



# UNIVERSITÀ DEGLI STUDI DI GENOVA

DIPARTIMENTO DI INGEGNERIA NAVALE, ELETTRICA,  
ELETTRONICA E DELLE TELECOMUNICAZIONI

CORSO DI STUDIO IN INGEGNERIA ELETTRONICA E  
TECNOLOGIE DELL'INFORMAZIONE

Tesi di laurea triennale

Febbraio 2024

**Progetto e implementazione di un sistema embedded per il  
riconoscimento di attività sciistica**

**Design and development of an embedded system for ski activity  
recognition**

Candidato: Filippo La Fauci

Relatore: Prof. Riccardo Berta

Correlatore: Dott. Matteo Fresta

## Sommario

Il presente elaborato è stato redatto con l'obiettivo di illustrare la progettazione e l'implementazione di un sistema embedded per il riconoscimento di attività sciistiche. Inizialmente, il sistema è finalizzato alla raccolta di dati rilevanti ottenuti dai sensori di una scheda elettronica Arduino, per poi successivamente addestrare un algoritmo di machine learning, capace di identificare autonomamente specifiche tecniche nel contesto dell'attività sciistica. Per raggiungere questo scopo, è stato impiegato un dispositivo Arduino Nano 33 BLE Sense, il quale raccoglie i valori relativi all'accelerometro, al giroscopio e al magnetometro durante una discesa lungo una pista da sci, focalizzandosi, nel nostro caso, sulle diverse tecniche sciistiche quali lo slalom, lo spazzaneve, la posizione a uovo e la spinta. I dati acquisiti vengono trasmessi a un dispositivo mobile connesso alla scheda elettronica attraverso la tecnologia Bluetooth Low Energy (BLE), tramite un'applicazione sviluppata mediante il framework Flutter. Infine, per la raccolta dei dati, essi vengono inviati a un API Framework denominato Measurify. Il mio interesse era riuscire ad approcciare un caso d'uso collegato a una pratica sportiva comune, non necessariamente agonistica, dimostrando che grazie a componenti hardware e software a basso costo è possibile ottenere risultati utili alla progettazione di un prototipo più strutturato, multi-sensore, in grado di analizzare qualitativamente la singola tecnica sciistica.

# Indice

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introduzione.....</b>                                     | <b>4</b>  |
| 1.1      | Flusso di lavoro .....                                       | 6         |
| <b>2</b> | <b>Metodi e strumenti utilizzati.....</b>                    | <b>7</b>  |
| 2.1      | Scheda di sviluppo Arduino .....                             | 7         |
| 2.2      | Applicazione sviluppata con Flutter .....                    | 8         |
| 2.3      | Measurify .....  | 11        |
| 2.4      | Postman .....  | 12        |
| <b>3</b> | <b>Sperimentazione e risultati .....</b>                     | <b>14</b> |
| 3.1      | Creazione e raccolta del dataset .....                       | 14        |
| 3.2      | Addestramento del modello di Machine Learning .....          | 16        |
| <b>4</b> | <b>Contributo personale e considerazioni conclusive.....</b> | <b>18</b> |
| <b>5</b> | <b>Riferimenti bibliografici .....</b>                       | <b>19</b> |
| <b>6</b> | <b>Ringraziamenti .....</b>                                  | <b>20</b> |

# 1 Introduzione

L'avvento dell'Internet of Things (IoT), grazie al collegamento di una vasta rete di dispositivi intelligenti, ha portato a una vera e propria rivoluzione in vari settori, tra cui quello sportivo. Questa trasformazione tecnologica ha notevolmente migliorato la capacità di raccolta, di elaborazione e utilizzo dei dati in tempo reale, aprendo nuove frontiere nell'allenamento personalizzato, nella prevenzione degli infortuni e nell'ottimizzazione delle prestazioni degli atleti. Nel mondo dello sport, l'IoT trova applicazione mediante l'uso di sensori avanzati e dispositivi indossabili, che forniscono dati rilevati in tempo reale, come il monitoraggio della frequenza cardiaca e l'analisi dell'accelerazione, come per esempio i nuovi smartwatch. Questa tecnologia consente agli allenatori di personalizzare gli allenamenti in base alle esigenze specifiche degli atleti, migliorando così l'efficacia degli stessi. Inoltre, ha reso possibile un coinvolgimento più intenso e informato degli appassionati, i quali possono godere di una visione più dettagliata e approfondita delle performance degli atleti.

La mia ricerca in questo contesto, mira a sviluppare un sistema che applichi il machine learning nel contesto della attività sciistica. Questa disciplina sportiva si basa sull'utilizzo di diverse tecniche eseguite lungo le discese, le quali dipendono strettamente dall'abilità dello sciatore.

In questo progetto sono state prese in considerazione le seguenti tecniche:

- **Spazzaneve:** la prima posizione che si insegna a uno sciatore è la posizione “a spazzaneve” in cui gli sci sono ruotati verso l'interno, con le punte vicine e le code lontane, le gambe sono inclinate verso l'interno in modo che gli sci siano in appoggio sulle lamine interne. Questa posizione frena lo sciatore che può scendere su piste poco ripide a bassa velocità; per fermarsi completamente è sufficiente aumentare l'inclinazione laterale delle gambe, mentre spostando il peso da una gamba all'altra, si curverà a sinistra o a destra.



*Figura 1.1. Tecnica Spazzaneve*

- **Slalom:** è la disciplina più tecnica dello sci alpino, la più utilizzata ed esteticamente interessante da vedere. Sulla pista vengono disegnati dagli sci degli archi quasi perfetti. Le curve sono molto veloci e richiedono rapidi cambi di direzione.



*Figura 1.2. Tecnica Slalom*

- **Posizione a Uovo:** è una posizione tipica dello sci alpino volta ad avere una maggiore aerodinamicità, lo sciatore cerca di diminuire la superficie di impatto del suo corpo con l'aria per aumentare la velocità.



*Figura 1.3. Tecnica Uovo*

- **Spinta:** è un movimento accessorio importante nelle competizioni perché nelle partenze dal “cancelletto” determina sovente il buon inizio della performance, mentre è determinante nei tratti di pista meno inclinati, in quanto permette di mantenere una velocità adeguata.

