VitaRun: Running-Injury Prevention System

Alex Gourlay, Jonny Midgen, Jacob Mitchell, Benjamin Pheifer, Norbert Wesely, Kenza Zouitene, Luidmila Zudina
Department of Electric and Electronic Engineering, Imperial College, SW2 7AZ
{ADG15, JM3515, JM4515, BP615, NW1515, KZ815, LZ5515}@ic.ac.uk
https://github.com/milazudina/MHML

***Abstract*— Running injuries resulting from incorrect pronation and high impact force lead to the development of chronic injury. Studies have shown that a high proportion of these injuries come from incorrect gait. Supplying runners with advice and training can reduce the risk of injury. This study introduces VitaRun, a mobile App, insole and online server, which looks to measure and interpret runners' step frequency and pronation with sensors and machine learning, and give the user recommendations on how to reduce risk of injury. The effectiveness of the app was tested on ten participants; each were asked to run with the insoles for two periods of five minutes, one run with and one without the app's audio feedback enabled. The study found that the feedback was statistically ineffective at improving the stride rate of participants, further improvements to the feedback system are discussed.**

I. INTRODUCTION

Running was the most popular sport activity in the UK in 2017. According to the Active Lives Survey, 15% of people surveyed had ran at least twice in the past month [1].The health and fitness benefits of running are well-established. It has been shown to improve mental health [2] and increase life-expectancy; people who run were shown to have a 45% lower risk of cardiovascular associated mortality [3].

While increased running training has great advantages, it is also invariably associated with more injuries [4], [5]. A study showed that for amateurs, across an 8 week training period before a race, at least 1 in 4 (25.9%) participants will have an injury significant enough to restrict running [6]. Risk factors for running-related injuries include lack of experience, previous injuries, running in overused shoes and bio-mechanic risk factors [7], [8]. Some of these are out of the runners control, but the runner can reduce the risk of injuries by controlling their **gait**, where gait is defined as motion achieved through the movement of human limbs [9]. It takes between several weeks to several months to return to running after an injury, hence it is important to focus on early intervention or even prevention of running injuries [9].

Our system aims to address this issue by monitoring selected gait features of the runner through force sensors and an IMU (Inertial Measurement Unit) embedded in an insole, and providing live actionable feedback during the run. It will also provide insights derived from this data and keep a history of runs, allowing the user to observe personal trends.

II. BACKGROUND

A. Bio-mechanic risk factors

Recent research suggests that runners who exhibit relatively large and rapid impact forces while running are at an increased risk of developing an overuse injury of the lower extremity due to a combination of high stress and frequency [9]. Furthermore, deviations from the normal in running mechanics can lead to injuries such as plantar fasciitis, shin splints and a range of knee problems [10].

Fig. 1. Different types of pronation, demonstrated on left foot

Following the advice of a kinesiologist, who researched injury prevention[11], we focus on the following features:

1) Pronation: **Pronation** is a natural movement that occurs during foot landing in walking and running in three planes of human movement and ensures shock absorption. To visualize pronation movement, stand behind a runner and watch how both the heel and the medial side of the ankle roll inward relative to the position of the lower leg [12]. It is well illustrated by Fig. 16 [13]. Due to the interconnection of the bones, joints and ligaments, each movement in pronation influences motion throughout the entire leg up to the hips. Three types are recognised: neutral pronation, under-pronation and over-pronation. In neutral type, the weight distributes fairly evenly among all of the toes with a slight emphasis on the big toe and second toe which are adapted to handle more of the load [14]. In **over-pronation**, push off load is more focused on the big toe and second toe, resulting in poor impact absorption and instability. The muscles of lower extremity compensate, leading to strains and micro-traumas. **Under-pronation** is the opposite and implies the lack of "inward roll" when the foot impacts the ground. The weight is then transferred to the outside of the foot and the smaller toes, which are not adapted to handle high forces. Under and over pronation affects 60% of runners [15], yet despite the wealth of resources available to runners today, studies have shown that they are generally unable to appropriately classify their pronation type [16], [17].

2) Impact force: Impact force refers to the force that is absorbed by the foot when striking the ground. Higher impact forces have been correlated with injury. Depending upon speed and landing geometry, impact forces vary in magnitude from approximately 1.5 to 5 times the body weight and recede after a very brief period of time ($< 30ms$) [9]. Studies show that runners who's stride patterns incorporate low levels of impact force and a moderately rapid rate of pronation are at a reduced risk of incurring overuse running injuries [9].

