A Machine Learning Approach to Modelling and Predicting Biomechanical Measures of Strain Across Bone

COSC470 Research Project Report

James Houghton
jho141@uclive.ac.nz

Supervised by
Dr James Atlas
james.atlas@canterbury.ac.nz

Department of Computer Science and Software Engineering
University of Canterbury
Christchurch, New Zealand

October 2021

Abstract

Accurately assessing bone fracture healing requires years of specialty training assessing radiographs which thereafter can still result in an incorrect diagnosis. Providing an ability to track and predict bone strength over time would allow for physicians to better assess patient recovery time. Our overall problem can be broken down into three sub-problems: determining what activity is occurring at a given time, analysing changes to that activity in subsequent occurrences, and correlating those changes with a biomechanical model. This research project presents activity recognition methods to address sub-problem one and identifies the time series extrinsic regression model to address sub-problem two. A time series forest model is compared with a convolutional neural network for activity recognition, with the time series forest model achieving an average leave one out cross validation accuracy of 85.7% on an interpolated dataset.

Acknowledgements

I would like to thank Dr James Atlas for his continuous guidance and support throughout the project. Dr Deborah Munro for her understanding and patience when discussing biomechanical engineering concepts. Simon Blue and Sebastian Jones for their intellectual discussions and collaboration in acquiring data for the project.

Contents

1 Introduction

Accurately assessing bone fracture healing requires years of specialty training assessing radiographs which thereafter can still result in an incorrect diagnosis. Improving the quality of data provided to a physician would allow for more accurate assessment of a fracture [1]. Furthermore, providing an ability to track and predict bone strength over time would allow physicians to better assess patient recovery time resulting in either a decrease in recovery time or if healing is not occurring, it allows the physician to do an intervention. Our overall problem can be broken down into three sub-problems: determining what activity is occurring at a given time, analysing changes to that activity in subsequent occurrences, and correlating those changes with a biomechanical model.

Each sub-problem, while related, requires individual methods to solve. Sub-problem 1 is to determine what activity is occurring at a given time. The data generated from the novel sensor is time-series strain data, therefore, traditional time series classification (TSC) models have been investigated and compared with a novel neural network architecture. The models implemented were trained to classify the activity occurring based on a fixed-length strain sensor signal.

Sub-problem 2 is to analyse changes in subsequent occurrences of the activity. Again, the data generated is time-series strain data, however, the learning task is different. Where TSC-based methods were used for sub-problem 1, time series regression-based methods are required for sub-problem 2. Analysing changes in subsequent activity occurrences is a task that requires learning a mapping between functional data and an external value, therefore we investigate time series regression and time series extrinsic regression methods in relation to sub-problem 2.

Sub-problem 3 is to correlate the activity changes found in sub-problem 2 with a biomechanical model of healing. This is crucial to effectively providing an overall solution. Recent bone healing models have been investigated to gain a deeper understanding of how to structure a pipeline that goes from strain sensor output to a value that quantifies the level of healing that has occurred.

This report is structured as follows. In Section 2, we review relevant literature and background concepts required for each of the three sub-problems. In Section 3 we discuss the prior work of the wider project. Section 4 describes the method used for sub-problem identification. Section 5 discusses sub-problem one. Section 6 discusses sub-problem two. Section 7 summarises the results and future work is discussed in Section 8.

2 Background and Related Work

2.1 Human Activity Recognition

Human activity recognition (HAR) is a popular topic within computer vision. HAR aims to recognize and classify the activities of one or more agents from a series of observations on the agents' actions. It has many applications in a variety of fields, such as human-computer interaction, video surveillance systems and healthcare monitoring systems [2, 3]. Most research has been targeted towards classifying human activity using camera and video.

Recently, there have been a number of studies that aim to classify human activities based solely on sensor output [4, 5]. These studies used convolutional neural networks (CNNs) to classify time series data obtained from a sensor that was either embedded in a transtibial prosthetic or carried by the subject. In [4], they develop a novel locomotion mode recognition model based on a one-dimensional signal from a strain gauge sensor in the subjects transtibial prosthetic. The CNN they developed was trained on raw data from the sensor which shows that CNNs have the capacity to extract relevant features from a raw signal. In [5], they use a triaxial smartphone accelerometer to gather data from subjects undergoing a particular activity. The three-dimensional acceleration data was transformed into vector magnitude data and then used as input to a 1D-CNN. In [6], they use a CNN to analyse vibroarthrographic signals obtained from a dataset published by [7]. In model construction, they convert the one-dimensional signals into two-dimensional image data in the form of spectrograms.

2.2 Time Series Data

Before discussing the methods available to analyse time series data, we introduce two formal definitions for univariate and multivariate time series data.

Definition 1: A univariate time series $X = [x_1, x_2, \ldots, x_T]$ is an ordered set of real values. The length of X is equal to the number of real values T.

Definition 2: A multivariate time series (MTS) $X = [X^1, X^2, \ldots, X^M]$ is a set of M different univariate time series $X^i \in R^T$.

There has been much research into working with time series data for both classification and forecasting tasks [8]. On the other hand, time series regression (TSR) has not received as much attention [9]. Recently, a subset of TSR, known as time series extrinsic regression (TSER), has been presented [citation]. In the following subsections we review the time series classification (TSC) task and the TSER task.

2.2.1 Time Series Classification

TSC remains a challenging problem within data science today. Increases in temporal data availability and the ever-growing applications of TSC have led to hundreds of algorithms being proposed in recent years [8]. A popular approach to TSC has been to use a nearest neighbour (NN) classifier along with a distance function. Dynamic time warping (DTW) has been shown to be a strong distance measure when used with a NN classifier [8]. Ensemble methods using NN classifiers with different distance functions have also been shown to outperform all of the ensemble's individual components [10]. This has led to recent contributions focusing on developing ensemble methods that significantly outperform NN with DTW. A new ensemble method called COTE (Collective Of Transformation-based Ensembles) was developed in [11]. It uses an ensemble of 35 different classifiers over different time series representations. This was extended to use a Hierarchical Voting system to become HIVE-COTE which was a significant improvement over COTE [12]. HIVE-COTE uses a probabilistic voting system with two additional classifiers and two additional representation transformation domains [13]. An issue with HIVE-COTE is it becomes computationally intensive when being used on a real big data mining problem [13]. Until recently, HIVE-COTE was the state-of-the-art method for the TSC task.

In 2020, Dempster et al. proposed the Random convolution kernel transform (Rocket) classifier which achieves state-of-the-art accuracy with less computational expense than existing methods [14]. Rocket uses a large number of convolution kernels to transform the time series and trains a ridge regression classifier. These kernels have random parameters, such as weights and length and, when applied to a time series produce a feature map. For each feature map, the maximum value and the proportion of positive values are computed. This gives two features per kernel which combined with the default 10,000 kernels produces 20,000 features of the input time series. Rocket achieves state-of-the-art performance when benchmarked on the 85 TSC datasets [15] and has been shown to be more accurate than both HIVE-COTE and InceptionTime [16]. Although Rocket was designed for classification tasks, it can be adapted to regression tasks by exchanging the ridge regression classifier with a ridge regression model.

2.2.2 Time Series Extrinsic Regression

Time Series Extrinsic Regression (TSER) is a task that aims to predict numeric values which are dependent on the entire time series [9]. This differs slightly from time series forecasting (TSF) in which the goal is to predict future values in the sequence. An example of TSF would be stock price prediction based on the previous stock prices. TSF usually assumes that the most recent values in the series are more closely related to future values than distant past values. This is a point of difference between TSER and TSF. TSER is closely related to TSC with the main difference being that TSC predicts a categorical class label and TSER predicts a continuous scalar value [9].

TSER could be considered an example of scalar-on-function regression (SoFR). This is part of a widely studied topic within statistics called functional regression [17]. Functional regression can be split into three categories: 1) scalar responses with functional predictors (SoFR); 2) functional responses with scalar predictors (function-on-scalar regression); and 3) functional responses on functional predictors (function-on-function regression). In the case of TSER, the functional data are the time series, where the values present are a function of time. Representing the time series data in its functional form allows SoFR to apply a basis function, such as a Wavelet or Fourier Transform, to reduce noise in the data. A regression model is then fitted to the smoothed data to predict a scalar value.

Examples of TSER in the machine learning community are sparse. However, there have been a number of specialized applications using photoplethysmogram (PPG) sensors. One such application aims to predict heart rate (HR) from the PPG sensor signal [18]. Similar to [6] for HAR, the methods rely on spectral analysis. [18] shows that these methods are not very accurate and proposes a new CNN-based approach that is significantly more accurate compared to the existing spectral methods.

