Bento Packing Activity Recognition from Human Motion Gesture using Machine Learning

Submitted By

Shohanur Rahman

ID: 181-35-280
Department of Software Engineering
Daffodil International University

Supervised by:

Mr. Shariful Islam
Lecturer
Department of Software Engineering
Daffodil International University

This Project report has been submitted in fulfillment of the requirements for the Degree of Bachelor of Science in Software Engineering.

APPROVAL

This thesis entitled on ***"Bento Packing Activity Recognition from Human Motion Gesture using Machine Learning"***, submitted by ***Shohanur Rahman*** (ID: 181-35-280) to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS

Chairman

Dr. Imran Mahmud
Associate Professor and Head
Department of Software Engineering
Daffodil International University

-- External Examiner

Abu Shamim Aminur Razzaque
Director
Computer Ease Limited

-- Internal Examiner 1

SK. Fazlee Rabby
Lecturer
Department of Software Engineering
Daffodil International University

-- Internal Examiner 2

Md. Rajib Mia
Lecturer
Department of Software Engineering
Daffodil International University

THESIS DECLARATION

I hereby declare that, this thesis has been done by me under the supervisor of **Mr. Shariful Islam**, Lecturer, Department of Software Engineering at Daffodil International University. I also declare that neither this thesis nor any part of this thesis has been submitted elsewhere for award of any degree.

……………………………………
Shohanur Rahman
ID: 181-35-280
Department of Software Engineering
Faculty of Science & Information Technology
Daffodil International University

Supervised by:

………………………………………
Mr. Shariful Islam
Lecturer
Department of Software Engineering
Faculty of Science & Information Technology
Daffodil International University

ACKNOWLEDGEMENT

The key focus of this thesis was on developing models that could be utilized to detect packing activity. Without the aid and support of a large number of individuals, it would not have been achievable.

I am very grateful to Daffodil International University for its guidance and constant monitoring by Shariful Islam sir, such as for providing me with the knowledge and help that is necessary to finish this study.

For additional guidance, I would like to express my appreciation to our honorable teachers MD Rajib Mia sir and Ms. Syeda Sumbul Hossain Maam.

I like to offer my respect to my parents, batch mates, and DIU members for their helpful cooperation and sympathy.

Shohanur Rahamn
ID:181-35-280

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BPARC = Bento Packing Activity Recognition Challenge;

AR= Activity recognition;

ML = Machine Learning;

AI = Artificial intelligence;

IoT = Internet of Things:

RF = Random Forest;

SVM = Support-vector;

LR = Logistic Regression;

RNNs = Recurrent Neural Networks;

Mo-cap = Motion capture;

ABSTRACT

The difficulty of identifying a body's behavior based on sensor data is known as activity recognition. It's among the most widely studied topics in the field of machine learning-based classification. Bento Packing Activity Recognition Challenge (BPARC) asked participants to recognize bento packing preparation using motion capture sensors. Thirteen motion capture body markers were used to capture the BPARC dataset. The markers were three axes (x, y, and z) with a total of thirty-nine axes. One of the most challenging difficulties to solve in this investigation was identifying complicated tasks as smaller activities that are part of larger activities. Using Random Forest, Support Vector Machine, and Logistic Regression, we've built a machine learning approach that extracts dynamical data for bento packing activity identification. The model we proposed for that kind of dataset has a classification accuracy of 76%, 62% and 63% for RF, SVM and LR, respectively.

CHAPTER 1: INTRODUCTION

Recognizing human activity is a popular research area nowadays. Because the sensors and hardware are very cheap and machine learning, deep learning, and IoT are improving so fast as well.

Machine learning (ML) is a popular approach for recognizing human activity [1]. Machine learning needs many extractions of features, which may take a long time [2]. But for classification problems, some machine learning algorithms work well and give higher accuracy. Deep learning (DL) algorithms give higher accuracy for image recognition, object classification, etc. [3] But deep learning requires a large amount of data in order to perform well with the model.

IoT and AI are combining to provide services and applications for individualized health care and treatment [4]. Monitoring workers/employees are one of these services that have gotten a lot of attention because it influenced the company's production and revenue. Those services keep track of what the individual is doing and send out alerts in case of an emergency.

Human-robot collaboration has been strongly influenced by HAR [5]. Human labor at a modest wage has become much more difficult to come by as people's lifestyles have improved. This is a major issue for a variety of companies that demand a large number of workers at a cheap wage. If robots can be employed to aid people with simple jobs, they might be a perfect answer to this problem. To do so, the robot must first comprehend the human's actions! Data on Bento Packaging Activity Recognition has been collected with this in mind. In the case of an accident when packing, smart gadgets will send alarms to an emergency station. It helps in decision-making, such as which job a specific individual should accomplish for bento packing to complete the meal preparation process.