3) Stride length/stride frequency: Reducing stride length leads to reducing impact force, thus preventing injuries, it can be done by increasing step frequency. Recommended frequency is 180 steps per minute, but a sudden change in frequency can lead to injuries. As such we normally suggest decreasing frequency gradually, eventually reaching a 5% to 10% reduction. This will lead to lower force absorption by ankle, knee and hip joints [18].

To achieve gold standard gait analysis an expensive motion capture system is required. However, an adequate level is achievable with a combination of signals from more simple sensors such as accelerometers and gyroscopes placed in an insole. As a result, there is a variety of systems on the market that provide basic gait analysis.

B. Market research

All similar systems currently available on the market focus on running efficiency and coaching, rather than injury prevention. For example, SHFT and Milestone Pod which provide feedback based on metrics such as landing position and brake effect [19], [20] and RunScribe which relies on on-shoe sensors to provide numerous metrics and analysis, but offers no real-time feedback [21].

Similar insole sensors available on the market are Moticon Sensor Insoles, which are marketed to researchers and clinicians, and require expensive software and training [22]. The closest alternative to our system is Retisense, that provides statistics after the run, but no coaching or real-time feedback [23]. The hardware side of this project relies on Retisense's Stridalyzer insoles, as producing a system of high quality, well-calibrated sensors in an insole was beyond its scope. The Stridalyzer native application shows pressure distribution, however it labels all subjects as "overpronating", which seems improbable.

III. Hypothesis

Given that pronation type can only be effectively altered through orthotic insoles, support running shoes, stretching and exercising; the theory chosen for testing was the effectiveness of the app feedback on stride frequency. The hypothesis was thus stated as so:

H_0: App feedback results in no change in stride frequency

H_1: App feedback results in an improvement in stride frequency.

Testing of the Hypothesis is discussed in the Experimental Findings section of this report.

IV. System Design

A. System Overview

An overview of the system is detailed in the system diagram in Fig. 2.

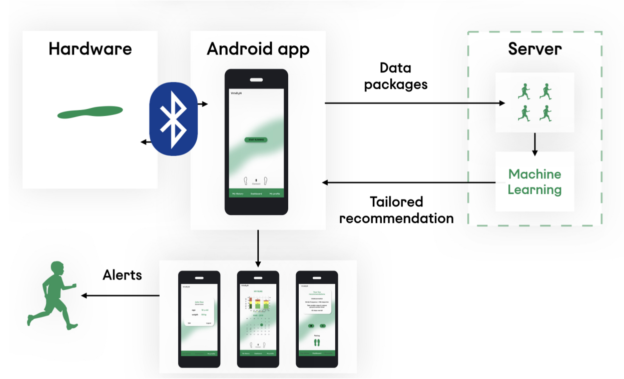

Fig. 2. Whole System Overview

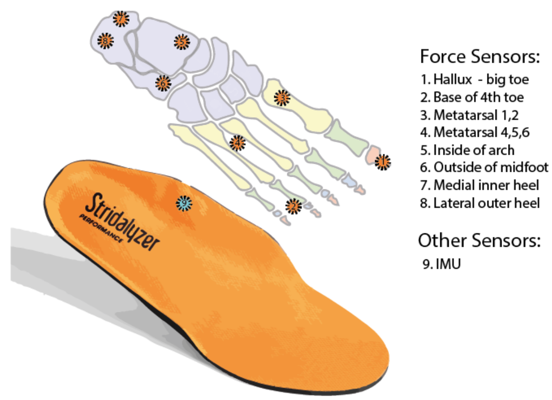

Fig. 3. Stridalyzer Insoles by Retisense

B. Hardware

The sensing element of the system was performed by a pair of 'Stridalyzer' smart insoles, purchased from the manufacturer Retisense, shown in Fig. 3. Each insole contained 8 pressure sensors and a 6 axis IMU. Given that an off-the-shelf solution was used, no extra hardware needed to be developed for the project. The focus was instead on integrating the insoles into the app. This involved securing a wireless connection to the insoles using Bluetooth Low Energy (BLE), whereby the insoles acted as the server and the Android device as the client. The VitaRun app performed a BLE scan for 8 seconds and if the MAC address of either of the insoles was found, then a reference to the found device was stored. Biomechanic data could then be streamed by subscribing to the appropriate BLE characteristic via a GATT service. Data was streamed at a frequency of 5Hz as determined by the insoles. The next step in integration involved importing and compiling the Stridalyzer native library supplied by the manufacturer into the Android project. This allowed the data stream to be converted from a byte array into a structure holding the pressure data values across the insole and accelerometer readings. Once conversion had occurred, the data was appended to a buffer to be sent through to the server once filled.

