

Predicting Epileptic Seizures with a Stacked Long Short-Term Memory Network

Jamie Pordoy¹, Ying Zhang², Nasser Matoorian³, Massoud Zolgharni⁴

¹jamiepordoy@hotmail.com; ²ying.zhang@uwl.ac.uk; ³nasser.matoorianpour@uwl.ac.uk; ⁴massoud.zolgharni@uwl.ac.uk
School of Computing and Engineering, University of West London, London W5 5RF, UK

Abstract- Despite advancements, seizure detection algorithms are trained using only the data recorded from past epileptic seizures. This one-dimensional approach has led to an excessive false detection rate, where common movements are incorrectly classified. Therefore, a new method of detection is required that can distinguish between the movements observed during a generalized tonic-clonic (GTC) seizure and common everyday activities. For this study, eight healthy participants and two diagnosed with epilepsy simulated a series of activities that share a similar set of spatial coordinates with an epileptic seizure. We then trained a stacked, long short-term memory (LSTM) network to classify the different activities. Results show that our network successfully differentiated the types of movement, with an accuracy score of 94.45%. These findings present a more sophisticated method of detection that correlates a wearers movements against 12 seizure related activities prior to formulating a prediction.

Key Words- *Epilepsy, accelerometer, LSTM, seizure detection*

1. INTRODUCTION

Epilepsy is a commonly occurring neurological disorder, affecting an estimated 60 million people (*i.e.*, close to 1% of the global population) worldwide, with an incidence rate of 40-70 new cases for every 100,000 adults [1]. Symptoms manifest in the form of recurrent, unprovoked epileptic seizures, where a hyper synchronous discharge of cortical neurons causes a paroxysmal alteration of one's neurologic function [2]. Epidemiological studies indicate that 20-30% of those diagnosed will not remain in long-term remission as they become refractory to anti-epileptic drugs (AEDs), having a form of drug resistant epilepsy (DRE) [3]. Those with DRE are at an increased risk for early mortality when compared to non-epileptics, with

sudden unexpected death in epilepsy (SUDEP) and spontaneous accidental injury documented as the preliminary causes of death [4].

Although the electroencephalogram (EEG) represents the gold standard for pre-surgical evaluation of epilepsy, short-term monitoring, and early detection of seizure onset [5], it is impractical for use in a non-hospital environment due to extensive operating and labour costs. Therefore, a focus on non-invasive, non-EEG detection methods is paramount as it is the sudden, unforeseen manner by which seizure occurs that poses one of the biggest threats to those diagnosed. If a seizures onset can be detected pre-ictal, then patient supervision and medical intervention can be applied sooner. This would reduce the likelihood of SUDEP as most documented cases are preceded by a generalized tonic-clonic (GTC) seizure [6]. The most promising method of non-EEG detection combines several types of sensory observation to form a multi-modal mechanism that measures various aspects of a patient's motor and autonomic functions [7]. Based on the sporadic and seemingly repetitive movements observed during a GTC seizure, the most promising multi-modal systems are accelerometer (ACM) centric, using machine learning algorithms to analyse differentials in velocity and direction.

1.1. Seizure Detection

A large number of studies in broader literature have validated the use of ACMs, due to their non-invasive and non-intrusive method of detecting GTC seizures [8]. By measuring the motion and velocity of patients in three-dimensions, ACMs have had varied levels of success, with many studies indicating high sensitivity but low specificity when predicting seizure onset [2]. There have been several studies to improve the performance of detection mechanisms centred around transitional accelerometry, notably the works of Van de

Vel *et al.* [11] and Beniczky *et al.* [10], attaining sensitivity scores of 89.7% and 95.71% respectively. Furthermore, a study by Lockman *et al.* [12] successfully detected 87.5% of seizures, however, on closer observation it was noticed that 204 non-seizure movements were also detected. Similarly, a study by Schulc *et al.* [13] produced a set of results on a sample size of 20 adults, measuring 100% sensitivity and 88% specificity. Whilst this study presented several novel findings for spectral and temporal analysis of ACM data, the predominant findings relevant to this study indicated that the false detection rate was affected by the choice of algorithm, as it was unable to distinguish between repetitive movements (*e.g.*, brushing teeth) from the convulsions observed during a GTC seizure. To counteract the recurring pattern of inaccurate classification and false predictions, several studies have attempted to solve this problem at an algorithmic level. Whilst the number of false predictions is well documented in ACM detection studies, there is an observable commonality regarding how models are trained. Algorithms are trained using sequential event prediction, where the measurements from past seizures are used to predict future occurrences. Although this training process has been successful in other domains, the vast spectrum of motions observed during a seizure can vary depending on each person's type of epilepsy and physiological characteristics.