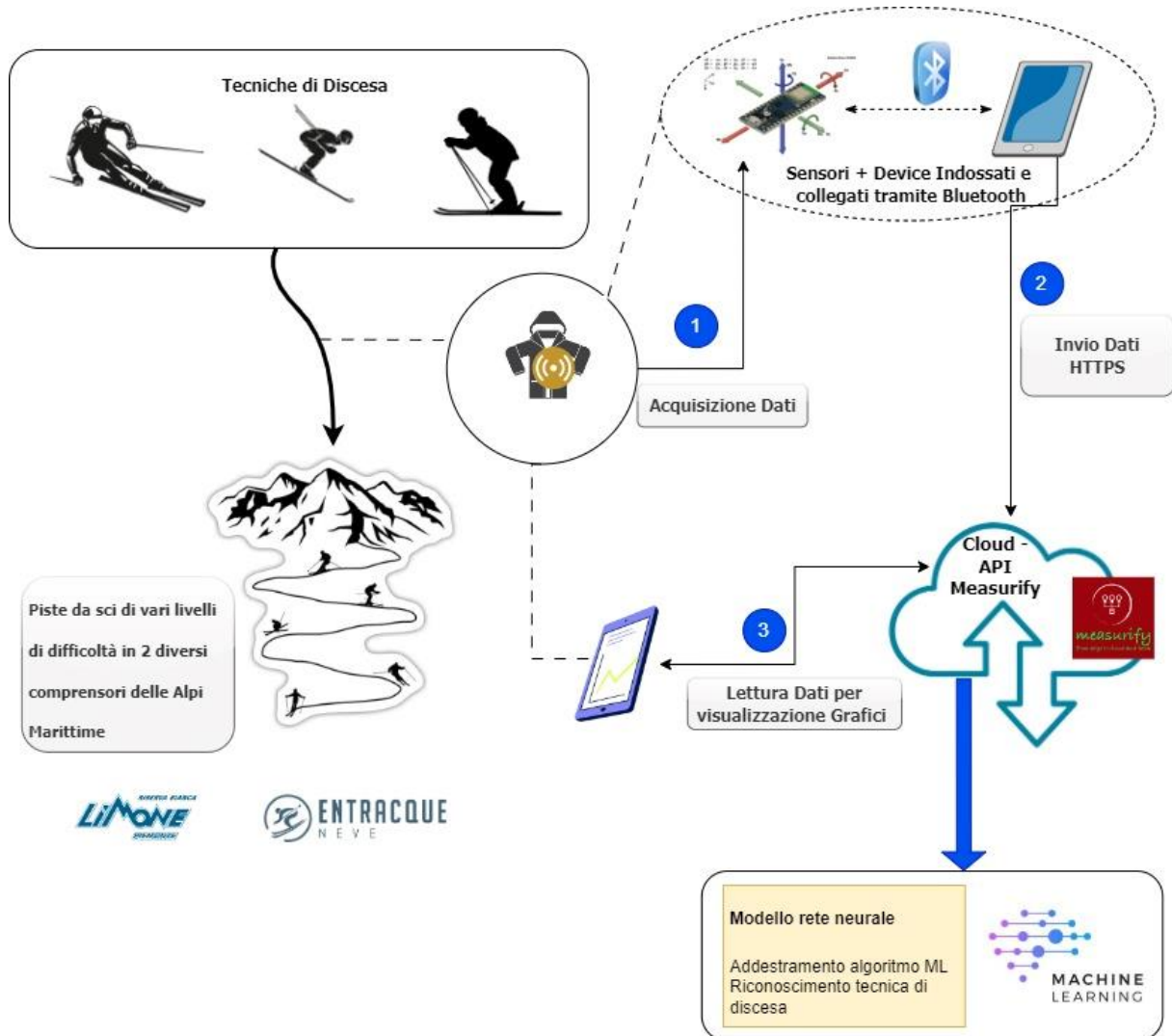


*Figura 1.4. Tecnica Spinta*

L'obiettivo primario è creare un sistema sofisticato in grado di riconoscere con precisione la tecnica impiegata da uno sciatore, in relazione anche al suo livello di abilità. Utilizzando dati provenienti da sensori integrati nella strumentazione indossata dallo sciatore, il machine learning sarà addestrato per identificare e distinguere diverse tecniche sciistiche. Questo approccio non solo migliorerà la comprensione delle prestazioni degli sciatori, ma consentirà anche una personalizzazione più accurata degli allenamenti, adattandoli al livello specifico di competenza di ciascun individuo.

## 1.1 Flusso di lavoro

Il progetto è suddiviso in varie fasi di lavoro, viene riportato di seguito uno schema che rappresenta il processo di funzionamento del sistema embedded di rilevamento delle tecniche sciistiche.



*Figura 1.5. Schema di sintesi del flusso progettuale*

Per implementare questo approccio innovativo, si fa uso di un sensore specializzato in grado di raccogliere dati fondamentali relativi ad accelerometro, magnetometro e giroscopio. Questi dati, una volta acquisiti, vengono trasmessi in modalità Bluetooth Low Energy all'apposito dispositivo mobile. Quest'ultimo, tramite un'applicazione (APP), permette la raccolta dei dati e il loro invio tramite API ai servizi Measurify in cloud.

Attraverso l'interfaccia di questa applicazione, l'utente può non solo interagire con l'Arduino per avviare e fermare le misurazioni, ma può anche visualizzare in tempo reale i dati raccolti mediante diagrammi cartesiani.

L'applicazione mobile si configura come elemento chiave per la gestione del ciclo di vita dei dati raccolti:

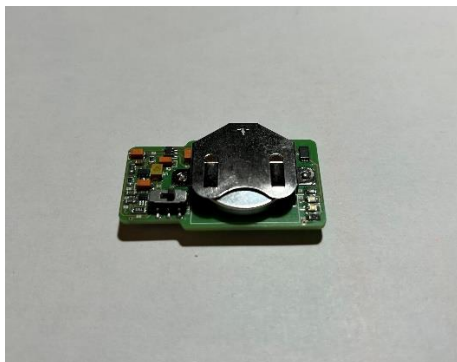
- acquisizione in formato grezzo,
- aggregazione in una serie temporale (singola discesa sciistica),
- aggiunta di una etichetta [tags] e invio al server con il database per la memorizzazione del dataset,
- consultazione dei dati in modalità real-time.

La fase successiva del processo coinvolge l'impiego del machine learning. Grazie ai dati precedentemente raccolti e memorizzati, è possibile creare un modello di machine learning avanzato che avrà il compito di classificare autonomamente le principali tecniche sciistiche, aprendo la strada a un riconoscimento preciso e personalizzato che potrà essere la base per poi valutare le abilità degli sciatori. L'utilizzo integrato di sensori, dispositivi mobili e machine learning può contribuire a un'analisi e valutazione delle performance nello sci.

## 2 Metodi e strumenti utilizzati

### 2.1 Scheda di sviluppo Arduino

Il sistema in esame è costituito da una scheda elettronica prodotta da Arduino che combina la potenza di un microcontrollore, con funzionalità Bluetooth Low Energy (BLE), e una serie di sensori integrati che consentono alla scheda di rilevare e misurare varie grandezze ambientali e fisiche, suddivise in tre categorie: IMU (Inertial Measurement Unit), Environment e Orientation. La combinazione di queste caratteristiche rende l'Arduino Nano 33 BLE Sense [1] la scelta migliore per progetti che richiedono movimento e orientamento nello spazio, interazione con dispositivi esterni tramite connessione Bluetooth ed elaborazione dati on board; inoltre, le sue dimensioni ridotte (18 x 45 mm) permettono di integrarla facilmente nelle attrezzature tecniche sportive. A questo fine è stato creato dal laboratorio Elios Lab una board personalizzata con batteria a bottone per garantire un'alimentazione autonoma. Nel caso in esame in cui l'attività sportiva viene svolta all'aperto e a basse temperature ambientali, alla fine della durata della batteria si può utilizzare un power bank per l'alimentazione via cavo e garantire per l'intera giornata l'efficienza costante del dispositivo. La tensione consigliata per la versione rev2 arriva fino a 5V; è importante verificare che il power bank fornisca una tensione entro questo valore.