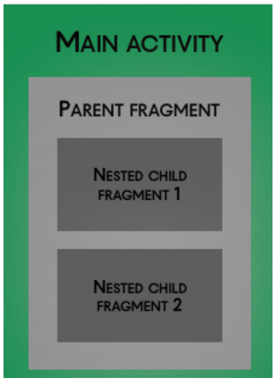

Fig. 4. Outline of the VitaRun App Architecture

C. App

1) App Architecture: The app was developed in Android Studio and constructed in separate modules called ’fragments’ which each provided a core UI function. The fragments all ran within a single main activity that held the state of many of the core functions of the app e.g. BLE connection or the current run information. This architecture allowed team members to work individually on components without editing the same scripts concurrently.

2) UI Design: The context of use, the intuitiveness and ease of implementation were considered during the design of the User Interface (UI). An initial mock up was made to describe the app flow from a users perspective and to predetermine the app architecture that was implemented in Android Studio. The app wireframes were developed using AdobeXd which allowed for dynamic prototyping.

The mobile app was divided into 3 main pages: ’My History’, ’Start a Run’ and ’My Profile’. It was designed following a quick glance ethos, the user can just glance at the app and get their tailored summary. Each screen is divided in the same manner to enable straightforward use. The consistency between each screen allows the user to effortlessly navigate VitaRun.

’Start a Run’ acts as the dashboard of the app. The VitaRun app launches into this page to start a run, observe the overall or post run summary. ’My History’ gives access to the users past data from their previous runs. It is divided into 2 fragments, containing a calendar with run events indicated on the relevant dates as well as an interactive graph displaying run data.

Finally, when the user first downloads the app they are prompted to create a profile. This allows VitaRun to associate all the data and machine learning classifications to a user. From then, they can always access their account from the My Profile tab, edit or log out of their profile.

3) Profile: All the data recorded and displayed in the app is user-specific. This means that to access their data, a user needs to be logged in. Upon startup, the app will check its locally stored information to see if a user is logged on. If not, the login activity will run. The user will be prompted to enter their username and password. Pressing ”Log In” will then send a request to the server with the entered details, to verify the users identity, a response will then grant or deny login. Upon successful login, their username will be set locally and their details will be shown on the Profile page. If the log in is unsuccessful, a notification will appear, informing them if, a) the password was incorrect, or b) the username does not exist. There is also a button below the login, offering the user to create a new profile. This button starts up an activity where the user enters all their details including a new username and password. They then click the create profile button and, so long as the username does not already exist, a new user directory will be created on the server, and they can proceed to use the app with their newly created profile. When logged in to the app, the Profile page Fig. 6 shows the user’s name, username, age and weight. The name, age and weight can all be updated by clicking the edit button, making the changes, and then clicking the save button. From this page the user can also log out, and all shared preferences will be cleared.

4) Feedback: The ’Start a Run’ screen Fig. 6 holds multiple fragments, one of them, the recommendations fragment needs to be updated at a given frequency. Every 15 seconds, the Recommendations Fragment requests the data from the server, it then passes the string to a method that takes the string and updates the VitaRun UI to give live recommendations to the runner. The audio feedback is a cut-down version of the method with only the stride length recommendations being read out every 4 minutes by a text-to-speech converter. The aim is to inform the user during their run and not distract them. The audio feature transfers the interesting information to the user without needing them to stop to get access to the stride length recommendation.

Steps Frequency (steps/min)	Audio Feedback
freq<165	"Try taking smaller steps and reducing ground contact time!"
165<freq<175	"You are just under the perfect running pace, try taking smaller steps and reducing ground contact time!"
175<freq<185	"Well done! Keep the same pace."
185< freq < 195	"WOW!! You are running at elite runners' stride rate!"
freq>195	"Are you sprinting? You should slow down"

Fig. 5. Types of audio feedback

5) Data Visualisation: Part of the user feedback is a visualisation of historic runs. This feature is a useful tool that allows the observation of trends in pronation over time, monitoring the progression of the condition and the effectiveness of intervention. An interactive calendar and stacked bar chart Fig. 6 were used to visualise the data from the server. Each run event is visualised as a symbol on the day that it occurred. When a day with an event is clicked, the graph adapts the view to show the runs on that day at the centre of the graph. The immediate runs before and afterwards are included for quick comparison of progress. Each stacked bar shows the number of steps that are over (red), normal (green) and under (yellow) pronation. The total height of each bar corresponds to the total number of steps from that run. The calendar used a specialist library [24] that allowed events to be added to each day. The visualisation required a library [25] that allowed stacked bar charts.

