(Sample of Approval Sheet)

APPROVAL SHEET

Title of Thesis: Detecting Makeup Activities using Internet-of-Things

Name of Candidate: Fatimah Alqurmti
Master of Science, 2019

Thesis and Abstract Approved: (……………………….)
Dr. Nirmalya Roy
Associate Professor
Department of Information Systems

Date Approved: ________________

ABSTRACT

Title of Document: DETECTING MAKEUP ACTIVITIES USING INTERNET-OF-THINGS

Fatimah Alqurmti, Master of Science in Information Systems, 2019

Directed By: Associate Professor, Nirmalya Roy
Department of Information Systems

This thesis focuses on identifying human activities for rendering make-up activities using sensors' data and a supervised machine learning approaches. We considered five make-up activities in our work, such as, applying cream, lipsticks, blusher, eyeshadow, and mascara. We collected the data from ten participants using two smart-watch built-in sensors, accelerometer and gyroscope. We preprocessed the data and trained with different predictive machine learning models and we evaluated make-up activity prediction built on using Naïve Bayes, Simple Logistic, k-nearest neighbors', and the random forest algorithms. We investigated the models' performance on three different datasets that differ by the environment they were collected in. The first dataset was collected from the participants using a controlled environment. In this staged setting, we provided the participants specific instructions on how to perform the five make-up activities. The second dataset was collected from the participants in an uncontrolled environment. We did not inform the participants with any prior instructions on how to perform the five activities and therefore, naturally they performed the make-up activities in their own way. Third, we synthetically generated a dataset by combining the existing datasets from the participants who were under both controlled and uncontrolled environments. Our results showed a 92.7 % accuracy for the controlled environment case given by the

Gradient Boosting classifier and an 89.20 % accuracy for the uncontrolled environment case shown by the Random Forest classifier. Finally, Random Forest classifier registered the highest accuracy 92%, for the hybrid case where both the datasets from controlled and the uncontrolled environments were combined. We believe that this early work on recognizing and discovering a multitude of make-up activities has potential application in assessing and training the performance of various stakeholders in the future work of fashion industry.

DETECTING MAKEUP ACTIVITIES USING INTERNET-OF-THINGS

By

Fatimah Alqurmti

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, Baltimore County, in partial fulfillment
of the requirements for the degree of
Master of Science
2019

Dedication

This thesis is dedicated to my mother, father, husband and my three little kids for all the time they have been waiting for me, and for all their continuous love and encouragement.

Acknowledgements

I would like to thank my advisor Dr. Nirmalya Roy, for his continuous guidance and direction. He was always available for my questions and pointed me toward avenues that helped me progress. Also, I would like to thank Dr. Zhiyuan Chen, and Dr. Ravi Kuber for their time serving as my committee members and for their valuable feedback.

I also would like to thank all the students at the Mobile Pervasive and Sensor Computing Lab for their willingness to help and their helpful insights.

Finally, I want to thank my family and my friends who always encouraged me. Without their warm support I could not have completed this work.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

The Internet-of-things is an emerging technology that has a high potential to impact various domains. The connectivity between different objects through the internet, along with the ability to remotely collect and exchange data, is what we refer to when using the term internet of things (IoT). This interaction between physical objects, such as devices, sensors, and cyber objects allowed for IoT-based applications across multiple fields. The fields IoT has impacted range from business and healthcare to security. [17] [18]. As the IoT brought potential improvement to different domains it simultaneously opened the door for more research directions.

Human Activity Recognition (HAR) is a research area that thrived with the development of IoT wearables. A variety of wearables equipped with different sensors are on the market that collect humongous data. This data, in turn, is an opportunity to conduct more research on HAR through the use of the wearable sensors and machine learning. HAR is a core building block for many interesting applications for improving people's lives. The implementations include daily life monitoring, elderly and youth care, localization, assistive tools, and virtual trainers. Activity recognition is done by taking the raw sensor's reading as input and processing it to predict human movement [19]. Using wearable's sensors, movement data can be collected from any part of the human body and then analyzed to predict body motion. This has the potential applications in a variety of systems. One of the fields that activity recognition played a basic role, is in the development of automated assistive tools that enable self-training.

Automated assistive tools, commonly referred to as virtual trainers, can be used as a stand-alone trainer or can be used to support real-person training by providing valuable feedbacks and assessments for the trainees' performance. These tools enable self-training in domains such as sports, art, and dance. Fashion is another promising domain to reap the benefits of assistive technology. IoT and Machine learning applications play a significant role in the development of automated assistive tools which enable self-training for fashion workforces. A variety of data science techniques could be developed to automate different fashion related activities such as applying makeup, haircuts, administering hair-coloring, nail polishing, etc. The make-up sector is a high-demand industry that can thrive by integrating automated support. The make-up workforce primarily concerns having beauty materials coupled with the ability to professionally apply them. This is considered a practiced skill that relies heavily on training.

Gaining new skills can be highly affected by the type and training-time. Incorporating automation to training can make training process more affordable and accessible for many people. This thesis aims to bring automation to the field of makeup. There has been no such previous effort in this direction. The proposed system would pave the way for an automated assistive tool for applying make-up activities. This thesis focuses on identifying human activities for rendering make-up activities using sensors' data along with a supervised machine learning approach. Five make-up activities were selected including: applying cream, lipsticks, blusher, eyeshadow, and mascara. The goal of developing this work to be a preliminary step toward further work in using machine learning to facilitate automated support for the make-up training industry.

1.1 Problem Significance:

the development of an assistive tool HAR, can be leveraged to a wide range of areas wherever training is required. The automated trainer system can help coaches in a variety of training types through receiving feedback from their trainees' performance. It can also enable automated training at an affordable cost where people reference it on an as-needed basis.

In this thesis research, we tried to leverage the applications of HAR to reach the fashion industry, which has recently received more attention from artificial intelligence scientists. The fashion industry is a high demand industry which has always been affected by global technological trends. Artificial intelligence can play a significant role in changing the way the fashion industry currently operates. Applying Artificial intelligence to the fashion industry allows for personal marketing, sale prediction. It can also allow for automated customization where the customer can explain his desire for a machine and prepare it within a short amount of time.

In addition, assistive and automated tools can line up the goals of the fashion industry. The make-up market is one of the most furnished fashion markets in product retailing and training demands. To address that, we worked on this thesis using machine learning techniques to enable automated training assistance for make-up application. We propose a system that uses machine learning algorithms to detect make-up applying activities. Such an effort provides the basis for a more advanced make-up applying automation tool

Chapter 2: Related Work

In this section, we will provide an overview of the work that has been done in human-activity detection for various domains such as adult daily living, health and diet, health security and monitoring, and automated assistive tools.

There is a substantial amount of work done in detecting daily living activities using various techniques of machine learning to support elderly living [1]. Detecting daily physical activities allows for further research in supporting elder people's healthy lifestyle. It supports elder individuals' independent living by designing health smart homes that facilitate health security and monitoring.

Fleury et. al. [1] conducted research on detecting daily living activities for the purpose of detecting early signs of independence loss in elderly people. The experiment used a health-smart-home environment's sensors to collect data for seven activities (hygiene, toilets, eating, resting, sleeping, communication, and dressing/undressing). The model gets trained on thirteen young and healthy participants using a support vector machine classifier and then tests the model on real data. The work showed that detecting abnormal activities could play a significant role in detecting the early signs of independence loss in elderly people.

Other research work has been done by Joha et. al. [2] in the classification of daily living activities such as walking, running, and cycling. The study aimed to identify the best sensors, algorithms, and approaches that could be used for activity recognition. A large number of sensors datasets were collected and used to build three models from the classifiers- custom decision tree, automatically generated decision tree, and artificial neural network. Various validation techniques such as leave one subject out, cross validation, and 10-fold cross validation were applied to justify the true accuracy. The overall accuracy showed that the automatically generated decision tree had the highest accuracy of 86%.

Health and security monitoring domain have also received attention from the machine learning communities. Falling is a common problem for elderly people which can lead to further consequences that negatively affect their health and well-being. Thus, the majority of activity detection research work has focused on fall-event detection to possibly identify and anticipate the falling event as well as suggesting a systematic approach to providing the right response for the falling case.

Haobo Li and et. al. [3] worked on creating a system to detect falling events and record the normal pattern for daily living activities for elderly people who are at a high risk of falling. Multimodal sensing that combines wearable sensors and radar sensor was used to increase the classification's accuracy. They additionally investigated the use of a hierarchical classification approach by dividing human activities into sub-groups. Ten activities were considered in the experiments. These activities were walk, walk while carrying an object, sitting, standing, pick up an object, tie shoelaces, drinking water, answer a phone call, fall, crouch, and stand back up. The model was trained on one participant out of twenty and then tested on the other nineteen participants. The results provided an accuracy evaluation for normal classification versus hierarchical classification for different sensors and found that hierarchical1 classification showed a performance improvement in different scenarios.

