TCSS 573: Internet of Things

# BJJ COPILOT

*Student:*
David P. Coughran Jr.
Sapper46@uw.edu

*Committee Chair:*
Prof. Eyhab Al-Masri
*Committee Member:*
Wenjun Yang

**School of Engineering and Technology**
March 14, 2025

**Abstract**

Research in computer science has devoted extensive time, resources, and energy to discover new ways to optimize human performance. This is true in both professional fields and sports. In this paper, we present the BJJ Copilot, a novel Node-RED-based web application that offers new training modalities for Brazilian Jiu Jitzu (BJJ) practitioners.

The BJJ Copilot is an interactive system that observes an athlete during training or a competitive match. BJJ Copilot captures images in real time and classifies them based on the positional relationship between the athlete and her opponent. BJJ Copilot uses this data to provide spoken instructions to the athlete during the session, just like a physical coach. BJJ copilot has ancillary features including keeping score of a match in accordance with the International Brazilian Jiu Jitzu Federation (IBJJF) rule set, monitoring athletes for out-of-bounds, and providing summary statistics to a backend database to examine which techniques require more practice.

# Contents

# 1 Introduction

Brazilian Jiu-Jitzu (BJJ) is a martial art and combat sport that focuses on ground fighting and grappling techniques. As a derivative of Japanese "jujutsu" and judo, its creators developed it to teach hand-to-hand combat to sailors in the Brazilian Navy. The core philosophy of BJJ is that a smaller, weaker person can successfully defend themselves against a larger opponent by using leverage, technique, and strategy rather than brute strength. BJJ does not incorporate strikes. Instead, it emphasizes positional control, joint locks, and chokeholds to force an opponent to submit.

Studying BJJ has an incredibly high transaction cost to learn. There are thousands of positions, maneuvers, techniques, sweeps, and key concepts. This means students can easily become overwhelmed trying to remember all these details in the face of an opponent who is actively attacking them. Anecdotally, instructors who run IBJJF schools estimate that 90 percent of beginners will quit within the first six months, and only 10 percent will progress from white to blue belt, the first of four belt promotions. This is an unfortunate outcome as BJJ has proven itself to be a healthy outlet for physical exercise, socialization, and building bonds between teammates.

One of the key drivers for athletes to quit BJJ are feelings of discomfort in a large training group. Students can feel "out of place" in an environment where they are learning a new skill, but mistakes are openly visible to everyone. Considering this, the researchers of this paper hypothesize that we can replicate 'coaching' with commercially available off-the-shelf technology. We can provide new BJJ learners with a different training venue in a private setting, without large groups, and during hours that fit their schedule. This has the potential to reduce transaction costs and keep new students in the sport.

In terms of the wider BJJ community, successfully creating a training application has immense potential to improve the sport. First, scoring and officiating matches remains an ambiguous undertaking. Different referees may score a match differently, and at high levels of competition, a single point may separate the winner from the loser. Second, unlike more traditional sports like baseball or football, in-depth analytics has not yet successfully penetrated BJJ. For example, there is no central database where someone could query "most successful submission techniques for adult competitors from 2023-2025."

Lastly, building a 'coaching' application to teach BJJ also constitutes an important step for research in computer science. Artificial intelligence (AI) algorithms and large language models (LLM) have made immense progress in just the last five years. We have incorporated them in carrying out more and more sophisticated tasks from writing papers to conducting in-depth statistical research. BJJ presents a challenge to this technology because so much of the sport is delicately nuanced, and the details that lead to success in a match are difficult to capture digitally. Using technology to better understand BJJ will inadvertently unlock better methodologies in fields such as medicine, life sciences, or other industries examining human movement patterns.

# 2 Related Work

At its core, the goal of BJJ Copilot is to optimize human behavior and performance. There are countless studies, many of which began here at the University of Washington Tacoma, that address this topic. In *NudgeRank: Digital Algorithm Nudging for Personalized Health*, the authors present an innovative system designed to foster healthier lifestyles through a relatively unobtrusive electronic reminder, or "nudge." Their system leverages a Graph Neural Network enhanced with a Knowledge Graph to deliver personalized health nudges to more than 1.1 million users daily. This system was deployed in Singapore through the Healthy 365 mobile app, a health initiative run by the Health Promotion Board. The study highlights the effectiveness of the system in improving physical activity metrics, like daily steps and moderate to vigorous physical activity, by using algorithmic nudges tailored to an individual's health data.[1]