Fig. 6. UI: Start Run, Recommendations, Profile and Historic Runs (clockwise from top left)

D. Back end

The back end tasks can be categorised into user profiles management, signal processing and pronation type classification. They are performed on the server. The server was written in Python and communicated with the mobile application through a RESTful (Representational State Transfer) API. Each buffer of samples is sent in JSON format from the app to the server via a POST method. The server then computes the step frequency of the current buffer, and accumulates a longer secondary buffer of data which is used for dividing

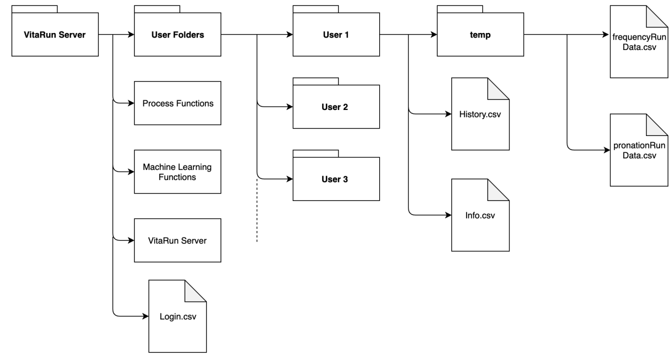

Fig. 7. Architecture of files within the server

samples into steps and identifying the pronation type. GET requests are then used to return data as JSON files to the application. The server was local, which made it only accessible from within the same Wi-Fi network.

1) User Data Management: Within the server, the files were stored within a custom designed architecture of files. This is shown in Fig. 7.

The first layer of the server contains one folder, 3 scripts and 1 csv file. The three scripts contain all the code that the server requires to run and to communicate with external devices. ”VitaRunServer” contains code that will send POST and GET requests to the outside world and depending on the messages it decodes it calls functions within the ”Machine Learning Functions” and ”Process Functions”. The ”Login.csv” contained all the username and password combinations and is used as a reference within the server for login certification. Within ”User Folders”, all users had their own folders which were created specifically for them, when they registered on the VitaRun application. Each user folder is created with the same architecture. ”History.csv” stores information from all the runs that that specific user has made. It gives the user the ability to request historical data. Personal information of the user, for example ’age’ and ’weight’ will be stored within ”info.csv”. The ”temp” folder contains two csv files which contain the running frequency data and pronation data for a run. These files will be re-written for every new run after the data has been used and condensed; where it is then stored in ”History.csv”.

2) Signal Processing: Raw data recorded by the insoles was sent to the server, where the ultimate aim was to characterise the run and classify pronation type. This was hard to achieve with unprocessed data shown in Fig. 8, hence preprocessing of data was required.

The goal of this step was threefold:

1) *Determine whether the person is running:* This was important, as leisure-time runners frequently stop their activity for a short period of time due to a red light, crowding on the pavement, small breaks due to tiredness, etc. These periods need to be excluded from any further processing as the ML algorithms are not trained to classify non-running steps and incidental misclassified stationary steps can decrease overall accuracy of

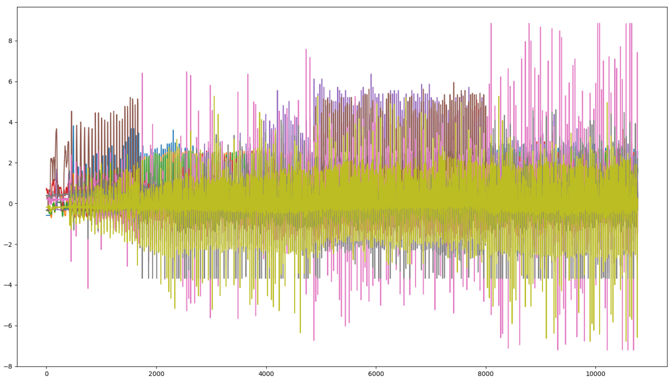

Fig. 8. Normalised input data from the Stridalyzer insoles

VitaRun.

2) *Measure real-time run frequency:* VitaRun provides in-run feedback on stride length, aiming to help runners achieve an ideal ca. 180 Spm rate. These recommendations rely on real-time frequency data.
3) *Split run into individual steps:* Pronation classification for entire runs is hardly feasible, a better strategy is to classify each step of the run individually and take the mode of labels as the label for the run. For this the ML classifier requires running data split into steps where each step is an array of equal dimensions. Furthermore, ideally to increase accuracy of classification, each step window should start at the same point of the step, e.g. row 1 should always store the data for the first moment of the step, where the shoe first touches the ground.