Another approach was used by Arkham Zahri Rakhmani et. al. [4] from the University of Gadjah Mada in Indonesia to prototype a full detection system using the built-in smartphone sensors, accelerometers, and gyroscopes. The system intended to identify fall incidents and distinguish them from daily living activities. In the case of fall incidents, the system sent an alarm to the family to facilitate a fast response. The study result showed that the system was able to detect falling events with 93.3% accuracy.

Activity recognition is also used to help reduce suicide cases. Researchers from Missouri University of Science and Technology introduced a system called SHARE [5]. The system aimed to infer self-harmful activities using accelerometer data from a smartphone that was secured to the participants' hands. The study investigated two types of activities: harmful activities such as cutting a hand, smothering, and hanging as well as non-harmful activities like drinking, lying down, and rolling. The study acquired data from four participants and trained the model using K-nearest algorithms to achieve an accuracy of 80% in distinguishing harm activities from non-harm activities.

Health and diet control is an area that also benefits from human activity recognition research work. Detecting diet-related activities can provide a person with insight about their diet behavior and encourage adopting healthy eating habits. Research in this area was spearheaded by Tauhidur Rahman and et. al. [6] to predict an “about-to-eat” moment then provide just-in-time eating intervention. To this effort, they developed a model to detect about-to-eat moment activities such as body vibration, jaw movement, hand gestures, etc. A set of ubiquities sensors were used to collect data from eight participants for five days. The participants also introduced a smartphone application which they used to log the start and the end of an eating event. The researchers built a model using various algorithms to distinguish two classes “about-to-eat” moments and “not-about-to-eat” moments. The study used various algorithms such as Random Forest and others to train the model. The performance of the Random Forest algorithm outperformed the other algorithms that were used in the experiments.

Aiming at the same research direction, this time at Northwestern University [7] conducted an study to detect and count swallow events when eating. The work aimed to understand the intake behavior and enabled to capture the poor eating behavior by detecting swallow actions. The researchers proposed a neural network framework called SwallowNet. The data was collected using a wearable necklace designed specifically for the experiment purposes. An automated feature learning approach was used to train the model on raw data. The system displayed an ability to accurately define eating patterns and passively detect swallow events.

Training and coaching areas also occurred in the scope of HAR research, which shows promise to provide automated trainers for activities that previously depended on real personal training such as dancing and tennis sports [8][9]. With the availability of virtual trainers, more affordable training opportunities will provide high performance and accurate feedback.

Abu Zaher Md Faridee et. al. proposed HappyFeet that model, assess and recognize dance activities on the floor [8]. The work focused on assessing Indian style dance. The data was collected from multi-channel sensors that were attached to a professional dancer. Two techniques were used to train the model. There were shallow learning classifier techniques and deep learning techniques. Among the shallow learning classifiers, the Random Forest algorithm performed the best. The neural network approach outperformed the shallow learning classifiers and provided a more accurate detection of the dancer's micro-steps. This work paves a way to develop a virtual dance trainer that can provide feedback to a dance learner.

SoccerMate is another application for HAR in sports profiling. This research was carried out by H M Sajjad Hossain et. al. [9] to provide a profiler to evaluate the performance of a soccer player. A built-in sensor in a smartwatch was used to collect data for soccer activities including possessing the ball, passing, kicking, sprinting, running, and dribbling. A neural network-based was utilized to train the model and achieved an overall accuracy of 86.54%. The work showed the promise in extending the use of wrist-worn smartwatches past providing basic analytical information providing more analytical results to evaluate the level of professionality of a player. This would be helpful to measure the performance as well as understanding the level of competency between players.

Activity recognition brought many interesting applications for many fields such as health, security, and automated assistance and it can bring more potential to other fields as it is the basic building block for many applications. This thesis will focus on applying activity recognition in the fashion field. We will work on detecting make-up activities using machine learning and sensor's data. This would be a preliminary step toward further work in using machine learning to facilitate automated support to the make-up training industry.

Chapter 3: Methodology

3.1 General approach

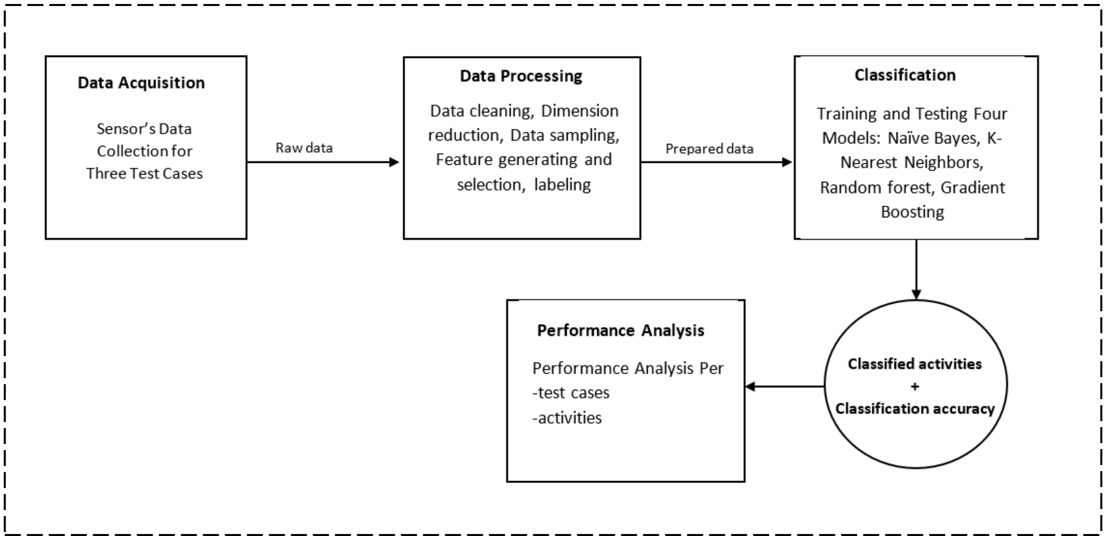


Figure 1: General approach overview

Figure 1 represent the general approach we followed in our experiment. we started by collecting the sensors' data from ten participants using the Microsoft Band smartwatch using the Companion smart-phone application. The Companion data collection smart-phone application collects the data from the watch and annotate the received data to with the corresponding activity and save them as CSV files. Our data consist of the reading from two sensors: accelerometer and gyroscope for applying five make-up activities. We have three different datasets collected in three different experimental set up. The first dataset was collected from participants inside a controlled environment. We gave participants specific instructions on how to perform the five make-up activities. The second dataset was collected from participants under an uncontrolled environment. We did not give them any instructions on how to perform the five activities and they did the applying make-up activities in their own way. Third dataset was synthetically generated by combining the existing datasets from the participants who were under both controlled and uncontrolled environments. The second step we did is preparing the raw data for classification. We checked missing values and reduced the dimensionality of the data by turning the three-dimension readings into one-dimensional through calculating the magnitude for both accelerometer and gyroscope's readings. We filtered the data using median filtering and one-minute time interval length using 50% overlapping. We used a python package named 'tsfresh' to generate 1588 features and reduced the features' size by appliying Principal Component Analysis (PCA) [20]. We conducted a small version classification process, using partial datasets, for three different feature sizes; three, ten, and twenty. By comparing the accuracies, we found that the best size of feature list is 10 to elicit the most performance out of the final models. Then we used WEKA software, an open-source toolkit of machine learning, to train and test four classification algorithms, Naïve Bayes, K-Nearest Neighbors, Random forest, Gradient Boosting on full datasets. We applied the same process for all the three dataset and created three test cases in our experiment; Test Case 1: classifying activities under a controlled environment; Test Case 2: classifying activities under an uncontrolled environment; Test Case 3: classifying activities under both controlled and uncontrolled environment. Finally, we analyzed the activity classification result we obtained from the four classifiers. We evaluate the classifiers' performance per test case and per activity.

3.2 Data collection

We collected data from ten participants for the five applying-make-up activities in this study. The activities included in the study were applying cream, lipsticks, blusher, eye shadow, and mascara. We divided the participants into two categories: participants who perform the experiments under a controlled environment and participants who perform the experiments in an uncontrolled environment. We assigned five participants for each group. For the participants inside the controlled environment, we gave them instructions on how each item should be applied. For the uncontrolled environment's participants, we allowed the participant to perform the five make-up activities in their own way with no instructions provided.