The entire paper is arranged in different chapters. In Chapter 2, the related works are described in order to extract the basic things founded in past research as well as in this framework. In Chapter 3, the dataset that is used for training and testing the model's accuracy is described and also discussed about the problems of this dataset with the pre-processing flow for overcoming those problems. The proposed approach, as well as the characteristics of the models that we created, are described in Chapter 4. With the help of multiple graphs and tables, we clarified the results and ultimate findings in Chapter 5. All of the findings and model efficiency are detailed in this section. In Chapter 6, a summary of the entire study is included as a conclusion.

CHAPTER 2: RELATED WORK

Numerous investigations have been conducted regarding the activity's recognition. Hossain et al. [6] proposed a traditional machine learning method using handcrafted features from the dataset. However, handcrafted features may not perform well with time-series data [7]. In addition, traditional machine learning algorithms are not suitable for recognizing complex activities with missing data [8].

Yoon et al. [9] proposed a multi-directional RNN that reduced missing data errors and increased the performance of the model by 35%. Swapnil et al. [10] suggested a deep convolutional network pipeline for detecting complicated activities with missing data handling. While the pipeline improved at larger processes, it did not significantly improve the classification of smaller tasks. In addition, training the pipeline necessitates a large amount of data.

Atsuhiro et al. [11] introduced a convolutional LSTM that calculates loss using several loss functions. The pipeline's ultimate categorization was determined by a majority vote. Even though the model used a qualified majority approach, loss functions occasionally reduced the true output. As a result, the voting mechanism forecasted incorrect activity.

Picard et al. [12] devised a system that used just mo-cap data to identify cooking actions. With training data, the model scored 95%, however, the validation score was much lower than the training score. Ankur A. [13] built a machine learning pipeline which performs well with 3D motion capture data but does not perform well with multiple axes motion capture sensors.

Various studies have been done concerning cooking and packing activity with some technical glitches in the above-related works. So, there is an adequate opportunity to design a model to properly identify cooking and packing activity.

In this investigation, connectivity has been created with a Random Forest, SVM and Logistic Regression model for sensors in each body part termed motion capture sensor. As an input to the model, features have been separated into a 100-second window. To polish missing data and handle varying sample rates, a controlled overlapping function has been utilized. Zeros have been used to fill in null values. Each model has been discussed in detail and compared to the others.

CHAPTER 3: DATASET DESCRIPTION

For doing this study, we downloaded data from the ABC Conference's Bento Packing Activity Recognition Challenge – 2021. The bento packing dataset has a total of ten activities, containing five inward and five outward activities. The activities are:

- Normal (inward & outward)
- Forgot to put ingredients (inward & outward)
- Failed to put ingredients (inward & outward)
- Turn over bento-box and (inward & outward)
- Fix/rearranging ingredients (inward & outward)

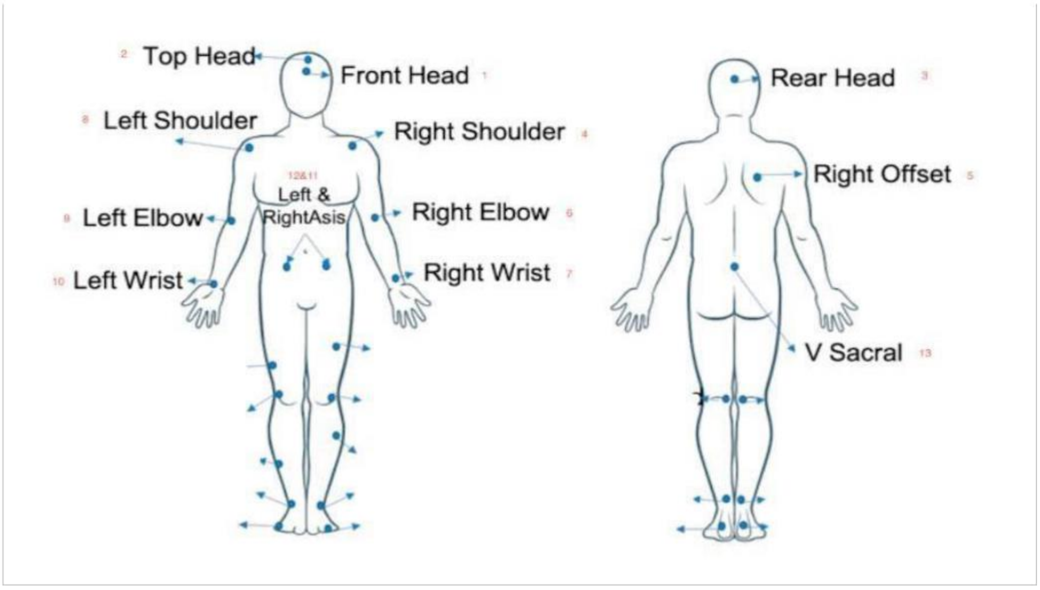

Figure 1: The body markers were implanted in the places shown in this picture. This picture depicts all thirteen body markers.

Only numerical values are included in the bento packing dataset. All of the data was collected using motion capture sensors. Four participants worked to record motion capture data and were informed how to fill the bento box using three different types of food. Participants were instructed to pack the boxes as naturally as possible, and they were given a script to follow in order to properly replicate all of the packing phases.