Objectives 1) and 2) tie together, as they both require the frequency of the run, which was achieved using FFT on a small chunk of recent data. As the raw data from the insoles is composed of multiple pressure values and is very noisy, first inputs of the individual sensors get summarised, then smoothed by a moving mean filter which was designed to eliminate only frequencies outside of the running frequency domain, shown in Fig. 9. The second challenge was the determination of a suitable window size for the FFT, as the larger the window, the more accurate the results are, but also the more the results lag behind the input values. For these reasons the length was chosen to be 256 cycles which is slightly longer than 5 steps or signal periods. Furthermore, due to the long signal periods compared to window size the FFT runs multiple times gradually decreasing window size to below 5 signal periods and frequency is determined based on these results to get more accurate readings. Fig. 10 shows local frequency of a run and the plot area with green background highlights the period the app considered as running, while white regions were identified as non-running and hence were discarded.

Objective 3) relies on the running periods determined by the function discussed above, but it also splits the data. The key challenge here was to determine where each step starts. Initially this was done using smoothing filters, peak recognition, and filtering of peaks which yielded good accuracy with rare false splits. This method was developed as the built-in step counter of the first pair of insoles was very unreliable, however the second pair worked surprisingly accurately, hence the latest version of VitaRun relies on this built-in function. Once the beginning of each running step is determined, the software extracts and returns 30 sample chunks of the normalised (but unprocessed) data. The window size for the steps was determined based on the sampling rate of the Stridalyzer insoles and human running characteristics. 30 samples at 50 Hz sampling rate is more than enough to record a step at 140 Spm which is the minimum frequency which can be considered as running, while on the other hand the same window is short enough to record only one step even at 200 Spm. Fig. 11 shows 3 steps extracted from the normalised data streams.

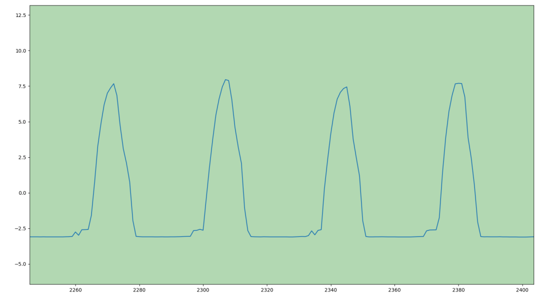

Fig. 9. Filtered data for FFT

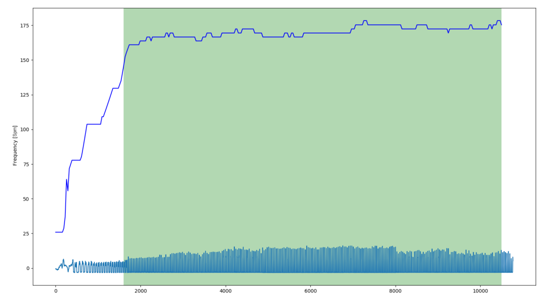

Fig. 10. Local frequencies over the run, and running period

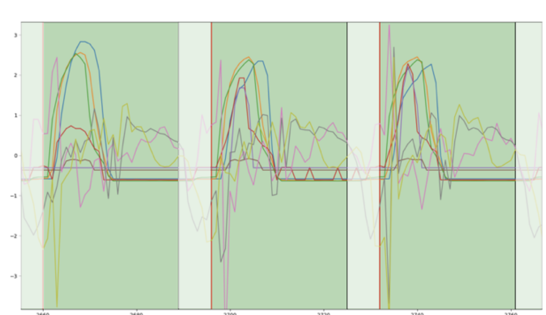

Fig. 11. Extracted steps form the normalised data streams

3) Pronation type classification: Each step of a run is classified as "normal", "over" or "under" pronation types. Classification is performed by an LSTM (Long Short-Term Memory) RNN (Recurrent Neural Network), trained on data

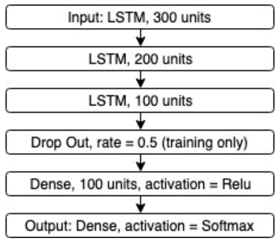

Fig. 12. LSTM model architecture

from the insole sensors and deployed on the server.To compensate for inaccuracies of the prediction, we return the mode of the predictions from the buffer. The details on the data collection, training and deployment can be found in the section below.

