Information We Can Extract About a User From ‘One Minute Mobile Application Usage’

Sarwan Ali
Department of Computer Science
Georgia State University
Atlanta, GA, USA
sali85@student.gsu.edu

Abstract—Understanding human behavior is important and has applications in many domains, such as targeted advertisement, health analytics, security, entertainment, etc. For this purpose, designing a system for activity recognition (AR) is important. However, understanding and analyzing common patterns become challenging since every human can have different behaviors. Since smartphones are easily available to every human being in the modern world, using them to track human activities becomes possible. In this paper, we extracted different human activities using the accelerometer, magnetometer, and gyroscope sensors of android smartphones by building an android mobile application. Using different social media applications, such as Facebook, Instagram, Whatsapp, and Twitter, we extracted the raw sensor values along with the attributes of 20 subjects along with their attributes (class labels) such as age, gender, and left/right/both hands application usage. We extract features from the raw signals and use them to perform classification using different machine learning (ML) algorithms. Using statistical analysis, we show the importance of different features in predicting class labels. Ultimately, we use the trained ML model on our data to extract unknown features from well-known activity recognition data from the UCI repository, highlighting the potential of privacy breaches using ML models. This security analysis could help researchers in the future to take appropriate steps to preserve the privacy of human subjects.

Index Terms—Activity Recognition; Data Security; t-SNE; Information Gain

I. INTRODUCTION

Nowadays, our mobile phones have evolved into powerful handheld computing devices. Holding a smartphone is now part of our daily activities. This is why many research studies explored smartphone sensors such as an accelerometer and gyroscope to detect human activities [1]. In general, one of the most significant tasks in pervasive computing is to provide precise and timely information about people’s activities and behaviors [2]. In fact, the study of human activities has emerged as one of the most active research areas in computer vision [3].

Authors in [4] discussed the following four challenges that people might face when studying human activities. First, human activities are concurrent; different actions might be happening at the same time. Secondly, human activities are intertwined. Thirdly, similar activities can be perceived in various ways. Finally, there might be more than one user in the studied environment. All of these human factors, alongside technological challenges, can contribute to the difficulties of accurately predicting human activities. They also differentiated between ‘Activity Recognition’ and ‘Activity Pattern Discovery’. They defined ‘Activity Recognition’ as the detection of human activities accurately using a predetermined activity model, which means that the model is built first, then it gets implemented into a suitable pervasive system. On the other hand, ‘Activity Pattern Discovery’ means that the data is collected first. Then models are applied to discover some information and unknown patterns about the activities, which means that the pervasive system is built first. Then the sensor data gets analyzed to uncover some patterns.

In this paper, we adopt the ‘Activity Pattern Discovery’ method and collect data from lower-level smartphone sensors to reveal some information about the user. First, we want to observe whether we can predict what type of mobile (social media) application the user is currently running. We chose the following four applications for our experiments: (Facebook, Instagram, Whatsapp, and Twitter). Then, after getting the sensor patterns (using different built-in sensors in the smart device), we want to analyze them further and identify hidden patterns (if there is any such pattern) to uncover additional information about the experiment subjects like: what is the gender of the subject? Do females have different behavior patterns than males? What age group the person belongs to? Can we predict the age of the subject while using some mobile application? And finally, we want to observe whether we can recognize if the user is using his left hand, right hand, or both hands while browsing the social media application. All of this information might help in recognizing if different users have different experiences while using any application and whether certain patterns disclose some private information about smartphone users. In the real world, such a system could be used to enhance user experience while designing any mobile application, for getting likes/dislikes of people, for targeted advertisement, for fall detection (specifically for older people), and for securing private information of the users based on the detected patterns from the sensor data.

Using the existing data (of 60 seconds of social media application usage), we show that the private information of users can be predicted using simple machine learning models. This behavior shows the vulnerability of the currently available activity data on the internet. Using the analysis we show in this paper, relevant authorities could take appropriate steps to
ensure the privacy of the data of people so that it can be avoided being misused.

In this paper, our contributions are the following:

1) We collect human activity data using three sensors of the android smartphones, namely accelerometer, magnetometer, and gyroscope, by building an android application.
2) By using our application (for data collection) for just 60 seconds, we were able to classify different attributes related to human activities with high accuracy.
3) Using statistical analysis (information gain), we show that some features are important for the classification of certain classes.
4) We show that the privacy of the existing data could be exploited very easily, which requires relevant authorities to take appropriate steps to avoid the misuse of activity information.

The rest of the paper is organized as follows: Section II contains the related work for the human activity recognition problem. Our main methodology is introduced in Section III. Details related to the experiments and dataset statistics are given in Section IV. Our results are reported in Section V. Finally, we conclude the paper in Section VI

II. RELATED WORK

Human activity recognition (HAR) is a well-studied problem in the literature [5], [6], [7], [8]. Authors in [5] use the K-nearest neighbor algorithm for HAR. Using machine learning-based algorithms for HAR is done in [7]. Authors in [9] used accelerometer data from wristwatches to detect different behaviors of humans. Similarly, authors in [10] use data from wristwatches to classify different human exercises using the ML models.