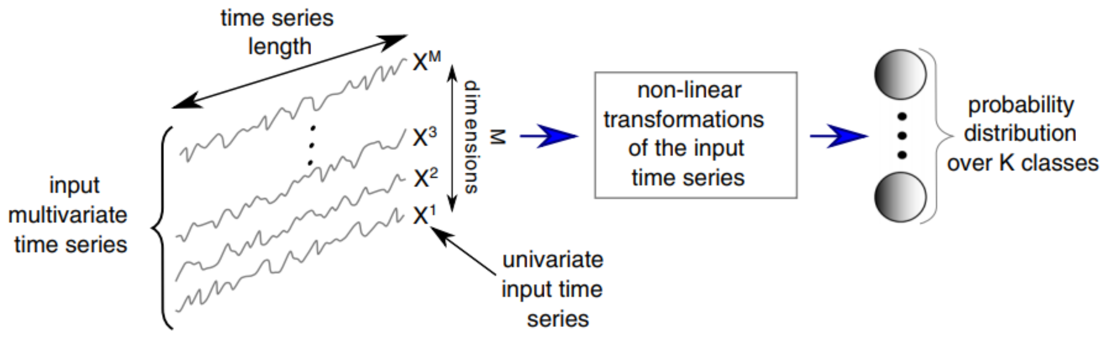
time series length
X^M
X^3
X^2
X^1
M dimensions
input multivariate time series
univariate input time series
non-linear transformations of the input time series
probability distribution over K classes

Figure 1: Deep learning framework for time series classification [13]

2.2.3 Deep Learning for Time Series Data

Deep Neural Network (DNN) architectures have seen many successful applications in recent years. CNNs have revolutionised computer vision, with some architectures reaching human-level performance in image recognition tasks [13]. There have also been several DNN architectures proposed for solving natural language processing (NLP) tasks such as learning word embeddings [19] and document classification [20, 21]. The sequential nature of data within NLP tasks provides a link with the TSC task and shows promise towards DNN methods being successful in TSC.

Deep neural networks are designed to learn a hierarchical representation of the input data. In the case of time series data, this is an ordered set of real values. A dataset is then a collection of (X, Y) pairs where Y is the label for the corresponding time series X. Figure 1 shows a general deep learning framework for the TSC task. DNNs consist of a collection of layers which in turn consist of neurons. Each layer takes as input the previous layer's output and applies a non-linear function to compute its own output. A set of parameters, called weights, are used to control the behaviour of the non-linear transforms. These weights then link the previous layer's output to the current layer's input. This process is referred to as feed-forward propagation.

A training dataset is used to learn the values of the weight parameters used by the network. The dataset is composed of a number of known input-output examples which allows the network to make a prediction based on the input and then evaluate its prediction against the known output. The prediction is evaluated using a cost function that computes the prediction error. Gradient descent is then used to update the weights in a backward pass over the network. This process is known as backpropagation.

A forward pass followed by backpropagation allows the network to update its weights such that the cost function is minimized. Repeating this process allows the network to learn a mapping through different representations of the input data to the output. The network is then tested on unseen data (usually referred to as the test set). A forward pass is performed on unseen input followed by a prediction. Performance of the model can then be measured using a metric such as accuracy. Testing on unseen data allows the developer to investigate how generalizable the model is.

2.2.4 Convolutional Neural Networks

Recent successes in different domains such as image recognition and NLP have provided new motivations for applying CNN architectures to time series data [13]. A CNN can be used for either a classification or regression problem. A CNN works by applying a sliding filter, called a convolution, to the time series. The filter applies a generic non-linear transformation to the data. An example would be to set the filter values equal to $[\frac{1}{2}, \frac{1}{2}]$ and convolve this with the time series to obtain a moving average with a sliding window of length 2. The result of such a convolution on a time series X can be considered as another time series C that underwent a filtering process. Applying several of these filters across a time series allows the network to learn multiple discriminative features of the time series.

A powerful property of CNNs is weight sharing. It allows the CNN to learn filters that are invariant across the time dimension. Filters are learned automatically by applying the convolution operation followed by a discriminative classifier or regression function.

The discriminative classifier or regression function is usually preceded by a pooling operation. Pooling operations can be either local or global. Local pooling operations such as average or max pooling take an input time series and applying an aggregation over a sliding window of the data. The result from such a pooling operation reduces the length of the time series by a fixed amount that is dependent on the stride and window size. Stride is the distance to move the window before applying the next pooling operation. Global pooling operations aggregate over the entire time dimension resulting in a scalar value.

To produce the prediction from the network a final discriminative layer is used. It takes as input a representation of the time series, which is the result of the convolutions, and returns either a probability distribution over the class labels (classification) or a scalar value (regression). This is usually accomplished using the softmax activation function for classification or a linear activation for regression.

2.2.5 Deep Learning algorithms for TSC and TSER

Deep learning models are capable of predicting both discrete labels (classification) and continuous values (regression). Fundamentally, the output of a neural network is a continuous value. To make this into a discrete output for classification tasks, an activation function, such as softmax, is used to compute the class probabilities, the highest of which is the network's class prediction. The loss function must also be changed when moving from a regression to a classification problem. For classification, the categorical cross entropy loss function may be used, but for regression this may be replaced by either the mean squared error or the mean absolute error.

Much of the algorithms that have been successful in the TSC task are able to be adapted to perform a regression task. In [13], they compared several neural network-based algorithms and found that Residual Networks (ResNet) were the best univariate time series model when benchmarked on the 85 univariate time series datasets [15]. Fully Convolutional Networks were found to be the most accurate deep learning model when tested on 12 multivariate time series datasets [22] and the second most accurate when using univariate time series data. In [9], they statistically compare and evaluate several algorithms (both classical regression and deep learning based) on the TSER task. They found that there is no statistical significance between the state-of-the-art time series algorithms and classical regression algorithms. They also ranked the algorithms based on the relative root mean squared error (RMSE). The results from this comparison showed that Rocket performed best overall with the lowest average RSME ranks followed by the other state-of-the-

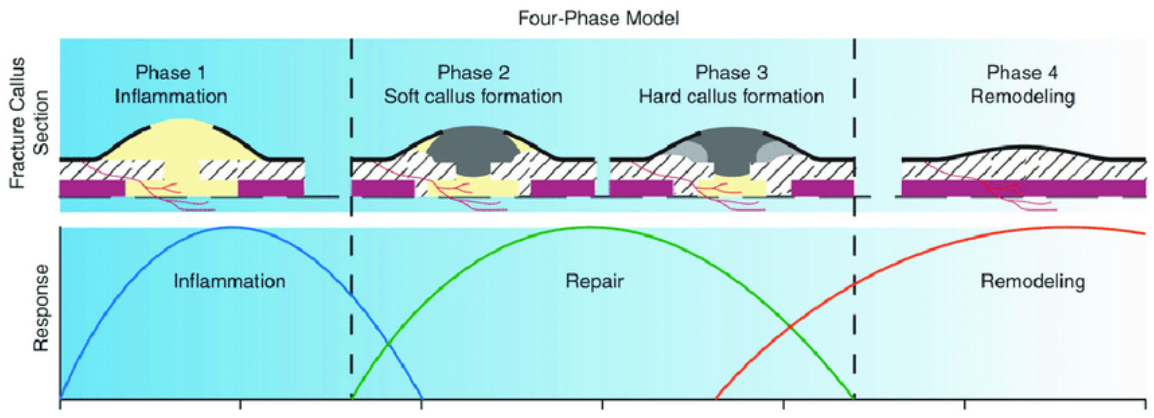

Figure 2: Biological phases of healing; inflammation, reparative, and remodelling. [23]

art algorithms. Their results showed that SoFR algorithms are also competitive as they are not statistically significant from the classical regression algorithms.

2.3 Modelling Bone Fracture Healing

Bone fracture healing is a physiologically complex process that involves both biological and mechanical aspects. Over the past decade, bone healing simulation and modelling have been used to better understand the mechanisms involved [23]. This has led to new methodologies of bone healing simulation such as conceptual modelling, biological modelling and mechanobiological modelling. The following subsections provide brief details of these model types as well as covering the different stages involved in fracture healing from both a biological and biomechanical perspective.

2.3.1 Biological phases of fracture healing

There are three biological phases of bone fracture healing: the inflammatory phase; the reparative phase; and the remodelling phase. Figure 2 shows the biological phases of healing. Each phase serves a distinct purpose during healing. The inflammatory phase is characterised by swelling and pain at the injury site. A haematoma (blood clot) is formed which immobilizes the fracture and acts as a scaffold for the inflammatory cells to begin the healing process. This process is similar to other biological cascades, where subsequent steps are predicated on previous steps [23].

The reparative phase of bone fracture healing begins approximately one week after the formation of the haematoma. Initially, a soft callus of woven bone is formed by a type of bone cell called osteoblasts found in bone marrow. This initial callus is called the provisional callus, and as calcification proceeds, it increases in rigidity to form the bony callus. It takes approximately one to two weeks for the osteoblasts to produce enough woven bone to reach its full-size. Calcification of the bony callus continues until bony fusion is achieved, with the new bony callus always being greater in cross-sectional area than the original bone [23].

The increase in cross-sectional area of the bone during the reparative phase results in a less mechanically efficient structure. The remodelling phase uses modelling and remodelling processes to restore the bones to their original shape and size whilst maintaining overall strength, therefore optimising mechanical efficiency.

Sufficient mechanical strength occurs after the woven bone callus has formed but before final calcification and determining when the bone is “fused enough” is the subject of this research project.

