi:10.5194/essd-12-875-2020

Välisuo, I., Zwinger, T., and Kohler, J. (2017). Inverse solution of surface mass balance of Midtre Lovénbreen, Svalbard. *Svalbard. J. Glaciol.* 63, 593–602. doi:10.1017/jog.2017.26

van Pelt, W. J. J., Oerlemans, J., Reijmer, C. H., Pohjola, V. A., Pettersson, R., and van Angelen, J. H. (2012). Simulating melt, runoff and refreezing on Nordenskiöldbreen, Svalbard, using a coupled snow and energy balance model. *Cryosphere* 6, 641–659. doi:10.5194/tc-6-641-2012

Varadharajan, C., Agarwal, D. A., Brown, W., Burrus, M., Carroll, R. W. H., Christianson, D. S., et al. (2019). Challenges in building an end-to-end system for acquisition, management, and integration of diverse data from sensor networks in watersheds: Lessons from a mountainous community observatory in East river, Colorado. *IEEE Access* 7, 182796–182813. doi:10.1109/access.2019.2957793

Wilkinson, M. D., Dumontier, M., Aalbersberg, G., Appleton, G., Axton, M., Baak, A., et al. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Sci. Data* 3, 160018. doi:10.1038/sdata.2016.18

Wong, B. P., and Kerkez, B. (2016). Real-time environmental sensor data: An application to water quality using web services. *Environ. Model. Softw.* 84, 505–517. doi:10.1016/j.envsoft.2016.07.020

Wyngaard, J. C. (1981). Cup, propeller, vane, and sonic anemometers in turbulence research. *Annu. Rev. Fluid Mech.* 13, 399–423. doi:10.1146/annurev.fl.13.010181.002151

Zhang, Z., Glaser, S. D., Bales, R. C., Conklin, M., Rice, R., and Marks, D. G. (2017). Technical report: The design and evaluation of a basin-scale wireless sensor network for mountain hydrology. *Water Resour. Res.* 53, 4487–4498. doi:10.1002/2016wr019619

Zhang, Z., Glaser, S., Watteyne, T., and Malek, S. (2022). Long-term monitoring of the sierra Nevada snowpack using wireless sensor networks. *IEEE Internet Things J.* 9, 17185–17193. doi:10.1109/jiot.2020.2970596