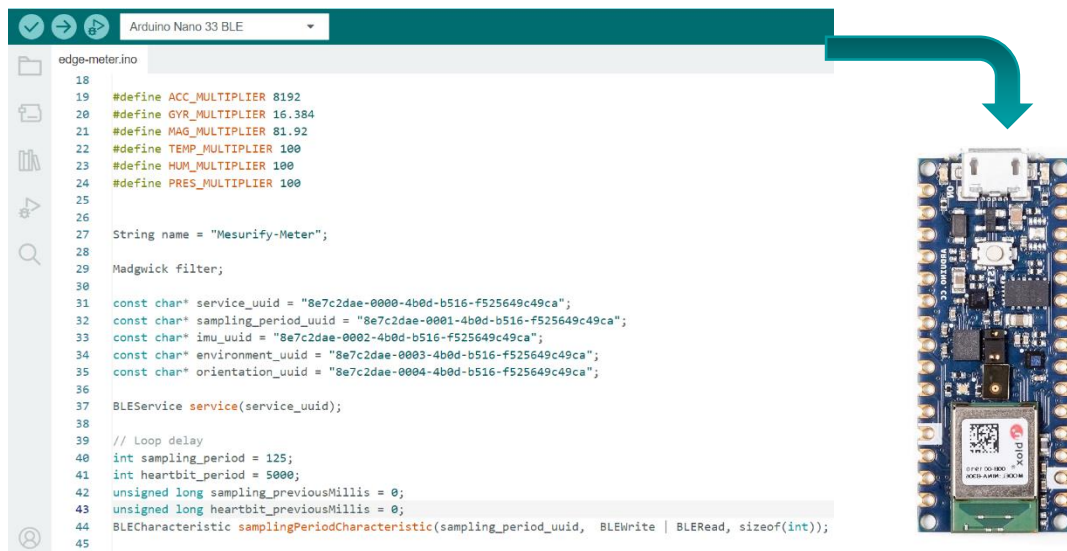


*Figura 2.1. Arduino con batteria custom*



*Figura 2.2. Arduino alimentato via cavo*

Arduino IDE [2] permette di implementare e caricare il codice sulla scheda in modo da gestire la connessione Bluetooth con il dispositivo mobile e l'invio dei dati. I valori che sono stati trasmessi riguardano la caratteristica IMU, la sola utilizzata in questo progetto, che può fornire informazioni sull'accelerazione lineare, sulla velocità angolare e sul campo magnetico consentendo di tracciare il movimento in tre dimensioni nello spazio XYZ. I valori rilevati su ognuno dei tre assi sono rappresentati in 9 float; tuttavia, la connessione BLE impone un limite di trasmissione dati di 20 byte per pacchetto, si è optato di convertire i valori da float32 a int16 per ridurre la finestra di invio da 36 byte (9 valori  $\times$  4 byte = 36 byte) a 18 bytes (9 valori  $\times$  2 byte = 18 byte). Questo è possibile grazie all'aggiunta nel codice, di costanti moltiplicative per ognuno dei nove valori. Altro aspetto fondamentale è la scelta relativa al periodo di campionamento, nel nostro caso pari a 125 millisecondi. È importante sottolineare che i valori del magnetometro, benché siano raccolti e inviati dalla scheda, non vengono presi in considerazione in quanto non rilevanti per la generazione del modello di machine learning e quindi per lo scopo di questa tesi.

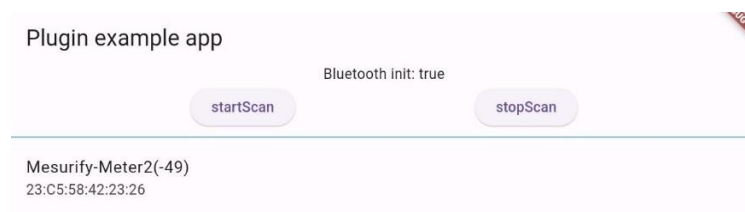


**Figura 2.3.** Codice caricato su Arduino, si evidenziano in particolare le costanti moltiplicative da riga 19 a riga 24 e il periodo di campionamento a riga 40

## 2.2 Applicazione sviluppata con Flutter

Flutter [3] è un framework open-source sviluppato da Google per la creazione di applicazioni cross-platform. Ciò significa che si può utilizzare un singolo codice sorgente per sviluppare app che possono essere eseguite su diverse piattaforme, tra cui Android, iOS, web e desktop. L'obiettivo principale di Flutter è fornire un modo efficace per costruire interfacce utente moderne e reattive. Flutter è scritto principalmente in Dart, un linguaggio di programmazione moderno sviluppato anch'esso da Google. Dart è progettato per essere facile da apprendere, offrendo un ambiente di sviluppo efficiente e una sintassi pulita.

Dopo aver configurato l'Arduino Nano 33 BLE Sense al fine di acquisire dati dai suoi sensori, si stabilisce una connessione Bluetooth con l'Arduino. Questo può essere fatto attraverso il codice Flutter che gestisce la connettività BLE e la comunicazione dati. Una volta stabilita la connessione, l'app Flutter può inviare richieste all'Arduino per ottenere i dati. L'Arduino riceve la richiesta, legge i dati dai suoi sensori e li trasmette all'app Flutter attraverso la connessione BLE. Questi dati possono includere informazioni sull'accelerazione, l'orientamento e altri parametri rilevati dai sensori.



**Figura 2.4.** connessione all'Arduino tramite Bluetooth

Per il caso in questione viene utilizzata la libreria `quick_blue`, un plugin BluetoothLE multiplatforma per Flutter. L'applicazione in uso è stata denominata Smart Collector; presenta alcune pagine iniziali adibite alla connessione del dispositivo di misura e una pagina di configurazione in cui sono presenti soprattutto il link e il token necessari al collegamento al servizio cloud per la raccolta dati.



**Figura 2.5.** Pagina di configurazione che permette il collegamento al server

La pagina principale è la Start Page in cui si sceglie il tag che si vuole analizzare (tecnica), la misurazione viene identificata con un nome, si seleziona la feature IMU per far sì che vengano raccolti i dati relativi alle grandezze prese in considerazione e infine si avvia la raccolta dati con un contatore in basso a sinistra che conferma i dati in arrivo; premendo il tasto “Stop and Send” si ferma la misurazione e se tutto si è svolto correttamente apparirà il messaggio “Values sended correctly”, ovvero i dati raccolti sono stati inviati al servizio cloud.

**Figura 2.6.** Raccolta e invio dati al server con messaggio di feedback

Successivamente si è pensato di rendere più veloce la visualizzazione e analisi delle serie temporali, è stata così creata un’ulteriore pagina, la Chart Page, in cui fosse possibile, dopo ogni misurazione, visualizzare l’andamento complessivo dei campioni raccolti in un grafico interattivo [4], oltre a consultare lo storico delle misurazioni effettuate dallo specifico tenant per poter ad esempio cogliere graficamente differenze con serie temporali precedenti. Questa funzione è disponibile anche nella Dashboard Measurify, tuttavia, l’implementazione direttamente sull’app ha permesso di eliminare i tempi di accesso al portale tramite browser con relative credenziali di autenticazione rendendo la user experience dell’utente più agile.

Il codice vede i suoi punti cardine in due funzioni asincrone, “getAllMeasurements” e “getSelectTimeseries”, la prima si occupa di recuperare e stampare la lista completa di identificatori di misurazioni da Measurify tramite una richiesta HTTP di tipo GET a un URL specificato e grazie a un token di autorizzazione.

