**CHAPTER 1**

**INTRODUCTION**

**1.1 OVERVIEW**

The explosion of mobility nowadays is setting a new standard for information technology industry. Mobile devices such as smart-phones and tablets more and more become popular, and hence, making people increasingly depend on them for their superior functionality. Such devices are commonly used for storage and retrieval of information like e-commerce and m-banking. However, they can be easily lost, stolen, or illegally accessed . That means sensitive or/and important information of users could be retrieved unexpectedly. Consequently, identification has evolved to become a more priority issue for developers. Currently, the most common methods are PIN and passwords which are not always effective considering security aspects. These limitations can be solved using approaches based on biometric such as face recognition , fingerprint etc.

However, as these methods require explicit action from the users, they are obtrusive and inconvenient in frequent use. Thus, a more friendly mechanism of identification is desired to be developed and aim to set a new standard in mobile security. Human gait has been introduced as a particular style and manner of moving human feet. In a more detail level view, the mechanism of human gait involves synchronization between the skeletal, neurological and muscular system of human body .

Therefore, gait characteristics will vary from people to people. Gait recognition has been studied as a behavioural biometric for decades. Its techniques could be typically divided

into 3 categories: Machine Vision Technology (WVT) , Floor Sensor Technology (FST) , and Wearable Sensor Technology (WST) . WST is recognized as the most approachable and newest of all. Sensors in WST are attached to human body in various positions, such as pockets, waist or shoes to record physical motions. WST takes advantage of mobile devices sensing capabilities including GPS, accelerometer, and gyroscope sensor, etc. Thus, it will provide developers an edge over improving various techniques in identification. In this paper, we propose identification method based on WST using an integrated accelerometer on mobile phone. Moreover, because segmentation of gait cycles is the most important process in any gait analysis, we also provide a novel algorithm to partition gait cycles when the device is placed at trouser pocket.

This project aims to authenticate people based on gait - the way of one's walking, data is collected from android operated smart-phone through accelerometer and upon evaluating the dataset on our machine learning model at azure Machine Learning, we can identify the person through his/her way of walking. Can be used in High Security Arenas or to uniquely identify a personal for any task viz. Tracking Activity, Assigning a unique ID to each person?

**1.2 MACHINE VISION TECHNOLOGY**

**Machine vision** (MV) is the technology and methods used to provide imaging-based automatic inspection and analysis for such applications as automatic inspection, [process control](https://en.wikipedia.org/wiki/Process_control), and robot guidance, usually in industry. Machine vision is a term encompassing a large number of technologies, software and hardware products, integrated systems, actions, methods and expertise. Machine vision as a [systems engineering](https://en.wikipedia.org/wiki/Systems_engineering) discipline can be considered distinct from [computer vision](https://en.wikipedia.org/wiki/Computer_vision), a form of basic [computer science](https://en.wikipedia.org/wiki/Computer_science). It attempts to integrate existing technologies in new ways and apply them to solve real world problems. The term is also used in a broader sense by trade shows and trade groups; this broader definition also encompasses products and applications most often associated with image processing.

**1.3 FLOOR SENSOR TECHNOLOGY**

This describes the development of a prototype floor sensor as a gait recognition system. This could eventually find deployment as a standalone system (e.g. a burglar alarm system) or as part of a multimodal biometric system. The new sensor consists of 1536 individual sensors arranged in a 3 m by 0.5 m rectangular strip with an individual sensor area of 3 cm/sup 2/. The sensor floor operates at a sample rate of 22 Hz. The sensor itself uses a simple design inspired by computer keyboards and is made from low cost, off the shelf materials. Application of the sensor floor to a small database of 15 individuals was performed. Three features were extracted: stride length, stride cadence, and time on toe to time on heel ratio. Two of these measures have been used in video based gait recognition while the third is new to this analysis. These features proved sufficient to achieve an 80% recognition rate.

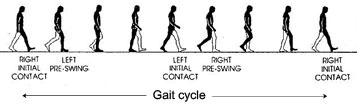
**1.4 WEARABLE SENSOR TECHNOLOGY**

Gait analysis using wearable sensors is an inexpensive, convenient, and efficient manner of providing useful information for multiple health-related applications. As a clinical tool applied in the rehabilitation and diagnosis of medical conditions and sport activities, gait analysis using wearable sensors shows great prospects. The current paper reviews available wearable sensors and ambulatory gait analysis methods based on the various wearable sensors.