E. Machine Learning

1) Data collection and labelling: The data for training was collected from amateur runners from Imperial Athletics and Cross Country Club. In order to get more diverse data from one runner, data was first collected during their normal run, after which they were asked to imitate overpronation and underpronation. This also allowed us to get similar distribution of classes. To assess their natural pronation type, the wear pattern on the soles of their running shoes was inspected. The runs were then divided into steps, and all steps were labelled with the type of the run to which they belong. In total, our training data set contains 20 runs from 8 users, adding up to around 7000 steps, divided into left and right foot. All six pressure sensors and accelerometer were sampled at 50Hz through the native Stridalyzer Insight App, resulting in 9 variables in total.

2) Model description: The choice to implement a Deep Learning model was driven by only partial availability of data from the sensors in the beginning of the project. Using deep learning allowed us to achieve preliminary results with only accelerometer data with minimal feature extraction, such as calculating total acceleration from three axes. To address multivariate time series classification, we have used an LSTM (Long Short-Term Memory) RNN (Recurrent Neural Network), which is widely used for classification of time-series such as accelerometer signals, and has previously demonstrated record results in problems, such as speech recognition [26]. To implement the model, Keras API was used [27]. Keras integrates with lower-level deep learning Tensorflow API and is second most adopted API after TensorFlow itself. After experimenting with multiple model architectures, our final version is a stacked LSTM neural network with an architecture, demonstrated in Fig. 12 [28]. Drop Out layer is used to prevent overfitting. In the final layer, Softmax function is used for multi-class classification. The model is compiled with Adam optimizer with the learning rate of 1e-4 and categorical-crossentropy loss function. The optimal batch size and number of epochs was determined by tuning and testing, as shown in ”Performance” section.

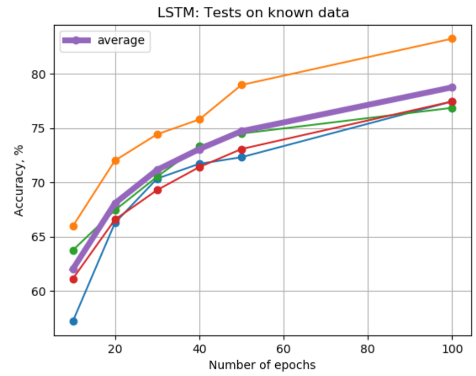

Fig. 13. Testing the model on known data

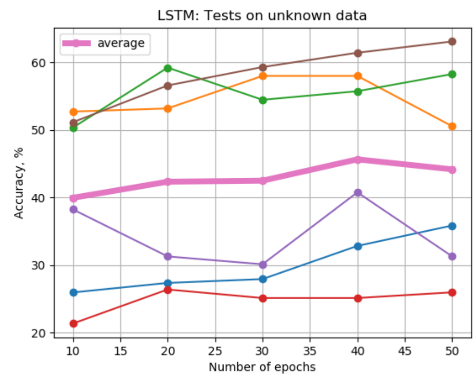

Fig. 14. Testing the model on unknown data

3) Performance: The performance of the model was assessed in two ways. First, accuracy was assessed on randomly drawn samples from the training data. That implies that the model was trained on all the runs from all the runners, and a random 20% was used to test the performance. The accuracy grows consistently, as is demonstrated in Fig. 13. Then, the accuracy was assessed after training the model on randomly drawn 7 out of 8 runners, and testing it on the labelled data from the 8th runner. The results are shown in Fig. 14 and indicate that the data set needs to be expanded to allow for better generalisation, and that performance varies significantly for some samples. The final model is trained for only 15 epochs to reduce the effects of overfitting. Fig. 15 shows the confusion matrix of the final trained model. The accuracy achieved is 55.4 %. Label 0 corresponds to ”normal”, 1 - ”over”, 2 - ”under” pronation.

4) Deployment: Originally, we have intended to deploy the model directly in the app. However, since the Tensorflow Mobile Library has been deprecated in early 2019 and TensorFlow Lite does not currently support RNN and LSTM in custom models, it was decided to deploy the model on the server.

V. Experimental Findings

Testing of the hypothesis was conducted on 10 users, 3 were VitaRun team members and all were amateur runners.

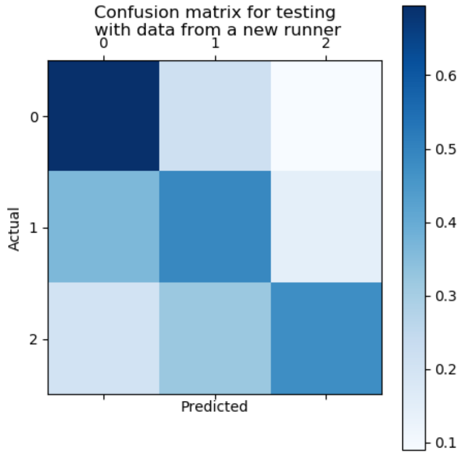

Fig. 15. Confusion matrix for testing on runner, not included in the training data set

Each runner was asked to run with the insoles present and with no audio feedback for a time period of around 5 minutes. The same runner was then asked to run again with audio feedback enabled. The audio feedback rate was increased beyond the app default to a rate of 1 voice command per minute. The average stride frequency of each run was stored upon completion and is shown in Table 1.