Research in this field also shows how there are challenges regarding how to best build the machine learning model. University of Washington researchers Larry Preuett and Dr. Ankur M. Teredesai published a paper titled *Reproducible Evaluation Framework for Sepsis Treatment Strategies Learned via Offline Reinforcement Learning*. They present a framework designed to evaluate sepsis treatment strategies developed using offline reinforcement learning (RL). Sepsis, a life-threatening condition caused by an extreme immune response to infection, requires timely treatment to prevent septic shock and death. Treatments often include intravenous fluids and vasopressors, but there is no consensus on the best treatment strategy, making the optimization of treatment protocols extremely difficult. The results show that none of the RL-based strategies outperform human physicians.[3]

While these papers study human behavior and decision-making more generally, researchers at the University of Tsukuba applied statistical inference techniques to an actual sport- judo. Extracting data from thousands of videos depicting high-level international Judo competitions, they developed the Extraction for Successful Movement (XSM) method to identify key positional factors that lead to successful throws. This research is significant because it was the first to consider “oppositional” forces. Stated differently, combat sports are inherently complicated because an opposing athlete can influence your position, posture, and movement. Any successful movement pattern must account for these forces.[2]

# 3 Proposed Internet of Things Solution

These previously cited papers offer suggestions on how to optimize our training application. First, like Nudge-Rank, BJJ Copilot should offer personalized technique suggestions suited to the age, gender, physical ability, and skill level of our mentored athlete. Second, coaching provided by our application must be flexible. In the chaos of a combat-sports match, very few engagements follow a strict step-by-step roadmap. Our goal should be to keep the athlete “within-the-rails” of a certain strategy. Lastly, like the XSM method, we must consider positions in their absolute totality. This means factoring for the relative position of our athlete’s opponent as well.

In terms of technical implementation, Node-RED provides a low-code, easily testable application solution to quickly integrate all the necessary requirements. The system also provides a ready-made graphical user interface (GUI) that can incorporate modifications with minimal input. With Node-RED, we identify the necessary sensors and actuators, and then deploy them against our imagined workflow. BJJ Copilot uses a total of three sensors (camera, thermometer, range finder) and one actuator (speaker). Figure (1) is a high level overview of the BJJ Copilot workflow.

On the hardware side, we use a Raspberry Pi to integrate all the necessary components. The Raspberry Pi functions as a standalone computer and runs Linux as our operating system, The Raspberry Pi also supports the various programming languages (Python, Java, HMTL, et al) the BJJ Copilot requires in its scripts. We use a GrovePi to quickly and efficiently connect sensors and actuators to the General Purpose Input/Output (GPIO) Pins of

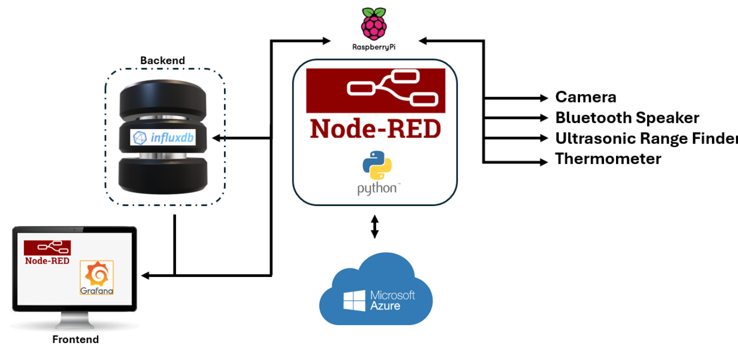


Figure 1: BJJ Copilot Overview

the Rasperry Pi. Acting in concert, the Raspberry Pi and GrovePi offer full connectivity to Wi-Fi, Bluetooth, USB Ports, HDMI, and Ethernet.

User interactions with the BJJ Copilot are minimal- the athlete simply uses the GUI to identify when she has begun training and when training has ended. When the flow enters an active training state (i.e. the user has clicked on "Begin Training"), the BJJ Copilot enters a repeating cycle. First, the Raspberry Pi uses the camera to capture a training image. The image is a 1.6 MB jpeg file that is 1920x1080 pixels in resolution. The Raspberry Pi stores the image locally in its storage, with future images taking the place of this file to reduce the overall file size of the application.

After the training image is taken, BJJ Copilot uses an API to connect, in the Cloud, with Microsoft Azure's Custom Vision Model to classify the image. The classification model itself is a machine learning model trained on eleven total positions, balanced between offense and defense, which represent the most prevalent positions in BJJ. Azure requires fifty pictures per position, so there are exactly 550 images loaded into the model. We note that these images focus on a single athlete who volunteered to be the training subject for this project and a second athlete who volunteered to be her training opponent. This means the BJJ Copilot is currently only configured to coach one athlete. Finally, the Azure model returns both a position classification and its projected confidence for that classification as a json object. BJJ Copilot sets a confidence threshold of 50 percent. Anything beneath that threshold causes BJJ Copilot to ignore the image and move on to another capture.

