gTTS (Google Text To Speech) library and play them as the predictions are ready to be outputted using Python's playsound library. Figure 9 contains the above described system in a diagram.

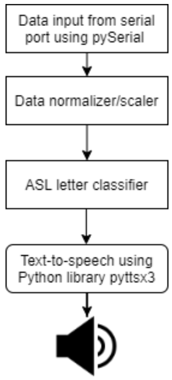


Figure 9: Software subsystem

6 TEST & VALIDATION

6.1 Fake Data Generation

In order to make progress on our project while we waited on the parts for our glove to arrive, we decided to generate fake data to test the machine learning models we are considering. We began by first going through each of our gestures and determining a range of values for each of the 14 measurements our glove will take. For the flex sensors on each of the fingers, the data that is being sent to our machine learning models will be in the form of angles (in degrees). For the values outputted by the accelerometer and gyroscope components of the IMU, we were not able to find good documentation or sample data that could guide us in modelling our fake data, so we decided to take random values from the range that the product specification indicates will outputted. For the magnetometer component of the IMU, we chose the ranges assuming the user will always be facing one direction with respect to the Earth's poles. In reality will not be true, but it will at least provide our preliminary tests with some IMU data to work with.

After these ranges were determined, for each generated data point, we obtained a random value from a normal distribution within the range that we had set, with a small probability of generating an outlier value. Outlier values are simply random numbers from a uniform distribution between 0 and 100. We generated a total of 100 data points for each of the 26 letters for the training data set and 50 data points for each of the 26 letters for the testing data set.

6.2 Preliminary Tests on Machine Learning Models

We did tests on the fake data we generated as well as real data collected from one person on the prototype of our glove after it was built. See figure 10 These results should be viewed sceptically since our fake data generation left out important information for the data coming out of the IMU. Additinonally,

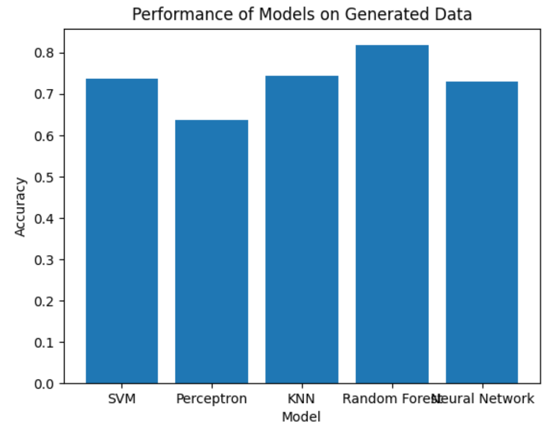


Figure 10: Performance of Models on Generated Data

6.3 Performance on Design Requirements

See table 1 for a comparison of the requirements we set for our project and what we were able to accomplish.

For the accuracy measurement, we tested this on both collected data and real-time data: collected data refers to the data that we gathered in order to train the machine learning model (segmented into a training and testing dataset) and real-time data refers to the information our glove measures while in use. We were able to get accuracies of 98.9% and 75.86% for the collected and real-time data, respectively. In theory, the two should not be that different, but in reality, we find that there is noticeably more variability in the data that is read in from the glove in real time compared to what we had collected. We believe this is because when we were collecting data, we made a conscious effort to make the gesture a certain way and collecting several data points in that one position, so the model is overfitting to those specific gestures. Efforts were made to tune our hyperparameters to avoid overfitting and was successful in improving our real-time accuracy slightly. See figure 11 for the confusion matrices of collected testing and real-time testing data.

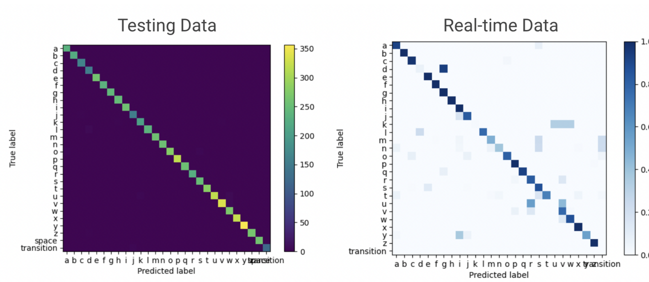


Figure 11: Confusion matrices of Collected Testing Data and Real-Time Data

The latency was measured by setting a software timer starting when the data is received by the computer program and ending when the classification is made. The average for this value was 63.89 ms/prediction, which is well below or original requirement of 100 ms.