To overcome this problem, detection algorithms need to be able to distinguish different types of non-seizure related movements from the rhythmic jerking and muscle spasms observed during a convulsion [14]. Therefore, detection algorithms need to be trained using data from past seizures and common activities that share similar positions in three-dimensional space. Although literature supports the idea that seizure detection algorithms need to differentiate the spasms and twitches observed during a seizure from common everyday movements, little has been done to achieve this [8]. Therefore, this paper proposes a novel, seizure detection method that applies activity recognition (AR) techniques to detect different types of movement and reduce the frequency of false positive predictions.

1.2. Activity Recognition

The principles of AR and classification of temporal movement have been applied to various facets of human analysis, such as medicine, behavioural

sciences, and physiotherapy [15]. Measuring acceleration in three-dimensional space enables ACMs to quantify human movement and locomotion. Current literature states that common activities such as walking, jogging, and sitting, can be classified using AR techniques to a consistent degree of accuracy between 83%-91% when using two or more ACM devices [16]. Further studies have extended the scope of recorded activities and the practical application of AR, using accelerometer-based systems for real-time fall detection [17], health & fitness monitoring [18], and energy efficiency [19]. There are several studies that have identified the thigh as the optimum location for ACM placement, although the activities used in these studies predominantly focus on ambulatory movement (*e.g.*, walking, jogging) [20]. However, for the detection of an epileptic seizure, ambulatory motion is negated, as the body stiffens and becomes unresponsive, with disequilibrium causing the patient to fall backwards [21]. Furthermore, studies by Ullah *et al.* [22] and Zebin *et al.* [23] have paved the way for deep learning AR, constructing neural networks with a 92% and 93% accuracy, respectively, when classifying commonly occurring activities. Whilst these studies have used a multi-sensor or embedded smartphone ACM to record activities, this study will use a single wrist-worn ACM and sequential deep learning model to detect different activities that share similar movements with an epileptic seizure.

1.3. LSTM

As defined by Hochreiter and Schmidhuber [24], a LSTM network is a recurrent neural network (RNN) architecture which can be used for learning long-term dependencies and modelling the temporal dynamics of time series data. LSTM units use gating to overcome the vanishing and exploding gradient problems encountered with traditional RNNs [25]. A common LSTM unit consists of a cell state, input gate, output gate and a forget gate. The cell state is used to store information based on previous and current intervals, whilst the three gating units act as artificial neurons that protect and regulate the cell state by using activation functions [26]. Furthermore, the output of an LSTM cell is called the hidden state, which can be expressed as h_{t-1} . The gates control the operations of the LSTM cell whilst the sigmoid and tanh activation functions identify the information to update and create new candidate vectors for the cell state [27].

Firstly, the forget gate uses a sigmoid activation function to keep or remove data from the cell state. Cell data is converted to a value between 0 and 1. If the value is equal to 0 then the data is removed from the cell state, else if it equals to 1 the data is kept. This can be expressed mathematically as

$$f_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (1)$$

Where h_{t-1} represents the cells hidden state and x_t is the weighted sum of the cells input data [23].

Next, the input gate layer uses a sigmoid function to determine where a new value will be stored within the cell state. Once identified, a tanh function initiates a new vector of type C_{t-1} where C represents the new candidate vector that will be added to the cell state [28].

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (3)$$

The previous cell state value C_{t-1} is then overwritten with the newly computed cell state value C_t . The forget gate value is then multiplied by the previous cell state where $f_t * C_{t-1}$. Then the input gate value is multiplied by the new cell state value $i_t * \tilde{C}_t$. The input and forget gate values are then combined to update the cell state [28]. This is expressed mathematically as

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (4)$$

Lastly, the output gate generates a filtered version of the cell state's output. A sigmoid function is used to filter the cell state and identify the output value [29]. Then the tanh function is used to translate the cell state value between -1 and 1 as shown in equation (6), expressing the cells hidden output as

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = o_t \cdot \tanh(C_t) \quad (6)$$

1.4. Stacked LSTM Networks

The success of deep learning studies, notably Szegedy *et al.* [30] and Simonyan & Zisserman [31] have indicated that a degree of depth to the underlying

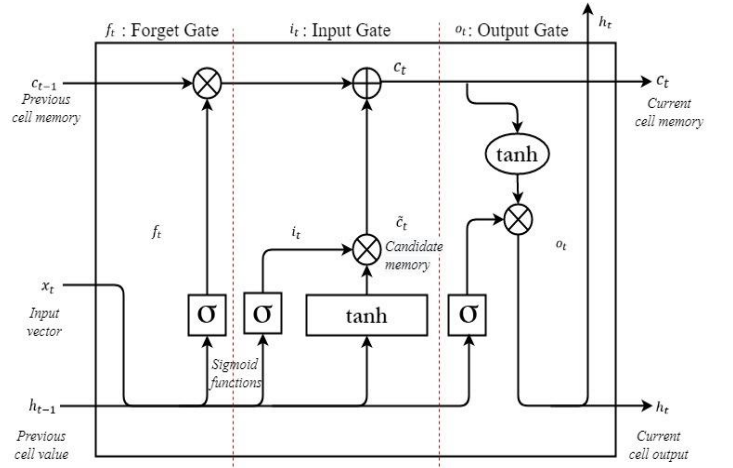


Figure 1: LSTM unit where C_{t-1} = previous cell state, h_{t-1} =cells hidden state, x_t = weighted sum of cell input, C = candidate vector, σ =sigmoid function, \tanh =tanh function, i_t =input gate, f_t =forget gate, o_t =output gate, C_t =cell state