With a successfully classified image, the next step for BJJ Copilot is to

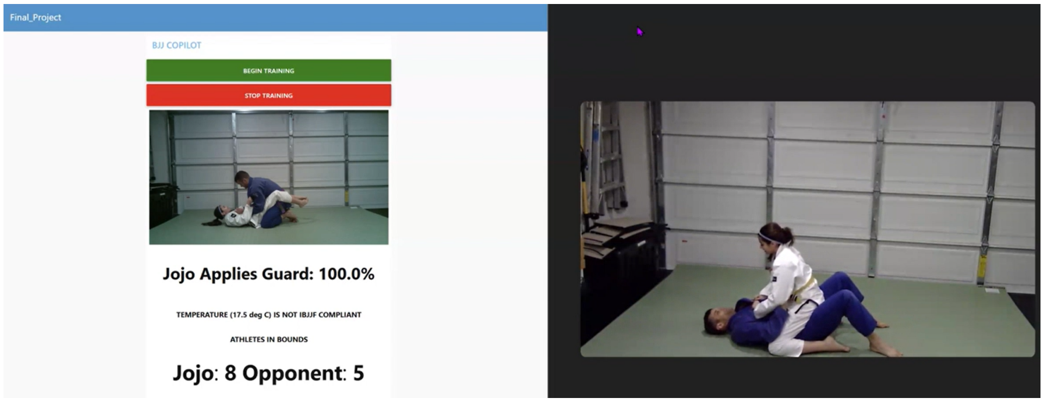


Figure 2: Points Adjudiction for BJJ Copilot

adjudicate points based on the relative position of the two athletes. We incorporate the IBJJF rule-set in the BJJ Copilot, though we recognize there are other federations using variants. For research purposes, this is only important to note the flow can account for all ways an athlete can score points: achieving a position, holding a position, or achieving a certain offensive position after being on defense. BJJ Copilot keeps the previous position classification cached so it can calculate all the combinations and permutations for points. For example, in Figure (2), we have juxtaposed two images. One is from the BJJ Copilot GUI and the other is from a standalone camera that recorded the match. The image in the GUI is approximately 2 seconds behind the image to the right. However, BJJ Copilot has both positions in memory. Therefore, it awards 3 points to Jojo (athlete in white) for achieving Mount (the position in the right image), and another 2 points for achieving this position from Guard (the position in the left image).

The next step in the flow is to provide the correct auditory prompt to the athlete so they know what to do in this position. Currently, we have configured the prompts based on the skill-set of a 13 year-old woman athlete with approximately 3 years of BJJ experience. The prompts are short and concise, being an average of four seconds in length. Additionally, the system understands Jojo's overall strategy for the match or training session. Her intent is to (1) get her opponent to the ground to neutralize the size advantage, (2) get a top position, and then (3) apply an armbar. This strategy ends with her in a position where she can leverage her legs against her opponent's weaker arms. We use a Bluetooth speaker, connected and specifically configured for the Raspberry Pi, to play .wav files. BJJ Copilot uses Aplay (Linux) to play the prompt.

BJJ Copilot's last step in the training cycle is to provide the athlete time to respond to the prompt. This gives the system time to update the GUI, reallocate memory, and clear caches for the next cycle. However, it is important to note that BJJ Copilot recognizes that a match or training session may not follow a strict script. The system's auditory prompts will correct for when positions fall outside of the expected outcomes of the strategy.

The athlete has two options to conclude the training session. First, she may click the "STOP TRAINING" button from the GUI. If she does this, the BJJ Copilot believes she was simply practicing moving through the various positions. It will upload the count to an edge-based backend Influx database that tracks the number of times the athlete held a position. This allows the athlete to query positions over a length of time in a Grafana backend dashboard. For analysis, the dashboard gives the athlete gives the balance between offense and defense, as well as which submission techniques have netted the most victories. However, if the training session ends in a submission (i.e. a jointlock or choke hold), the BJJ copilot takes an additional step. It assumes the training session was a sparring match, meaning it will also upload points and the submission type into the backend. The following image in Figure (3) demonstrates the visualization in Grafana.