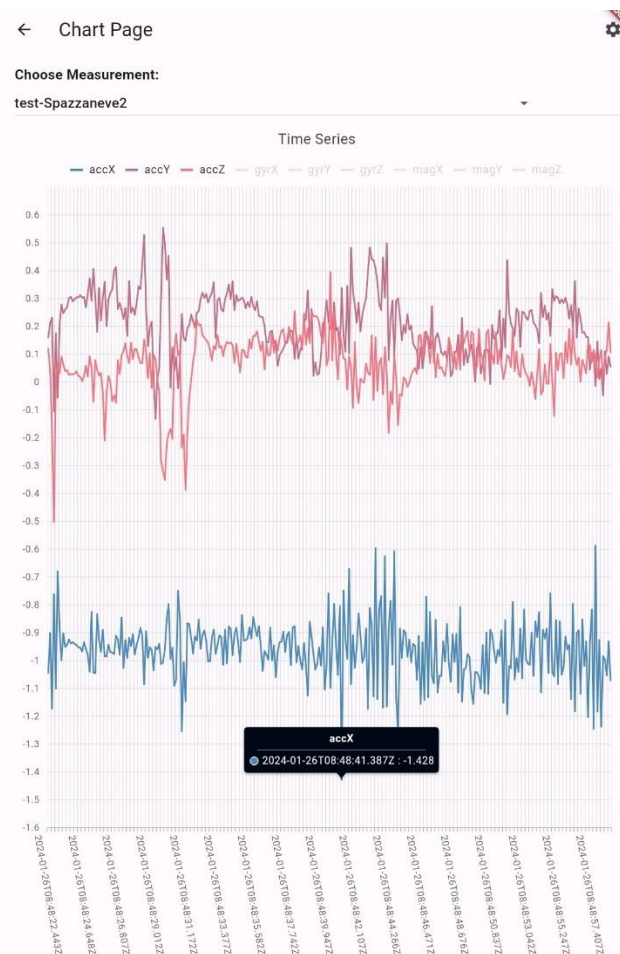
```

143 Future<List<String>> getAllMeasurements() async {
144   List<String> allMeasurements = [];
145   var headers = {'Authorization': widget.globals.deviceToken};
146   var request = http.Request(
147     'GET',
148     Uri.parse(
149       widget.globals.url+'measurements?select["_id"]&limit=-1');
150   request.headers.addAll(headers);
151   http.StreamedResponse response = await request.send();
152
153   if (response.statusCode == 200) {
154     String jsonResponse = await response.stream.bytesToString();
155     Map<String, dynamic> measurements = jsonDecode(jsonResponse);
156
157     allMeasurements.addAll(
158       measurements["docs"].map((measure) => measure["_id"].toString()).cast<String>());
159   } else {
160     print(response.reasonPhrase);
161   }
162   return allMeasurements;
163 }
164 }
165 }

```

**Figura 2.7.** Codice della funzione *getAllMeasurements*

Per quanto riguarda la seconda funzione viene inviato lo stesso tipo di richiesta, ma lo scopo in questo caso è ottenere tutti i valori sui 9 assi di ogni serie temporale, fino a formare la lista completa di tutte le serie memorizzate. Nel grafico finale sarà poi possibile scegliere in modo interattivo gli assi su cui focalizzarsi oscurando quelli non necessari, per esempio nel nostro caso i tre assi relativi ai dati sul magnetometro.



**Figura 2.8.** Chart Page con grafico interattivo rappresentante una serie temporale

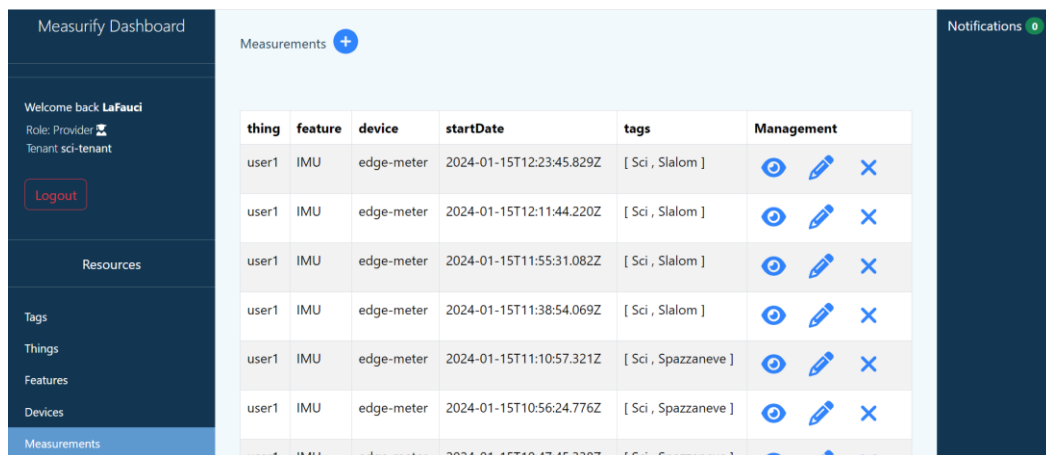
## 2.3 Measurify

Measurify [5] è un API cloud (di tipo RESTful) sviluppato dall'Elios Lab dell'università di Genova per la raccolta e l'elaborazione di dati provenienti da varie sorgenti dati, compresi sensori sul campo in ambito IoT; è risultato fondamentale come server remoto per la memorizzazione dei dati e il successivo impiego per la creazione del modello di machine learning atto a classificare autonomamente le principali tecniche sciistiche.

Gli elementi essenziali comuni nell'ambiente IoT sono: **Thing, Feature, Device, Measurement e Tag**:

- **Thing** è il soggetto (una persona, un ambiente) per la quale si sta facendo una misurazione.
- **Feature** descrive la quantità (tipicamente fisica) che si sta misurando tramite un Dispositivo.
- **Device** è uno strumento (hardware/software) che fornisce misure di una certa Feature riguardanti una Thing.
- **Measurement** rappresenta una misura effettuata da un dispositivo per una certa quantità su una determinata cosa.
- **Tag** è una etichetta che può essere aggiunta a una misurazione, utile per distinguere un gruppo di misurazioni da un altro.

Nel caso in questione, il soggetto della misurazione quindi la Thing sarà lo sciatore, che vorrà usufruire dei sensori caratteristici dell'IMU (Feature), tramite la scheda Arduino (Device); il risultato finale saranno una serie di misurazioni caratterizzate da questi tre elementi e contraddistinte da una StartDate e da un Tag tra i 4 possibili: Slalom, Spazzaneve, Posizione a Uovo, Spinta.