Converting unstructured data into a fixed-length numerical representation is an interesting and ongoing research problem. It has been studied in many fields, such as graphs [11], and electricity consumption [12], [13], electroencephalography and electromyography sequences [14], Networks [15], and biological sequences [16], [17], [18], [19], [20]. For time series data, several authors proposed machine learning and deep learning-based methods for time series classification [21], [22], [23], [24]. However, since deep learning-based models are “data hungry”, they cannot be applied when we have limited data [25].

III. PROPOSED METHOD

In this section, we discuss the proposed approach in detail, which includes data collection and generating fixed-length numerical representation from the raw sensor data.

A. Mobile Application Usage

In the first step, we designed an android mobile application that takes some basic information (we use this information as class labels later on) from the user, such as the type of social media application they want to use, their gender, age, and the hand in which they hold the smartphone while browsing the social media applications (this information is going to help us in training our classification algorithms). When they click on the start button, the application starts collecting the information from three different sensors, namely the accelerometer, magnetometer, and gyroscope, in the background. The user minimizes our application and starts using the required social media application for 60 seconds. The social media application that users can use is Facebook, Instagram, Whatsapp, and Twitter. After 60 seconds, the application stops collecting the data and sends a message to the user that the data collection process is stopped. The final data for each sensor is added to separate excel sheets located in the download folder of the mobile. The user then sends the data to us for further analysis.

B. Data Processing and Feature Extraction

Once the data are received, we then start extracting the features. Because the raw sensor data can be of varying lengths, and the machine learning (ML) algorithms require a fixed-length vector as input, we did not use the raw data in the analysis and chose specific features only to get a fixed-length feature vector. Although there are methods that could be used to make the (raw) signal of a fixed length (such as data padding). However, this approach will only increase the dimensionality of the data (which will cause a curse of dimensionality) and hence the classifiers’ runtime. Furthermore, from the information gain analysis (explained in the later section), we discovered that only a few features are important to predict different labels from our data. This means that analyzing the raw signal would not help much in improving the classifier’s performance and would have only added a huge computational cost. The features selected are Mean, Median, Mode, and Quantiles (Divide data into 3 intervals with equal probability), Population Standard Deviation, Sample Standard Deviation, and Variance (total 9 features). Several ways have been proposed in the literature to use a combination of different statistical features to get the fixed-length representation from the raw signals [14]. However, there is no standard way defined to select specific types of features. Therefore, we selected the standard features used for statistical analysis. In this way, we got 9 features for the x, y, and z-axis each from different sensors. Therefore, there will be a total of $9 + 9 + 9 = 27$ features for each sensor data. In total, we had 81 Features that included 27 Magnetometer, 27 Accelerometer, and 27 Gyroscope features. The total entries (data samples) that we collected from all 29 users are 112, which contains samples for the different users using different social media applications (with each sample having 81 features).

As for choosing the sensors for the data collection, authors in [1] state that in the field of activity recognition, the accelerometer sensor has acquired the greatest attention in research. They also stated that when aiming to enhance and boost the performance of activity recognition tasks, the gyroscope and magnetometer have been integrated alongside the accelerometer sensor. In addition, most of the old android devices only had these three sensors. Therefore, we decided
to use these basic sensors in order to reach a larger audience for the data collection.

C. Classification Step

After the feature extraction step, we run different machine learning (ML) algorithms to help us correctly identify the user’s behavior during mobile application usage. We ran seven classification algorithms: Support vector machines (SVM) with a linear kernel, Naive Bayes (NB), Multilayer perceptron (MLP) where we had 10 hidden layers, K-nearest neighbors (KNN) where K=5, Random Forest (RF) with 100 as the number of trees, Logistic regression (LR) with a ‘l2linear’ solver because it is the best choice for small datasets, and Decision tree (DT). We aimed to compare and contrast the performance of each algorithm in the hope that we find one that will outperform the others. In Figure 1, we provide an overview of our system components working together to achieve the goal of predicting different class labels.

![Figure 1: Overview of the system components.](image)

IV. EXPERIMENTAL SETUP

In this section, we describe the implementation detail and dataset statistics used for the experimentation. For mobile application development in android studio (using the java code), we simply used the built-in sensor libraries for Accelerometer, Gyroscope, and Magnetometer. For the ML classification models, we use the python language.

We use the standard 5-fold cross-validation approach for the experiments and report the average results of 5 runs. We use different evaluation Metrics to measure the performance of classifiers. The metrics used are Accuracy, Precision, Recall, F1 (weighted), F1 (Macro), F1 (Micro), and ROC AUC (one-vs-rest). For computing the information gain (statistical analysis), we use Weka software.

A. Dataset Statistics

We collected the data from 29 subjects, 17 of which were males and the other 12 were females. Also, 12 of the 29 subjects under the study used right-handed, 10 preferred using left-handed, and 7 preferred using mobile applications from both hands. To better illustrate a summary of the collected labels, we provide the pie charts in Figure 2. Similarly, the signal patterns for different sensors and axes are shown in Figure 3.