Although, in real life each make-up activity could take less than a minute to be applied, we wanted to collect data for five minutes duration with each activity. To accomplish this, we asked the participants to continue each activity for five-minute period and to follow a certain sequence of the activities. We further stipulated that the participants perform three of the activities (applying blush, eyeshadow, and mascara) on one cheek and one eye, because we wanted to focus on the exact movement of performing each activity. This also helped eliminate any unnecessary movement of the participant's hand.

We collected data using the Microsoft-band smart-watch and the Companion smart-phone application. The participants wore a Microsoft band in their hand while performing the activities. The band captured the participants' hand movements for each activity using two sensors, an accelerometer and gyroscope. We used the Companion smart-phone application to collect the data from the watch and synchronized the accelerometer and gyroscope data. We demoed the user in the controlled environment to make sure they follow it correctly. For annotating the received data, we used the companion application to annotate the data and we also recorded video while performance and track the time of the activates to ensure assigning data to its corresponding activities.

3.3 Understanding the data

Building the right model for a dataset requires exploring the data itself and understanding the nature of it. This data analysis can be approached by statistically summarizing the data as well as visualizing the data.

The datasets we are working with, consists of six attributes for each activity. There are three axis readings from the accelerometer sensor and the gyroscope sensor with corresponding timestamp and activity label. Table 1 shows a sample of the collected data for the applying cream activity.

	timestamp	datatime	acc-x	acc-y	acc-z	gyro-x	gyro-y	gyro-z	class
0	1.549570e+12	2/7/2019 13:48	-0.650146	0.443115	0.781250	67.256096	29.939026	-18.658537	applying cream
1	1.549570e+12	2/7/2019 13:48	-0.554443	0.459229	0.649902	18.658537	-66.554880	13.628049	applying cream
2	1.549570e+12	2/7/2019 13:48	-0.423584	0.541504	0.454834	-37.682926	141.707320	-2.347561	applying cream
3	1.549570e+12	2/7/2019 13:48	-0.427979	0.498779	0.516846	-73.353660	128.445130	-10.670732	applying cream
4	1.549570e+12	2/7/2019 13:48	-0.418213	0.433838	0.627197	-59.512196	-50.792683	11.280488	applying cream
5	1.549570e+12	2/7/2019 13:48	-0.385254	0.333496	0.820801	-46.585365	75.518295	14.329268	applying cream
6	1.549570e+12	2/7/2019 13:48	-0.444580	0.272461	1.149902	-9.817074	150.213420	1.280488	applying cream
7	1.549570e+12	2/7/2019 13:48	-0.727295	0.266846	1.124268	46.676830	156.219510	-33.262196	applying cream
8	1.549570e+12	2/7/2019 13:48	-0.742188	0.340088	0.939697	117.408540	33.932926	-7.256098	applying cream
9	1.549570e+12	2/7/2019 13:48	-0.590820	0.440674	0.614990	69.237810	-81.493904	25.213415	applying cream
10	1.549570e+12	2/7/2019 13:48	-0.416504	0.574463	0.325439	-5.213415	132.926830	26.493902	applying cream
11	1.549570e+12	2/7/2019 13:48	-0.410645	0.450928	0.550049	-79.329270	-90.884150	23.993902	applying cream
12	1.549570e+12	2/7/2019 13:48	-0.393799	0.459473	0.625244	-111.219510	-27.804878	8.079268	applying cream
13	1.549570e+12	2/7/2019 13:48	-0.389404	0.310791	0.768799	-92.560974	63.536587	14.939025	applying cream
14	1.549570e+12	2/7/2019 13:48	-0.512695	0.169922	1.233643	-13.780488	126.371956	-12.378049	applying cream

Table 1: Representation of the raw data for the applying cream activity

Understanding the data also requires looking at the peaks of the data and checking the data dimensions and datatypes. Checking class distribution can provide a preliminary idea about how the raw data is distributed among classes and facilitate later steps to fix any imbalance, where a certain class can have significantly more values than the other classes. One representation that can be done to understand the data distribution among one dimension (attribute) of the data is to plot a histogram for the attribute values. It helps to identify outliers' values. Figure 2 shows the histogram plot for the applying cream activity's raw six attributes data. The histogram shows some outliers exist for acc-x and acc-z attributes. It also gave us an idea that our data is having mostly a Gaussian univariate distribution.

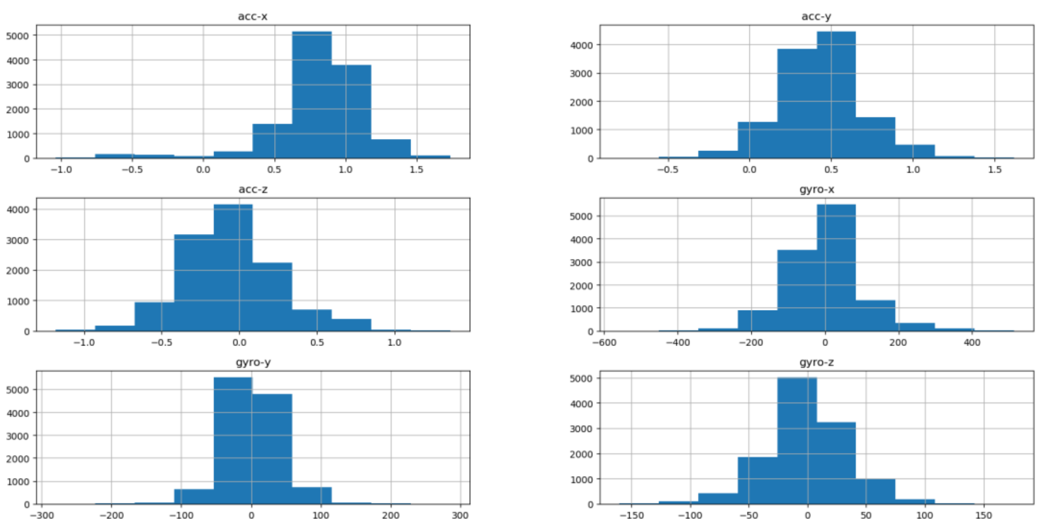


Figure 2: Raw values distribution of the applying cream activity

Figure 3 shows the accelerometer raw signals for the five activities. We can see that each activity has a different pattern that distinguishes it from other activities.

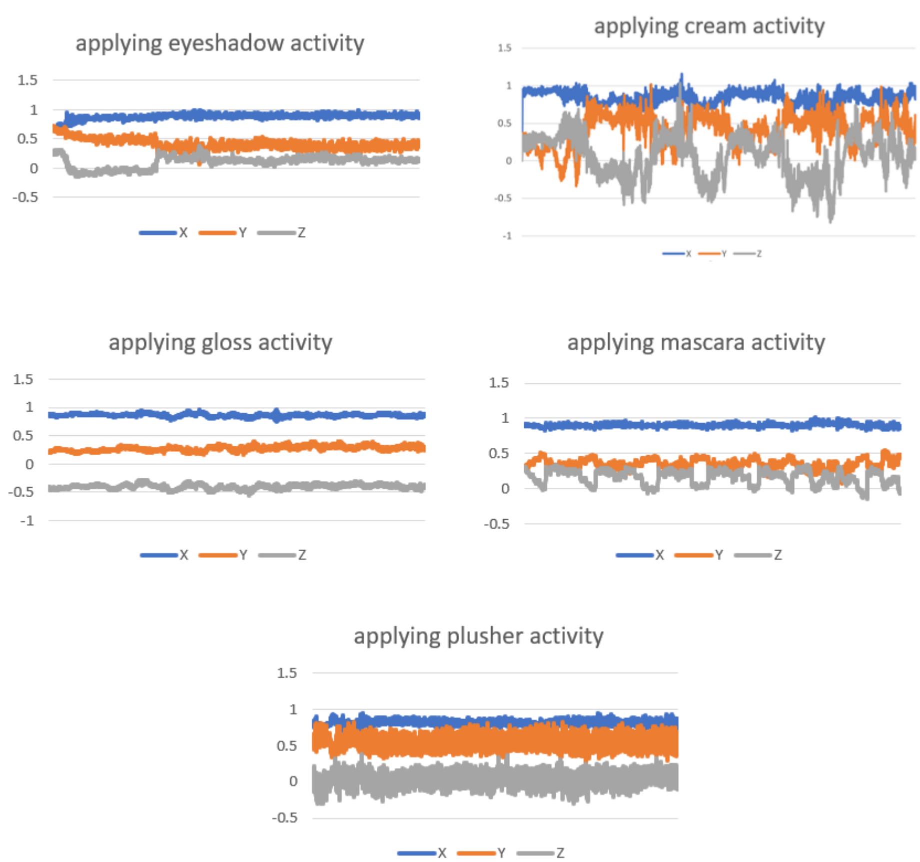


Figure 3: Accelerometer raw signals for the five activities

3.4 Data Preprocessing

Data preparation is an essential step in machine learning. It focuses on creating the right data version out of the raw data so can allow the machine learning algorithms to learn their best from the fed data. In data preparation, we worked on checking missing values, remove noisy data, format and sample the data. It can involve transferring the cleaned data to a version that is more useful to the machine learning algorithms with engineered features. New features can be engineered out of the cleaned data using scaling, attribute decomposition or attribute aggregation.