The frequency of output was 0.862 s/gesture, which is higher than our goal of 0.5 s/gesture. However, this is a value that can be adjusted– this value was obtained under the requirement that our system makes eight of the same consecutive predictions before outputting. Therefore, we can lower that number in order to allow faster output rates.

Lastly, our glove weighs in at 79 g, which is significantly less than or initial goal of 200 g.

6.4 Tuning the System

Similar to what we found in testing our models on our generated data, the Random Forest Classifier performed the best for our use case. See figure 12 for a comparison of model performance on our final iteration of the glove.

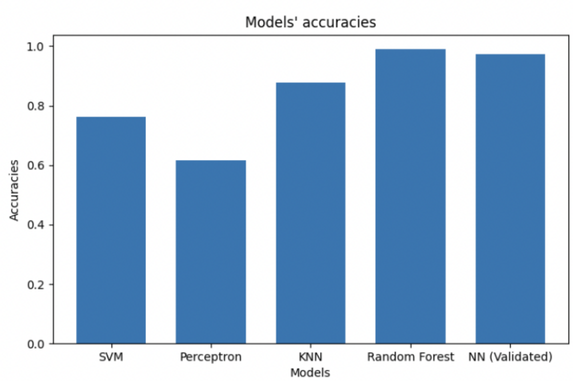


Figure 12: Model Comparison on Final Iteration of Glove

Figure 13 shows the hyperparameters we tuned for the Random Forest Classifier. Restricting the minimum for the number of samples in a leaf node or minimum number of samples to split an internal node can potentially prevent the model from overfitting to the training data but increasing it significantly can lead to significant underfitting, hence the consistent downward trend in accuracy. Setting the number of trees has more of an interesting effect on accuracy. At first increasing the number of trees in the classifier significantly improves the accuracy of prediction, but that accuracy plateaus at around 100 trees, which is the number we ultimately went with.

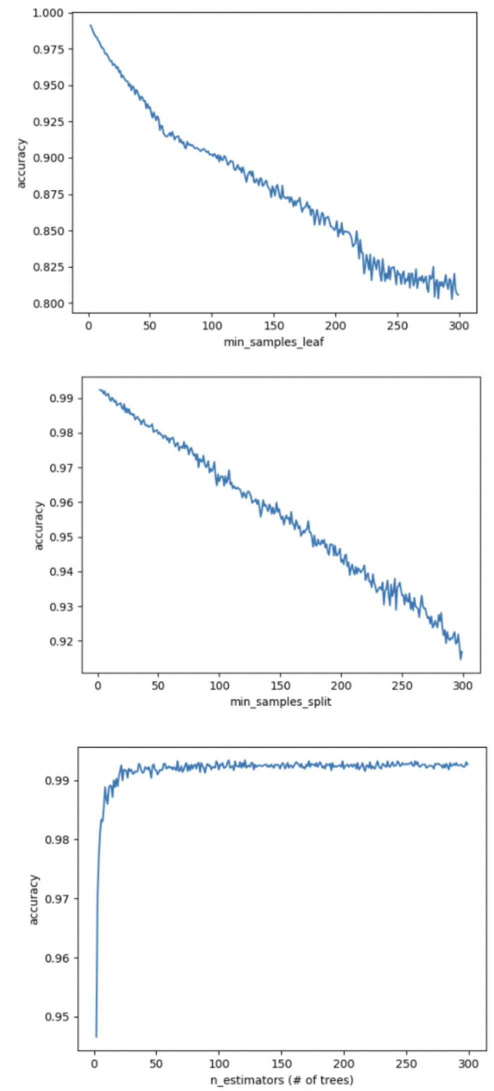


Figure 13: Performance as a Result of Tuning Hyperparameters

We also tested our glove's performance while changing the speed of the system as well as the user's signing rate. To perform the test changing the speed of the system, we kept the user signing at one gesture per second and adjusted the number of consecutive classifications required for an output. This performed the best at requiring 8 consecutive predictions. A higher number resulted in the model missing letters from time to time because it would not receive the required number of consecutive letters. On the other hand, a lower value would result in extraneous intermediate outputs since it would make predictions more often than the user was signing.