2.3.2 Biomechanical stages of fracture healing

Mechanical factors such as strain, pressure, stability, and fluid velocity are important parameters which act as stimuli for tissue formation during bone healing [23]. A standard way to study fracture healing has been to mechanically test in the lab the healing bone in bending and in torsion and compare it with the stiffness of intact bone [24].

In 1977, White et al. defined four biomechanical stages of healing [25]. Stage I is characterized by seeing no stiffness across the bone, and failure occurs through the original fracture line. This stage corresponds to the inflammatory and the early reparative phases of healing. For Stage II, substantial stiffness is encountered; however, the original fracture still remains as the failure point. Stage II corresponds to the middle of the reparative phase when a bony callus has formed but has not become as large or calcified as it will become. During Stage III, there is no further increase in stiffness, but the failure point is now partially through the fracture site and partially through intact bone. In Stage IV, failure occurs through the intact bone instead of the callus at the fracture site. Stages III and IV correspond to the late reparative and early remodelling phases.

2.3.3 Model types

Researchers have developed various models to describe bone healing, including conceptual, biological, and mechanobiological. These are described in further detail below.

2.3.4 Conceptual/Mechanical

Conceptual models such as, [26] describe the healing process in terms of bone-forming and bone-resorbing processes. Another, [27] describes the process solely in terms of strain. These models are best used for gaining a deeper understanding of bone healing processes. Conceptual models provide approximations of the temporal aspect of healing by using mathematical models based on parameters in a conceptual model. For example, when analysing the healing response in terms of strain, as in [27], the healing response can be seen as a mathematical function of strain dependent on time. These simple models are crucial to formulating and understanding the healing process. This research project uses conceptual models to gather data.

2.3.5 Biological

Biological models use systems of partial differential equations (PDEs) to describe the change in concentration and density of bone cells. These models focus on the cellular activity of the healing bone and the growth factors that regulate these activities. PDE-based biological models focus on the evolution of bone healing biology but neglect mechanical factors, causing them to be poor for explaining mechanical issues such as the effect the rate of loading has on healing. However, they are useful for understanding cellular changes during bone healing.

2.3.6 Mechanobiological

Mechanobiological modelling processes start from the initial phase of healing at the fracture site and consider both mechanical and biological properties. Finite element analysis (FEA) is used to calculate mechanical parameters such as stress, strain, pressure, and fluid velocity around the fracture site. These parameters are then used to simulate the biological processes by solving the PDEs corresponding to the biological processes of interest. Different values of mechanical loading and deformation lead to different tissue types being formed. By allowing tissue formation to occur through updating of the tissue type based on conditions, material properties and geometry are revised in each iteration and used for calculations in the subsequent step. This type of fracture healing modelling is highly informative, but computationally expensive. Developing a model of this type would allow simulated data to be generated which could then be used to develop a machine learning bone fracture healing model. Our project has access to an FEA model of a sheep spine with a signal input that simulates gait.

2.3.7 Sheep Gait Analysis

Sheep gait is a promising avenue to analyse different forms of bone strengthening. Recent studies [28, 29, 30] have shown that analysing gait in sheep can provide valuable insight into the strengthening process of the observed bone. Furthermore, gait analysis is a widely studied topic for many animals [30] which gives rise to the possibility of using this project's novel sensor in other animals. Our research uses sheep gait analysis for two reasons: 1) Dr Munro's spinal fusion research uses sheep as a model due to their anatomical similarities to humans [31]; therefore, our available gait data is entirely sheep-based; and 2) the simulated finite element analysis (FEA) model is a reconstructed 3-dimensional scan of a sheep spine. Gaining deeper insight into sheep gait will be a valuable resource for aiding the design and calibration of future simulations and will thereafter result in an improved machine learning-based bone fracture healing model.

Analysing sheep gait is a useful tool for monitoring bone healing and general bone health [30]. Features are able to be extracted using various gait analysis techniques which characterise bone healing, such as peak vertical force [28]. These features are highly informative about the amount of healing that has occurred, and in one particular study, the level of union was subsequently confirmed using radiographs [32]. They investigated the dependence of gait conditions on the amount of callus formation during bone transport with an external bone fixation device attached. Six merino sheep were used in two groups: 3 in the control group, and 3 in the distraction group. Gait parameters were recorded using a force plate that the sheep were guided over. The sheep were monitored approximately weekly from surgery to one-year-post surgery by recording the ground reaction force (GRF) against time as the sheep walked over the force plate. The results showed that, as healing occurs, the gait parameters exponentially converge towards the values of healthy sheep, and that the walking profile of a healthy sheep differs significantly from that of one carrying an injury. Furthermore, a statistical model was able to be fitted that correlated force data to healing progress.

Sheep gait analysis has also been used to identify differences between healthy sheep and ones that have sustained a spinal cord (SC) injury [33]. The researchers analysed the gait of 17 healthy sheep and one injured sheep walking on a treadmill. A 3D model of the gait was created using infrared reflective markers on the sheep and 6 cameras to monitor the space they moved in. Parameters such as angular motion and velocities of the hock joint were extracted from the model. The 3D kinematics of the limbs were extracted over the gait cycle and values were compared pre- and post-operatively. The results showed that meaningful features related to bone healing could be extracted from analysis of sheep gait. Results such as these will help others gain a more detailed understanding useful for defining new measures of bone healing and applying these to other applications.

Other methods for analysing sheep gait include recording force-based parameters from either a pressure-sensitive walkway or a force plate [30, 33] or using accelerometers placed at different locations on a sheep and analysing the stance and swing phase duration of the gait cycle [34]. In [29], peak vertical force (PVF) was recorded for 21 clinically healthy sheep that were directed to walk over a pressure sensitive walkway. The sheep were divided into three groups based on age, G1, G2, and G3, for younger to older sheep respectively. A significant difference in the recorded force was found between groups G1 and G3, in both the forelimbs and hindlimbs. This showed that young healthy sheep differed from older sheep in the vertical forces they exerted when walking.

Accelerometer data has also been used to discriminate between sound and simulated lame gait movement in sheep [34]. Lameness was simulated by restricting the front right leg of the sheep using a bandage. Triaxial accelerometer data was collected through the use of collar, leg, and ear–attached accelerometers. The data was partitioned into 10-second mutually exclusive behaviour windows and then input to a quadratic discriminant analysis (QDA) model. The final classification model was trained to classify five activities: sound walking, sound grazing, sound lying, sound standing, and lame walking. The ear-attached accelerometer achieved the best accuracy of 82%, concluding that a tri-axial accelerometer attached to the ear-tag of a sheep could successfully discriminate between sound and lame activities in sheep.

3 Prior Work

A novel microelectronic strain gauge sensor has been developed by Dr Deborah Munro and her team in UC's mechanical engineering department [31]. Initially the sensor was designed to determine the strength and stability of a posterolateral spinal fusion in an in vitro sheep model. The sensor was attached to a titanium rod that was surgically implanted at the fusion site and then simulated healing was performed using a compound known as bone cement (polymethylmethacrylate). It was hypothesised that the strength of the healing bone would be directly correlated to the strain across the titanium rod. The results showed that there is a correlation between strain and spinal fusion as well as promise to develop a computer algorithm to track and predict bone fracture healing.

SENG402 Project 20 began developing the computer algorithm hypothesised above [35]. The algorithm developed was aimed at classifying activities based solely on the novel strain sensor data stream. The strain data that was used in the project was gathered from a drill press set up that aimed to simulate moments generated by sheep movement such as walking. Initially, fourteen activities were to be simulated and then classified by the algorithm. This number of activities was determined to be excessive and was reduced to six activities in the subsequent trials. The data recorded from the sensor are strain measurements as a function of time. This allowed methods from Time Series Classification (TSC) and Human Activity Recognition (HAR) to be used to classify the activities contained within the time series.

A Time Series Forest (TSF) classifier was implemented using Scikit-Learn's Sktime library. TSF was then used to classify the different activities being simulated. The activities simulated were: 1) Resting with no strain; 2) Crouching; 3) Standing with normal strain; 4) Standing with heavy strain; 5) Walking with slow stride; 6) Walking with fast stride. Leave-one-out cross-validation (LOOCV) was used to assess the model. It involved leaving one instance from training data as validation and predicting with all permutations of the data. TSF achieved a LOOCV average accuracy of 0.809. A k-nearest neighbours (kNN) method was also used which achieved a LOOCV average accuracy

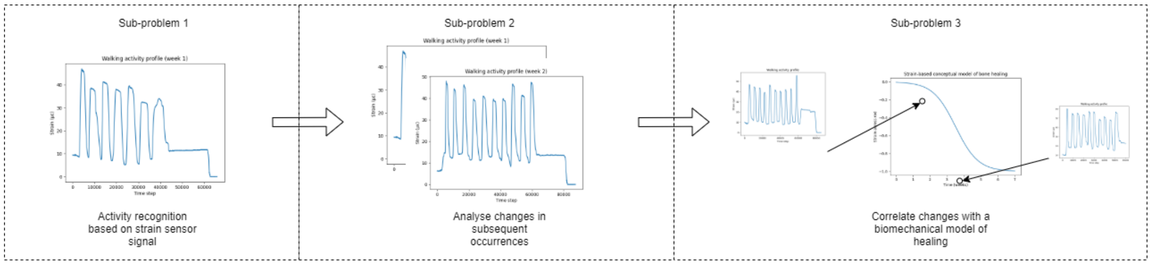
Sub-problem 1
Walking activity profile (week 1)
Activity recognition based on strain sensor signal
Sub-problem 2
Walking activity profile (week 1)
Walking activity profile (week 2)
Analyse changes in subsequent occurrences
Sub-problem 3
Strain-based conceptual model of bone healing
Correlate changes with a biomechanical model of healing

Figure 3: Hypothesised general data pipeline.

of 0.857. kNN achieved a higher accuracy than TSF, however, it was not investigated further, and the reason is unknown. The results showed that it is possible to classify different activities based on a strain signal obtained from the novel sensor developed by Dr Munro.