Twenty infrared cameras were utilized to track the markers and thirty body markers were placed to capture the data. Each marker was measured in three dimensions (x, y, and z). A 50-70s window was used to segment sensor data. Each activity was repeated five times by the participants. The dataset had 200 .csv files in total. Each .csv file must contain at least one subject and one activity. The sampling rate for motion capture data was taken as 100 Hz.

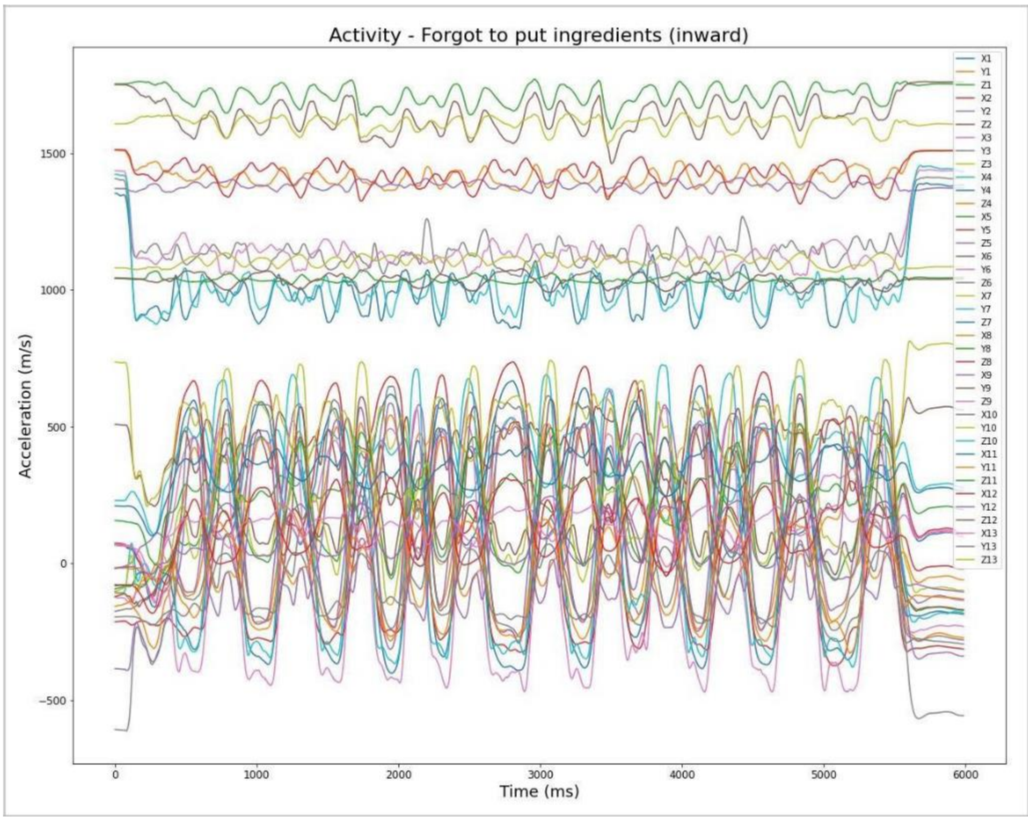

Figure 2: The training data file for the activity 'Forgot to put ingredients (inward)' is represented by the graph. This graph illustrates how exact motion capture data from thirteen body markers were displayed inside the dataset, presenting all three axes (x, y, and z) related to the sensor's time and acceleration.

The dataset we investigated had two main flaws. The first was an out-of-synchronized data sequence, while the second was null values in data files. Furthermore, some activities looked to be quite similar to others, making it difficult to distinguish between them.

3.1 Data Pre-processing

As previously stated, this dataset contains several significant flaws. In order to overcome those issues and get the maximum performance out of the model, some pre-processing has been conducted. First, the overlap rate to step for the sliding window has been transformed to solve the data overlap problem. The data were segmented using a 100-window rate and 0% overlap.

The dataset contains 18774 null values, as previously stated. Zeros were used to substitute null values. For training the model, manual label encoding was performed to appropriately encode the activities.

Activity Name	Label Number (Training data)
Normal(inward)	1
Normal(outward)	2
Forgot to put ingredients (inward)	3
Forgot to put ingredients (outward)	4
Failed to put ingredients(inward)	5
Failed to put ingredients(outward)	6
Turn over bento-box (inward)	7
Turn over bento-box (outward)	8
Fix/rearranging ingredients (inward)	9
Fix/rearranging ingredients (outward)	10

Table 1: The labeling of the activities is shown in this table. There were two types of activities: inward and outward.

Some features from the original dataset were extracted to train the model. Standard division, average, maximum, and minimum was the extracted features. The model's activity id column was also excluded for better performance. The dataset was partitioned into two 70:30 pieces for training and testing. Python 3.9.0 was used for all of the pre-processing and feature extraction.

CHAPTER 4: METHODOLOGY

In this investigation, three machine learning architectures were virtually applied to the training process. Random Forest, Support Vector Machine (SVM) and Logistic Regression.