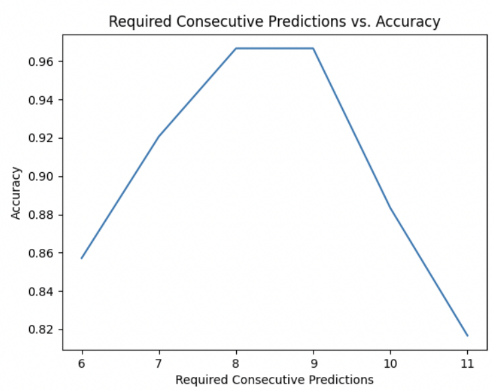
For the user signing rate test, we kept the system requiring 8 consecutive predictions while changing the speed at which the user is signing. As expected, this performs the best at one gesture per second since 8 consecutive classifications was determined to be the best at a signing rate of one gesture per second.

See figure 14 for a plot of the Speed vs. Accuracy tests

Table 1: Design Requirements

Requirement	Specification	Performance
Accuracy	90%	98.9% / 75.86%
Latency	<100 ms/prediction	63.89 ms/prediction
Frequency	0.5 s/gesture	0.862 s/gesture
Craftsmanship	<200 g	79 g

(using averages from multiple trials). As demonstrated with these tests, the number of consecutive classifications we require should be set according to how fast we expect our target users to sign. While the average ASL user signs at around 0.5 s/sign, we decided to keep this parameter of our system at 8 consecutive classifications since the glove makes movement slower and is quite fragile in its current stage.



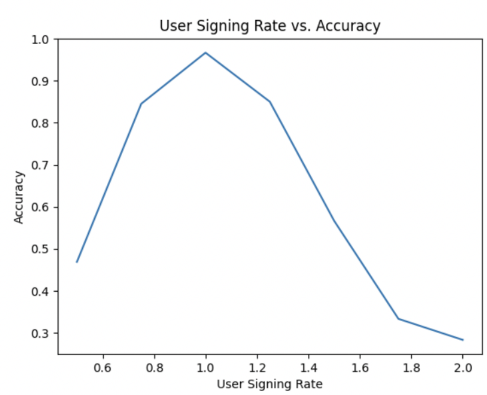


Figure 14: Speed vs. Accuracy Plots (Averages across 3 Trials)

7 PROJECT MANAGEMENT

7.1 Schedule

Our schedule can be viewed in Figure 7. In general, we plan to spend the first half of the semester building the glove, and the second half of the semester training, testing and refining the ML model.

7.2 Team Member Responsibilities

Sophia is leading the hardware and construction part of the project. She is responsible for building the physical glove, making repairs and researching how to improve the design to create a robust, yet comfortable product. Her secondary responsibility includes gathering subjects for training/testing the ML models and helping conduct and compare experiments of competing design choices.

Rachel is primarily responsible for data collection, serial streaming of data, and normalizing the data as to reduce the noise from the data read in from the sensors. Her secondary responsibility also includes gathering subjects to collect training and testing data as well as analyzing the performances and costs/benefits of each ML model we are considering.

Stephanie is in charge of training and tuning the machine learning models. She also has a secondary responsibility of gathering subjects to collect training/testing data and helping determine how much data we want to read in and at what rate to get the best results (in terms of accuracy) without compromising the latency of our product.

7.3 Budget

The budget for our project is $600. We only used $194.67 of it. A breakdown of our components purchased can be found in Table 2.

7.4 Risk Management

The primary risks to the success of our project involved the reliability of our sensors. The sensors are how we detect the motion and pose of the hand. If those are not accurate, then the rest of the pipeline will not be accurate.

To mitigate the risk of faulty sensors, we bought multiple extra flex sensors, IMUs and Arduino Nanos. If we find any of these components to be faulty or if they get damaged during construction or testing, they can easily be replaced. Throughout the process we found that the flex sensors would frequently come loose or stop giving variable readings, so having extra flex sensors we could replace the broken ones was helpful in creating a working final product.