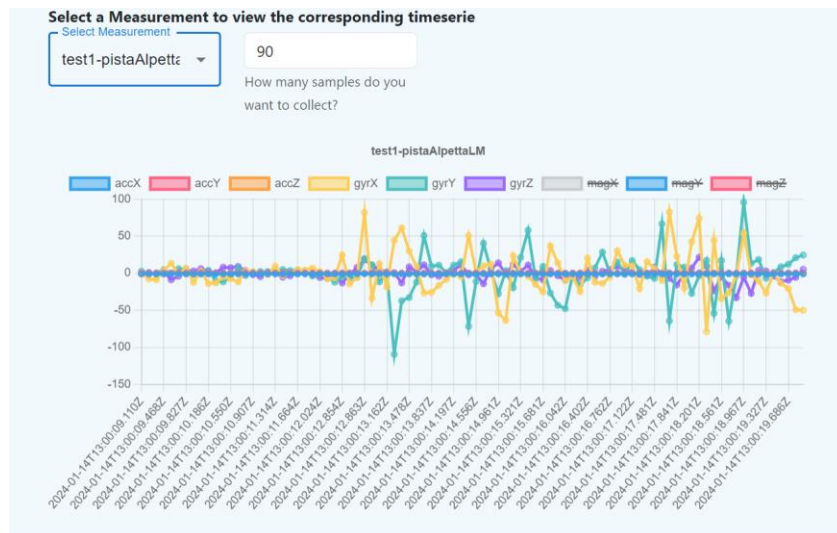


The screenshot shows the Measurify Dashboard. On the left is a dark sidebar with navigation links: Welcome back LaFauci, Role: Provider, Tenant sci-tenant, Logout, Resources, Tags, Things, Features, Devices, and Measurements (highlighted). The main area has a 'Measurements' header with a plus icon. Below it is a table with columns: thing, feature, device, startDate, tags, and Management. The table contains 7 rows of measurement data. On the right is a 'Notifications' sidebar with a green circle and the number 0.

| thing | feature | device     | startDate                | tags                 | Management |
|-------|---------|------------|--------------------------|----------------------|------------|
| user1 | IMU     | edge-meter | 2024-01-15T12:23:45.829Z | [ Sci , Slalom ]     |            |
| user1 | IMU     | edge-meter | 2024-01-15T12:11:44.220Z | [ Sci , Slalom ]     |            |
| user1 | IMU     | edge-meter | 2024-01-15T11:55:31.082Z | [ Sci , Slalom ]     |            |
| user1 | IMU     | edge-meter | 2024-01-15T11:38:54.069Z | [ Sci , Slalom ]     |            |
| user1 | IMU     | edge-meter | 2024-01-15T11:10:57.321Z | [ Sci , Spazzaneve ] |            |
| user1 | IMU     | edge-meter | 2024-01-15T10:56:24.776Z | [ Sci , Spazzaneve ] |            |
| user1 | IMU     | edge-meter | 2024-01-15T10:47:45.338Z | [ Sci , Spazzaneve ] |            |

**Figura 2.9.** Dashboard di Measurify [6] con le misurazioni raccolte

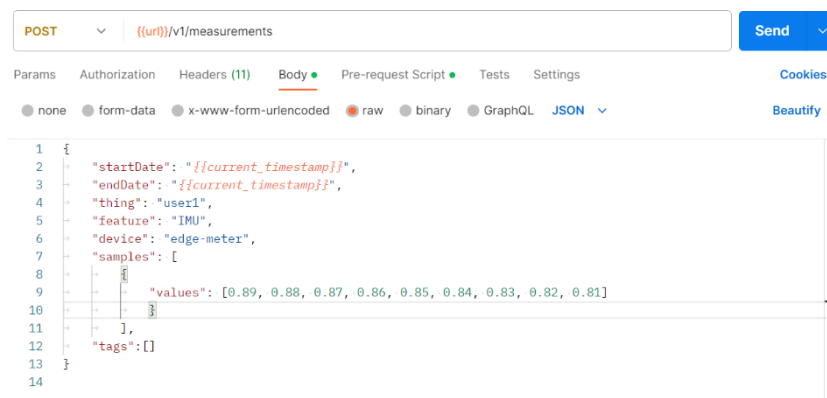
Inoltre, una volta effettuata una misurazione sarà possibile, tramite il tool **Visualize Timeseries**, visualizzare i dati ottenuti in real-time su un grafico che tiene in considerazione tutte e 9 le variabili in gioco, con la possibilità di focalizzarsi singolarmente su una o più di esse e sul numero di campioni che si vuole prendere in esame rispetto al totale registrato. Questa funzione di Measurify è stata essenziale nell'analisi delle tecniche effettuata dopo la raccolta dei dati ed è stata fondamentale nella divisione dei campioni per l'addestramento del modello di machine learning.



**Figura 2.10.** Tool di visualizzazione delle serie temporali presente nella Dashboard

## 2.4 Postman

Postman [7] rappresenta uno strumento fondamentale per lo sviluppo e il testing di API, fornite da servizi web o applicazioni, rivelandosi di grande utilità per sviluppatori e team. Nel suo ambiente organizzato, noto come "Workspace", è possibile gestire progetti, richieste API e collezioni. È inoltre possibile specificare il tipo di richiesta (GET, POST, ecc.), l'URL, gli header, i parametri e il corpo della richiesta. Ciò consente di simulare interazioni con l'API come se fosse chiamata da un'applicazione.



**Figura 2.11.** Post della prima misura inserita

Nel mio caso l'uso di Postman è stato solo preliminare e didattico, prima di effettuare dei test di invio dati tramite Flutter e Measurify, volevo familiarizzare con questo tipo di procedura e con le richieste HTTP così ho provato a caricare dei dati manualmente tramite Postman per testarne il funzionamento e i valori ritornati, dopo aver fatto il login e ottenuto il token con durata limitata, ho provato il comando "POST a new measurements" inserendo due misure, successivamente con il comando "GET ALL measurements" e "GET measurements only from a specific feature (filtered)" ho ottenuto correttamente quello che avevo inserito.

```

"docs": [
  {
    "visibility": "private",
    "stage": "final",
    "tags": [],
    "_id": "eb7374711483c49786cca38636711442b578864ca3f6262abd8795b30fad2a13",
    "startDate": "2023-11-26T11:06:10.251Z",
    "endDate": "2023-11-26T11:06:10.251Z",
    "thing": "user1",
    "feature": "IMU",
    "device": "edge-meter",
    "samples": [
      {
        "values": [
          0.89,
          0.88,
          0.87,
          0.86,
          0.85,
          0.84,
          0.83,
          0.82,
          0.81
        ]
      }
    ]
  }
]

```

```

"totalDocs": 2,
"limit": 10,
"totalPages": 1,
"page": 1,
"pagingCounter": 1,
"hasPrevPage": false,
"hasNextPage": false,
"prevPage": null,
"nextPage": null

```

**Figura 2.12.** Risposta alla GET per la prima misura

Avrei potuto usare Postman anche per verificare i dati inviati sulle piste da sci, però l'utilizzo della Dashboard Measurify e della ChartPage di Flutter è risultato più veloce e pratico per il contesto in questione.

### 3 Sperimentazione e risultati

#### 3.1 Creazione e raccolta del dataset

Dopo essersi concentrati sullo sviluppo di tutti gli strumenti necessari alle misurazioni sia lato hardware che lato software, si è iniziato a testare le variabili in gioco utilizzando la strumentazione fornita. Il primo passo è stato predisporre tutto il materiale nella maniera più efficace possibile, quindi si è caricato il codice sorgente dell'applicazione sviluppata con Flutter su un tablet Android compatibile e ci si è accertati che la connessione con il dispositivo Arduino funzionasse correttamente, in quanto, in caso di errori o inconvenienti, sulle piste da sci, sarebbe stato difficile potervi porre rimedio; è stata creata un'apposita tasca all'interno della giacca che fosse in grado di contenere solo l'Arduino Nano e soprattutto limitasse al minimo le sue oscillazioni e i suoi movimenti durante la discesa, seguendo solo l'andamento del corpo; a fronte di queste esigenze si è scelto di posizionare la scheda ad altezza petto con l'asse delle ascisse parallelo al terreno, ovviamente per tutte le misurazioni effettuate si è seguito lo stesso schema operativo: in prima battuta utilizzando la batteria a bottone, poi utilizzando il collegamento seriale a un power bank; il tablet è stato posizionato in una tasca della giacca in modo tale che fosse vicino al dispositivo e non ci fossero problemi di connessione di alcun tipo.