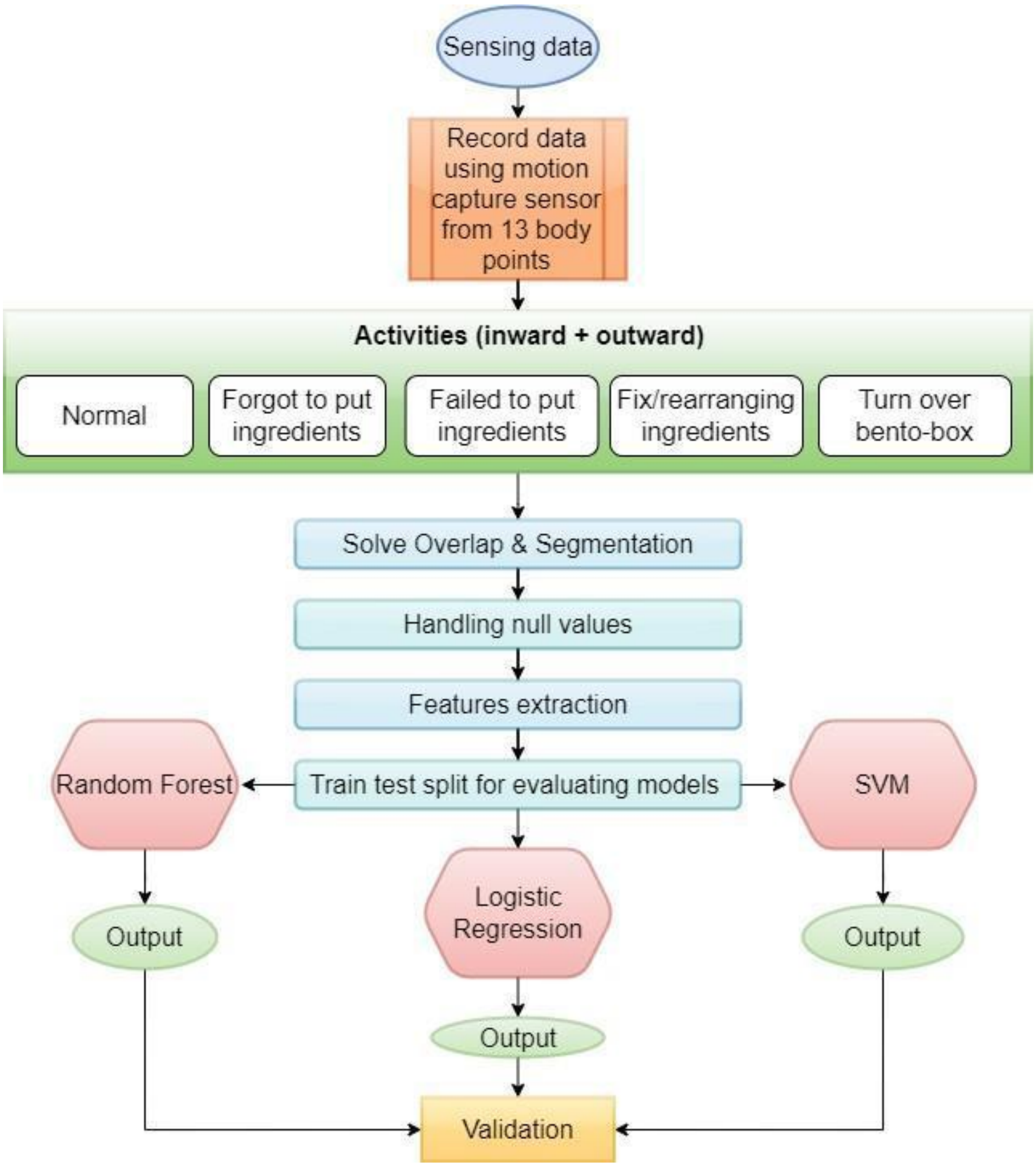

Figure 3: The proposed methodology's workflow is illustrated in this diagram.

As illustrated in Fig. 3, the following technique is based on three models (Random Forest, SVM, and Logistic Regression). The results of these three models were drastically different, despite the fact that their designs were comparable. The models were trained and evaluated using motion capture data from thirteen body locations.

There were two sections to the activities. After separating the dataset into two pieces, there were 10 actions in total (inward and outward). A data overlapping function was utilized to fix overlapping issues in the data flow. The Python programming language was used to create the function. Zeros have been used to fill in any missing or null values. The data folder contains some noisy results due to a sensor malfunction during recording. As a consequence, several characteristics from raw data have been extracted in order to get the most performance out of the models.

While feeding the model, all data must be in numerical format. The dataset's labels, on the other hand, were in text format. Encoding techniques were used to transform text labels to numerical labels. Label encoding was used to encode the data. The data was divided into 70:30 ratios for train and test and segmented into 100 second windows. The models that were used for training and verifying accuracies are detailed in the next section.