network architecture will improve overall performance. To increase the depth of an LSTM network, cells can be stacked as layered matrices. As LSTM networks process sequential time series data, stacking additional layers adds a level of abstraction of input observations [32]. Whilst stacking additional layers can improve network performance, this only works up to a certain number of stacks [33]. Generally, once two stacks are passed, accuracy levels become saturated, as the network becomes susceptible to the degradation problem [34]. Therefore, finding the optimum number of stacks is paramount, as the number of layers increase, it becomes harder to optimise the network.

2. Methods and Materials

In this study, we collected data from 10 participants (eight healthy adults and two with refractory epilepsy who experience GTC seizures) using a single wrist-worn tri-axial ACM and gyroscope. Spanning a 48-hour period, participants were instructed to wear an accelerometer on their dominant hand and carry out the activities and movements listed in Table 1.

Common gestures	Non-seizure related activities
Walking	Lie-down
Jogging	Watching TV
Sitting	Sleeping
Standing	Brushing teeth
Upstairs	Making sandwich
Downstairs	

Table 1: List of the activities recorded



Figure 2: An experiment using participant 4, where he is simulating the muscle movements of a GTC seizure.

This study used a variation of the list by Lockman *et al.* [12] who identified several movements that have similar wrist mechanics to a GTC seizure, and the common activities listed in the Wireless Sensor Data Mining (WSIDM) dataset [35].

2.1. Seizure Simulation

The notion of using healthy participants to simulate an epileptic seizure has been explored in several studies, most notably by Conradsen *et al.* [36] who used 10 healthy subjects to simulate the motor manifestations observed during a GTC seizure. A trained clinician was used to observe the simulation process, concluding that the observed visual movements were very similar to those of a real GTC seizure. Furthermore, simulation has been used in several multi-modal detection studies, with the movements acquired successfully training seizure detection models [37].

Therefore, based on the successful use of simulated seizures, it is our feeling that the combination of simulated and non-simulated seizures will provide our model with the training data required to differentiate different types of activity. For this study, each participant simulated 4 GTC seizures, two in a supine sleeping position, and a further two in a lateral position to recreate lying down on one's side. Based on current observations it was our feeling that these two positions provided an accurate representation of the positions a patient would experience during a seizure's onset. Video recordings were used to guide participants in accurately recreating the muscle twitches and convulsions seen in a GTC seizure [38], with an allotted time of 80 -110 seconds per simulation.

Participant	Age	Gender	Seizure Duration
1	34	Male	3:20
2	35	Female	4:35
3	45	Male	4:0
4	28	Male	3:34
5	58	Female	3:48
6	34	Male	4:20
7	33	Female	4:13
8	25	Male	4:01
9	29	Female	6:20
10	35	Male	8:10

Table 2: Seizure duration for participants. Participant 1 - 8 are healthy adults, whilst participants 9 - 10 are diagnosed with epilepsy.

2.2. Model Type

Due to the complexities of time series forecasting, we needed to develop a model with the control ability to regulate the flow of long-term dependencies whilst handling the recurring issues RNN's experience during back propagation such as the vanishing gradient problem. Based on the success of Hochreiter and Schmidhuber [24], it was clear that we required a model designed to use multiplicative, hidden gating units to manage long-term memory sequences such as a gated recurrent unit (GRU) or LSTM network. Therefore, an LSTM network with Adam optimisation was chosen as literature dictates longer memory sequences, separate update and forget gates, and a higher degree of accuracy when using long-term dependencies. To further improve our model's performance and depth, we used a stacked architecture as this enables each layer to process sequential data at different timescales [33].

2.3. Feature Selection and Preprocessing

To train our LSTM network, we converted a one-dimensional array to a three-dimensional array. We then split our dataset into a series of fixed length sequences as LSTMs require three-dimensional input [39]. A 25Hz sampling frequency was used to capture the data, recording an estimated 20 records of three-dimensional data every second. Each fixed length sequence spans a 10 second time segment and contains 200 records per fixed length sequence. We recorded 12 GTC seizures from the two adults diagnosed with epilepsy, spanning 14 minutes and 30 seconds. A further 34 seizures were simulated by the remaining participants, totalling 32 minutes and 10 seconds. After

preprocessing, we had 280 fixed length sequences of simulated and non-simulated seizure data to train our model.