The following image in Figure (4) gives an overview of the flow in Node-RED. We have added labels to give a synopsis of the function that each group of nodes performs. Of note, there are two ancillary functions in the flow which provide a support role to the main program. First, the thermometer checks the temperature with IBJJF compliance for sanctioned competitions. Second, the a laser range finder detects when athletes are approaching the edge of the mat and directs them to return to the center for safety.

# 4 Decisions and Observations

BJJ Copilot performs all the tasks and functions we identified during the initial problem framing. The application successfully replicates coaching in real-time, providing performance cues in keeping with a previously established strategy. BJJ Copilot also aggregates data for backend analysis. This includes the time of training sessions, points scored vs the opponent, total number of positions held, offensive vs defensive balance, and a breakdown of

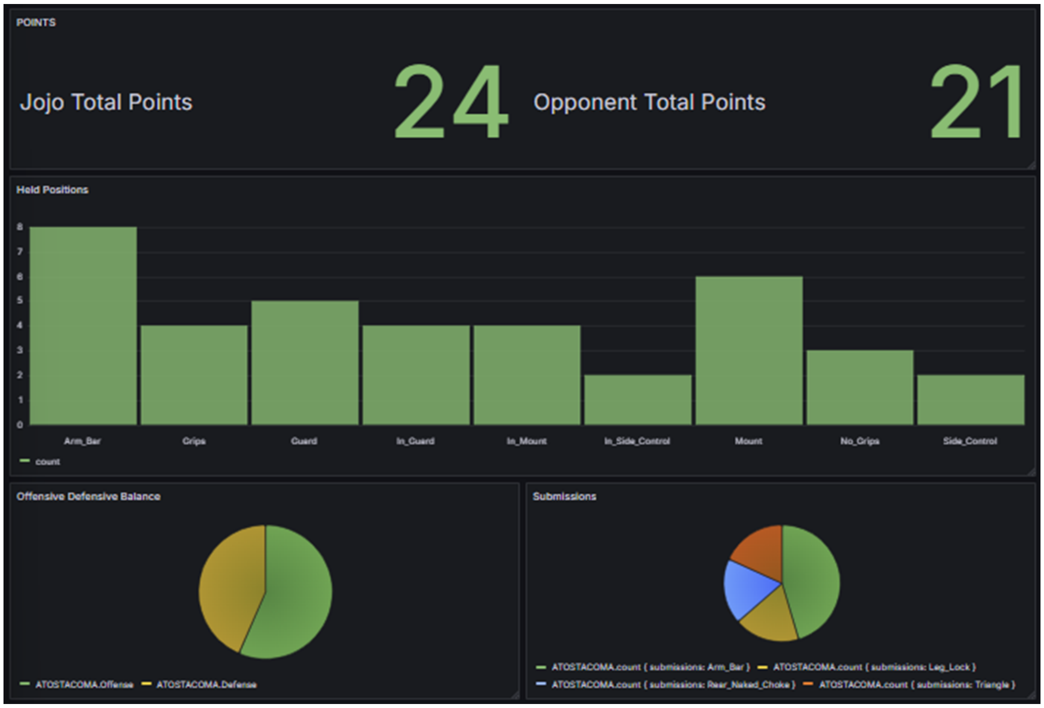


Figure 3: Grafana Visualization

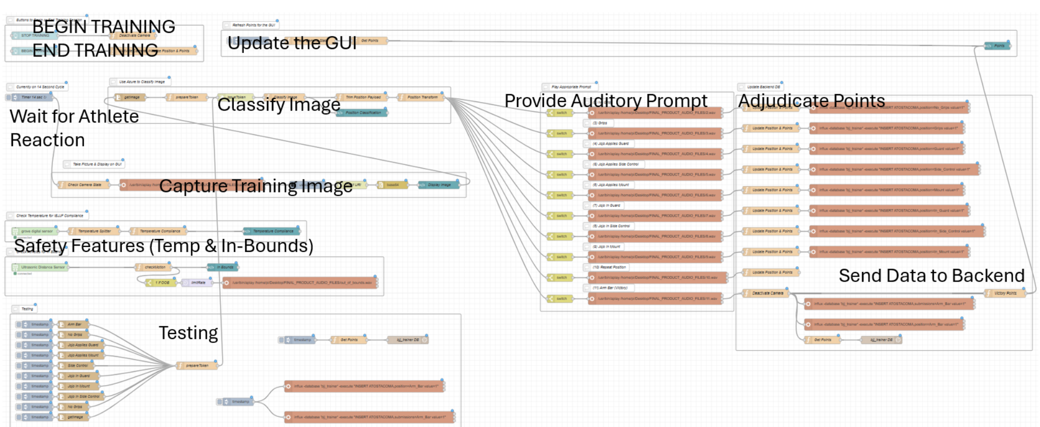


Figure 4: Node-RED Flow for BJJ Copilot

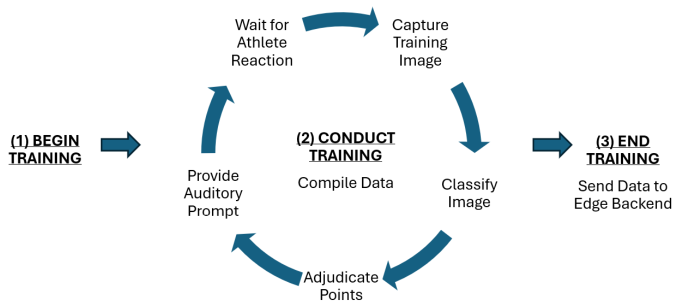


Figure 5: BJJ Copilot Cycle

successful match-winning submissions. BJJ Copilot successfully adjudicates points based on the IBJJF rule set. In short, this system can provide a realistic training environment for a BJJ practitioner when they are away from their school or coach.

# 5 Discussion

The key technical challenges in this endeavor were managing the various workflows and balancing hardware limitations against BJJ Copilot's performance. One unexpected outcome we encountered was "slower can be better."

In Figure (5), you can see we made a conscientious decision to place all these steps in a single cycle. BJJ Copilot takes one picture, classifies it, adjudicates points, gives the auditory prompt, and then waits for the athlete's reactions. Initially, we did not believe that placing all these steps into a singular cycle would provide the best outcome. Our first approach was much different. We programmed the flow so that the Raspberry Pi took pictures on a self-contained loop that maximized the output of the camera. This was one picture every 1.8 seconds, which was as fast as the system to go. In this way, we could have a directory that always has the most up-to-date training image, irrespective of any other flows. Unfortunately, this drew computational speed away from the other functions and led to a breakdown in message traffic that quickly queued up behind other nodes. We also tried to deploy the Azure model locally. Our thought was that doing so would also allow classifications to return at a much faster rate. Unfortunately, this too led to degraded system performance. Verbal prompts began to queue up, and BJJ Copilot gave instructions that were not aligned to the athlete's current position.