*Figura 3.1. Posizione del sensore durante la raccolta del dataset e invio dati alla fine di ogni misurazione*

Dopo questa fase preliminare si è potuto passare alla fase di testing e raccolta dati vera e propria. Le misurazioni sono state svolte in due comprensori differenti, Entracque e Limone Piemonte, sia per poter testare le variabili in piste diverse sia perché entrambe le stazioni sciistiche presentano caratteristiche adatte per provare le tecniche che devono essere valutate. I test sono stati svolti su 4 tags: Slalom, Spazzaneve, Posizione a Uovo e Spinta; le misurazioni registrate sono state circa 42 corrispondenti ad altrettanti discese. Lo schema di lavoro è stato il seguente: attivazione della scheda a inizio pista dopo averla posizionata correttamente; connessione della scheda al tablet; scelta del tag da analizzare e del nome della misurazione che deve essere sempre diverso onde evitare confusione nelle serie temporali da analizzare; registrazione dei dati; a fine discesa stop della registrazione e invio dei dati al server remoto. I campioni ottenuti per ogni misurazione sono risultati molteplici in quanto non sono state prese in considerazione azioni istantanee, ma tecniche sciistiche che hanno richiesto un accumulo importante di dati per ogni test. È fondamentale sottolineare l'importanza delle riprese video effettuate per ogni misurazione in quanto, per ovvie ragioni, l'efficacia dei test non è stata verificata in real-time sulle piste, ma è stata accertata solo dopo aver raccolto tutti i dati, a tal proposito il supporto video delle discese da parte di un'altra persona è stato di notevole aiuto nell'interpretazione dei dati raccolti. Occorre tener presente che la pratica sciistica e la particolare scelta di analisi effettuata, a differenza di molti altri sport è caratterizzata da una notevole quantità di variabili imprevedibili e diversificate che hanno reso il compito più arduo sebbene più completo.

| Nome Misura             | Tag        | Data       | Comprensorio | Nome Pista                | Lungh Stimata Percorsa | Dislivello | Curve/Spinte Video | Durata Video | Durata Misurazione | Note   |
|-------------------------|------------|------------|--------------|---------------------------|------------------------|------------|--------------------|--------------|--------------------|--|
| test1-pista1            | SLALOM     | 14/01/2024 | Entracque    | Canalino                  | 100 mt                 | 30 mt      | 4 totali           | 00:00:29     | 00:00:47           | primo test su pista, poche curve lente per testare la neve |
| test1-pistaAlpettaLM    | SLALOM     | 14/01/2024 | Limone       | Alpetta                   | 800 mt                 | 180 mt     | 8                  | 00:00:22     | 00:01:14           |  |
| test2-pistaAlpettaLM    | SLALOM     | 14/01/2024 | Limone       | Alpetta                   | 800 mt                 | 180 mt     | 8                  | 00:00:25     | 00:01:12           |  |
| test3_pistaAlpettaLM    | SLALOM     | 14/01/2024 | Limone       | Alpetta                   | 800 mt                 | 180 mt     | 10                 | 00:00:27     | 00:01:15           |  |
| test-pistaAlpettaneraLM | SLALOM     | 14/01/2024 | Limone       | Alpetta Olimpica          | 550 mt                 | 200 mt     | 37 totali          | 00:00:56     | 00:01:18           | Slalom stile speciale curve molto strette                  |
| test1-pistaPancaniLM    | SLALOM     | 14/01/2024 | Limone       | Pancani parte alta        | 350 mt                 | 90 mt      | 10 totali          | 00:00:23     | 00:00:44           |  |
| test2-pistaPancaniLM    | SLALOM     | 14/01/2024 | Limone       | Pancani parte bassa       | 1100 mt                | 250 mt     | 21                 | 00:00:36     | 00:01:48           |  |
| test3-pistaPancaniLM    | SLALOM     | 14/01/2024 | Limone       | Lagone Belvedere          | 900 mt                 | 200 mt     | 12                 | 00:00:29     | 00:01:35           |  |
| test-pistaBluLM         | SLALOM     | 14/01/2024 | Limone       | Pian del Sole             | 770 mt                 | 160 mt     | 24 totali          | 00:00:46     | 00:01:18           |  |
| test2-pistaBluLM        | SLALOM     | 14/01/2024 | Limone       | Pian del Sole             | 480 mt                 | 110 mt     | 15                 | 00:00:32     | 00:01:02           |  |
| test3-pista1            | SPAZZANEVE | 14/01/2024 | Entracque    | Canalino                  | 180 mt                 | 30 mt      | 4                  | 00:00:21     | 00:00:51           |  |
| test4-pistababyEN       | SPAZZANEVE | 14/01/2024 | Entracque    | Baby                      | 175 mt                 | 25 mt      | 6 totali           | 00:00:39     | 00:01:02           |  |
| test5-pista1EN          | SPAZZANEVE | 14/01/2024 | Entracque    | Canalino                  | 180 mt                 | 30 mt      | 9 totali           | 00:00:39     | 00:01:10           |  |
| test5-pistababyEN       | SPAZZANEVE | 14/01/2024 | Entracque    | Baby                      | 175 mt                 | 25 mt      | 6 totali           | 00:00:40     | 00:01:02           |  |
| test6-pista1EN          | SPAZZANEVE | 14/01/2024 | Entracque    | Canalino                  | 180 mt                 | 30 mt      | 8 totali           | 00:00:35     | 00:01:00           |  |
| test6-pistaLuciaENsp    | SPAZZANEVE | 15/01/2024 | Entracque    | Santa Lucia Tapis Roulant | 110 mt                 | 10 mt      | 4 totali           | 00:00:30     | 00:00:46           |  |
| test7-pistaLuciaEN      | SPAZZANEVE | 15/01/2024 | Entracque    | Baby                      | 175 mt                 | 25 mt      | 6 totali           | 00:00:46     | 00:01:02           |  |

**Figura 3.2.** Tabella Excel con tutte le informazioni necessarie allo studio delle misurazioni

Entrando nel dettaglio sono state effettuate 12 misurazioni di Slalom in 7 piste diverse, 11 di spazzaneve in 3 piste diverse, 11 di Spinta in 5 piste diverse e 5 di Uovo in 3 piste diverse, sono infine state effettuate 3 misurazioni che in una stessa raccolta di dati inglobassero tutte e 4 le tecniche sciistiche. Ogni misurazione è stata analizzata nel dettaglio creando un dataset completo che potesse dare più informazioni sull'ambiente di testing, sui valori raccolti e semplificasse l'analisi delle serie temporali per la fase successiva relativa al machine learning. È stato quindi creato un documento Excel che annotasse **Lunghezza Stimata Percorsa**, **Dislivello**, **Curve/Spinte Video**, **Durata Video**, **Durata Misurazione** ed eventuali note nel caso di particolarità o irregolarità.