2.4. Network Architecture

This methodology presents a stacked LSTM neural network for multi-class classification of fixed length sequences. Figure 3 shows the architectural schematics of our network, consisting of two fully connected layers and two stacked LSTM layers with 64 cells in each. Data normalisation was initiated using the hidden layers linear discriminant function and a rectified linear unit (ReLU) activation, feeding the data to the LSTM cells. The network was developed using the Python programming language, with TensorFlow and Keras deep learning libraries. The following components were used to develop our model.

2.5. L2 Regularisation

To prevent our network from overfitting or incurring generalisation errors, we added an L2 regularizer. Regularisation is the process of modifying an algorithm to reduce its generalisation error without altering its training error value [40]. The L2 regularizer uses the squared magnitude of the coefficient as penalty term for the network's loss function [41]. This is expressed mathematically as

$$\lambda \sum_i w_i^2 \quad (7)$$

where λ represents the regularizers coefficient and w is the value of our models manipulatable parameters (weights and biases).

2.6. Loss Function

This study used a cross-entropy loss (log loss) function for single label categorisation. An L2 regularizer was added to the cross-entropy loss function to reduce overfitting and improve the networks performance and optimisation capabilities [42]. The following formula is a mathematical representation of our network's loss function with added L2 regularizer where w represents the network's parameters, y_i the true class labels and \hat{y}_i the predicted labels [43].

$$LF = -\frac{1}{N} \sum_{i=1}^N [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] + \lambda \sum_i w_i^2 \quad (8)$$

2.7. Softmax Function

The softmax (normalised exponential) function is the final layer of our network and takes an input vector of K values and transforms it into a vector of K values that have a combined sum of 1 [44]. Each K value has a number between 0 and 1, so that they can be used as probability measures for multi-class classification [45]. The softmax function can be written as

$$\sigma(\vec{z})_{i=1}^K = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (9)$$

Where \vec{z} represents the input vector of K values, K the number of activities applied classes and e^{z_j} is the standard exponential function to each K value [44].

2.8. Adaptive Moment Estimation

Adaptive Moment Estimation (Adam) is a stochastic optimisation algorithm that calculates adaptive learning rates for each parameter [46]. Adam combines the advantages of Adaptive Gradient (AdaGrad) and Root Mean Square Propagation (RMSProp) for first-order gradient-based optimisation of stochastic objective functions [47]. Adam computes an exponential moving average of the gradient, squared gradient and parameters β_1 and β_2 . As defined by Kingma & Ba [48], Adam's mathematical structure for stochastic optimisation is express using the pseudo code in Table 3.

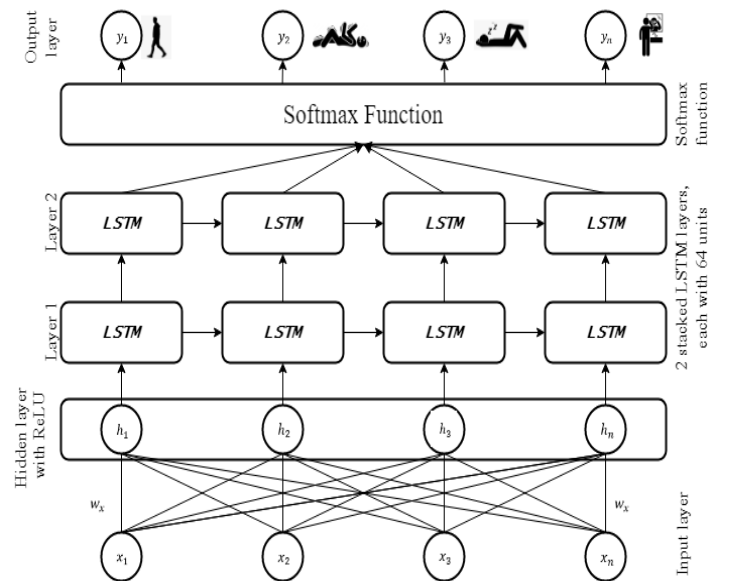


Figure 3: Architectural diagram of our stacked LSTM network with softmax function

Require: α : Stepsize
Require: $\beta_1, \beta_2 \in [0,1)$: Moment estimates exponential decay rates
Require: $f(\theta)$: Stochastic objective function with parameters θ
Require: θ_0 : Initial vector
 $m_0 \leftarrow 0$ (Initialize 1st moment vector)
 $v_0 \leftarrow 0$ (Initialize 2nd moment vector)
 $t \leftarrow 0$ (Timestep Initialization)
while θ_t not converged **do**
 $t \leftarrow t + 1$
 $g_t \leftarrow \nabla_{\theta} f_t(\theta_{t-1})$ (Get gradients from stochastic function where timestep = t)
 $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (Update 1st moment estimate)
 $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ (Update 2nd moment estimate)
 $\hat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ (Calculate bias-corrected 1st moment estimate)
 $\hat{v}_t \leftarrow v_t / (1 - \beta_2^t)$ (Calculate bias-corrected 2nd moment estimate)
 $\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon)$ (Parameters updated)
end while
return θ_t (Resulting parameters)

Table 3: Pseudo code for the ADAM optimisation algorithm

3. Experiments and Results

This section lists the experiments that were performed and the results that were obtained when evaluating the performance of our stacked LSTM network.