To prepare the collected data for modeling, we started by combining each participant's accelerometer and gyroscope data for all the five activities in one excel file. The resulting 50 data excel files were then processed as five files for each one of the ten participants. In this section we will go through the steps we did for preparing our data.

3.4.1 Checking Missing Values and Outliers

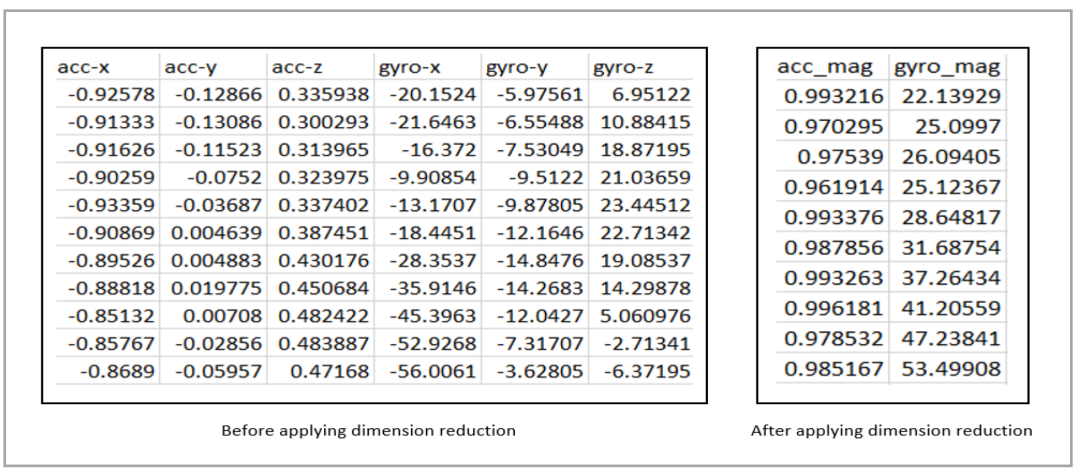
Initially, we checked the data for the existence of any missing values and outliers. There were no missing values in the data, but outliers did exist, as it clearly appears in the histogram attribute's representation in Figure 2. To remove the outliers, we simply removed any value that was beyond -3 or +3 standard deviation of each data attribute.

3.4.2 Dimension Reduction/reduce dimensionality

Due to the large amount of data, we followed an additional step to reduce the dimensionality of our data. We turned the axis values of the accelerometer into a one value by applying the magnitude of the acceleration using the following formula

$$A = \sqrt{a_x^2 + a_y^2 + a_z^2}$$

The same process was applied to three axis values from gyroscope. This step allowed us to cut down data preprocessing steps by making the process less computationally less strenuous and time consuming. Figure 4 shows the data before and after dimension reduction for a sample of our dataset.



acc-x	acc-y	acc-z	gyro-x	gyro-y	gyro-z
-0.92578	-0.12866	0.335938	-20.1524	-5.97561	6.95122
-0.91333	-0.13086	0.300293	-21.6463	-6.55488	10.88415
-0.91626	-0.11523	0.313965	-16.372	-7.53049	18.87195
-0.90259	-0.0752	0.323975	-9.90854	-9.5122	21.03659
-0.93359	-0.03687	0.337402	-13.1707	-9.87805	23.44512
-0.90869	0.004639	0.387451	-18.4451	-12.1646	22.71342
-0.89526	0.004883	0.430176	-28.3537	-14.8476	19.08537
-0.88818	0.019775	0.450684	-35.9146	-14.2683	14.29878
-0.85132	0.00708	0.482422	-45.3963	-12.0427	5.060976
-0.85767	-0.02856	0.483887	-52.9268	-7.31707	-2.71341
-0.8689	-0.05957	0.47168	-56.0061	-3.62805	-6.37195

Before applying dimension reduction

acc_mag	gyro_mag
0.993216	22.13929
0.970295	25.0997
0.97539	26.09405
0.961914	25.12367
0.993376	28.64817
0.987856	31.68754
0.993263	37.26434
0.996181	41.20559
0.978532	47.23841
0.985167	53.49908

After applying dimension reduction

Figure 4: Sample data before and after applying dimension reduction

3.4.3 Applying median filtering

Noisy data is unwanted data that mislead the algorithm to find patterns in the data. Treating the noisy data can benefit the training process to become faster and to reduce the complexity of the model and to improve its accuracy. We applied Median filtering to the accelerometer and gyroscope magnitude to get smoother signals and to eliminate the noise in the dataset.

3.4.4 Applying features and sampling the data

Machine learning algorithms cannot be applied to raw data. It is essential to extract features out of the collected data. Features are a meaningful summarization of the raw data that can allow the machine learning algorithms to find patterns. Prior to extract features, we need to fragment the data into fixed time windows to enable extracting features for each single window of data.

For sampling our data, we segmented the data into one-minute time interval length with 50% overlapping. We found it is a suitable length to show a pattern of the hand's movement for each activity.

For feature extraction we used a python package called tsfresh. This package that can be used for feature extraction from time series features. It is a powerful tool that allows for extracting many characteristics from the data. The downside of it is time-consuming process and needs to be executed on a device with high computation capability to reduce the execution time for feature extraction in case of handling large amount of data. We applied the tsfresh features generator to our data and it resulted in 1588 features for each single window of the data.

3.4.5 Feature selection

It is important for the machine learning model to be trained on a set of features that are relevant to the required target to ensure better accuracy and reduce the overfitting problem. Doing so also allows the model to train faster. Since we have plenty of features after applying the tsfresh generator to our data, it is important to ensure the protection of features that would positively contribute to model training and discard features that are not needed.

We applied two approaches for selecting the best features for our model, Recursive Feature Elimination and Principal Component Analysis. We will inspect the two methods later, while training the model, to see which one performs better with the model.

First, we applied Recursive Feature Elimination (RFE) which select best features by training a machine learning model on the whole set of the features and recursively removing attributes that contribute less to the training accuracy. It rebuilds the model on those which features remain and continue doing the process till reaching the required number specified by the user. We applied the RFE method using a Logistic Regression algorithm and selected the best ten features which contributed the most to model accuracy.

Second, we applied Principal Component Analysis (PCA). This is another technique which is used to summarize features and reduce the dimensionality of the data using linear algebra. In PCA, users can choose the number of dimensions, principle component, that they are willing to compress the original data to. It is required for the data to be scaled before applying the PCA, so we started by scaling our data to apply PCA. We applied PCA multiple times with different principle components including 3,5,7, 10 and 20. The various PCA results will be evaluated in the model training phase to check which principle component will contribute better to the training process [11]

3.5 Building Models

After preparing the data, the next step is to build the machine learning models, so it may train and test models on our data. In our work, we are following a classification approach where we built various classifier models to classify five make up activities.

Activity 1: Applying cream

Activity 2: Applying lipsticks

Activity 3: Applying blusher

Activity 4: Applying eyeshadow

Activity 5: Applying mascara

We used four machine learning classification algorithms to build our models: Naïve baize, k-nearest neighbors' algorithm, Random forest algorithm, and Gradient Boosting. We selected the algorithms based on the problem we want to solve and after reviewing various papers that solve similar problems as ours. Below there is a brief overview of each algorithms we used in our experiments.

Naïve Bayes

This classification is a machine learning algorithm that was built based on Bayes Theorem.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

In Bayes Theorem we want to find the P(A) given that B has occurred. A is the hypothesis and B is the evidence. The theorem is called Naïve because it assumes that attributes are independent, and they do not affect the existence of each other [12].

K-Nearest Neighbors' Algorithm

This algorithm is a classification algorithm which can be used for regression as well as classification. For classification, it works by finding the most similar instances for the new instant x in the entire training dataset. The algorithm subsequently guesses based on their classification. The class that appears most frequently among similar instances for instance x will be chosen to be the class for instance x.

It is also known as the lazy learning algorithm because it requires no learning before building the model. All the learning required happens at the same time as the prediction. Also, K-nearest algorithm is a non-parametric algorithm as it does not make any assumptions about the distribution of the data. [13] [14].

Random Forest Algorithm

Random Forest is an ensample algorithm which means it combines more than one algorithm. It consists of multiple decision trees that were created from a randomly selected subset of the training data. The algorithm decides on the final class for an instance after aggregating the voting results from the various decision trees. It provides better accuracy than a single tree because aggregating votes of more than one tree prevents noisy data from affecting the result. Another reason for this accuracy is that Random Forest algorithms enforce each decision tree to be unique, so they may elicit the best votes out of the trees [15].