A. Results

	Average Stride Frequency (steps/min)	
Sample	No Feedback	Feedback
A	171.5	173.6
B	174.4	178.5
C	168.4	168.5
D	160.0	165.4
E	170.5	170.6
F	167.4	166.4
G	176.3	186.0
H	178.2	181.5
I	172.5	174.9
J	174.9	171.4
mean	171.4	173.7
s.d.	5.27	6.63

Table 1: Average Stride frequency of users before and after mid-run audio feedback.

Given that testing was performed on a small number of repeat participants and that the experiment involves a categorical dependent and continuous independent variable, a Wilcoxon Matched-Pairs test was conducted at the 10% significance level. The values of stride frequency were converted to their absolute differences from the ideal frequency of 180 Spm before testing in order to assess whether an improvement was made. The test yielded a p-value **0.33**, meaning that the null hypothesis is accepted at the 10% level, and so testing would conclude that the audio feedback was not effective in improving the runners' stride rate.

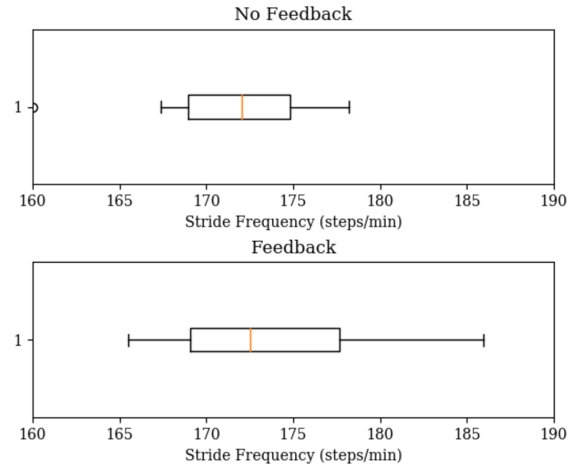

Fig. 16. Boxplots of experimental data

B. Discussion

Although providing audio feedback did not result in a statistically significant improvement in stride frequency overall, it can be seen that audio feedback did result in an improvement in certain samples. In samples B and D an increase of around 5 Spm towards 180 was observed which would suggest a positive response. All test runners were running with a below ideal stride rate before feedback which would suggest that generally amateur runners run with too infrequent strides. Only with feedback did samples G and H run with a stride rate above 180.

Testing was done on a small and biased sample set. All participants were in the age range of 20-24 and so the analysis only account for effectiveness on that age range. Factors may have affected the validity of the results include the fact that:

- The two runs were conducted immediately after each other. This could have resulted in test users being more tired or, conversely, better practised in the 2nd run.
- The two runs were not conducted over the *exact* same path.

VI. Future Work

To expand on the project and make it a product that can be truly trusted by its users, encryption of data transfer is required. The method that was originally chosen at the start of this project was to use a hashing algorithm. Although with further thought and advice from the department, the decision was made to change the encryption method to asymmetric encryption. This means that each user will have a public and private key, as well as the server. The method works where as the public keys are shared easily and all data that is sent will be encrypted using the receiver of the information public key. This means that only the private key connected to the public key used to encrypt the sent data will be able to decrypt the data. To generate the two keys, on the creation of each user the RSA algorithm will be used. The creation of the servers two keys will also be done with the RSA algorithm. VitaRun is aimed at running enthusiasts and therefore it is important that VitaRun is able to compete with competitors. This means that future work is going to require implementation of common place running features like distance travelled, speed, GPS location and a visual map of running route. As well as including a social aspect of the app so that users can compete with their friends and compare running statistics.

VII. Conclusion

In conclusion, this project has successfully designed a mobile application to help amateur runners prevent injuries by giving feedback on their gait and stride frequency. This feedback is given given before, during and after the run. The system includes insoles, the android application and the server.

The system detects abnormal gait patterns and stride frequencies with visual and audible feedback to allow the user to adjust their running pattern and reduce their risk of injures to their lower body from their running.

Machine learning was used to analyse the complex data patterns, that the application uses to measure and detect any pronation issues with the users running technique. As well as the speed of the runner stride.