Parameters	Value
Training/test data split	75:25
Learning rate	0.0025
Number of epochs	50
L2 loss	0.0015

Table 4: Optimisation parameters and network specifications

3.1. Experiments

The experiments conducted were designed to assess varying aspects of our network's ability to classify different types of movement. Table 4 lists the parameters and specifications used to train our network prior to experimentation. The formulas for each experiment are defined using the following acronyms.

TP: True Positive, FP: False Positive, TN: True Negative and FN: False Negative.

Accuracy measures the number of correct positive observations against the number of total observations [48]. This is expressed mathematically as

$$Accuracy = \frac{TP + FN}{TP + TN + FP + FN} \quad (10)$$

Precision represents the number of correct positive predictions against the total number of positive predictions that were made [48]. Precision is defined as

$$Precision = \frac{TP}{TP + FP} \quad (11)$$

Recall represents the ratio of observations that are correctly predicted as positive against the total number of positive observations. This is expressed as

$$Recall = \frac{TP}{TP + FN} \quad (12)$$

F-measure is defined as the weighted harmonic mean of precision and recall. The F-measure was used to assess the overall performance of the precision and recall experiments and present a balanced average of the two measures combined [50].

$$F - Measure = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (13)$$

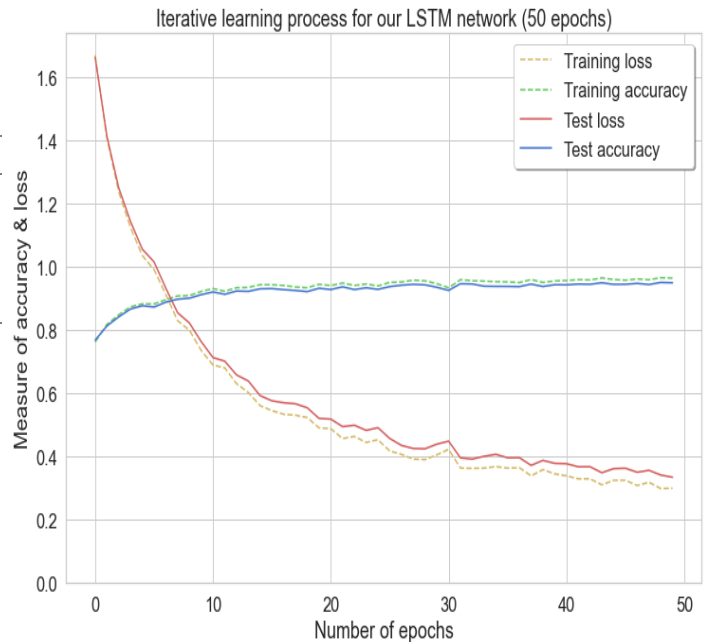


Figure 4: Learning and loss curve for stacked LSTM network

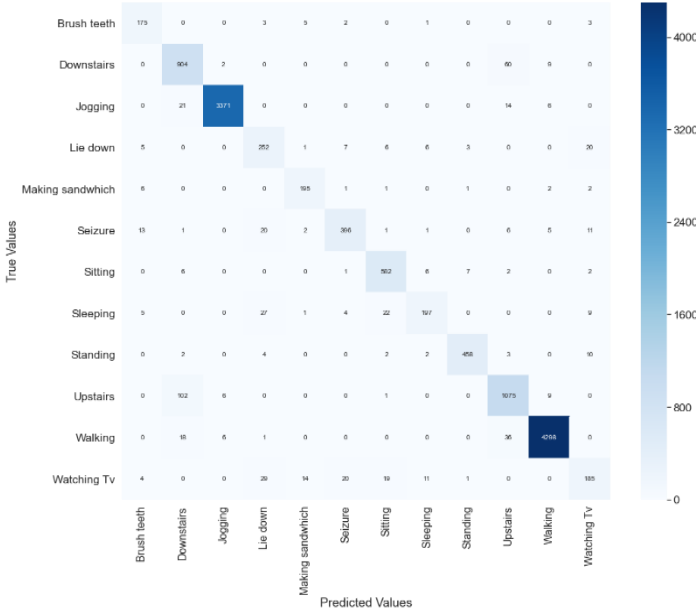


Figure 5: Multi-label confusion matrix with diagonal elements representing the number of correct predictions.

3.2. Results

This section presents the experimental results of our LSTM network. Figure 4 is a mathematical representation of our LSTM network’s incremental learning process, whilst Table 5 shows a 10-increment overview of the training accuracy and loss incurred. Performance is evaluated using classification accuracy, whilst the network is optimised using cross-entropy loss. As accuracy increases, the loss value decreases until they both stabilise at 35 and 50 epochs, respectively. After 50 epochs, our network achieved an overall accuracy of 94.45% with a loss of 0.357.