4.1 Random Forest

Random Forest has been divided into two stages. The first was to construct a randomized forest by mixing a set of decision trees. The second step was to make forecasts for each of the trees that had been built in the previous stage [14]. In this proposed model, the needed number of trees (n_estimators) was 500. In this framework, the entropy criteria were applied. The engine can employ an infinite number of processors because n_jobs is set to -1 in this model.

```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=None, max_features='auto',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=500,
                       n_jobs=-1, oob_score=False, random_state=None, verbose=0,
                       warm_start=False)
```

Figure 4: The design of the Random Forest model, which has been used to train bento packing.

4.2 Support Vector Machine (SVM)

Both classification and regression tasks were solved using the SVM classification technique [15]. SVM uses a strategy to transform data, and the model then selects the best potential outputs depending on this modification [16]. Kernel trick was the name for this approach.

```
Out[8]:
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
    kernel='linear', max_iter=-1, probability=False, random_state=0,
    shrinking=True, tol=0.001, verbose=False)
```

Figure 5: The Support Vector Machine (SVM) model's architecture, which was used to train bento packing actions.

SVM with a linear kernel has been employed in this study. The SVM had a cache size of 200 and a random state of zero. With default probability, auto gamma (False) was used to train the model. The polynomial kernel method's degree was enabled by default.

4.3 Logistic Regression

Based on independent factors, logistic regression was used to determine categorical factors [17]. Using uninterrupted datasets, the algorithm can solve odds and authenticate new information.

A conventional scalar function was employed in this experiment. A pipeline for training the Logistic Regression model has been constructed using the standard scalar. The standard scaler scales the data before it is fed into the model, allowing the model to perform better. The model's numerical input values were scaled by the pipeline.

CHAPTER 5: RESULT ANALYSIS

Random Forest, SVM and Logistic Regression algorithms were used to compile the motion capture data from the original dataset. The findings were examined from thirteen sensor data points: front head, top head, rear head, right shoulder, right offset, right elbow, right wrist, left shoulder, left elbow, left wrist, left asis, right asis and v sacral, all at the same time, to assist for inward and outward activity categorization. The files were broken into two portions for training the architectures (train: test/validation). 70% of the data was used for training and the remaining 30% was used for testing and validation.

To see if the trained model was overfitted, the acceleration data was checked using the cross-validation (CV) method. The train data has been organized in a random format (RF) in this CV strategy. This RF approach has worked on the principle of separating particular samples before splitting them into frames for model training. As a result, when the model was trained, the accuracy of the original samples for prediction was maintained. It has been extremely evident how well the trained models could predict and evaluate the accuracy for all activity recognition individually using this method of cross-validation.

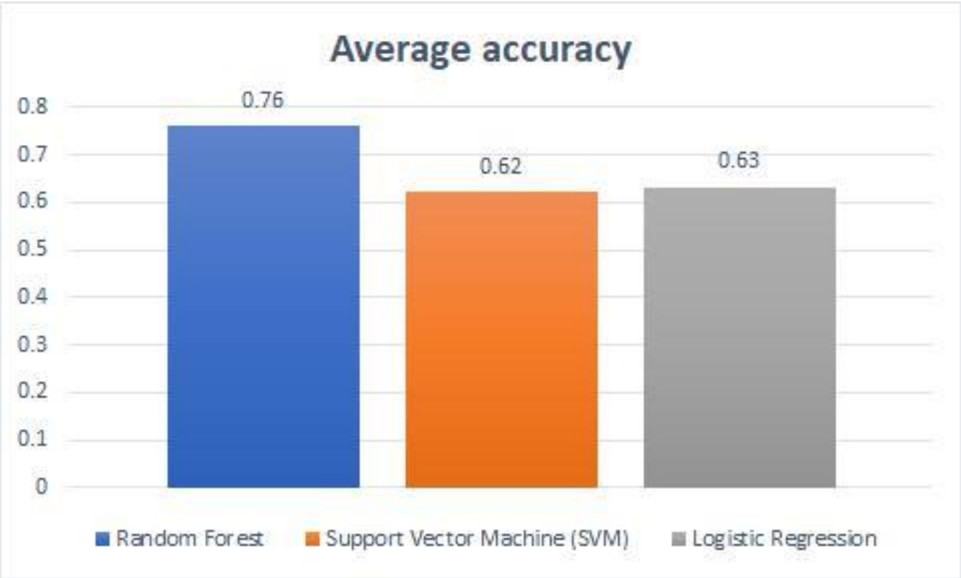

Figure 6: In this graph, the average accuracy of all three models are compared. As indicated inside the graph, Random Forest validation accuracy is 14% greater than SVM validation and 13% higher than Logistic Regression validation.

500 decision trees were utilized to train random forest models. Because of the first and infinite processors, the value of n jobs was -1. In addition, the SVM model was utilized to train the model as a linear kernel.

The python language was utilized to train the three models. Keras, sklearn, and TensorFlow were some of the libraries included. The window was 100s in size, with a 2.40 GHz Intel 5-core CPU, 8GB of RAM, and a GPU of the NVIDIA GeForce 940MX.