Gradient Boosting

Gradient boosting is an ensemble algorithm that utilizes boosted decision trees to build a predictive model. It works based on the boosting theory where weak learners/models are repeatedly used, forcing the algorithm to handle the classification of harder instances. Each time, a new learner is added and encouraged to become an expert in classifying instances that were miss-classified by earlier learners. The result of each learner is recorded to help with improving the performance of the final learner. The training process ends when there is no such improvement in the performance on the external validation datasets [16].

In this work we built a multi-classifiers model to classify the five make up activities for three test cases.

Test case 1: classifying activities under a controlled environment

In this test case we built a multi-classifier model to classify five make up activities from datasets that were collected in a controlled environment. We gave the participants specific instructions on how to apply the activities. Table 2 shows the instructions that were given to participants on how to apply each activity.

Activity	Instruction
Activity 1: Applying cream	Apply to the whole face in a circular movement that starts from right cheek and continues to forehead, left cheek, and lastly the chin.
Activity 2: Applying lipsticks	Apply to the lower lip first, then to the upper lip.
Activity 3: Applying blusher	Apply forward and backward in straight movements.
Activity 4: Applying eyeshadow	Apply forward and backward in straight movements starting from right to left.
Activity 5: Applying mascara	Apply only for upper lashes.

Table 2: Controlled environment's applying makeup instruction for each activity

Test case 2: classifying activities under an uncontrolled environment

In this test case we built a multi-classifier model to classify the five make up activities from datasets that were collected under an uncontrolled environment setting. We did not give any instructions on how to perform the five activities and they did the applying make-up activities in their own way.

Test case 3: classifying activities under both a controlled and an uncontrolled environment.

In this test case we built a multi-classifier model to classify the five make up activities from a dataset that combined both the controlled environment dataset and the uncontrolled environment dataset.

We started by applying the preliminary models on two groups of data with two different sampling windows, a half-minute window and 1-minute window. We also applied the preliminary models to datasets that use different techniques for dimension reduction RFE and PCA. The models performed better when using the 1-minute window dataset and the PCA dimension reduction technique. So, we decided to build the final models on 1-minute sampled data which had been reduced using the PCA dimensionality reduction technique.

For building our final model, we used four algorithms to train and test the model: Naïve baize, k-nearest neighbors' algorithm, Random forest algorithm, and Gradient Boosting to train different models on our 1-minute sampled data. We trained and tested the model using datasets with different dimensionalities: three dimensions dataset, ten dimensions dataset, and twenty dimensions dataset. We applied 10-fold cross validation and adjusted some hyperparameters to get the most out of the used algorithms. For the K-Nearest Neighbors' Algorithm we used K=5 and for Random forest algorithm we used seed=7. Finally, we applied the Gradient Boosting using Random forest algorithm with seed=7.

Chapter 4: Result and Discussion

This chapter presents the result for our experiment. Section 4.1 shows comparison of the model's performance when using different dimensionality sizes to determine the best feature size to be used. Section 4.2 illustrates the model's performance when applied on ten dimension's datasets for three test cases. Section 4.3 shows performance analysis per activity for all three test cases.

4.1 Result for feature size analysis

We trained and tested four models on three different datasets that differ in dimension size. We sought to identify the best dimensions, feature, and size to be used for the final models. The models' performance varied for all test cases based on the algorithm and number of dimensionalities that were used. We averaged the performance of the models to find out which dimension size worked the best. Models performed the best when using the ten-feature size. We concluded that it was the best size to elicit the most performance out of the final models. Table 3 and figure 5 shows the general models' performance for all three test cases when using three, ten, and twenty-dimensional feature sizes.

Classifier	**Feature size (PCA)**		
	(PCA) =3	**(PCA) =10**	**(PCA) = 20**
Naïve Bayes	55.90%	62.07%	68.30%
K-Nearest Neighbors' Algorithm	84.97%	88.47%	85.67%
Random forest algorithm	82.70%	90.67%	89.17%
Gradient Boosting	81.87%	86.97%	84.30%
Average Accuracy	**76.36%**	**82.05%**	**81.86%**

Table 3: Models' performance comparison using various dimensionality for all cases

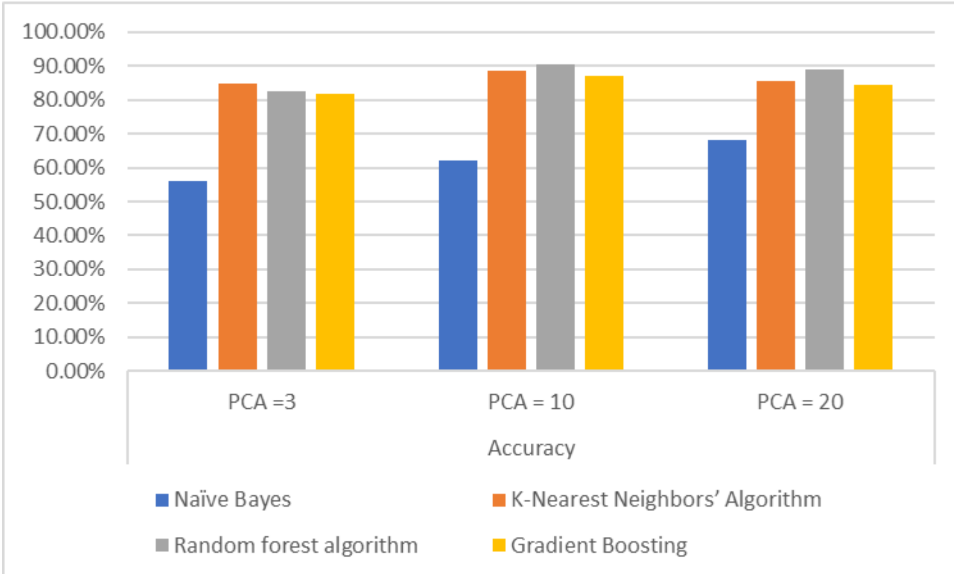
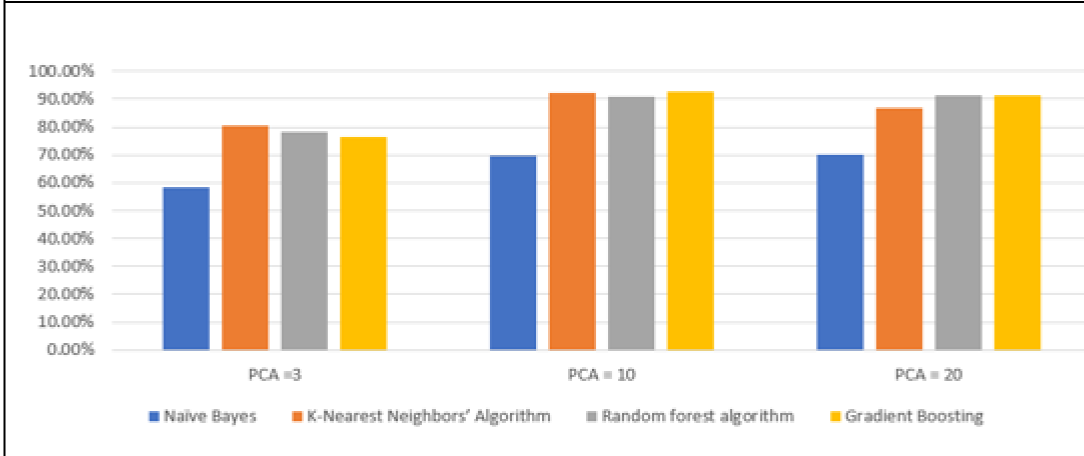


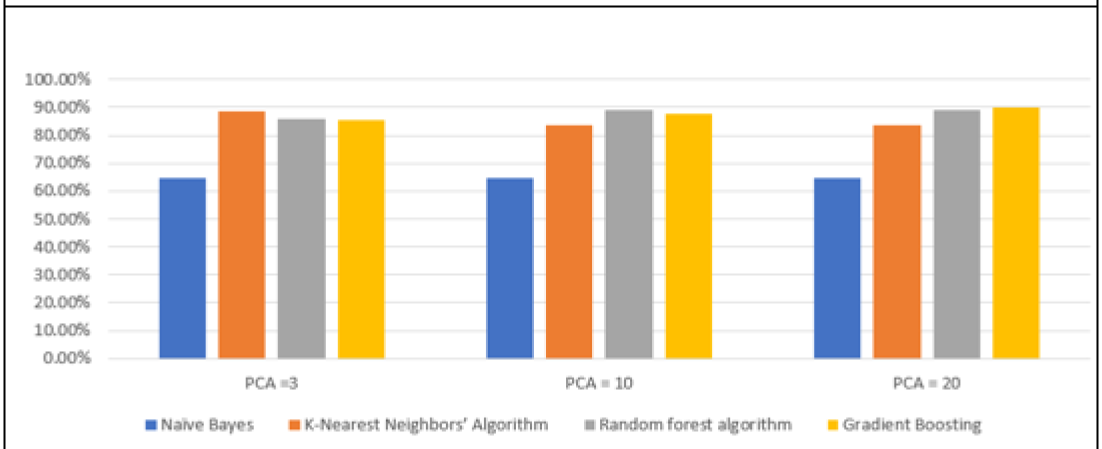
Figure 5: Models' performance for all test cases together when using various dimensionality sizes

We can see that the general models' performance is the highest when using the ten-dimension size and is the lowest when using the three-dimension size. There was a performance improvement when we increased the dimension size to ten, but when we increased the dimension size from ten to twenty, we did not see a high level of improvement. In response, we indicated that ten feature size is the threshold size to get the best performance out of our models for all the three test cases. Figure 6 shows how each model performed when using different dimensionalities for each test case.