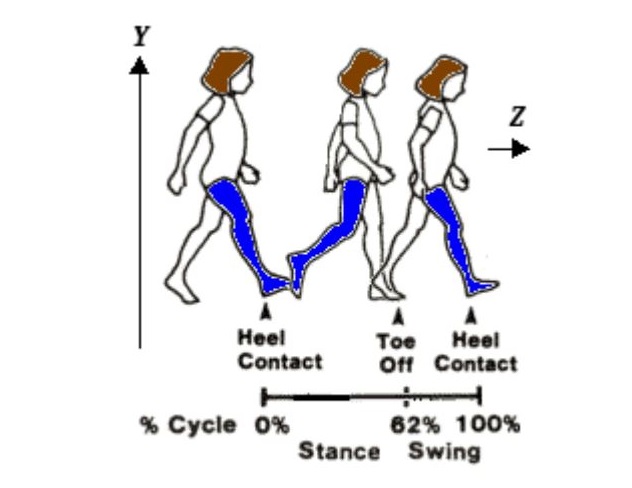
After an introduction of the gait phases, the principles and features of wearable sensors used in gait analysis are provided. The gait analysis methods based on wearable sensors is divided into gait kinematics, gait kinetics, and electromyography. Studies on the current methods are reviewed, and applications in sports, rehabilitation, and clinical diagnosis are summarized separately. With the development of sensor technology and the analysis method, gait analysis using wearable sensors is expected to play an increasingly important role in clinical applications.

**1.5 GAIT ANALYSIS**

Gait cycle is the time between successive foot contacts of the same limbs. Thus, one gait cycle begins when the reference foot contacts the ground and ends with subsequent floor contact of the same foot.



**Fig 1.1 Gait Cycle**

Human gait is periodic movement. The period within walking is called the gait cycle. A gait cycle is composed of one right leg step and one left leg step.Figure 1 shows the y and z directional forces. The x directional force is the left/right axis that goes straight out through

**Fig. 1.2 Gait Stance & Swing**

the figure. Even when walking in a straight line, a person exerts a slight left/right force. A novelty of this work is the use of a gyroscope to measure the rate of change of the angles about the three axes. The x gyro force is the rate of change aroundthe x axis; y gyro force about the y axis and z gyro force about the z axis. Although an accelerometer can be orientated in any way, the orientation throughout this work is exactly as described in Figure 1 and this paragraph. Together, the accelerometer and gyroscope produce six values at any one instant in time (hardware discussed later).

Each gait cycle contains the unique characteristics of a person's walk, and these unique attributes compose the gait signature. In order to robustly identify a person's gait signature, there must be a method for extracting gait cycles from a continuous signal of walking data.

* Step length is the distance between the heel contact point of one foot and that of the other foot.
* Stride length is the distance between the successive heel contact points of the same foot. Normally, stride length = 2 x step length

|  |  |
| --- | --- |
|  |  |

Fig. 1.3 Gait Parameters

Cadence is the rate at which a person walk, expressed in steps per minute. The average cadence is 100 - 115 steps/min. Thus, if you let your character take 10 steps in 156 to 180 frames (using 30 frames/sec), the character's cadence is within a normal range.

**1.6 PERIODICITY**

A gait cycle is the smallest repeating unit in the signal produced by the accelerometer/gyroscope. The goal is to take a signal of walking data, identify x periods from that signal, where x is proportional to time, and then extract the most important features of the period and store the feature vector as one example for the given test subject. Then, an algorithm trains on the test examples and predicts a person based on a given gait cycle. Given the orientation and axes, our accelerometer/gyroscope combination device should produce six data points for any single point in time.

**CHAPTER 2**

**DATA COLLECTION**

**2.1 COLLECTION METHOD**

People with age ranging from 18 to 40 (students and instructors) participated in volunteer data collection program, both male and female participated and logged the data in a whole span of week at various time of the day, walking with a constant pace at a smooth surface. A upload utility was provide to aggregate the data at a central repository.

The data is being collected with the help of Accelerometer Log android application available on Google Play Store with due permission. This application logs data from device’s accelerometer in 3 coordinates axes - x, y and z and stores them on device’s external storage. The format used for logging accelerometer data is comma separated values(CSV).The app starts logging upon launching and the device needs to be put in the following manner for effective logging of accelerometer data.

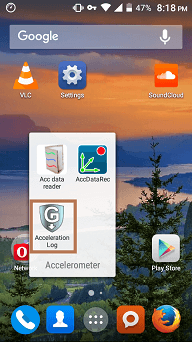


**Fig. 2.1 Position of device**

**2.2 STEPS FOR COLLECTION**

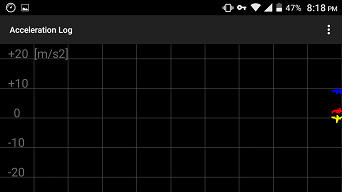
### On Android, Step to collect the data

### 1. Open the Accelerometer App



**Fig. 2.2 Accelerometer App**

**2.** Accelerometer Data will start display in the form of a graph



**Fig.2.3 Accelerometer Data Graph**

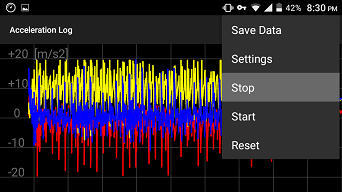
 3. Put your phone instantaneously in pocket in position described :