Project 20 was almost entirely focused on the first sub problem: activity classification. Project 20 was limited by its dependence on the quality of the protocol and physical experimentation. Since the data was gathered from an experimental set up, it was not possible to prove that the protocol directly matched an in vivo scenario. The implementation of the models using Sktime did not allow for the easy addition of neural network-based models, such as CNNs. Another limitation of Project 20 was the data pre-processing stage. The data that was obtained from the drill press set up had varying length between instances. Although this is natural due to some activities running longer than others , it was an issue for the models that were implemented. TSF, for example, requires all the time series data to be of equal length. Linear interpolation was used to handle the unequal length data. However, this transformation alters the time scale of the data, effectively removing its impact.

4 Method

The data output by the novel microelectronic strain sensor is univariate time series with strain as the response. A goal for the wider project is to wirelessly record data from the sensor that is implanted in a sheep spinal fusion as it walks over a pressure sensitive walkway. Sheep are notoriously hard to handle and require training to be directed down the walkway [30]. This may result in the sheep standing still on the walkway before walking over sporadically. Therefore, it is expected that when sheep move over the walkway the strain data recorded will contain multiple activities and these activities will require segmentation to accurately predict the stability of the healing bone. This gives us our first sub-problem: to determine the activity occurring at a given time. Determining the activity occurring is a crucial step in the data pipeline as it allows for subsequent occurrences of the same activity to be identified, which is necessary for a solution to sub-problem 2.

The sheep will be directed down the walkway approximately once per month until the spinal fusion reaches maturity. The data generated is now a multivariate time series (MTS) with strain as the response (Sub-problem 2 in Figure 3). It is hypothesised that, as the fusion matures, the strain output will decrease. Sub-problem 2 is to analyse changes in subsequent occurrences of an activity. Although the data is still time series based, the target value has changed. Where for sub-problem 1 the target was a label, here the target is an external continuous value, more specifically, it is a numerical value for the level of calcification that has occurred. This makes time series extrinsic regression (TSER) models ideal for this sub-problem. In Dr Munros initial study, bone cement was used to simulate healing of a posterolateral spinal fusion in an in vitro sheep model. This gave ground truth labels for the amount of calcification that had occurred. Data of this type is required for sufficiently training a TSER model to address sub-problem 2.

Correlating the changes found by the TSER model with a biomechanical model of bone fracture healing is the focus of sub-problem 3. This sub-problem is beyond the scope of this project, however, is relevant to understanding what other models in the pipeline are required to produce. For example, this helped in the identification of TSER models as a possible solution to sub-problem 2. Figure 3 shows a hypothesised general data pipeline and where each sub-problem fits within.

4.1 Sub-problem one analysis

Sub-problem one is determining the activity occurring as the sheep is directed down the walkway. Since the novel strain sensor generates time series data and an activity is a discrete label, we have a time series classification (TSC) problem. An application of the TSC task is human activity recognition (HAR). The HAR task is to recognise and classify the activities of one or more agents from a series of observations on the agents' actions. While sub-problem one is focused on determining the activity occurring for a sheep, the methods used in HAR have provided valuable insight into developing models to address sub-problem one.

The data recorded from the novel sensor combined with a discrete label makes sub-problem one fundamentally a TSC task. After reviewing the literature surrounding the TSC task, HAR and, deep learning architectures for TSC, it was determined to proceed forward by reimplementing the traditional models used in Project 20 and comparing these with a novel neural network-based architecture.

4.2 Sub-problem two analysis

The sheep will be directed down the walkway approximately once per month until the spinal fusion reaches maturity. Therefore, each month a new activity profile will be generated. Upon successful recognition of the activities occurring in the profile, changes in subsequent occurrences of the same activity are to be analysed. This is the goal of sub-problem 2: to analyse changes in subsequent occurrences of the same activity. Specifically, the task is to map a multivariate time series input to an external continuous value for the level of healing that has occurred. Predicting a continuous target value based on a time series input is the goal of the time series regression (TSR) task.

The TSR task is similar to the TSC task with the main difference being that the target value is continuous in the TSR task. An example of the TSR task is time series forecasting (TSF) in which the goal is to predict the next value in the series. Where the TSF task predicts future values in the time series, sub-problem 2 is aiming to predict a value external to the input series. Recently, Tan et al. introduced the time series extrinsic regression (TSER) task: predicting an external continuous value that is dependent on the entire input time series [9]. This leads to the investigation of the TSER task and how it can be applied to sub-problem 2.

5 Sub-problem one: Determining the activity

5.1 Introduction

Sub-problem one is focused on determining the activity occurring as the sheep moves over the walkway based solely on the strain sensor output. This is a time series classification (TSC) task. There have been many methods proposed for the TSC task, for example k-nearest neighbours coupled with a distance function has been shown to be a strong TSC method [8]. Recently, neural network-based models, such as Residual networks (ResNet) [13] or Fully Convolutional Networks (FCNs) [22] have been used for the TSC task.

An application of the TSC task is human activity recognition (HAR). The HAR task is to recognise and classify the activities of one or more agents from a series of observations on the agents' actions. While sub-problem one is focused on determining the activity occurring for a sheep, the methods used in HAR have provided valuable insight into developing models to address sub-problem one. Project 20 began exploring traditional statistical models for this task. A time series forest (TSF) model was implemented and evaluated on a simulated activity dataset. During the project, other models, such as k-nearest neighbours and shapelet transform were explored, but not included in the final analysis. Recently, neural network architectures have been used for HAR [4, 5]. In 2019, Feng et al. proposed a one-dimensional convolutional neural network for classifying locomotion mode in amputee patients [4]. Their results showed that convolutional neural networks have the ability to extract relevant feature information from strain gauge data. Project 20 began implementation of the traditional statistical models, here we reimplement the TSF model and a further two statistical models and compare these with a neural network-based model.

5.2 Prior work data collection

Project 20 simulated 3 − 5 instances of fourteen activities that were considered typical for a sheep using a drill press setup. The data collection procedure aimed to simulate moments generated by sheep movement such as walking. Initially, fourteen activities were to be simulated and then classified by the algorithm. In Project 20, this number of activities was determined to be excessive and was reduced to six activities in the subsequent trials, however, here we use the full set of fourteen activities. For an exact protocol description, please refer to the appendix in Project 20's final report [35]. The data recorded from the sensor are strain measurements as a function of time.

5.3 Data interpolation

During pre-processing in Project 20, interpolation is used to ensure all the time series are of equal length. While this is used elsewhere in the time series analysis [4], it is limited by its obfuscation of the timescale when creating equal length time series. Linear interpolation estimates the values in between known values by fitting a linear line between the two known values. This gives a new estimated value between two known time points, at a fictitious time point. The addition of the estimated value alters the time scale, and therefore how the models interpret the data. For example, one model used was Nearest Neighbours with Dynamic Time Warping, which is largely unaffected by the altered time scale as the distance measure it uses transforms the time scale. However, the Time Series Forest (TSF) model is affected by the interpolation of the data as the time scale is a feature that it can use. Furthermore, one aim of this project is to extend on the work done in Project 20 by comparing the traditional models with novel neural network architectures such as

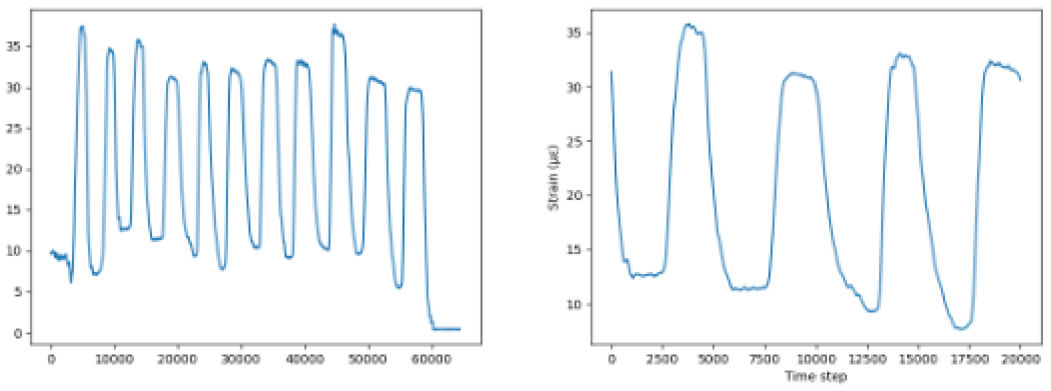

Figure 4: Comparison of the simulated walking activity profile and the window extracted.

a CNN. Any data augmentation, such as linear interpolation, prior to being input to the network may cause the network to recognise the augmentation as a feature instead of true features in the data.