5.1 Random Forest

Random Forest was the first model used in this investigation. For the model, a total of 500 trees were constructed. After feature extraction, 70% of the data from the dataset was utilized to train the model. 30% of the data was utilized for testing and validation once the model was trained. The random forest has an average accuracy of 76 %. 0.77 was the average f1 score.

	precision	recall	f1-score	support
1	0.79	0.77	0.78	161
2	0.79	0.74	0.76	172
3	0.74	0.69	0.71	163
4	0.80	0.69	0.74	163
5	0.79	0.80	0.80	140
6	0.82	0.88	0.85	156
7	0.71	0.83	0.77	164
8	0.79	0.75	0.77	173
9	0.69	0.80	0.74	158
10	0.85	0.80	0.83	180
accuracy			0.77	1630
macro avg	0.78	0.78	0.77	1630
weighted avg	0.78	0.77	0.77	1630

Table *2*: This table shows the Random Forest model's output report for each activity. The best accuracy is achieved by activity 10 (fix/rearranging ingredients - outward), while the lowest accuracy is achieved by activity 9 (fix/rearranging ingredients – inward).

The model accurately predicted Label 10 (fix/rearranging ingredients - outward) 144 times, giving it the highest prediction score. Furthermore, both labels 3 (forgot to put ingredients - inward) and 5 (failed to put ingredients - inward) were properly predicted 112 times, resulting in the model's lowest score prediction. The model shows that the Random Forest performs better for outward actions based on this research.

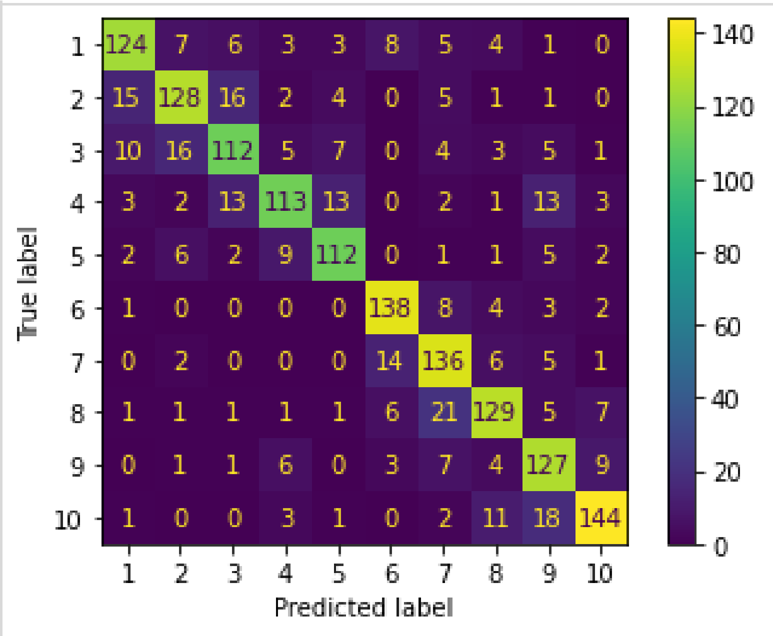

Figure 7: The confusion matrix for all 10 activities in the Random Forest model. The X-axis represents the actual labels, while the Y-axis represents the expected labels. Furthermore, 1 stand for "normal (inward)", 2 stands for "normal (outward)", 3 stands for "forgot to put ingredients (inward)", 4 stands for "forgot to put ingredients (outward)", 5 stands for "failed to put ingredients (inward)", 6 stands for "failed to put ingredients (outward)", 7 stands for "turn over bento-box (inward)", 8 stands for "turn over bento-box (outward)", 9 stands for "fix/rearranging ingredients (inward)", 10 stands for "fix/rearranging ingredients (outward)".

5.2 Support Vector Machine (SVM)

The second model used in this investigation was the Support Vector Machine (SVM). The model was trained using a linear kernel. After feature extraction, 70% of the data from the dataset was utilized to train the model. 30% of the data was utilized for testing and validation once the model was trained. The SVM has an average accuracy of 62 %. 0.63 was the average f1 score.

	precision	recall	f1-score	support
1	0.61	0.66	0.64	161
2	0.54	0.62	0.58	172
3	0.54	0.53	0.54	163
4	0.65	0.61	0.63	163
5	0.75	0.66	0.70	140
6	0.69	0.83	0.76	156
7	0.59	0.60	0.59	164
8	0.58	0.59	0.58	173
9	0.66	0.57	0.61	158
10	0.72	0.63	0.67	180
accuracy			0.63	1630
macro avg	0.63	0.63	0.63	1630
weighted avg	0.63	0.63	0.63	1630

Table 3: The output report of each activity for the Support Vector Machine (SVM) model is shown in this table. The best accuracy is achieved by activity 5 (failed to put ingredients – inward), while the lowest accuracy is achieved by activities 2 (normal – outward) and 3 (forgot to put ingredients – inward).