****

**Fig 2.4 Device Configuration**

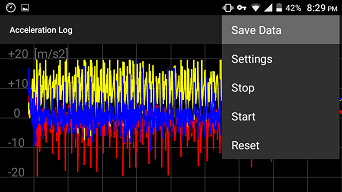
4. Start walking naturally - slow, normal or fast

5. After finishing walk touch options and tap Stop



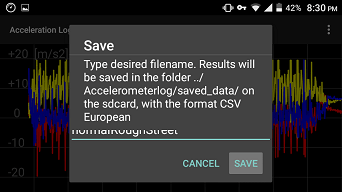
**Fig. 2.5 Stopping Logging**

6. After finishing walk touch options and tap save data



**Fig. 2.6 Saving Logged data**

7. Name the file in conventions [mentioned below](http://vkku.me/GBAS/#namingFiles)



**Fig.2.7 Naming file**

**2.3 NAMING CONVENTION**

Your Name initials - vkku for Vivek, vtsl for Vatsal and alike.

* Name your pace - factor of : Fast, Normal, Slow
* Name your walking surface - factor of : Smooth, Rough
* Optional(Recommended) : give a initial of place you recorded data at - example : MDA, MIT etc.

Now, your file name should look like:  vkkuNormalSmooth or vtslFastRoughMDA (Recommended).

**2.4 DATA LOGGING ALLEY**

The following figure depicts alley used for data logging, 60 M in one way distance. Each subject was told to take 08 return trips of alley.



**Fig.2.8 Walking Alley Fig.2.9 Subject completing trip**

**CHAPTER 3**

**AZURE MACHINE LEARNING PLATFORM**

**3.1 INTRODUCTION**

This project utilizes machine learning approach to identify subject upon execution of a classification model.

**3.1.1 Azure Machine Learning**

Azure Machine Learning is an interactive, visual workspace to easily build, test, and iterate on a predictive analysis model.**Datasets** and analysis **modules**, are connected to form an **experiment**, which you run in Machine Learning Studio. To iterate on model design, edit the experiment, and run it again. When ready, **training experiment** can be converted to a **predictive experiment**, and then can be published as a **web service** so that your model can be accessed by others.

**Fig 3.1 Work Flow of Azure Machine Learning**

**3.1.2 Definitions**

* PROJECTS - Collections of experiments, datasets, notebooks, and other resources representing a single project
* EXPERIMENTS - Experiments that you have created and run or saved as drafts
* WEB SERVICES - Web services that you have deployed from your experiments
* NOTEBOOKS - Jupyter notebooks that you have created
* DATASETS - Datasets that you have uploaded into Studio
* TRAINED MODELS - Models that you have trained in experiments and saved in Studio
* SETTINGS - A collection of settings that you can use to configure your account and resources.

**3.1.3** **Components of an experiment**

An experiment consists of datasets that provide data to analytical modules, which you connect together to construct a predictive analysis model. Specifically, a valid experiment has these characteristics:

* The experiment has at least one dataset and one module
* Datasets may be connected only to modules
* Modules may be connected to either datasets or other modules
* All input ports for modules must have some connection to the data flow
* All required parameters for each module must be set