5.4 Data segmentation

To reduce the protocols dependence on interpolation, data segmentation methods were investigated. Segmenting the data into activity specific windows of a fixed size gives equal length data without modification of the time scale. This also allows each window to be of a specific activity instead of a transition period which, in future, could allow for a model to segment the entire data stream recorded from the sensor into mutually exclusive activity windows. Segmenting the data used in Project 20 will also give the developed models more data to train on. For example, instead of interpolating one 20-second-long walking activity profile, it can be broken up into four 5 second mutually exclusive windows each with an instance of the walking activity profile.

Using the fourteen activities and data from Project 20, a set of window data, W, was created based on the following requirements:

1. Equal time for each activity and;
2. The returned window contains useful activity profile information.

The first three instances of each activity were used in the creation of W, the same as the LOOCV procedure. Initially, an automated window creation method was attempted, however, this did not produce segmented data that met the requirements. Instead, a manual approach was used. The first instance of each activity was plotted and inspected for when the activity profile began and finished and compared with the data generation protocol described in Project 20. The lead in and lead out sections of the data were removed, leaving only the window that corresponds to an activity. From these windows approximately 10 seconds of data was extracted. Figure 4 shows the before and after window extraction for a walking activity.

5.5 Model development

The models discussed in this section are specifically addressing sub-problem 1: to determine the activity occurring based on a strain signal recorded from the novel sensor. Project 20 began developing traditional statistics models to address this problem. Here, we extend on the traditional models and investigate a one-dimensional convolutional neural network using the same dataset as in Project 20 and the window dataset described above. The traditional models will first be described, followed by the convolutional neural network.

5.5.1 Traditional Models

Project 20 investigated three statistical models: Time Series Forest (TSF), Shapelet Transform (ST), and k-Nearest Neighbour with Dynamic time warping (kNN-DTW), however, only results for the TSF model were presented. The TSF and kNN-DTW were retrained for comparison with the CNN model, but the ST model was not due to its computational complexity. Where the TSF and kNN-DTW models trained in seconds, the ST model took minutes. kNN-DTW finds the distance between the train data and the test data. The k training instances which are closest to the test data are selected, and the most frequent class is returned as the prediction. TSF is a feature-based classifier that extracts features from intervals of each time series and then trains a classifier based on these features. The model performs a prediction using a majority ensemble voting method. TSF is competitive against 1NN-DTW and has computational complexity that is linear in the length of the time series, making it a promising candidate method.

5.5.2 Convolutional Neural Network

In addition to the traditional models implemented in Project 20, a CNN was developed for comparison. Recent studies in human activity recognition have implemented one dimensional CNN (1D-CNN) architectures that were able to extract meaningful features of the data [4]. The CNNs implemented are designed for processing signal-based data, usually in the form of accelerometer data [5]. Other studies have analysed time series data by first converting to a spectrogram using frequency analysis [2].

The data generated from the novel sensor is signal-based data and, therefore, CNN architectures are a promising starting point for exploring and comparing neural networks to the traditional models evaluated in Project 20. CNNs were designed for being trained with image data with each image having a height, width, and depth (number of channels). However, the strain sensor data is signal-based data and, therefore, cannot be directly input to a CNN. A one-dimensional CNN architecture was adapted from Feng et al. for analysing the strain sensor data [4]. A toy 1D-CNN model was first developed to extend the data pipeline implementation from Project 20 to work with CNN-based models. The toy 1D-CNN was not developed further, nor compared against the traditional models. The architecture developed by Feng et al. was used to classify the locomotion mode of an amputee patient based on output from a strain sensor implanted in a transtibial prosthetic [4]. The strain signal was recorded as the patient walked through a predefined circuit. The model developed by Feng et al. was successful in extracting features from raw interpolated strain sensor data and, therefore, was deemed promising for recognising activities from the novel strain sensor.

5.5.3 CNN architecture and implementation

The 1D-CNNs implemented here are simplified versions of the model developed by Feng et al. Since the window dataset and the interpolated dataset have different lengths, two versions of the 1D-CNN were implemented. The window version, Window-CNN, added two initial layers to the architecture used by Feng at al. to handle the increased length of the window data. The filters of each convolution layer were updated, and the final fully connected layer (layer 12 in table 1) was removed to decrease training time and reduce overfitting. The interpolated version, Interp-CNN, is the same as the Window-CNN without the first two layers. The filters and fully connected layers are the same as for Window-CNN. For both the Window-CNN and Interp-CNN, the Rectified Linear Unit (Relu) activation functions were used for both the convolutional layers and the fully connected layer. The output layer uses the softmax activation function. Details of the architecture used by Feng et al. and the two implemented here are shown in Tables 1 - 3. Both models were implemented in Tensorflow 2.4.1.

Layers	Type	Number of Neurons	Convolutional kernel size	Stride
0	Input layer	1000 x 1	-	-
1	Convolution layer	1000 x 4	5	1
2	Max-pooling layer	200 x 4	-	5
3	Convolution layer	200 x 8	5	1
4	Max-pooling layer	40 x 8	-	5
5	Convolution layer	40 x 16	3	1
6	Max-pooling layer	20 x 16	-	2
7	Convolution layer	20 x 32	3	1
8	Max-pooling layer	10 x 32	-	2
9	Convolution layer	10 x 64	3	1
10	Max-pooling layer	5 x 64	-	2
11	Fully connected layer	320	-	-
12	Fully connected layer	100	-	-
13	Output layer	3	-	-

Table 1: Details of the Feng et al. architecture. [13]

5.5.4 Model training and testing

Both the traditional and CNN-based models were trained using the same LOOCV as in Project 20. The LOOCV procedure involved creating three splits of the dataset. The dataset consisted of three instances of each of the fourteen activities giving 42 total examples. In each split, one instance of each activity was reserved for testing while the remaining two were used for training. The three splits ensured that each activity instance will belong to the test set once, and therefore the average accuracy across all three test splits gives an evaluation of model performance. The CNNs used cross entropy as the loss function, and the Adaptive Momentum Estimation (Adam) optimizer was used for training to minimize loss. The number of epochs was 10, and the learning rate was 0.001. The average accuracy on the 3 test splits is used to evaluate and compare the CNN models with the traditional models.

Layers	Type	Number of Neurons	Convolutional kernel size	Stride
0	Input layer	20000 x 1	-	-
1	Convolution layer	20000 x 4	9	1
2	Max-pooling layer	5000 x 4	-	4
3	Convolution layer	5000 x 8	9	1
4	Max-pooling layer	1000 x 8	-	5
5	Convolution layer	1000 x 16	5	1
6	Max-pooling layer	200 x 16	-	5
7	Convolution layer	200 x 32	5	1
8	Max-pooling layer	40 x 32	-	5
9	Convolution layer	40 x 64	3	1
10	Max-pooling layer	20 x 64	-	2
11	Convolution layer	20 x 128	3	1
12	Max-pooling layer	10 x 128	-	2
13	Convolution layer	10 x 256	3	1
14	Fully connected layer	320	-	-
15	Output layer	14	-	-

Table 2: Details of the window-CNN architecture.

Layers	Type	Number of neurons	Convolutional kernel size	Stride
0	Input layer	1000 x 1	-	-
1	Convolution layer	1000 x 16	5	1
2	Max-pooling layer	200 x 16	-	5
3	Convolution layer	200 x 32	5	1
4	Max-pooling layer	40 x 32	-	5
5	Convolution layer	40 x 64	3	1
6	Max-pooling layer	20 x 64	-	2
7	Convolution layer	20 x 128	3	1
8	Max-pooling layer	10 x 128	-	2
9	Convolution layer	10 x 256	3	1
10	Fully connected layer	320	-	-
11	Output layer	14	-	-

Table 3: Details of the Interp-CNN architecture.

5.6 Results

5.6.1 Interpolated data

The accuracy for each split and the average LOOCV accuracy for the interpolated data are shown in table 4. The bold values indicate the best accuracy for that column. The time series forest (TSF) model performs the best in each test split and achieves the best average LOOCV accuracy of 0.857. In comparison, the interp-CNN achieved an average accuracy of 0.595. The TSF implemented in Project 20 achieved an average LOOCV accuracy of 0.809, however, the better performance of the TSF model here is due to randomization in how the TSF model is initialized. To obtain reproducible results, the same seed was used for all models, therefore resulting in a different accuracy between our TSF model and Project 20's. Project 20 also reported that a k-nearest neighbours (kNN) model achieved an average LOOCV accuracy of 0.857, however, it was concluded that the TSF model and the kNN model has similar accuracies and so were not investigated further. Table 4 shows a far larger gap in the average LOOCV accuracy between TSF and kNN than was found in Project 20. This indicates that the kNN model's success in Project 20 may have been due to a particular initialization of the method as opposed to it successfully extracting discriminatory features from the interpolated strain sensor data. The exact effects of model initialization have not been explored fully here but is an important piece of future work as it will help in model comparison.