|           |     |            |           |                 |        |       |  |          |          |  |
|-----------|-----|------------|-----------|-----------------|--------|-------|--|----------|----------|--|
| test-All  | ALL | 15/01/2024 | Entracque | Baby e Canalino | 500 mt | 70 mt |  | 00:00:56 | 00:01:27 | 20s SPAZZANEVE, 10s SPINTA, 12s SLALOM, 14s UOVO |
| test2-All | ALL | 15/01/2024 | Entracque | Baby e Canalino | 500 mt | 70 mt |  | 00:01:04 | 00:01:27 | 15s SLALOM, 9s UOVO, 24s SPAZZANEVE, 8s SPINTA   |
| test3-All | ALL | 15/01/2024 | Entracque | Baby e Canalino | 500 mt | 70 mt |  | 00:00:56 | 00:01:13 | 9s SPINTA, 12s UOVO, 15s SLALOM, 20s SPAZZANEVE  |

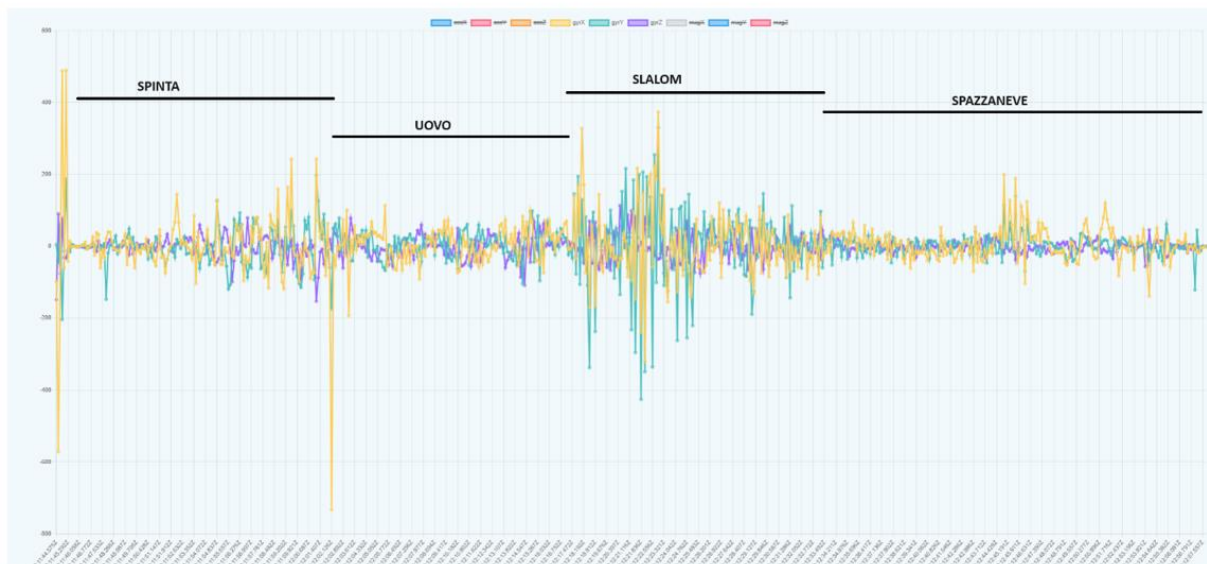
**Figura 3.3.** Informazioni relative alle 3 misurazioni con tutte e 4 le tecniche in sequenza



### 3.2 Addestramento del modello di Machine Learning

La fase successiva alla raccolta del dataset è stata l'analisi meticolosa delle serie temporali, sono stati nuovamente visionati tutti i video per poter individuare gli istanti temporali in cui iniziassero le misurazioni delle tecniche analizzate in quanto tra l'inizio di una raccolta e la sua fine erano sempre presenti vuoti temporali dovuti alla sistemazione degli strumenti all'interno della tenuta sciistica; è stato quindi scelto di rendere uniforme per ogni serie temporale l'esclusione di questi campioni iniziali e finali (ovvero i primi 10 e gli ultimi 10). Per quanto riguarda la finestra di campioni da mandare all'algoritmo di Machine Learning per addestrarlo si è optato di considerare una lunghezza pari a 80 campioni, ciò significa che ogni serie temporale viene divisa in blocchi di 80 campioni e per ogni blocco viene assegnata la corrispettiva etichetta (tecnica), quello che avanza viene scartato. Questo numero è stato scelto in quanto le misurazioni erano spesso lunghe e questo intervallo poteva raccogliere buoni risultati, ricordiamo infatti che, con riferimento alle tesi precedenti in cui sono state utilizzate azioni, questa si differenzia perché noi non andiamo ad analizzare azioni ma tecniche, una tecnica non è qualcosa di ripetuto nel tempo come calciare un pallone ripetutamente, ma è qualcosa che viene mantenuto nel tempo, quindi i criteri di selezione del dataset sono differenti. Ovviamente come in tutti i dataset alcune misurazioni presentavano delle incorrettezze e, di conseguenza, queste sono state considerate nel dettaglio e divise ad hoc.

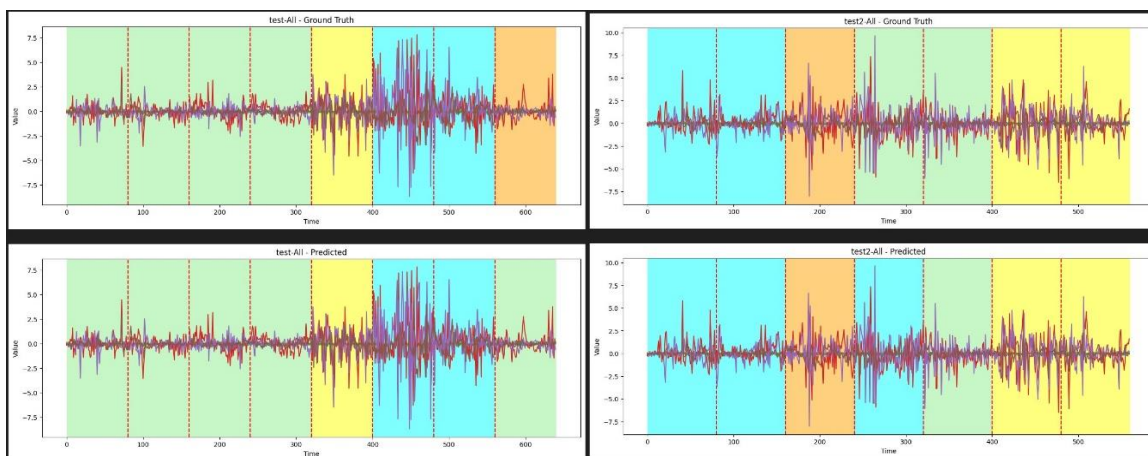
**Salterei i primi 10 campioni di ognuna e gli ultimi 10 campioni di ognuna.**



*Figura 3.4. Analisi delle serie temporali tramite Dashboard Measurify*

Il passo finale è stata la scrittura da parte del Laboratorio Elios Lab del codice necessario a creare il modello di classificazione e l'utilizzo del modello di ML denominato CNN (Convolutional Neural Network), la scelta di questo algoritmo è dovuta al fatto che lo spazio occupato risultava accettabile (232 Kbyte), considerando infatti il fine ultimo, ovvero il caricamento su Arduino per il riconoscimento real-time. Per ragioni di semplicità e per il fatto che le condizioni della neve quest'anno sono risultate molto scarse, si è scelto di inserire come misurazioni di training tutti i test effettuati sulla singola tecnica e tenere come testing set le 3 misurazioni totali contenenti tutte le 4 tecniche in sequenza, verificando, quindi, l'accuratezza del modello grazie a queste ultime; di fatto il risultato è lo stesso che si sarebbe ottenuto verificando il modello caricato su Arduino alla fine di ogni misurazione.