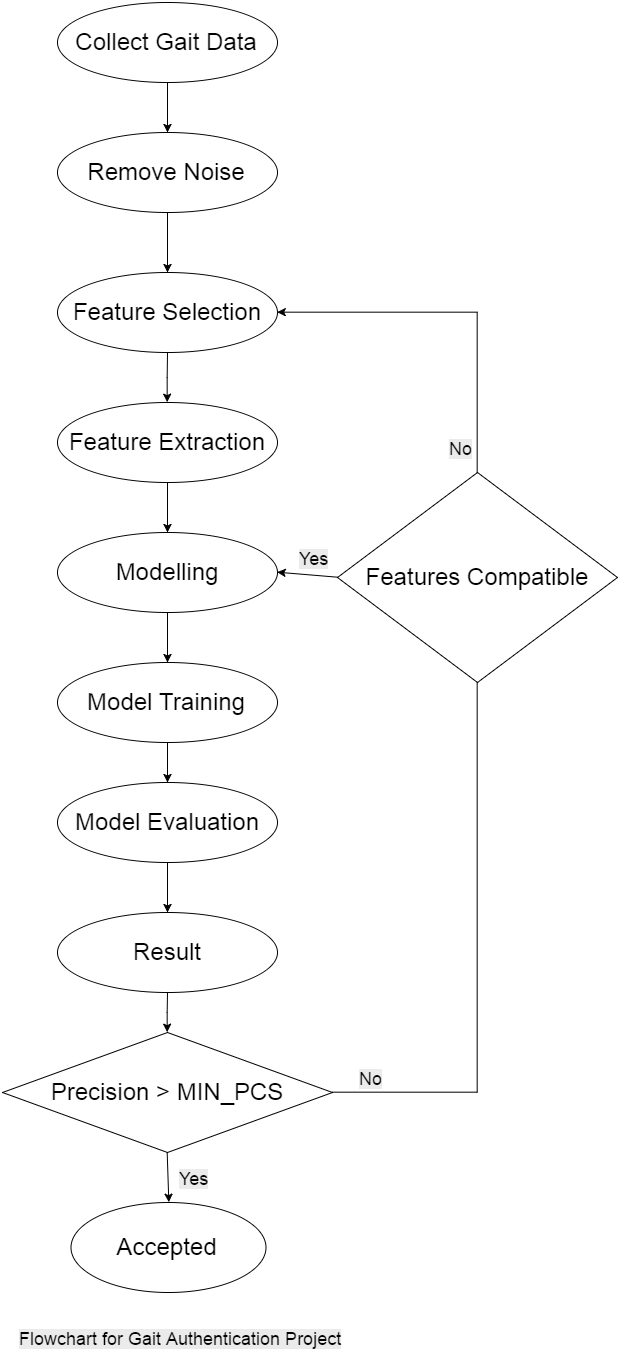
**3.2 LANGUAGE USED**

R statistical programming language is used to tune model in Azure ML and various scripts are being executed to tune parameters, train data and visualize data.

**CHAPTER 4**

**GAIT PATTERN RECOGNITION**

**4.1 FLOWCHART**



**Fig 4.1 Flowchart for Gait Authentication Project**

**4.2 COLLECTING GAIT DATA**

People with age ranging from 18 to 40 (students and instructors) participated in volunteer data collection program, both male and female participated and logged the data in a whole span of week at various time of the day, walking with a constant pace at a smooth surface. A upload utility was provide to aggregate the data at a central repository.

The data is being collected with the help of Accelerometer Log android application available on Google Play Store with due permission. This application logs data from device’s accelerometer in 3 co-ordinates axes - x, y and z and stores them on device’s external storage. The format used for logging accelerometer data is comma separated values (CSV).

**4.3 FEATURE SELECTION AND EXTRACTION**

There are several ways to prepare the raw sensor data before using it for biometrics. Some gait-based biometric work utilizes the data within the time domain, but other successful systems map the time-series sensor data into examples using a sliding window approach, which permits the use of conventional classifier induction systems that cannot handle time-series data.

We are using sliding window approach to extract features for a dataset bounded by the time slot of the window i.e. the actual time person has logged the data through given means - android application. Several features are computed with the dataset using custom R scripts which include:

* Average : Average sensor value (each axis)
* Standard Deviation : Standard deviation (each axis)
* Average Absolute Difference : Average absolute difference between the 200 values and the mean of these values (each axis)
* Average Resultant Acceleration : For each of the sensor samples, take the square root of the sum of the squares of the x, y, and z axis values, and then average them.

Each set of data is being tagged with initials of the person logging the data ex. vtsl for Vatsal and vkku for Vivek. These extracted features are being added to the current dataset which is being passed for evaluating on the classification model. The whole dataset is being converted to a comma separated file(CSV) for evaluation. The computation model is implemented  with the help of cloud based platform - Microsoft Azure Machine Learning. The tagged dataset upon evaluation will contain a field - Scored Labels which is used to decide final result.

**4.4 MODELLING**

A Machine Learning algorithm - decision jungle is used to predict the output labels, this is further discussed in next chapter in detail.

**4.5 RESULT**

Final labels are being predicted upon evaluating the machine learning model, and are examined for values less than threshold (60% actual probability) if found to have less precision, model parameters returned and model is evaluated and trained again.

**CHAPTER 5**

**MACHINE LEARNING ALGORITHM**

**5.1 ALGORITHM USED**

The classifier algorithm used for this project is Microsoft Research’s implementation of Decision Forest - Decision Jungles. Problem with decision forest is that given enough data, the number of nodes in decision trees will grow exponentially with depth which is being addressed by Decision Jungles. This ensembles of rooted decision directed acyclic graphs (DAGs).Unlike conventional decision trees that only allow one path to every node, a DAG in a decision jungle allows multiple paths from the root to each leaf.

During training, node splitting and node merging are driven by the minimization of exactly the same objective function, here the weighted sum of entropies at the leaves. Results on varied datasets show that, compared to decision forests and several other baselines, decision jungles require dramatically less memory while considerably improving generalization.