The accuracy of the interp-CNN is indicative that the model has failed to learn any meaningful features from the interpolated dataset. A large drawback of neural network-based models is that they typically require a large amount of data to be trained sufficiently. The dataset that interp-CNN was trained on consisted of 28 examples out of 42 total. Furthermore, the number of output classes was large relative to the dataset size making it hard for the network to learn discriminatory features from the data. Given that Feng et al. was able to achieve average 5-fold cross validation accuracies between 0.91 and 0.94 [4] on a larger dataset, it is reasonable to assume that increasing the training dataset size for the interp-CNN would improve performance. Another possible remedy for improving the interp-CNN model performance is to reduce the number of parameters in the network. This can be achieved by altering the filter sizes and the number of neurons present in the final fully connected layer.

5.6.2 Window data

The accuracy for each split and the average LOOCV accuracy for the window dataset, W, are shown in table 5. The bold values indicate the best accuracy for that column. As for the interpolated data, the TSF model performs the best, achieving an average LOOCV accuracy of 0.786. The window-CNN model performs the worst of the three models, achieving an average LOOCV accuracy of 0.548. The window-CNN, like the interp-CNN, has failed to learn any meaningful features from the data, and this is further shown by the 50% accuracy for splits two and three of the window data. The kNN model achieves an average LOOCV accuracy of 0.69 using the window dataset. This an increase when compared to the interpolated data.

The window-CNN has failed to extract any meaningful features from the data. The window-CNN was a larger network than interp-CNN due to the increased length of the time series. Similar to the interp-CNN, increasing the amount of training data available to the network would improve performance.

Model	Accuracy per split			Average Accuracy
	Split 1	Split 2	Split 3	
Time Series Forest	0.857	0.929	0.786	0.857
Interp-CNN	0.643	0.571	0.571	0.595
k-Nearest Neighbours	0.571	0.571	0.571	0.571

Table 4: Accuracies, using the interpolated data, on the testing set for each dataset split and the average accuracy. Bold indicates the best accuracy of each column.

Model	Accuracy per split			Average Accuracy
	Split 1	Split 2	Split 3	
Time Series Forest	0.786	0.923	0.643	0.786
Window-CNN	0.643	0.5	0.5	0.548
k-Nearest Neighbours	0.714	0.786	0.571	0.690

Table 5: Accuracies, using the window data, on the testing set for each dataset split and the average accuracy. Bold indicates the best accuracy of each column.

5.6.3 Comparison of Window data models and Interpolated data models

Both the TSF and interp-CNN trained on the interpolated data outperform the TSF and window-CNN trained on the window dataset. However, the kNN model trained on the window data outperformed the kNN model trained on the interpolated data. TSF is the best performing model for both datasets, however, neither of the CNN-based models were fine tuned during development.

Both models were trained for 10 epochs and used a learning rate of 0.0001. Having such a small learning rate is likely to have impacted the model's ability to learn features within such short number of epochs. Furthermore, the learning rate was not altered for later layers in the network. In the architecture proposed by Feng et al. a learning rate of 0.8 is used in the final fully connected layer of the network [4].

6 Sub-problem two: Analysing changes

6.1 Introduction

Sub-problem two is to analyse changes in subsequent occurrences of the same activity. Sheep will be directed down the walkway approximately once per month until the spinal fusion reaches maturity. Therefore, an activity profile of the sheep moving through the walkway will be generated every month, resulting in a multivariate time series (MTS). Dr Munro hypothesises that, as the fusion matures, the strain output will decrease, indicating that the fusion is strengthening. These changes in strain will also be present in subsequent occurrences of the same activity allowing for a model to be constructed to predict these changes. Where for sub-problem 1 the target value is a discrete label, the target for sub-problem 2 is an external continuous value, more specifically, it is a numerical value for the level of calcification that has occurred at the healing site.

Predicting a continuous value from a time series input is a time series regression (TSR) task. An important distinction to make between the TSR task and sub-problem 2 is that the target value for sub-problem 2 is external to the time series input. Recently, Tan et al. introduced a subset of the TSR task called time series extrinsic regression (TSER) [9]. The TSER task is to predict an external continuous value that is dependent on the entire input time series as opposed to a future value of the input series. This makes TSER models a promising candidate for addressing sub-problem 2: analysing changes in subsequent occurrences of the same activity. Here the external value being predicted is the level of calcification in the healing bone.

In Dr Munros initial study [31], bone cement was used to simulate healing of a posterolateral spinal fusion in an in vitro sheep model. This gave ground truth labels for the amount of calcification that had occurred. Data of this type is required for sufficiently training a TSER model to address sub-problem 2. Acquiring in vivo data of this type is beyond the scope of this project, however, methods for generating simulated healing data are investigated.

6.2 Model Identification

A literature survey was conducted to identify suitable model types for sub-problem 2. Initially, time series classification methods were investigated as it was hypothesised that a discrete value for healing could be used in training, therefore making the task a classification problem, however this was not the case. Recently, Tan et al. introduced the TSER task as a subset of TSR and related it directly to current TSC-based models [9]. TSER is closely related to TSC with the main difference being that TSC predicts a categorical class label and TSER predicts a continuous scalar value [9].

TSER could be considered an example of scalar-on-function regression (SoFR). This is part of a widely studied topic within statistics called functional regression [17]. Functional regression can be split into three categories: 1) scalar responses with functional predictors (SoFR); 2) functional responses with scalar predictors (function-on-scalar regression); and 3) functional responses on functional predictors (function-on-function regression). In the case of TSER the functional data are the time series, where the values present are a function of time. Representing the time series data in its functional form allows SoFR to apply a basis function, such as a Wavelet or Fourier Transform, to reduce noise in the data. A regression model is then fitted to the smoothed data to predict a scalar value.

Examples of TSER in the machine learning community are sparse. However, there have been a number of specialized applications using photoplethysmogram (PPG) sensors. One such application aims to predict heart rate (HR) from the PPG sensor signal [18]. Similar to [6] for HAR, the methods rely on spectral analysis. Reiss et al. show that these methods are not very accurate and proposes a new CNN based approach that is significantly more accurate compared to the existing spectral methods [18].

6.3 Data requirements

Following the identification of TSER models as a proposed method for sub-problem two, the data requirements for sub-problem two were to be established. To train a TSER model effectively, the strain measurements obtained from the novel sensor need to be associated with the level of calcification that has occurred in the healing bone. Dr Munros initial study used data of this type, but more data is required for training a TSER model. Furthermore, it was determined that, initially, the data used to train a TSER model was to be of one activity type: walking. Within the wider project, this gave two options available for acquiring the required data: 1) Collect simulated healing data using an *in vitro* sheep model or, 2) Use a finite element analysis (FEA) model of a sheep spine to simulate forces acting on the spine as healing proceeds.

Due to time constraints, it was infeasible to generate enough *in vitro* sheep data to train a TSER model, therefore, the FEA model was investigated. The FEA model was developed by mechanical engineering master's student Sebastian Jones and is a reconstructed 3D-scan of a healthy sheep spine. An FEA model works similar to the mechanobiological models. The mechanobiological modelling process starts from the initial phase of healing at the fracture site. Finite element analysis (FEA) is used to calculate mechanical parameters such as stress, strain, pressure, and fluid velocity around the fracture site. These parameters then regulate the biological processes by solving the PDEs corresponding to the biological processes of interest, which here is the level of