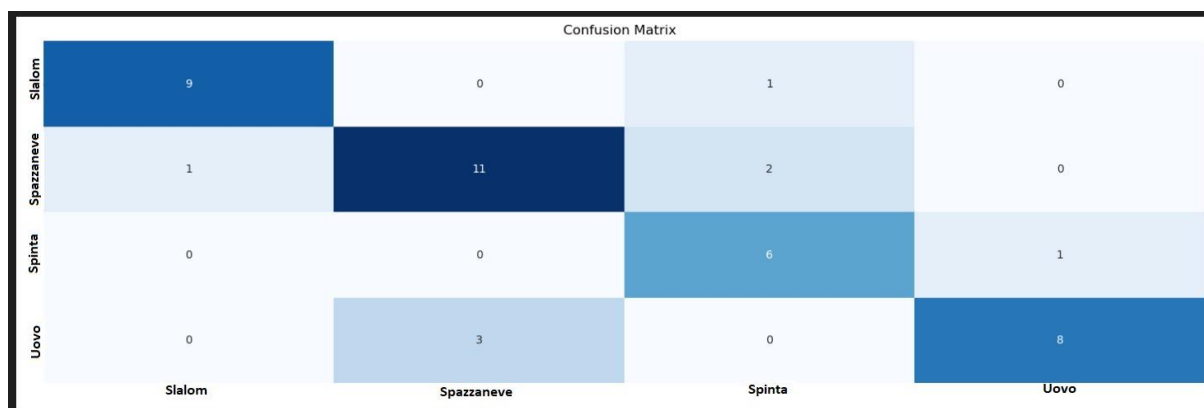


**Figura 3.5.** Serie temporali reali sopra e serie predette dal modello CNN sotto, la stessa colorazione nei blocchi da 80 campioni equivale a identificazione corretta; azzurro Slalom, verde Spazzaneve, giallo Spinta e arancione Uovo

Le prestazioni del modello hanno raggiunto un grado di accuratezza superiore all'80% con 193 serie temporali da 80 campioni utilizzate come training set e 42 serie come test set, i risultati ottenuti possono considerarsi soddisfacenti dal momento che, come detto in precedenza, la raccolta dati è stata effettuata su tante piste diverse e le variabili da considerare sono molteplici in questo tipo di misurazioni.

| models | accuracy | precision | recall   | f1_score |
|--------|----------|-----------|----------|----------|
| CNN    | 0.809524 | 0.820106  | 0.809524 | 0.810714 |

**Figura 3.6.** Risultati ottenuti dal modello CNN



**Figura 3.7.** Confusion Matrix con i valori "reali" sulle ordinate e i valori "predetti" sulle ascisse, se coincidono sulla diagonale il risultato è ottenuto

## 4 Contributo personale e considerazioni conclusive

La scelta di questo progetto è nata dal desiderio di unire una parte del percorso universitario degli ultimi tre anni a una mia grande passione sportiva. La possibilità di monitorare l'atleta durante il suo allenamento, ponendo l'attenzione sulle diverse tecniche dello sci alpino, ha colto da subito la mia attenzione, anche perché ho potuto personalmente testare e utilizzare tutti gli strumenti necessari al funzionamento del sistema in questione.

Non è stata una strada priva di difficoltà. La comprensione del sistema nella sua totalità e del collegamento tra le varie fasi, in particolare per quanto riguarda Flutter, poi risolta con l'aiuto del laboratorio Elios; l'interpretazione dei dati e la loro raccolta, le prove in pista (quest'anno la neve scarsa e le alte temperature non hanno facilitato la raccolta dei dati) con invio in tempo reale dei risultati. Comunque, il risultato è stato raggiunto ed è promettente in quanto potrebbe permettere agli atleti, in un prossimo futuro, di poter perfezionare la tecnica di discesa.

Il mio contributo personale nell'evoluzione della piattaforma è stato modificare il codice sorgente dell'applicazione caricata sul tablet, implementando la pagina di visualizzazione delle serie temporali su un grafico, con il collegamento al servizio centrale di raccolta dati; inoltre, la parte fondamentale è stata l'acquisizione dei dati, effettuando numerose misurazioni in diverse giornate, in diverse tipologie di piste e con una diversa consistenza della neve per ampliare il dataset e rendere il sistema più flessibile ai diversi aspetti che caratterizzano questo sport; infine lo studio e l'analisi dei video e delle serie temporali per poter interpretare i dati raccolti e aiutare il laboratorio all'implementazione del modello di riconoscimento delle tecniche sciistiche.

Questo progetto è considerato un punto di partenza per applicazioni future in particolare potrebbe essere utile aumentare il numero di sensori utilizzati e quindi poter gestire più sensori nelle varie fasi di raccolta dati in modo tale da analizzare più precisamente non solo le tecniche in questione, ma anche, per esempio, le corrette posizioni da assumere durante una discesa, sia a livello base per poter imparare, sia a livello agonistico per poter migliorare le proprie prestazioni in allenamento. Ovviamente più dati diversi vengono raccolti, più ampio sarà il dataset e il sistema potrà raggiungere un livello maggiore di affidabilità ed efficienza.

## 5 Riferimenti bibliografici

[1] Arduino Nano 33 BLE Sense Rev2: <https://docs.arduino.cc/hardware/nano-33-ble-sense-rev2>

[2] Arduino IDE: <https://www.arduino.cc/en/software>

[3] Flutter: <https://flutter.dev>

[4] [https://pub.dev/packages/syncfusion\\_flutter\\_charts](https://pub.dev/packages/syncfusion_flutter_charts)

[5] Measurify: <https://measurify.org>

[6] DB Measurify: <https://tracker.elioslab.net>

[7] Postman: <https://www.postman.com>

## 6 Ringraziamenti

Desidero ringraziare il mio relatore il prof. Riccardo Berta per avermi concesso l'opportunità di realizzare un progetto che fosse didattico ma anche molto stimolante e divertente. Un ringraziamento sentito va al mio correlatore il dott. Matteo Fresta per essere stato disponibile in qualsiasi orario di qualsiasi giornata e per avermi aiutato e seguito in ogni fase e difficoltà di questo lavoro.

Ringrazio infinitamente i miei genitori e mia sorella per essere stati un supporto costante in questi anni in ogni mia decisione e per essere stati parte di questa tesi aiutandomi a realizzare i video nei comprensori sciistici. Ringrazio tutti i miei amici e tutte le persone che mi hanno accompagnato in questo percorso di studi perché senza di voi questo importante traguardo della mia vita non sarebbe stato lo stesso.

Volevo ringraziare, infine, la società Entracque Neve S.R.L., con il maestro Roberto Gosso e il suo staff per avermi messo a disposizione le piste e gli impianti a uso esclusivo per i test finali.