Test case 1: classifying activities under controlled environment



Test case 2: classifying activities under uncontrolled environment



Test case 3: classifying activities under both controlled and uncontrolled environment

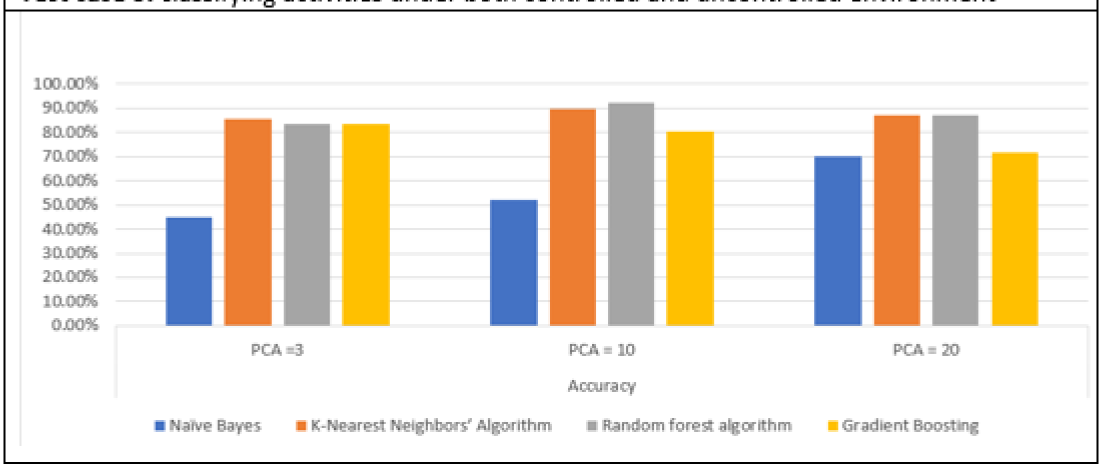


Figure 6: Comparison on how models performed for each test case when using various dimensionality sizes

Generally, we see performance improvement in all test cases when improving the dimensionality size from to three to twenty. We can see that the Naïve Bayes algorithm performed the lowest and was outperformed by the other algorithms on all three test cases and with the three dimensionality sizes. The three algorithms (KNN, Random Forest, and Gradient Boosting) shows various performance among the three test cases when using different dimensionality.

4.2 Result for models' performance when using ten-feature size

We trained and tested our final models on three datasets with a ten-feature size to create three test cases for our experiment. Each dataset represents a test case in our analysis. Tables 4,5, and 6 show the resulting model performance for the three test cases. Figure 7 provides a graphical representation of the outcomes.

	Classifier	Accuracy	Precision	Recall	F-measure	ROC AUC
Test Case 1: Classifying Activities in a Controlled Environment	Naïve Bayes	69.5 %	0.72	0.69	0.69	0.90
	K-Nearest Neighbors' Algorithm	92.4 %	0.92	0.92	0.91	0.98
	Random forest algorithm	90.8 %	0.91	0.91	0.91	0.99
	Gradient Boosting	92.7 %	0.92	0.92	0.92	0.99

Table 4: Models' performance on test case 1 (feature size= 10)

	Classifier	Accuracy	Precision	Recall	F-measure	ROC AUC
Test Case 2: Classifying Activities in an Uncontrolled Environment	Naïve Bayes	64.50 %	0.66	0.64	0.64	0.90
	K-Nearest Neighbors' Algorithm	83.50 %	0.83	0.83	0.83	0.97
	Random forest algorithm	89.20 %	0.89	0.89	0.89	0.98
	Gradient Boosting	87.80 %	0.87	0.87	0.87	0.98

Table 5: Models' performance on test case 2 (feature size= 10)

	Classifier	Accuracy	Precision	Recall	F-measure	ROC AUC
Test Case 3: Classifying Activities in both Controlled and Uncontrolled Environment	Naïve Bayes	52.20	0.56	0.52	0.51	0.84
	K-Nearest Neighbors' Algorithm	89.50	0.89	0.89	0.89	0.98
	Random forest algorithm	92.00	0.92	0.92	0.92	0.99
	Gradient Boosting	80.40	0.80	0.80	0.80	0.87

Table 6: Models' performance on test case 3 (feature size= 10)

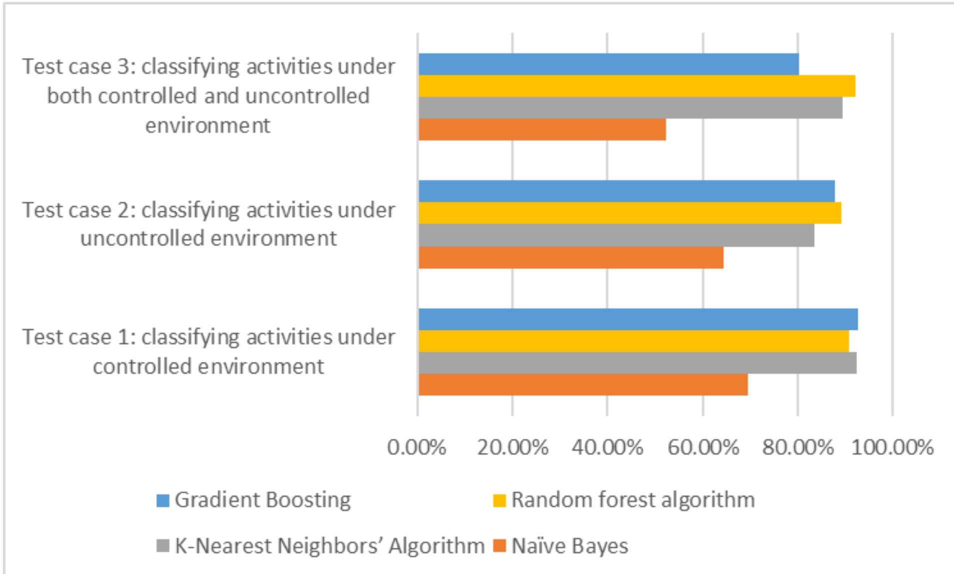


Figure 7: Models' accuracy analysis for all test cases (feature size= 10)

The accuracy in classifying the activities varied for each test case. All models classified the activities in the highest accuracy when applied on the controlled environment, test case1, with an average accuracy of 86.53%. We can attribute that to a minimal variety of hand's movement patterns between participants in the controlled environment, test case 1. The models' showed the second-best total performance when applied on the uncontrolled environment, test case 2, with an average accuracy of 81.25%. The third test case had the lowest average accuracy with an accuracy of 78.52%. The average accuracy for the model's performance in all three test cases was brought down by the low performance of individual models, such as Naïve Bayes, which performed the lowest in all test cases. Although test case 3 had the lowest averaged model accuracy, it has a high accuracy coming from individual model's performance such as Random forest. In test case 3, Random Forest Algorithm attained the highest accuracy among all test cases with an accuracy of 92%. This accuracy is only outperformed by Gradient Boosting when applied in test case 1 with accuracy of 92.7%. Algorithms performed differently in different test cases. We can see that Gradient boosting performed lower in test case 3 than in test case 1 and test case 2. Random Forest increased its performance in test case3 higher than its performance in test case 1 and 2 and provided the highest performance among the other models in test case 3. K-Nearest Neighbors' Algorithm showed a good performance in all test cases and provided its highest performance in test case 1 with an accuracy of 92.4%. Figure 8 shows the confusion matrix for the three-best performance for all three cases.

```
 b  c  d  e   <-- classified as
 1  0  0  5 |  a = applying cream
77  6  4  1 |  b = applying eyeshadow
 2 83  4  0 |  c = applying gloss
 4  0 68  0 |  d = applying mascara
 0  0  0 92 |  e = applying plusher
```