The algorithm accurately predicted Label 6 (Failed to put ingredients - outward) 130 times, obtaining the model's highest prediction score. Label 3 (forgot to put ingredients - inward) on the other hand, which has the lowest score prediction from the model, was accurately predicted 87 times. The model shows that the SVM performs better for outward actions based on this data. The validation accuracy of SVM was lower than that of the Random Forest technique.

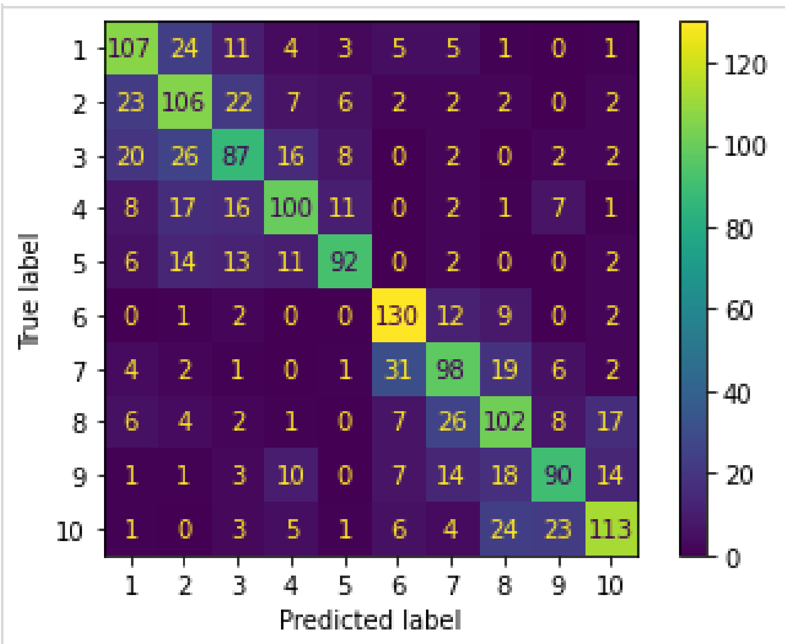

Figure 8: This is the Support Vector Machine (SVM) model's confusion matrix for all ten activities. The actual labels are shown on the X-axis, while the anticipated labels are shown on the Y-axis.

5.3 Logistic Regression

The last model used in this investigation was Logistic Regression. The model was trained using a standard scaler pipeline. After feature extraction, 70% of the data from the dataset was utilized to train the model. 30% of the data was utilized for testing and validation once the model was trained. The random forest has an average accuracy of 63 percent. 0.63 was the average f1 score. Which performed somewhat better than the SVM algorithm but not as well as the Random Forest approach.

	precision	recall	f1-score	support
1	0.67	0.64	0.65	161
2	0.61	0.56	0.59	172
3	0.56	0.52	0.54	163
4	0.61	0.61	0.61	163
5	0.70	0.76	0.73	140
6	0.69	0.78	0.73	156
7	0.61	0.59	0.60	164
8	0.58	0.56	0.57	173
9	0.62	0.61	0.62	158
10	0.65	0.69	0.67	180
accuracy			0.63	1630
macro avg	0.63	0.63	0.63	1630
weighted avg	0.63	0.63	0.63	1630

Table 4: The output report for each activity for the Logistic Regression model is shown in this table. The best accuracy is achieved by activity 5 (failed to add ingredients – inward), while the lowest accuracy is achieved by activity 3 (forgot to put ingredients – inward).

The model accurately predicted Label 10 (fix/rearranging ingredients - outward) 125 times, giving it the highest prediction score. Label 3 (forgot to add ingredients - inward) on the other hand, which has the lowest score prediction from the model, was predicted 85 times accurately. The model shows that the Logistic Regression delivers the best results for outward actions based on this investigation. In addition, the model outperforms SVM but not the Random Forest algorithm.