In the end, we found we could optimize BJJ copilot by keeping all the Conduct Training tasks from Figure (1) in the same cycle. Most importantly, we realized we had to give the athlete time to carry out the prompt or at least create a change in state. In this sense, slowing the system down led to better outcomes in terms of latency and consistency of instruction. Trials and informal experimentation revealed that putting the Conduct Training tasks on a 14-second cycle created the best fusion between system, athlete, and the GUI. In the end, this could be key as it opens computational resources for future improvements to the system.

# 6 Conclusion

BJJ Copilot is a successful step in using commercially available technology to optimize human performance. This research is significant because BJJ is a nuanced martial art; it contains many subtleties that are difficult for digital sensors to penetrate. Weight, balance, relative position, and strategy are difficult to codify and wrap into an algorithm. However, the methods in which BJJ Copilot overcomes these challenges offer a road map for endeavors in other fields.

For future work, we recommend taking the "coach" we have virtualized in BJJ Copilot, and making him better. Currently, BJJ copilot will offer a prompt for every position, regardless of whether the athlete really needs it. The best coaches understand when the situation needs time to develop, or if the athlete is innovating away from the previously established strategy. This is in keeping with the idea that the next evolution of BJJ Copilot should be more intuitive. Instead of giving prompts on a set interval, it should only offer them when needed. For example, the application should allow athletes to recall the appropriate moves on their own, and only give a prompt if they get "stuck" in a certain position.

# 7 Education Statement

Dave Coughran is a Master of Computer Science and Systems Candidate at the University of Washington- Tacoma. He holds a Masters of Public Policy from the Harvard Kennedy School of Government and a Bachelors in Mechanical Engineering from the U.S. Military Academy at West Point. Dave is a twenty-three year public service veteran with experience in engineering, special operations, software design, econometrics, and diplomacy.

# References


[1] Chiam, J., Lim, A., and Teredesai, A. Nudgerank: Digital algorithmic nudging for personalized health. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (2024), pp. 4873–4884.

[2] Kato, S., and Yamagiwa, S. Statistical extraction method for revealing key factors from posture before initiating successful throwing technique in judo. *Sensors 21*, 17 (2021), 5884.

[3] Preuett, L. D. Learning personalized health recommendations via offline reinforcement learning. In *Proceedings of the 18th ACM Conference on Recommender Systems* (2024), pp. 1355–1357.