All information was stored within a server designed for this project and the application is able to access whichever data the user requires on the app. This is done via get and post requests.

The results of our test concludes that the audible feedback requires more work and maybe a different approach to produce a greater effect on the runner.

VitaRun's future work includes including traditional features that competitor running applications have, for example distance travelled and GPS tracking of running path. As well as making all information inside VitaRun asymmetrically encrypted. These two features will enable VitaRun to be a more complete product for all runners.

References

[1] Active Lives Adult Survey 16/17 Report. Technical report, 2017.

[2] Paul Morris. Not just a run in the park: A qualitative exploration of parkrun and mental health. Technical report, 2017.

[3] Duck-Chul Lee, Russell R. Pate, Carl J. Lavie, Xuemei Sui, Timothy S. Church, and Steven N. Blair. Leisure-time running reduces all-cause and cardiovascular mortality risk. *Journal of the American College of Cardiology*, 64(5):472481, 2014.

[4] Chi Pang Wen, Jackson Pui Man Wai, Min Kuang Tsai, and Chien Hua Chen. Minimal Amount of Exercise to Prolong Life. *Journal of the American College of Cardiology*, 64(5):482–484, aug 2014.

[5] R N van Gent, D Siem, M van Middelkoop, A G van Os, S M A Bierma-Zeinstra, and B W Koes. Incidence and determinants of lower extremity running injuries in long distance runners: a systematic review. *British journal of sports medicine*, 41(8):469–80; discussion 480, aug 2007.

[6] I Buist, S W Bredeweg, B Bessem, W van Mechelen, K A P M Lemmink, and R L Diercks. Incidence and risk factors of running-related injuries during preparation for a 4-mile recreational running event. *British journal of sports medicine*, 44(8):598–604, jun 2010.

[7] Dennis Y. Wen. Risk factors for overuse injuries in runners. *Current Sports Medicine Reports*, 6(5):307–313, Oct 2007.

[8] Maarten P. van der Worp, Dominique S. M. ten Haaf, Robert van Cingel, Anton de Wijer, Maria W. G. Nijhuis-van der Sanden, and J. Bart Staal. Injuries in Runners; A Systematic Review on Risk Factors and Sex Differences. *PLOS ONE*, 10(2):e0114937, feb 2015.

[9] Alan Hreljac. Impact and Overuse Injuries in Runners. *Med. Sci. Sports Exerc*, 36(5):845–849, 2004.

[10] Carl W Chan and Andrew Rudins. Subject Review Foot Biomechanics During Walking and Running. Technical report, 1994.

[11] Siobhan Anderson. Injury prevention research. https://siobhananderson.myportfolio.com/ (Accessed: 2019-01-28).

[12] Joseph E. Muscolino. *Kinesiology : the Skeletal System and Muscle Function.* Elsevier Health Sciences, 2014.

[13] Pronation type. https://www.justrunlah.com/2015/08/01/do-you-know-whats-your-pronation-type/. (Accessed: 2019-01-29).

[14] Kiera Carter. Pronation Explained. www.runnersworld.com/gear/a20795259/pronation-explained, (Accessed: 2019-01-29).

[15] Cespedes Andrea. The degrees of overpronation in running, 2019. Last accessed 20 March 2019.

[16] *Do pronators pronate? Presented at the 6th Symposium on Footwear Biomechanics, Queenstown, New Zealand, July 2003.*, 2003.

[17] PhD Tracy A. Dierks. Pronation in runners: Implications for injury — Lower Extremity Review Magazine.

[18] Bryan C Heiderscheit, Elizabeth S Chumanov, Max P Michalski, Christa M Wille, and Michael B Ryan. Effects of step rate manipulation on joint mechanics during running. *Medicine and science in sports and exercise*, 43(2):296–302, feb 2011.

[19] What is shft. www.shft.run/about. (Accesed: 2019-01-28).

[20] Milestonepod. www.milestonepod.com/. (Accessed: 2019-01-28).

[21] Runscribe. www.runscribe.com/. (Accessed: 2019-01-28).

[22] Moticon. www.moticon.de. (Accessed: 2019-01-28).

[23] Stridalyzer insoles. https://www.retisense.com/. (Accessed: 2019-01-28).

[24] SundeepK. Compactcalendarview, 2019. Last accessed 20 March 2019.

[25] PhilJay. Mpandroidchart, 2019. Last accessed 20 March 2019.

[26] Sepp Hochreiter and Jrgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[27] François Chollet et al. Keras. `https://github.com/fchollet/keras`, 2015.

[28] Jason Brownlee. Stacked long short-term memory networks, Jul 2017.