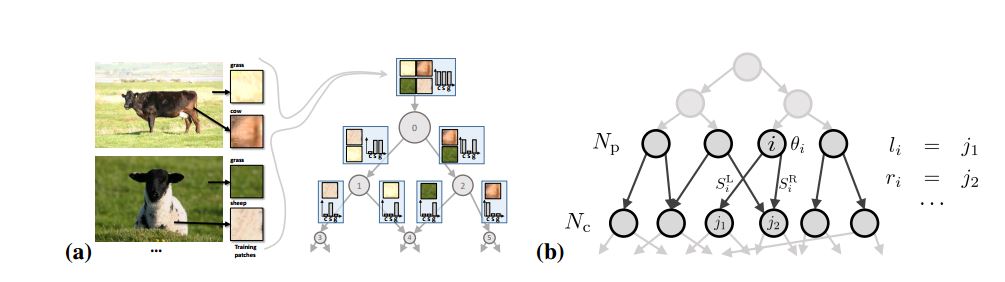
**5.2 RANDOM FOREST ALGORITHM**

Randomized decision trees and forests have a rich history in machine learning and have seen considerable success in application, perhaps particularly so for computer vision. However, they face a fundamental limitation: given enough data, the number of nodes in decision trees will grow exponentially with depth. For certain applications, for example on mobile or embedded processors, memory is a limited resource, and so the exponential growth

of trees limits their depth, and thus their potential accuracy. This paper proposes decision jungles, revisiting the idea of ensembles of rooted decision directed acyclic graphs (DAGs), and shows these to be compact and powerful discriminative models for classification. Unlike conventional decision trees that only allow one path to every node, a DAG in a decision jungle allows multiple paths from the root to each leaf.

We present and compare two new node merging algorithms that jointly optimize both the features and the structure of the DAGs efficiently. During training, node splitting and node merging are driven by the minimization of exactly the same objective function, here the weighted sum of entropies at the leaves. Results on varied datasets show that, compared to decision forests and several other baselines, decision jungles require dramatically less memory while considerably improving generalization.

**5.3 FORESTS AND JUNGLES**

**** Before delving into the details of our method for learning decision jungles, we first briefly discuss how decision trees and forests are used for classification problems and how they relate to jungles. Binary decision trees, A binary decision tree is composed of a set of nodes each with an in-degree of 1, except the root node. The out-degree for every internal (split) node of the tree is 2 and for the leaf nodes is 0.

**Fig.5.1 Motivation and Notation. An example use of a rooted decision DAG for classifying image patches as belonging to grass, cow or sheep classes.**

Using DAGs instead of trees reduces the number of nodes and can result in better generalization. For example, differently coloured patches of grass (yellow and green) are merged together into node 4, because of similar class statistics. This may encourage generalization by representing the fact that grass may appear as a mix of yellow and green. (b) Notation for a DAG, its nodes, features and branches. See text for details. input instance that reaches that node should progress through the left or right branch emanating from the node.

Prediction in binary decision trees involves every input starting at the root and moving down as dictated by the split functions encountered at the split nodes. Prediction concludes when the instance reaches a leaf node, each of which contains a unique prediction. For classification trees, this prediction is a normalized histogram over class labels. Rooted binary decision DAGs. Rooted binary DAGs have a different architecture compared to decision trees and were introduced by Platt et al. As a way of combining binary classifier for multi-class classification tasks.

More specifically a rooted binary DAG has: (i) one root node, with in-degree 0; (ii) multiple split nodes, with in-degree ≥ 1 and out-degree 2; (iii) multiple leaf nodes, with in-degree ≥ 1 and out-degree 0. if we have a C-class classification problem, here we do not necessarily expect to have C DAG leaves. In fact, the leaf nodes are not necessarily pure, And each leaf remains associated with an empirical class distribution.

**5.4 CLASSFICATION DAG V/S CLASSIFICATION TREES**

We explain the relationship between decision trees and decision DAGs using the image classification task illustrated in Figure as an example. We wish to classify image patches into the classes: cow, sheep or grass. A labelled set of patches is used to train a DAG. Since patches corresponding to different classes may have different average intensity, the root node may decide to split them according to this feature. Similarly, the two child nodes may decide to split the patches further based on their chromaticity. This results in grass patches with different intensity and chromaticity (bright yellow and dark green) ending up in different subtrees. However, if we detect that two such nodes are associated with similar class distributions (peaked around grass in this case) and merge them, then we get a single node with training examples from both grass types. This helps capture the degree of variability intrinsic to the training data, and reduce the classifier complexity.

**5.5 LEARNING DECISION JUNGLES**

We train each rooted decision DAG in a jungle independently, though there is scope for merging across DAGs as future work. Our method for training DAGs works by growing the DAG one level at a time.1 At each level, the algorithm jointly learns the features and branching structure of the nodes. This is done by minimizing an objective function defined over the predictions made by the child nodes emanating from the nodes whose split features are being learned.