Further, after looking at the flex sensor specification sheets more in depth, the manufacturers recommend reading the voltage after passing it through an op-amp which acts as an impedance buffer. If our readings had proven too unstable, this is a path we were prepared to take. However,

Table 2: Bill of materials

Description	Model #	Manufacturer	Quantity	Cost @	Total
Microcontroller	Arduino Nano	Arduino	1	\$14.98	\$14.98
Flex Sensors	182	Adafruit	8	\$12.95	\$103.6
Pack of 12 Gloves	n/a	Donfri	1	\$11.99	\$11.99
A-Male to Mini-B USB Cable	n/a	AmazonBasics	1	\$7.01	\$7.01
9 DoF IMU	ICM20948	Adafruit	2	\$14.95	\$29.90
Embroidery Thread	n/a	Pllieay	1	\$6.99	\$6.99
Arduino Nano BLE	Arduino Nano BLE	Arduino	1	\$20.20	\$20.20
					\$194.67

Note: Shipping costs are not included in the calculations.

we found that the flex sensors gave pretty consistent readings without the presents of an op-amp.

Lastly, we are normalizing the data the computer receives and filtering out any outliers. Normalizing the data ensures that our data is always within the same range, correcting for some minor variation in sensor data.

We had also initially planned on ordering a PCB, but settled on using a protoboard to complete our circuitry instead. While it would make the circuitry cleaner, we could risk the PCB taking too long to get manufactured and delivered. Additionally, any changes to our circuitry would mean we would have to order another PCB to support those changes. Using a protoboard instead of a PCB mitigates the risk of delayed delivery since we can solder one together virtually whenever we need.

8 ETHICAL ISSUES

Sign language interpreting gloves have been built in the past without input from ASL signers. It's important that ASL gloves accurately represent the language. For example, facial expressions are an important part of ASL, however that aspect of the language is inherently disregarded with a solution like a sign recognizing glove. We are well aware of this fault in our design and plan to consult actual ASL users during our development to collect and if time permits, integrate their feedback.

Another ethical concern is mis-translation. If an ASL speaker uses our glove and the messages are not translated correctly, or if a malicious party interfers to translate phrases incorrectly, others could perceive the ASL speaker in a negative light. To mitigate this risk, before production, the product should be tested with ASL speakers to gather their feedback.

Another ethical issue we faced when first figuring out our implementation details was privacy. As previously mentioned, computer vision is a viable option for detecting and identifying ASL gestures. However, using a camera as part of our system could have privacy implications since anything that is captured in the scene could be a privacy violation. Since we decided to take a sensor approach instead, this issue is no longer relevant.

However, privacy could still be an issue in a different way. The volume of the audio output could be adjusted so that information directed at one person could be heard by an entire group of people. This issue could be addressed by adding a feature to allow the user to vary the output volume depending on how many people they are addressing and how near they are to their conversational partners.

Lastly, this product could also create some class concerns if it is something only available to a more wealthy subset of the population. Additionally, this device could reduce people's will to learn ASL. If this product goes to market, it's important it is also advertised as an educational tool so that it is accessible to all classes and encourages people to get excited about learning ASL.

9 RELATED WORK

Gesture-recognizing gloves are not uncommon nowadays. During our research into implementation details, we have found quite a few similar projects. Here are a few that we feel closely resonated with our project.

Sign Language Glove[4] This project was built by two students from Cornell University. Similar to our objective, they wanted the glove to recognize ASL letters. Their finished product uses sensors to collect data, mainly flex sensors, an IMU, and contact sensors. They set up an off-glove circuitry on a breadboard to communicate between the sensors and the computer, which will train and test the data. The classifier models were then trained on data specific to each user.

The flaws within this design are quite obvious. The first and foremost being the glove is trained for specific users, and new users will have to train a set of data on it before they can use it. This greatly degrades user experience and can be confusing to new users if this were a commercial product, as users would likely expect a pre-configured product. The second drawback is the off-glove circuitry which adds complexity to the physical component and is inconvenient to set up and carry around. This also may impede movement while gesturing.

We are attempting to improve upon this project by finding a larger set of data to train on to generalize to a broad audience. We are also constructing a printed circuit board to aggregate our hardware components to help reduce the