Epoch	Accuracy	Training Loss
01	0.74448025226	1.75442266464
10	0.88748329877	0.82598221302
20	0.92881274223	0.53437787294
30	0.93407714366	0.44142538309
40	0.94460594654	0.38219016790
50	0.94452738761	0.35765498876
Outcome	0.94452738761	0.35765498876

Table 5: Accuracy and loss results with 10 epoch increments

To evaluate how well our model can distinguish the different types of movement, we used a multi-label confusion matrix (see Figure 5). The diagonal elements represent the true positive values indicating the number

of times our model made a correct prediction [51]. The remaining values indicate incorrect classifications. Horizontal values represent false negatives, whilst vertical values represent false positives. Table 6 lists the results of our precision and recall experiments. Precision results represent the proportion of positive predictions that were made. The network distinguished GTC seizures from other non-seizure related movements 91% of the time. A recall value of 90% was observed when identifying seizure activities, which indicates the number of times a positive value was correctly identified for that activity. Furthermore, our model struggled to differentiate the watching TV activity from other types, resulting in a precision and recall value of 69% and 71% respectively.

Activity	Precision	Recall
Brush teeth	0.84	0.87
Downstairs	0.95	0.84
Jogging	0.99	0.99
Lie-down	0.84	0.70
Make sandwich	0.89	0.87
Seizure	0.91	0.90
Sitting	0.86	0.97
Sleeping	0.84	0.63
Standing	0.92	0.98
Upstairs	0.87	0.93
Walking	0.98	0.99
Watching TV	0.69	0.71

Table 6: Precision and recall results for each activity

Table 7 shows the F-measure values of our model. We have calculated the harmonic mean for each activity, to present a balanced measure of precision and recall.

Activity	F-Measure
Brush teeth	0.85
Downstairs	0.89
Jogging	0.99
Lie-down	0.77
Make sandwich	0.88
Seizure	0.90
Sitting	0.91
Sleeping	0.72
Standing	0.95
Upstairs	0.90
Walking	0.99
Watching TV	0.70

Table 7: F-measure scores for each activity experiment

4. Discussion

Whilst further investigation is required to determine if this method of detection has incurred a lower false positive rate than other commercial systems, there is sufficient evidence to make an educated assumption that this has been attained. Of the 456 seizures predicted, 396 (87%) were correctly identified. Our model also eliminated 11 other possible activities prior to formulating this final prediction. Whilst traditional detection algorithms are susceptible to false predictions when an unforeseen or repetitive movement occurs (brushing teeth), we believe that our model has overcome this limitation and can only improve over time as more activities and repetitive movements are added to our ever growing dataset.

Although it is unclear why activities such as watching TV and lying down yielded a lower precision and recall value, we speculate that it was due to other activities such as sleeping, sharing a similar position in three-dimensional space. This is further validated as distinct activities that do not share any similarities such as walking, jogging, and standing show higher levels of precision and recall. However, at this stage of understanding further research is warranted to prove whether these performance measures are a result of similar or distinct movements. Furthermore, identifying the specific movements or activities that are predominantly responsible for triggering false alarms could positively impact those diagnosed with refractory epilepsy and constitutes further research and investigation to validate the kinds of conclusions that can be drawn from this study.

5. Conclusion

This paper presents a deep learning method for ACM-based detection of epileptic seizures. Although the model used for this study was based on a known technique for time-series forecasting, its application for the detection of epileptic seizures is novel and requires further investigation. The system developed for this study can distinguish different types of movement and presents a more sophisticated method of detection as multiple possibilities (activities) are eliminated before a positive identification is made. Whilst traditional detection mechanisms depend solely on the data recorded prior to and during a seizure's onset, the method presented in this study is multifaceted, and analyses activities from several categories of

movement (common, seizure and non-seizure) prior to formulating a conclusion. By using a stacked LSTM network, we tested the hypothesis that activity recognition could be used to detect epileptic seizures and improve on current levels of performance as common movements and activities are accounted for when making a prediction.

To conclude, our stacked LSTM network successfully distinguished a seizures onset from other types of seizure related and non-seizure related movements, outputting an overall classification accuracy of 94.45%.