Consider the set of nodes at two consecutive levels of the decision DAG (as shown in Fig. 1b). This set consist of the set of parent nodes Np and a set of child nodes Nc. We assume in this work a known value for M = |Nc|. M is a parameter of our method and may vary per level. Let θi denote the parameters of the split feature function f for parent node i ∈ Np, and Si denote the set of labelled training instances (x, y) that reach node i. Given θi and Si , we can compute the set of instances from node i that travel through its left and right branches as S L i (θi) = {(x, y) ∈ Si | f(θi , x) ≤ 0}

**5.6 SWAP FRAMES ATTACK**

Experiments and results,this section compares testing accuracy and computational performance of our decision jungles with state-of-the-art forests of binary decision trees and their variants on several classification problems.

**5.7 CLASSSIFICATION AND DATASETS**

We focus on semantic image segmentation (pixel-wise classification) tasks, where decision forests have proven very successful. We evaluate our jungle model on the following datasets:

**1. Kinect body part classification [29] (31 classes)**

We train each tree or DAG in the ensemble on a separate 1000 training images with 250 example pixels randomly sampled per image. Following, 3 trees or DAGs are used unless otherwise specified. We test on (a common set of) 1000 images drawn randomly from the MSRC-5000 test set. We use a DAG merging schedule of |N D c | = min(M, 2 min(5,D) · 1.2 max(0,D−5)), where M is a fixed constant maximum width and D is the current level (depth) in the tree.

**2. Facial features segmentation [18] (8 classes including background)**

We train each of 3 trees or DAGs in the ensemble on a separate 1000 training images using every pixel. We use a DAG merging schedule of |N D c | = min(M, 2 D).

**3. Stanford background dataset [12] (8 classes)**

We train on all 715 labelled images, seeding our feature generator differently for each of 3 trees or DAGs in the ensemble. Again, we use a DAG merging schedule of |N D c |= min(M, 2 D).

**4. UCI data sets [22]**

We use 28 classification data sets from the UCI corpus as prepared on the libsvm data set repository.2 For each data set all instances from the training, validation, and test set, if available, are combined to a large set of instances. We repeat the following procedure five times: randomly permute the instances, and divide them 50/50 into training and testing set. Train on the training set, evaluate the multiclass accuracy on the test set. We use 8 trees or DAGs per ensemble. Further details regarding parameter choices can be found in the supplementary material. For all segmentation tasks we use the Jaccard index (intersection over union) as adopted in PASCAL VOC. Note that this measure is stricter than e.g. the per class average metric reported in. On the UCI dataset we report the standard classification accuracy numbers.

**5.8 BASELINE ALGORITHMS**

We compare our decision jungles with several tree-based alternatives, listed below. Standard Forests of Trees. We have implemented standard classification forests, as described in and building upon their publically available implementation.

**Baseline 1: Fixed-Width Trees**

* As a first variant on forests, we train binary decision trees with an enforced maximum width M at each level, and thus a reduced memory footprint. This is useful to tease out whether the improved generalization of jungles is due more to the reduced model complexity or to the node merging. Training a tree with fixed width is achieved by ranking the leaf nodes i at each level by decreasing value of E(Si) and then greedily splitting only the M/2 nodes with highest value of the objective. The leaves that are not split are discarded.

**Baseline 2: Fixed-Width Trees**

* A related, second tree-based variant is obtained by greedily optimizing the best split candidate for all leaf nodes, then ranking the leaves by reduction in the