Gradient boosting performance in Test Case 1

```
  a  b  c  d  e   <-- classified as
 67  0  0  0  5 |  a = applying mascara
  4 82  0  0  3 |  b = applying lipstick
  1  0 84  4  1 |  c = applying cream
  0  0  3 91  0 |  d = applying plusher
  6  7  4  0 74 |  e = applying eyeshadow
```

Random Forest performance in Test Case 2

```
   a   b   c   d   e   <-- classified as
 119   0   4   5   0 |   a = applying mascara
   1 141   1   3   1 |   b = applying cream
   4   1 135   4   0 |   c = applying lipstick
  13   2   6 121   3 |   d = applying eyeshadow
   1   7   0   1 142 |   e = applying plusher
```

Random Forest performance in Test Case 3

Figure 8: Confusion matrix analysis for the best model's performance in the three test cases

4.3 Models' Performance Analysis Per Activity

We did a comparison of the total models' classification accuracy for individual activities per test case. We calculated the models' average accuracy for classifying certain activity for each test case. The comparison result that we obtained is shown in table 7

Test Case	**Activities**				
	Applying Cream	Applying Lipsticks	Applying blusher	Applying Eyeshadow	Applying Mascara
Test case 1	83.53	91.3	93.1	75.3	85.45
Test Case2	77.2	89.55	82.45	75.48	81.7
Test case 3	74.65	88.38	81.45	70.33	77.73

Table 7: Activity classifying accuracy for each test case

Figure 9 provides a graphical representation for the result we obtained from our activity performance analysis. Each bar represents the average accuracy obtained for classifying an activity by all algorithms in a certain test case.

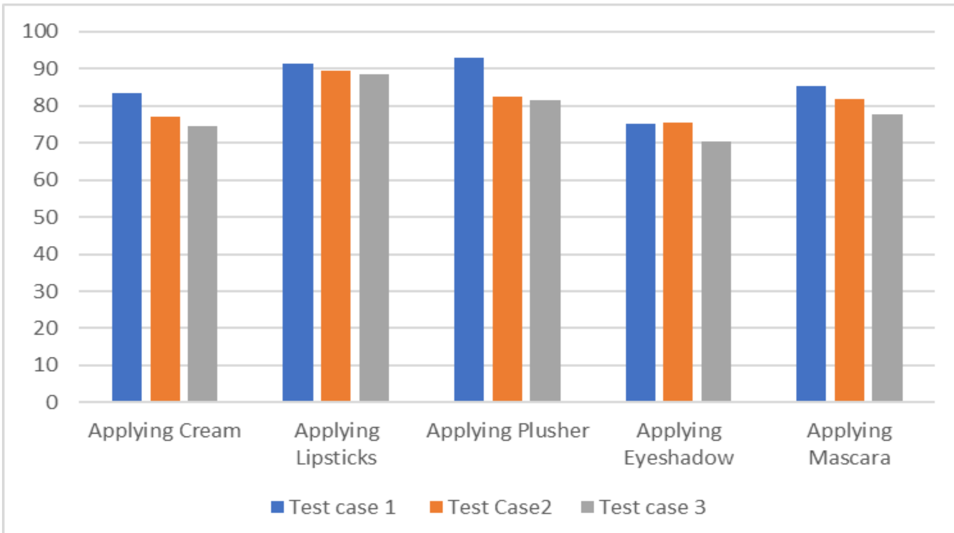


Figure 9: Activity classifying accuracy for each test case

We can see that most activities were best classified in test case 1Test case 1 obtained the highest performance in all models for all activities because of the low variance in an activity applying sensor's reading among various participants, due to the instruction that were given for all the participants in test case 1 on how to apply each activity. We can also see that applying blusher activity classification shows a visible improvement in test case1 compared to test case 2 and 3. The lower accuracy for classifying the applying blusher activity in both test case 2 and 3 might be due to the high variance in sensor data for applying blusher because of the variety in how people apply blusher when not given strict guidelines. Tables 8, 9, and 10 show details on how each classifier, has performed when classifying each activity for each test case.

	Classifier	Activity				
		Applying Cream	Applying Lipsticks	Applying blusher	Applying Eyeshadow	Applying Mascara
Test Case 1 (Controlled Environment)	Naïve Bayes	57.8 %	79.8 %	86.2 %	42.9 %	81.9 %
	K-Nearest Neighbors’	90.0 %	100%	93.6 %	89.0 %	86.1 %
	Random forest	93.3 %	92.1 %	96.8 %	81.3 %	81.9 %
	Gradient Boosting	93%	93.30%	95.80%	88.00%	91.90%

Table 8: Activity classifying accuracy for test case 1

	Classifier	Activity				
		Applying Cream	Applying Lipsticks	Applying blusher	Applying Eyeshadow	Applying Mascara
Test Case 2 (Uncontrolled Environment)	Naïve Bayes	47.4 %	83.6%	54.4 %	63.0 %	75.0 %
	K-Nearest Neighbors’	89.5 %	98.2 %	91.2 %	68.5 %	69.6 %
	Random forest	87.7 %	90.9 %	91.2 %	85.2 %	91.1 %
	Gradient Boosting	84.2 %	85.5 %	93.0 %	85.2 %	91.1 %

Table 9: Activity classifying accuracy for test case 2

	Classifier	Activity				
		Applying Cream	Applying Lipsticks	Applying blusher	Applying Eyeshadow	Applying Mascara
Test Case 3 (Controlled and Uncontrolled Environment)	Naïve Bayes	32.0 %	79.9 %	52.3 %	43.4 %	53.9 %
	K-Nearest Neighbors’	92.5 %	97.9 %	92.1 %	78.6 %	85.9 %
	Random forest	95.9 %	93.8 %	94.0%	83.4 %	93.0 %
	Gradient Boosting	78.2 %	81.9 %	87.4 %	75.9 %	78.1 %

Table 10: Activity Classifying Accuracy for test case 3

Chapter 5: Conclusion

This thesis introduces a novel application for human activity recognition (HAR) using machine learning techniques. Sensor data and machine learning algorithms are accounted for classifying five different make-up activities. Three different experiments are carried out over three different datasets. These datasets have been collected under three various controlling environments- controlled environment, uncontrolled environment, semi-controlled environment. In the controlled environment, the participants are given activity performing instructions where as in the uncontrolled environment, the participants execute the activities without any restrictions. Combining both datasets, semi-controlled environment dataset is formed. The datasets are collected from ten participants wearing a smartwatch. After that, the datasets are preprocessed by cleaning, filtering, apply features and reduce the dimensionality. In terms of dimensionality reduction appoaches, the data analysis reveals that Principal Component Analysis with feature size ten suits best for this research work. Finally, leveraging four machine learning models, experiments are carried out over the preprocessed diastases.

After applying the models, it is found that proposed approach is capable of classifying the five make-up activities that were involved in this study: applying cream, applying blusher, applying eyeshadow, applying lipsticks, and applying mascara. The accuracy in classifying the activities varied for each experiment. All models classified the activities in the highest accuracy when applied on the controlled environment, with an average accuracy of 86.53%. This high accuracy can be attributed to the minimal variety of hand's movement patterns among participants under the controlled environment. The models' showed the second-best total performance when applied on the uncontrolled environment, with an average accuracy of 81.25%. Experiments with the semi-controlled environment generates the lowest average accuracy with an accuracy of 78.52%. It is found that the Naive Bayes performed worst and KNN performed well among the selected models. Random Forest Algorithm attained a high accuracy in test case 3 with an accuracy of 92%. This accuracy is only outperformed by Gradient Boosting when applied in test case 1 with accuracy of 92.7%.

The accuracy in classifying individual activity varied based on the dataset and the algorithm used. Applying blusher and lipsticks were highly classified in all three experiments and the most over the controlled environment collected dataset. Applying cream and mascara were best classified in test case1 compared to test cases 2 and 3. K-Nearest Neighbors' Algorithm did the best performance in classifying the applying lipsticks activity in all three test cases, while other activities were best classified by different algorithms in different test cases. From the obtained experimental results, it can be concluded that classifying make-up activities using shallow machine learning algorithms and sensor-driven data is possible and establishes the core building blocks for a future make-up supportive tool or virtual training system.

Chapter 6: Work Contribution

This work is instrumental to the fashion industry by contributing to a fundamental step in a potential applying make-up training system. Training is in high demand for the make-up workforce. Applying make-up is a skill that requires training and practice. For learning how to apply make-up, people start by either self-training themselves or are trained by someone who is more experienced with applying make-up. To attain professional skills in applying make-up, suitable professional tutoring or coaching is needed. The available choices in the make-up training industry is dependent on a real-person instructing, training, and evaluating trainees and lacks any automatic support. Machine learning has the potential to make automatic support for make-up training possible, as it already did to facilitate training support in other industries such as sports and dancing. The current literature on machine learning applications lacks any work in support of the make-up training industry.