I. REFERENCES

- [1] Sander J. The epidemiology of epilepsy revisited. *Curr Opin Neurol.* 2003;16(2):165-170. doi:10.1097/01.wco.0000063766.15877.8e.
- [2] Stafstrom C, Carmant L. Seizures and Epilepsy: An Overview for Neuroscientists. *Cold Spring Harb Perspect Med.*, 2015; 5(6):a022426-a022426. doi:10.1101/cshperspect.a022426.
- [3] French J. Refractory epilepsy: clinical overview. *Epilepsia*, 2007; 48(1):3-7.
- [4] Stafstrom C.E, Carmant L. Seizures and epilepsy: an overview for neuroscientists. *Cold Spring Harb Perspect Med*, 2015;5(6): doi:10.1101/cshperspect.a022426
- [5] Rosenow F, Lüders H. Presurgical evaluation of epilepsy, *Brain*, 2001; 124(9):1683-1700.
- [6] Kloster R, Engelskjøn T. Sudden unexpected death in epilepsy (SUDEP): a clinical perspective and a search for risk factors, *Journal of Neurology, Neurosurgery & Psychiatry*, 1999;67:439-444.
- [7] Leijten F S S. Multimodal seizure detection: A review. *Epilepsia*, 2018; 59(S1): 1-66.
- [8] Freidman D, Kazl C. Practical Neurology. <https://practicalneurology.com/articles/2018-nov-dec/seizure-detection-and-sudep-prevention>, 2018; Accessed July 8, 2020.
- [9] Van de Vel A, Cuppens K, Bonroy B *et al*, Non-EEG seizure-detection systems and potential SUDEP prevention: State of the art. *Seizure*, 2013; 22(5):345-355. doi:10.1016/j.seizure.2013.02.012.
- [10] Van de Vel A, Cuppens K, Bonroy B *et al*. Long-term home monitoring of hypermotor seizures by patient-worn accelerometers. *Epilepsy & Behavior*, 2013; 26(1):118-125.
- [11] Beniczky S, Polster T, Kjaer T *et al*. Detection of generalized tonic-clonic seizures by a wireless wrist accelerometer: A prospective, multicenter study. *Epilepsia*, 2013; 54(4):e58-e61.
- [12] Lockman J, Fisher R S, Olson D M. Detection of seizure-like movements using a wrist accelerometer. *Epilepsy & Behavior*, 2011; 20(4):638-641.
- [13] Schulc E, Unterberger I, Saboor S *et al*. Measurement and quantification of generalized tonic-clonic seizures in epilepsy patients by means of accelerometry—An explorative study. *Epilepsy Res.*, 2011; 95(1-2):173-183.
- [14] Kumar A, Maini K, Arya K *et al*. Simple Partial Seizure. <https://www.ncbi.nlm.nih.gov/books/NBK500005/>, 2020. Accessed July 8, 2020.
- [15] Bussmann J, Martens W, Tulen J *et al*. Measuring daily behavior using ambulatory accelerometry: The Activity Monitor. *Behavior*