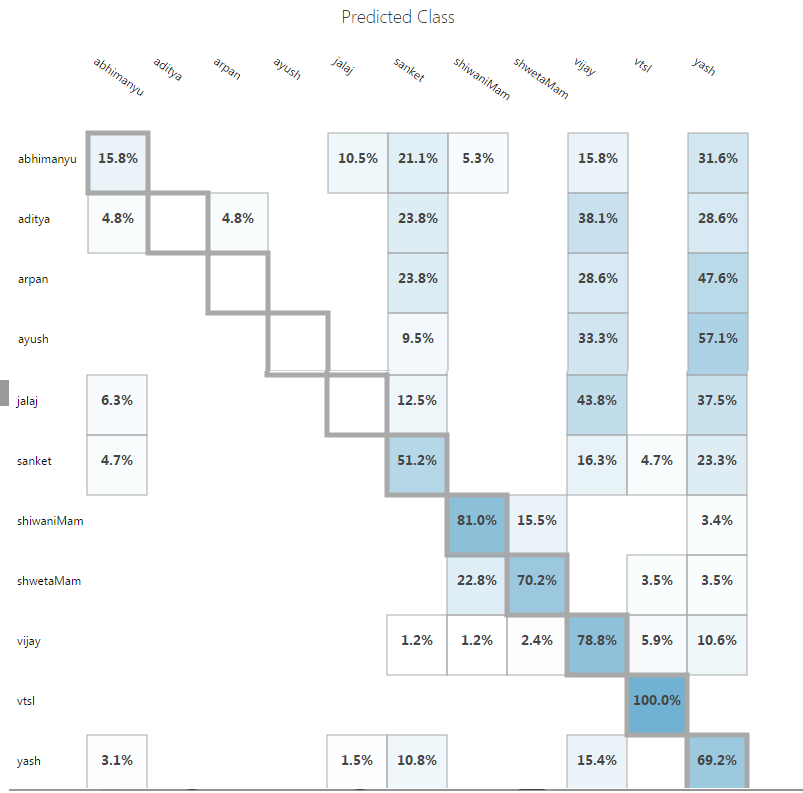
**5.9 DAGs VISUALIZATION**



**CHAPTER 6**

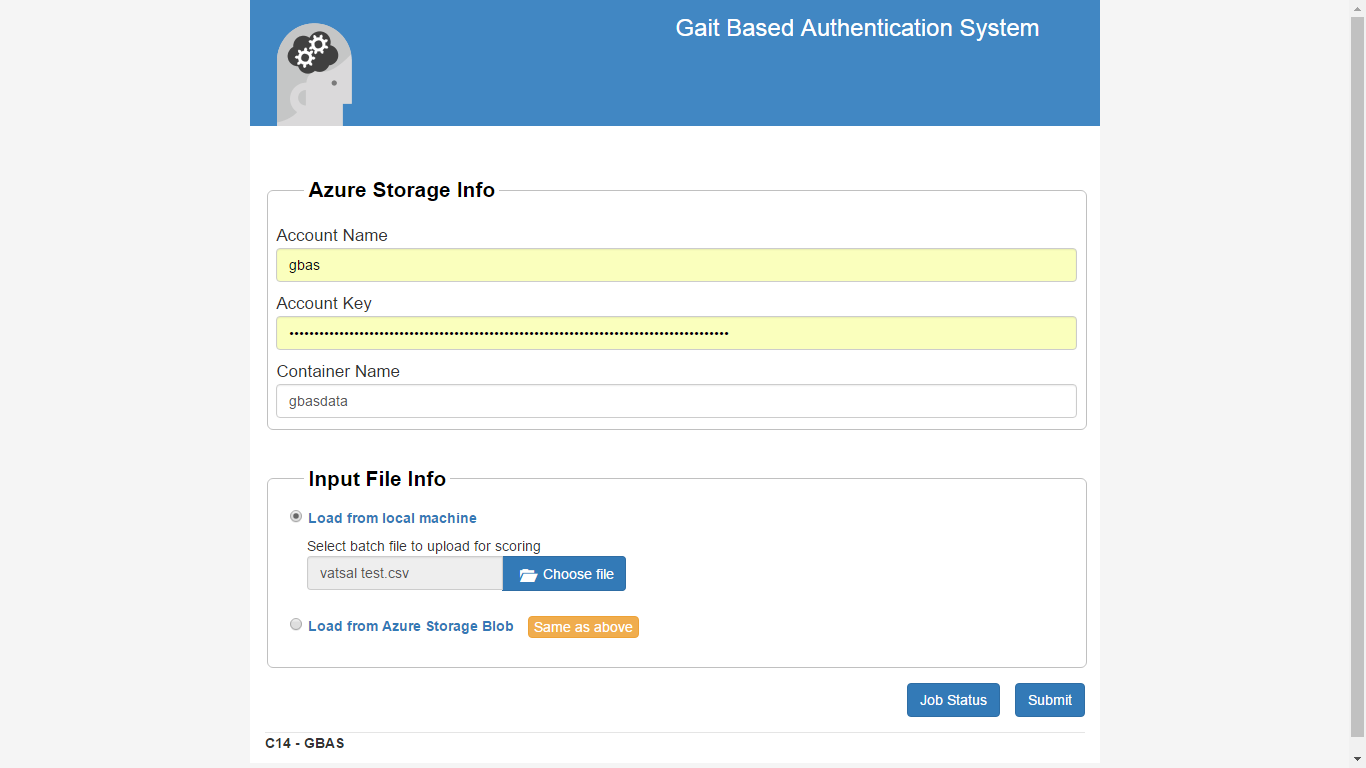
**RESULTS AND CONCLUSION**

This project successfully completed with good precision and accurate labelled outputs. The machine learning model was out performing and scored labels with great accuracy upon expanding the dataset, hence providing more robustness.