This paper worked on identifying basic make-up applying activities using machine learning techniques. The work involved three experiments where each experiment differed in the type of environment, they were conducted in. The controlled environment experiment simulated a situation where make-up was applied according to an instructor's direction and evaluated the model's performance in recognizing the make-up activities. The un-controlled environment experiment attempted to recognize the activities when no rules where enforced. It investigated the model's robustness in recognizing an activity when it was performed in a variety of ways by different users. The third experiment is a combination of the two previous experiments where models were trained and evaluated on both controlled and uncontrolled datasets. It investigated the model's ability to recognize activities in a semi-controlled environment where an activity has two sets of patterns. Some activity patterns were similar according to the instructions, and other patterns were different due to the uncontrolled environment. The third experiment was the most simulating of a real-life situation when applying make-up. Most activities are applied in a common way with slight personal differences in how they are performed. The ability to recognize the applying make-up activity in an environment that simulates reality is an essential step toward facilitating a virtual training system. Such a system could be used as a stand-alone trainer, evaluator, or be used to instruct and evaluate trainees. The scope of this paper was to enable recognizing five applying make-up activities using machine learning, as a preliminary step toward further work in using machine learning to facilitate automatic support to the make-up training industry.

Chapter 7: Limitations and Future Work

This study has a limitation of not including a second side for the activities that required to be applied twice for two sides of the face such as applying blusher, applying eye shadow, applying mascara. Further improvement upon this work would include the second part of the activities and investigate the models' ability to identify the activity and the applied-on side. Also, in this study we avoided working with activities that had similar body movement patterns, such as applying lipsticks and filling eyebrows are avoided. Instead, activities that had distinct body movement patterns that could be identified using wrist-worn sensors are investigated. Future work can include adding new applying make-up activities such as filling eyebrows, applying eye liner, applying highlighter, and applying contour. To increase the ability to recognize similar activities and distinguish them from each other, additional finger sensors can be used in addition to the wrist sensors to track the fingers' tiny movement patterns for each activity and distinguish them accurately. Another machine learning technique, such as a Neural network, could be used to assess ability to provide better accuracy for recognizing the various applying make-up activities.

References

[1] A.Fleury, Norbert Noury, Michel Vacher, "Supervised Classification of Activities of Daily Living in Health Smart Homes using SVM", presented at the 31st Annual International Conference of the IEEE EMBS Minneapolis, pp. 2-6, September 2009. Available: https://ieeexplore-ieee-org.proxy-bc.researchport.umd.edu/stamp/stamp.jsp?tp=&arnumber=5334931&tag=1

[2] J.Parkka,M.Ermes,P.Korpipaa,J.Mantyjarvi,J.Peltola,andI.Korhonen, ‘‘Activity classification using realistic data from wearable sensors,’’ IEEE Trans. Inf. Technol. Biomed., vol. 10, no. 1, pp. 119–128, Jan. 2006. Available: https://ieeexplore-ieee-org.proxy-bc.researchport.umd.edu/stamp/stamp.jsp?tp=&arnumber=1573714

[3] H. Li, A. Shrestha, F. Fioranelli, J. L. Kernec and H. Heidari, "Hierarchical Classification on Multimodal Sensing for Human Activity Recogintion and Fall Detection," 2018 IEEE SENSORS, New Delhi, 2018, pp. 1-4. Available: http://ieeexplore.ieee.org.proxy-bc.researchport.umd.edu/stamp/stamp.jsp?tp=&arnumber=8589797&isnumber=8589503

[4] A.Z. Rakhman, L.E. Nugroho, Widyawan, and Kurnianingsih, “Fall Detection System Using Accelerometer and Gyroscope Based on Smartphone,” in The 1st International Conference on Information Technology, Computer and Electrical Engineering (ICITACEE), Semarang, Indonesia, 2014, pp. 99–104. Available: https://ieeexplore-ieee-org.proxy-bc.researchport.umd.edu/document/7065722

[5] L. Malott, P. Bharti, N. Hilbert, G. Gopalakrishna and S. Chellappan, "Detecting self-harming activities with wearable devices," 2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops), St. Louis, MO, 2015, pp. 597-602.Available: http://ieeexplore.ieee.org.proxy-bc.researchport.umd.edu/stamp/stamp.jsp?tp=&arnumber=7134105&isnumber=7133953

[6] T.Rahman, M.Czerwinski,and R.Gilad-Bachrach,"Predicting “About-to-Eat” Moments for Just-in-Time Eating Intervention" 2016 ACM ,Montréal,QC,Canada. Available: https://userpages.umbc.edu/~nroy/courses/shhasp18/papers/abouttoeat_DH16.pdf

[7] Dzung Tri Nguyen, E. Cohen, M. Pourhomayoun and N. Alshurafa, "SwallowNet: Recurrent neural network detects and characterizes eating patterns," 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Kona, HI, 2017, pp. 401-406.Available: http://ieeexplore.ieee.org.proxy-

bc.researchport.umd.edu/stamp/stamp.jsp?tp=&arnumber=7134105&isnumber=7133953

[8] A. Faridee, S. Ramamurthy, H. M. S. Hossain, and N. Roy, “Happy Feet: Recognizing and Assessing Dance on the Floor,”2018 HotMobile, Tempe, AZ, USA. Available: http://mpsc.umbc.edu/wp-content/uploads/2018/05/HotMobile18.pdf

[9] H. M. S. Hossain, M. A. A. H. Khan and N. Roy, "SoccerMate: A personal soccer attribute profiler using wearables," 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Kona, HI, 2017, pp. 164-169. Available: http://ieeexplore.ieee.org.proxy-bc.researchport.umd.edu/stamp/stamp.jsp?tp=&arnumber=7917551&isnumber=7917470

[10] R. Goodrich, “Accelerometer vs. Gyroscope: What's the Difference?,” *LiveScience,* 30-May-2018. [Online]. Available: https://www.livescience.com/40103-accelerometer-vs-gyroscope.html.

[11] K. Naik, “Principle Component Analysis (PCA) using sklearn and python,” YouTube, 02-Jul-2018. [Online]. Available: https://www.youtube.com/watch?v=QdBy02ExhGI.

[12] R. Gandhi, “Introduction to Machine Learning Algorithms: Linear Regression,” *Towards Data Science*, 27-May-2018. [Online]. Available: https://towardsdatascience.com/introduction-to-machine-learning-algorithms-linear-regression-14c4e325882a.

[13] J. Brownlee, “K-Nearest Neighbors for Machine Learning,” *Machine Learning Mastery*, 15-Apr-2016. [Online]. Available: https://machinelearningmastery.com/k-nearest-neighbors-for-machine-learning/.

[14] D. Soni, “Introduction to k-Nearest-Neighbors - Towards Data Science,” *Medium*, 12-Mar-2018. [Online]. Available: https://towardsdatascience.com/introduction-to-k-nearest-neighbors-3b534bb11d26.

[15] S. Patel, “Chapter 5: Random Forest Classifier - Machine Learning 101,” *Medium*, 18-May-2017. [Online]. Available: https://medium.com/machine-learning-101/chapter-5-random-forest-classifier-56dc7425c3e1.

[16] J. Brownlee, “A Gentle Introduction to the Gradient Boosting Algorithm for Machine Learning,” *Machine Learning Mastery*, 09-Sep-2016. [Online]. Available: https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/.

[17] Y. Shaikh, V. K. Parvati and S. R. Biradar, "Survey of Smart Healthcare Systems using Internet of Things (IoT): (Invited Paper)," 2018 International Conference on

Communication, Computing and Internet of Things (IC3IoT), Chennai, India, 2018, pp. 508-513. Available: http://ieeexplore.ieee.org.proxy-bc.researchport.umd.edu/stamp/stamp.jsp?tp=&arnumber=8668128&isnumber=8668098

[18] Kristijan, “Emerging Technologies: The Internet of Things,” TFOT. [Online]. Available: https://thefutureofthings.com/13229-emerging-technologies-the-internet-of-things/.

[19] X.Su, H.Tong, P.JI, "Activity Recognition with Smartphone Sensors,” 2014 TSINGHUASCIENCEANDTECHNOLOGY pp.235-249 Volume 19, Number 3.Available: http://hixing.weebly.com/uploads/5/2/5/3/52538413/06838194.pdf

[20] “tsfresh,” tsfresh. [Online]. Available: https://tsfresh.readthedocs.io/en/latest/. [Accessed: 30-Jul-2019].