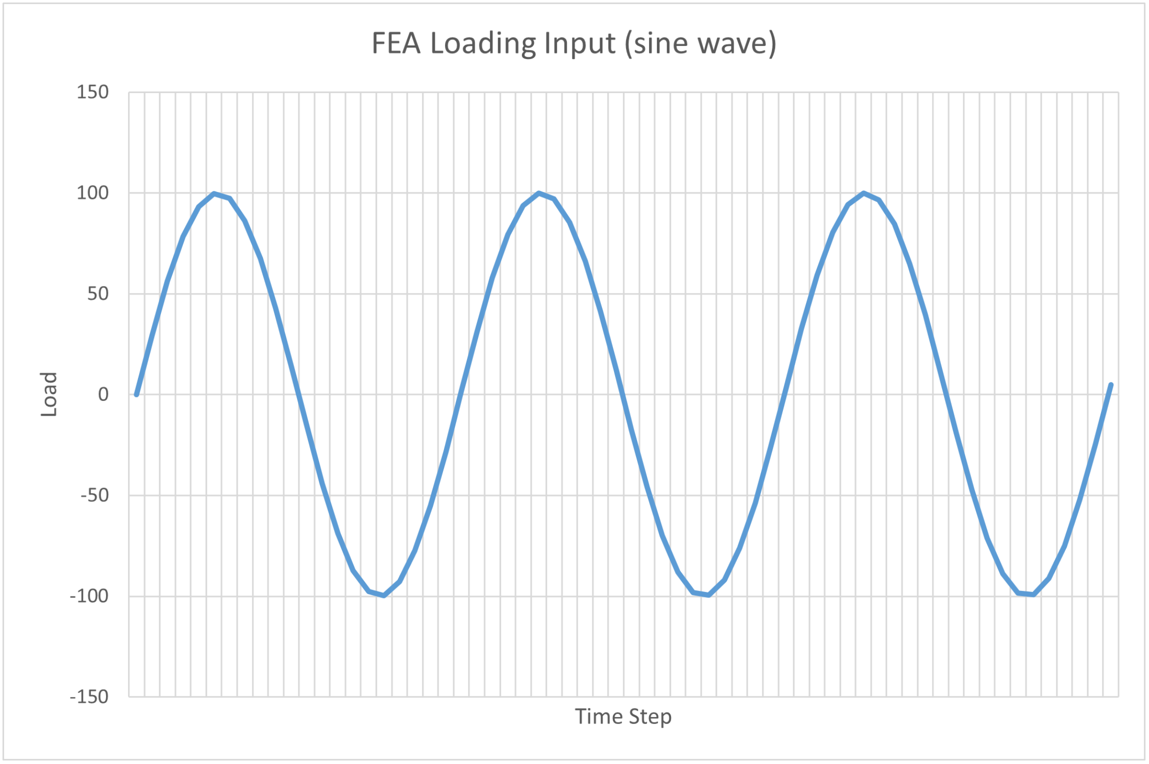

Figure 5: Example loading conditions input to the FEA model.

calcification at the healing site. Different values of mechanical loading and deformation lead to different outcomes in the level of calcification that has occurred. The values that determine this are called the loading conditions.

The loading conditions of the FEA model describe the forces acting on the spine when undergoing a certain activity. Since the initial TSER model is focused on only walking, the loading conditions input to the FEA model should be similar to those observed when a sheep is walking. In general, walking is the most common activity the sheep will be performing when being monitored. Furthermore, recent studies [28, 29, 30] have shown that analysing gait in both sheep and humans can provide valuable insight into the strengthening process of the observed bone. Gait in both sheep and humans is periodic, with popular methods for analysing gait focusing on parameters such as, stance-swing phase duration or peak vertical forces [28]. This periodicity gives a starting point for generating loading conditions for the FEA model.

The loading conditions implemented are based on a sine wave and an example input can be seen in figure 5. Equation 1 shows a basic sine wave.

$$f(t) = A * sin(ft + \phi) \tag{1}$$

The amplitude (A) of the sine wave is analogous to the vertical forces the sheep imparts on the ground when walking, the frequency (f) is analogous to duration of the gait cycle and the phase (ϕ) is analogous to a delayed starting time. These values are all able to be adjusted to produce many

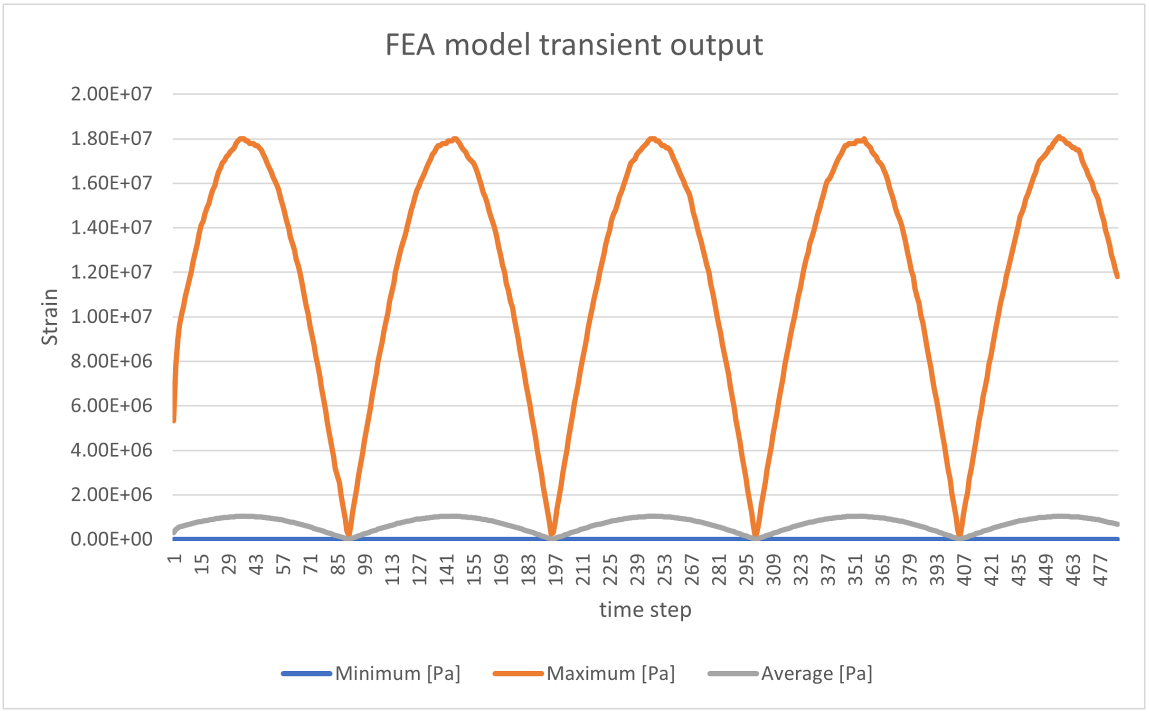

Figure 6: Example FEA model output based on sine wave loading conditions.

different loading conditions. This is a key requirement for the loading conditions. Consider gait in either humans or sheep, individuals in both groups will have a particular gait cycle, producing different values for the parameters that can be derived from the gait cycle. For example, changes in the stride length or peak vertical force (PVF) may be observed between individuals [30]. Therefore, adjusting the input parameters to the sine wave is necessary for generating useful data from the FEA model.

Using these loading conditions, the FEA model can solve for the strain at the position of the sensor implanted in the titanium rod at each time step. This will then give a transient response, in terms of strain, which is dependent on the function used as the loading condition. Figure 6 shows an example output for the loading condition input in figure 5.

Future work using these loading conditions is to generate healing-over-time data for a walking-based activity for different sheep.

7 Summary

This research project investigates a machine learning approach to modelling and predicting biomechanical measures of strain across bone. The overall problem can be broken down into three distinct sub-problems: 1) determining the activity occurring, 2) analysing changes in subsequent occurrences of an activity, and 3) correlating those changes to a biomechanical model of healing.

A goal for the wider project is to wirelessly record data from the sensor that is implanted in a sheep spinal fusion as it walks over a pressure sensitive walkway. The sheep may begin standing still on the walkway before walking over sporadically. Therefore, it is expected that when sheep move over the walkway the strain data recorded will contain multiple activities and these activities will require segmentation to accurately predict the stability of the healing bone. Sub-problem one is to determine the activity occurring at a given time. Combining the univariate time series data recorded from the novel microelectronic strain with a discrete label for the activity gives a time series classification (TSC) problem. An application of the TSC task is human activity recognition (HAR) where the goal is to recognize the activity of an agent based on a sequence of observations of that agents' actions. This work has implemented three models for recognising simulated sheep activities based on the novel strain sensor output. A time series forest (TSF), k-Nearest Neighbours (kNN), and a 1D-CNN were trained using two versions of the simulated data. The first version of the data used linear interpolation before being input to the models, and the second version extracted activity windows from the simulated data. The TSF model performed the best overall, achieving an average LOOCV accuracy of 0.857 using the interpolated data. The implemented 1D-CNN models were not as successful, achieving an average LOOCV accuracy of 0.595 and 0.548 on the interpolated and window data respectively. Future work on this sub-problem will involve fine tuning the 1D-CNN models and gathering more training data to improve performance.

The sheep will be directed down the walkway approximately once per month until the spinal fusion reaches maturity. Therefore, each month a new activity profile will be generated. Analysing changes in subsequent occurrences of the same activity is the goal of sub-problem 2. Specifically, the task is to map a multivariate time series input to an external continuous value for the level of healing that has occurred. Predicting a continuous target value based on a time series input is the goal of the time series regression task. A subset of the time series regression task is the time series extrinsic regression task. The time series extrinsic regression task is to predict an external continuous value that is dependent on the entire time series. A model of this type will be required for addressing sub-problem 2 with the external continuous value being the level of calcification that has occurred at the healing site. A finite element analysis model was developed by mechanical engineering master's student Sebastian Jones to simulate forces acting on a sheep spine. The finite element analysis model was investigated with regard to generating simulated data pertaining to healing-over-time for use in developing a time series extrinsic regression model.

8 Future Work

Sub-problem one is focused on determining the activity occurring based on the strain data recorded from the novel microelectronic strain sensor. Fine tuning the CNN-based models and increasing the amount of training data available to them is an area of future work for sub-problem one. Furthermore, acquiring *in vivo* data and comparing this with the simulated procedure used here is another area of future work.