- Research Methods, Instruments, & Computers*, 2001; 33(3):349-356. doi:10.3758/bf03195388.
- [16] Mantyjarvi J, Himberg J, Seppanen T. Recognizing human motion with multiple acceleration sensors. *IEEE International Conference on Systems, Man, and Cybernetics*, 2001; 747-752.
 - [17] Gjoreski H, et al. RAReFall — Real-time activity recognition and fall detection system. *IEEE International Conference on Pervasive Computing and Communication Workshops*, 2014; 145-147.
 - [18] Mathie M J, Celler B G, Lovell N H et al. Classification of basic daily movements using a triaxial accelerometer. *Med Biol Eng Comput.*, 2004; 42(5):679-687. doi:10.1007/BF02347551.
 - [19] Ahmadi-Karvigh S, et al. Real-time activity recognition for energy efficiency in buildings. *Applied Energy*, 2018; 211. 146-160.
 - [20] Bao L, Intille S. Activity Recognition from UserAnnotated Acceleration Data. *Lecture Notes Computer Science*, 2004; 1-17.
 - [21] Manckoundia P, Mourey F, Pérennou D et al. Backward disequilibrium in elderly subjects. *Clin Interv Aging*, 2008; 3(4):667-672.
 - [22] Ullah M, Ullah H, Khan S D et al. Stacked Lstm Network for Human Activity Recognition Using Smartphone Data, *8th European Workshop on Visual Information Processing (EUVIP)*, 2019; 175-180.
 - [23] Zebin T, Sperrin M, Peek M et al. Human activity recognition from inertial sensor time-series using batch normalized deep LSTM recurrent networks, *IEEE Engineering in Medicine and Biology Society (EMBC)*, 2018; 1-4, doi: 10.1109/EMBC.2018.8513115.
 - [24] Hochreiter S, Schmidhuber J. Long Short-term Memory. *Neural computation*, 1997; 9(8): 1735-1780.
 - [25] Greff K, Srivastava R K, Koutnik J et al. LSTM: A search space odyssey. *IEEE Trans. Neural Netw. Learn. Syst.*, 2017; 28(10):2222-2232.
 - [26] Sakinah N, Tahir M, Badriyah T et al. LSTM With Adam Optimization-Powered High Accuracy Preeclampsia Classification, *International Electronics Symposium (IES)*, 2019; 314-319. doi: 10.1109/ELECSYM 2019.8901536.
 - [27] Ding G, Qin L. Study on the prediction of stock price based on the associated network model of LSTM. *International Journal of Machine Learning and Cybernetics*, 2020; 11(6): 1307-1317.
 - [28] Olah C. Understanding LSTM Networks. <https://colah.github.io/posts/2015-08-Understanding-LSTMs>, 2015; Accessed July 1, 2020.
 - [29] Borovkova S, Tsiamas I. An ensemble of LSTM neural networks for high-frequency stock market classification. *Journal of Forecasting*, 2019; 38: 600-619. <https://doi.org/10.1002/for.2585>.
 - [30] Szegedy C, Vanhoucke V, Ioffe S et al. Rethinking the Inception Architecture for Computer Vision, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016; 2818-2826.
 - [31] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition, 2014; arXiv 1409-1556.
 - [32] Brownlee J. Stacked Long Short-Term Memory Networks, <https://machinelearningmastery.com/stacked-long-short-term-memory-networks/>, 2017; Accessed 1 July 2020.
 - [33] Wang J, Peng B, Zhang X. Using a stacked residual LSTM model for sentiment intensity prediction, *Neurocomputing* 2018; 322:93-101.
 - [34] Eckhardt K. Choosing the right Hyperparameters for a simple LSTM using Keras, <https://towardsdatascience.com/choosing-the-right-hyperparameters-for-a-simple-lstm-using-keras-f8e9ed76f046>, 2018; Accessed 21 June 2020.
 - [35] Kwapisz J R, Weiss G M, Moore S A. Activity Recognition using Cell Phone Accelerometers, *Proceedings of the Fourth International Workshop on Knowledge Discovery from Sensor Data*, 2010; 10-18.
 - [36] Conradsen I, et al. Automatic multi-modal intelligent seizure acquisition (MISA) system for detection of motor seizures from electromyographic data and motion data. *Computer Methods and Programs in Biomedicine*, 2012; 107(2), 97-110.
 - [37] Conradsen I, et al., Multi-modal intelligent seizure acquisition (MISA) system — A new approach towards seizure detection based on full body motion measures, *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2591-2595.
 - [38] My Tonic Clonic/Grand Mal Seizure https://www.youtube.com/watch?v=Nds2U4CzvC4&list=PLB8D6FCEECE94001C&ab_channel=JosephStruska, 2011. Accessed June 22, 2020.
 - [39] Lee R. Computer and Information Science. Springer, 2019; 87-88.
 - [40] Goodfellow I, Papernot N, Huang S et al. Attacking Machine Learning with Adversarial Examples. <https://openai.com/blog/adversarial-example-research>, 2017. Accessed 22 June 2020.
 - [41] Chopra R, England A, Alaudeen M. Data Science with Python: Combine Python with Machine Learning Principles to Discover Hidden Patterns in Raw Data. Birmingham, UK: Packt Publishing Ltd, 19th 2019.
 - [42] Zafar I, Tzanidou G, Burton R et al. Hands-On Convolutional Neural Networks with TensorFlow. Packt Publishing Ltd, 28th Aug. 2018.
 - [43] Gaddam S H R. Log loss function math explained. <https://towardsdatascience.com/log-loss-function-math-explained-5b83cd8d9c83>, 2020. Accessed 07 June 2020.
 - [44] Wood T. What is the Softmax Function? <https://deeptai.org/machine-learning-glossary-and-terms/softmax-layer>, 2019. Accessed 05 July, 2020.
 - [45] Cao J, Su Z, Yu L et al. Softmax Cross Entropy Loss with Unbiased Decision Boundary for Image Classification, *Chinese Automation Congress (CAC)*, 2018; 2028-2032. doi: 10.1109/CAC.2018.8623242.
 - [46] Zhong H, Chen Z, Qin C et al. Adam revisited: a weighted past gradients perspective, *Frontiers of Computer Science*, 2020; 14:1-16.
 - [47] Yu Y and Liu F. Effective Neural Network Training With a New Weighting Mechanism-Based Optimization Algorithm, *IEEE Access*, 2019; 7:72403-72410.
 - [48] Kingma D P, Ba J L. Adam: A Method for Stochastic Optimization, International Conference on Learning Representations (ICLR), 2015; abs/1412.6980.
 - [49] Salmon B P, Kleynhans W, Schwegmann C P et al. Proper comparison among methods using a confusion matrix, *IEEE International Geoscience and Remote Sensing Symposium*, Milan, 2015; 3057-3060.
 - [50] Iyer V, Iyengar S S. Modeling Unreliable Data and Sensors: Using F-measure Attribute Performance with Test Samples from Low-Cost Sensors, *IEEE 11th International Conference on Data Mining Workshops*, 2011; 15-22, doi: 10.1109/ICDMW.2011.124.
 - [51] Tharwat A. Classification Assessment Methods: a detailed tutorial. *Applied Computing & Informatics*, 2018; doi: 10.1016/j.aci.2018.08.003.