**6.1 CONFUSION MATRIX**

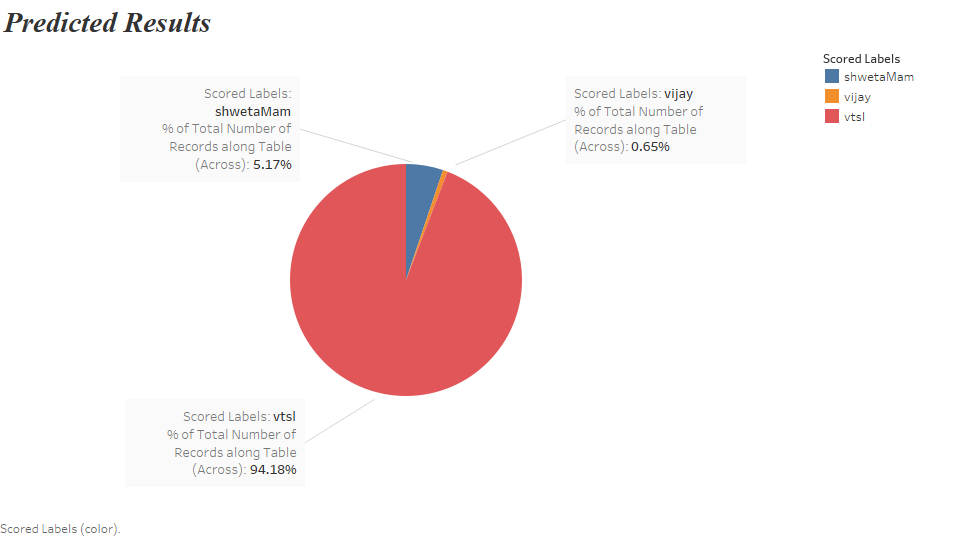
**Table 6.1 Confusion Matrix**

**6.2 WEBAPP**

****

**Fig.6.1 This webapp can be located at :** [**http://gbasbatch.azurewebsites.net/**](http://gbasbatch.azurewebsites.net/)

**6.3 SCORED PROBABLITIES**

** Fig. 6.2 Scored Probabilities Visualization by Tableau**

**CHAPTER 7**

**FUTURE ASPECTS**

Gait as a means of identification is explored and found to be reliable in this paper. The available commercial smart-phone embedded accelerometer was able to carry out identification with modest accuracy. As stated before, the advantage of this method to other biometric systems which could be implemented on mobile phones, is the unobtrusive operation which gives a high user friendliness. To make biometric gait recognition using embedded accelerometers a technology suitable for practical use, further research on feature extraction and comparison is required. However the achieved results are promising and the proposed approach contains potential for enhancement.

We plan to improve our identification and authentication systems in several ways. With respect to data collection, we intend to increase the number of users in the data set, collect more data per user. Instead of our own database we will try to evaluate our model on various online available data sources. More experimentation is needed regarding the placement of the device, instead of placing it in definite position and configuration we will try this placement to be natural so as to increase of user friendliness to a level up. The attack resistance of biometric gait recognition should also be analyzed. Studies by Gafurov and Mjaaland show that it is difficult for an attacker to imitate another person. This needs to be confirmed for the special scenario of mobile phones.

**REFERENCES**

[1] “FVC2006: the Fourth International Fingerprint Verification Competition,” http://bias.csr.unibo.it/fvc2006/results/O\_res\_db2\_a.asp, [Online; accessed 28-March-2017].

[2] Abeni, P. Baltatu, and R. D’Alessandro, “A face recognition system for mobile phones,” ISSE 2006 — Securing Electronic Business Processes, pp. 211–217, October 2006

[3] “FVC2006: the Fourth International Fingerprint Verification Competition,” http://bias.csr.unibo.it/fvc2006/results/O\_res\_db2\_a.asp, [Online; accessed 28-March-2017]

[4] D. J. Fish and J. Nielsen, “Clinical assessment of human gait”, in Journal of Prosthetics and Orthotics 2, April 1993

[5] L. Lee, “Gait Analysis for classification”, Massachusetts Institute of Technology-artificial Intelligence Laboratory, 2003

[6] J. Suutala, K. Fujinami and J. Roning, “Gaussian process person identifier based on simple floor sensor”, in Proceedings of the 3rd European Conference on Smart Sensing and Context, 2008.

[7] A. Annadhorai, E. Guenterberg, J. Barnes and K. Haraga, “Human identification by gait analysis”, in Proceedings of the 2nd International Workshop on Systems and Networking Support for Health Care and Assisted Living Environments, 2008

[8] Jamie Shotton, Toby Sharp and Pushmeet Kohli, “Decision Jungles: Compact and Rich Models for Classification”

[9] gbas data upload utility, <https://script.google.com/macros/s/AKfycbxEiWElCj9-CpHwsnGP9HfrcIuaZgGOMfquRidZQmbhkJ36NdM/exec>

[10] J. R. Quinlan. Induction of decision trees. Machine Learning, 1986.