Sub-problem two is focused on analysing changes in subsequent occurrences of an activity. This project identifies the time series extrinsic regression (TSER) task as analogous to sub-problem two. Acquiring enough data to develop a TSER-based model is necessary to addressing sub-problem two.

References

[1] M. Kanayama, B. W. Cunningham, J. C. Weis, L. M. Parker, K. Kaneda, and P. C. McAfee, "Maturation of the posterolateral spinal fusion and its effect on load-sharing of spinal instrumentation. an in vivo sheep model*," *JBJS*, vol. 79, no. 11, 1997.

[2] S. Ranasinghe, F. Al Machot, and H. C. Mayr, "A review on applications of activity recognition systems with regard to performance and evaluation," *International Journal of Distributed Sensor Networks*, vol. 12, no. 8, p. 1550147716665520, 2016. [Online]. Available: https://doi.org/10.1177/1550147716665520

[3] M. Vrigkas, C. Nikou, and I. A. Kakadiaris, "A review of human activity recognition methods," *Frontiers in Robotics and AI*, vol. 2, no. 28, 2015. [Online]. Available: https://www.frontiersin.org/article/10.3389/frobt.2015.00028

[4] Y. Feng, W. Chen, and Q. Wang, "A strain gauge based locomotion mode recognition method using convolutional neural network," *Advanced Robotics*, vol. 33, no. 5, pp. 254–263, 2019. [Online]. Available: https://doi.org/10.1080/01691864.2018.1563500

[5] L. Song-Mi, Y. Sang Min, and C. Heeryon, "Human activity recognition from accelerometer data using convolutional neural network," in *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*, Conference Proceedings, pp. 131–134.

[6] D. Kraft and G. Bieber, "Vibroarthrography using convolutional neural networks," 2020. [Online]. Available: https://doi.org/10.1145/3389189.3397993

[7] R. M. Rangayyan, S. Krishnan, G. D. Bell, C. B. Frank, and K. O. Ladly, "Parametric representation and screening of knee joint vibroarthrographic signals," *IEEE Transactions on Biomedical Engineering*, vol. 44, no. 11, pp. 1068–1074, 1997.

[8] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh, "The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances," *Data Mining and Knowledge Discovery*, vol. 31, no. 3, pp. 606–660, 2017. [Online]. Available: https://doi.org/10.1007/s10618-016-0483-9

[9] C. W. Tan, C. Bergmeir, F. Petitjean, and G. I. Webb, "Time series extrinsic regression," *Data Mining and Knowledge Discovery*, 2021. [Online]. Available: https://doi.org/10.1007/s10618-021-00745-9

[10] J. Lines and A. Bagnall, "Time series classification with ensembles of elastic distance measures," *Data Mining and Knowledge Discovery*, vol. 29, no. 3, pp. 565–592, 2015. [Online]. Available: https://doi.org/10.1007/s10618-014-0361-2

[11] A. Bagnall, J. Lines, J. Hills, and A. Bostrom, "Time-series classification with cote: The collective of transformation-based ensembles," in *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, Conference Proceedings, pp. 1548–1549.

[12] J. Lines, S. Taylor, and A. Bagnall, "Hive-cote: The hierarchical vote collective of transformation-based ensembles for time series classification," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*, Conference Proceedings, pp. 1041–1046.

[13] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: a review," *Data Mining and Knowledge Discovery*, vol. 33, no. 4, pp. 917–963, 2019. [Online]. Available: https://doi.org/10.1007/s10618-019-00619-1

[14] A. Dempster, F. Petitjean, and G. I. Webb, "Rocket: exceptionally fast and accurate time series classification using random convolutional kernels," *Data Mining and Knowledge Discovery*, vol. 34, no. 5, pp. 1454–1495, 2020. [Online]. Available: https://doi.org/10.1007/s10618-020-00701-z

[15] H. A. Dau, A. Bagnall, K. Kamgar, C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, and E. Keogh, "The ucr time series archive," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 6, pp. 1293–1305, 2019.

[16] H. Ismail Fawaz, B. Lucas, G. Forestier, C. Pelletier, D. F. Schmidt, J. Weber, G. I. Webb, L. Idoumghar, P.-A. Muller, and F. Petitjean, "Inceptiontime: Finding alexnet for time series classification," *Data mining and knowledge discovery*, vol. 34, no. 6, pp. 1936–1962, 2020.

[17] P. T. Reiss, J. Goldsmith, H. L. Shang, and R. T. Ogden, "Methods for scalar-on-function regression," *International Statistical Review*, vol. 85, no. 2, pp. 228–249, 2017. [Online]. Available: https://doi.org/10.1111/insr.12163

[18] A. Reiss, I. Indlekofer, P. Schmidt, and K. Van Laerhoven, "Deep ppg: Large-scale heart rate estimation with convolutional neural networks," *Sensors (Basel, Switzerland)*, vol. 19, no. 14, p. 3079, 2019.

[19] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," 2013.

[20] Y. Goldberg, "A primer on neural network models for natural language processing," 2015.

[21] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," 2014.

[22] M. G. Baydogan and G. Runger, "Learning a symbolic representation for multivariate time series classification," *Data Mining and Knowledge Discovery*, vol. 29, no. 2, pp. 400–422, 2015. [Online]. Available: https://doi.org/10.1007/s10618-014-0349-y

[23] M. S. Ghiasi, J. Chen, A. Vaziri, E. K. Rodriguez, and A. Nazarian, "Bone fracture healing in mechanobiological modeling: A review of principles and methods," *Bone Reports*, vol. 6, pp. 87–100, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352187217300128

[24] N. A. S. R Bruce Martin, David B Burr, *Skeletal Tissue Mechanics*, 1998.

[25] r. A. A. White, M. M. Panjabi, and W. O. Southwick, "The four biomechanical stages of fracture repair," *Journal of bone and joint surgery. American volume*, vol. 59, no. 2, p. 188, 1977.

[26] D. G. Little, M. Ramachandran, and A. Schindeler, "The anabolic and catabolic responses in bone repair," *Journal of bone and joint surgery. British volume*, vol. 89, no. 4, p. 425, 2007.

[27] D. S. Elliott, K. J. H. Newman, D. P. Forward, D. M. Hahn, B. Ollivere, K. Kojima, R. Handley, N. D. Rossiter, J. J. Wixted, R. M. Smith, and C. G. Moran, "A unified theory of bone healing and nonunion: Bhn theory," *The bone & joint journal U6 - Journal Article*, vol. 98-B, no. 7, p. 884, 2016.

[28] J. W. Fernandez, N. Alves, A. Veloso, S. Amado, P. Morouço, R. Silva, I. S. Dimas, and A. C. Maurício, "Sheep gait biomechanics and the assessment of musculoskeletal conditions: A systematic review," *Applied mechanics and materials*, vol. 890, pp. 248–259, 2019.

[29] F. S. Agostinho, S. C. Rahal, F. A. P. Araújo, R. T. Conceição, C. A. Hussni, A. O. El-Warrak, and F. O. B. Monteiro, "Gait analysis in clinically healthy sheep from three different age groups using a pressure-sensitive walkway," *BMC Veterinary Research*, vol. 8, no. 1, p. 87, 2012. [Online]. Available: https://doi.org/10.1186/1746-6148-8-87

[30] P. Seebeck, M. S. Thompson, A. Parwani, W. R. Taylor, H. Schell, and G. N. Duda, "Gait evaluation: A tool to monitor bone healing?" *Clinical Biomechanics*, vol. 20, no. 9, pp. 883–891, 2005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0268003305001221

[31] *Correlation of Strain on Instrumentation to Simulated Posterolateral Lumbar Fusion in a Sheep Model*, ser. ASME International Mechanical Engineering Congress and Exposition, vol. Volume 3: Biomedical and Biotechnology Engineering, 11 2016, v003T04A008. [Online]. Available: https://doi.org/10.1115/IMECE2016-65696

[32] J. Mora-Macías, E. Reina-Romo, J. Morgaz, and J. Domínguez, "In vivo gait analysis during bone transport," *Annals of Biomedical Engineering*, vol. 43, no. 9, pp. 2090–2100, 2015. [Online]. Available: https://doi.org/10.1007/s10439-015-1262-2

[33] S. Safayi, N. D. Jeffery, S. K. Shivapour, M. Zamanighomi, T. J. Zylstra, J. Bratsch-Prince, S. Wilson, C. G. Reddy, D. C. Fredericks, G. T. Gillies, and M. A. Howard, "Kinematic analysis of the gait of adult sheep during treadmill locomotion: Parameter values, allowable total error, and potential for use in evaluating spinal cord injury," *Journal of the Neurological Sciences*, vol. 358, no. 1, pp. 107–112, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0022510X15005274

[34] J. Barwick, D. Lamb, R. Dobos, D. Schneider, M. Welch, and M. Trotter, "Predicting lameness in sheep activity using tri-axial acceleration signals," *Animals*, vol. 8, no. 1, 2018.

[35] J. Orchard and J. Atlas, "Seng402: Project 20," Report, 2020.