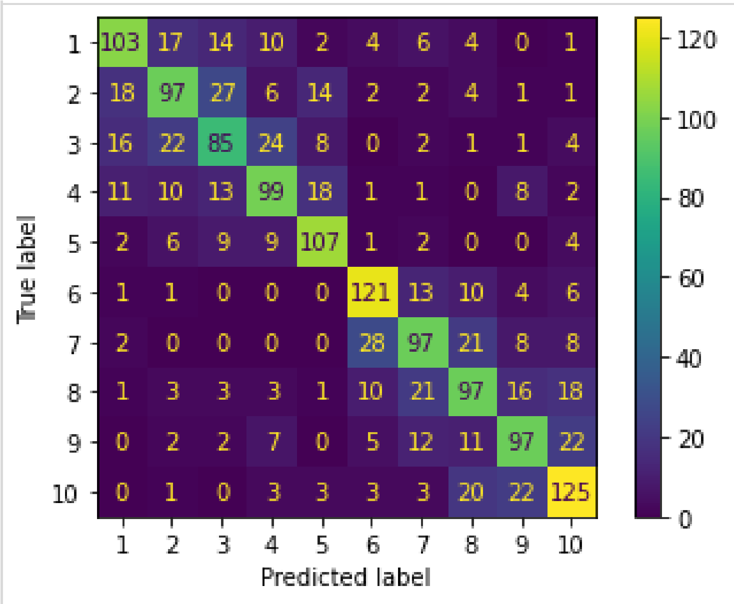

Figure 9: This is the confusion matrix for all 10 tasks in the Logistic Regression model. The X-axis reflects the actual labels, while the Y-axis illustrates the predicted labels. Furthermore, 1 denotes "normal (inward)", 2 denotes "normal (outward)", 3 denotes "forgot to put ingredients (inward)", 4 denotes "forgot to put ingredients (outward)", 5 denotes "failed to put ingredients (inward)", 6 denotes "failed to put ingredients (outward)", 7 denotes "turn over bento-box (inward)", 8 denotes "turn over bento-box (outward), 8 denotes “turn over bento-box (outward)”, 9 denotes “fix/rearranging ingredients (inward)”, 10 denotes “fix/rearranging ingredients (outward)”.

CHAPTER 6: CONCLUSION

In this investigation, we showed how we used sophisticated machine learning approaches to classify bento packing activities using motion capture sensor data. We ran across a lot of issues at the time of training the models. In addition, the pre-processing stage for preparing training data was very difficult. They've previously been mentioned in the section on Dataset Descriptions. We overcame those challenges and attained an average accuracy of 76%, 62% and 63% from Random Forest, Support Vector Machine (SVM) and Logistic Regression, respectively. In the future, these strategies will aid in the resolution of difficult activity difficulties. Additionally, these models will operate well with time-series datasets.

REFERENCES

1. Biswal, A., Nanda, S., Panigrahi, C.R., Cowlessur, S.K. and Pati, B., 2021. Human Activity Recognition Using Machine Learning: A Review. Progress in Advanced Computing and Intelligent Engineering, pp.323-333.

2. Khalid, S., Khalil, T. and Nasreen, S., 2014, August. A survey of feature selection and feature extraction techniques in machine learning. In 2014 science and information conference (pp. 372-378). IEEE.

3. LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. *nature*, *521*(7553), pp.436-444.

4. Alshamrani, M., 2021. IoT and Artificial Intelligence Implementations for Remote Healthcare Monitoring Systems: A Survey. Journal of King Saud University-Computer and Information Sciences.

5. Thomas, C., Matthias, B. and Kuhlenkötter, B., 2016, June. Human-robot collaboration–new applications in industrial robotics. In International Conference on Competitive Manufacturing (pp. 293-299).

6. Hossain, T. and Inoue, S., 2019, May. A Comparative study on missing data handling using machine learning for human activity recognition. In *2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)* (pp. 124-129). IEEE.

7. Liu, R., Meng, G., Yang, B., Sun, C. and Chen, X., 2016. Dislocated time series convolutional neural architecture: An intelligent fault diagnosis approach for electric machine. *IEEE Transactions on Industrial Informatics*, *13*(3), pp.1310-1320.

8. Emmanuel, T., Maupong, T., Mpoeleng, D., Semong, T., Banyatsang, M. and Tabona, O., 2021. A Survey On Missing Data in Machine Learning.

9. Yoon, J., Zame, W.R. and van der Schaar, M., 2018. Estimating missing data in temporal data streams using multi-directional recurrent neural networks. *IEEE Transactions on Biomedical Engineering*, *66*(5), pp.1477-1490.

10. Saha, S.S., Sandha, S.S. and Srivastava, M., 2021. Deep Convolutional Bidirectional LSTM for Complex Activity Recognition with Missing Data. In Human Activity Recognition Challenge (pp. 39-53). Springer, Singapore.

11. Zhu, W., Lan, C., Xing, J., Zeng, W., Li, Y., Shen, L. and Xie, X., 2016, March. Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 30, No. 1).

12. Schmidt, J., Marques, M.R., Botti, S. and Marques, M.A., 2019. Recent advances and applications of machine learning in solid-state materials science. npj Computational Materials, 5(1), pp.1-36.

13. Agarwal, A., 2006. Machine learning for image-based motion capture (Doctoral dissertation, Institut National Polytechnique de Grenoble-INPG).

14. Hartshorn, S., 2016. Machine learning with random forests and decision trees. Kindle edition.

15. Kwok, J.T.Y., 1998, August. Support vector mixture for classification and regression problems. In Proceedings. Fourteenth International Conference on Pattern Recognition (Cat. No. 98EX170) (Vol. 1, pp. 255-258). IEEE.

16. Suthaharan, S., 2016. Support vector machine. In Machine learning models and algorithms for big data classification (pp. 207-235). Springer, Boston, MA.

17. Bujang, M.A., Sa'at, N. and Bakar, T.M.I.T.A., 2018. Sample size guidelines for logistic regression from observational studies with large population: emphasis on the accuracy between statistics and parameters based on real life clinical data. The Malaysian journal of medical sciences: MJMS, 25(4), p.122.