To compute the importance of different features, we use a statistical method called Information Gain (IG). The IG computes the importance of each feature with respect to the class label. More formally: $IG(Class, position) = H(Class) - H(Class|position)$, where $H = \sum_{i \in Class} -p_i \log p_i$ is the entropy, and $p_i$ is the probability of the class $i$.

![Fig. 2: Gender, hand, and age distribution. Figure is best seen in color.](image)

![Fig. 3: Time series signals for Accelerometer sensor.](image)

V. RESULTS AND DISCUSSION

In this section, we first show the classification results for different class labels. After that, we show the procedure for data prediction for unknown data using our trained models.

A. Gender Prediction

Table I shows a summary of the ML algorithms for gender prediction (best values are shown in bold). SVM performed the best on this task. Figure 4 illustrates the information gain values for different features. A feature having a higher information gain value means that it is contributing more towards the prediction of the class label (gender). We can observe that a large number of features are contributing towards the prediction of gender, because of which, we are getting the classification accuracy of 98.2% (for SVM).

**TABLE I: Performance of the ML algorithms on gender prediction. Best values are shown in bold.**

| ML Alg. | Acc. (%) | Prec. (%) | Recall (%) | F1 (Weighted) | F1 (Macro) | ROC-AUC | Runtime (sec.) |
|--------|----------|-----------|------------|---------------|------------|---------|----------------|
| SVM    | 0.982    | 0.984     | 0.982      | 0.981         | 0.980      | 0.981   | 0.011          |
| NB     | 0.791    | 0.814     | 0.794      | 0.778         | 0.754      | 0.749   | 0.014          |
| MLP    | 0.979    | 0.981     | 0.979      | 0.979         | 0.976      | 0.976   | 0.248          |
| KNN    | 0.905    | 0.917     | 0.905      | 0.902         | 0.893      | 0.886   | 0.014          |
| RF     | 0.941    | 0.949     | 0.941      | 0.940         | 0.935      | 0.932   | 0.132          |
| LR     | 0.976    | 0.979     | 0.976      | 0.976         | 0.974      | 0.972   | 0.013          |
| DT     | 0.851    | 0.862     | 0.851      | 0.850         | 0.838      | 0.842   | 0.008          |

1) Hand Prediction: It was really difficult to accurately predict the hand used by the subjects as shown in Table II (best values are shown in bold). To help us understand why that happens, we used information gain to reveal if there is a correlation between the class label (hand) and the data features. Generally, when information gain value for more features is higher, it implies that those features are (positively) related to the class label, and hence they will contribute more towards increasing the predictive accuracy of the classifiers. However, as shown in Figure 5, there are only a few attributes/features...
with larger information gain. As a result, just a few features actually aid the classifiers in distinguishing between distinct hands.

### TABLE II: Performance of the ML algorithms on hand prediction. Best values are shown in bold.

| ML Algo | Acc.   | Prec.  | Recall | F1 weigh. | F1 Macro | ROC - AUC | Train. runtime (sec.) |
|---------|--------|--------|--------|-----------|----------|-----------|-----------------------|
| SVM     | 0.404  | 0.432  | 0.404  | 0.408     | 0.578    | 0.548     | 0.015                 |
| NB      | 0.404  | 0.504  | 0.404  | 0.412     | 0.393    | 0.562     | 0.016                 |
| MLP     | 0.416  | 0.430  | 0.416  | 0.406     | 0.555    | 0.210     | 0.012                 |
| KNN     | 0.330  | 0.415  | 0.330  | 0.332     | 0.518    | 0.017     | 0.012                 |
| RF      | 0.410  | 0.440  | 0.410  | 0.405     | 0.547    | 0.140     | 0.012                 |
| LR      | 0.423  | 0.441  | 0.423  | 0.413     | 0.556    | 0.019     | 0.012                 |
| DT      | 0.434  | 0.461  | 0.434  | 0.433     | 0.563    | 0.010     | 0.012                 |

### TABLE III: Performance of the ML algorithms on application prediction. Best values are shown in bold.

| ML Algo | Acc.   | Prec.  | Recall | F1 weigh. | F1 Macro | ROC - AUC | Train. runtime (sec.) |
|---------|--------|--------|--------|-----------|----------|-----------|-----------------------|
| SVM     | 0.365  | 0.402  | 0.363  | 0.363     | 0.358    | 0.583     | 0.010                 |
| NB      | 0.221  | 0.267  | 0.221  | 0.218     | 0.209    | 0.480     | 0.011                 |
| MLP     | 0.521  | 0.574  | 0.521  | 0.520     | 0.515    | 0.688     | 0.192                 |
| KNN     | 0.206  | 0.249  | 0.206  | 0.197     | 0.196    | 0.479     | 0.021                 |
| RF      | 0.477  | 0.542  | 0.477  | 0.478     | 0.473    | 0.680     | 0.132                 |
| LR      | 0.379  | 0.430  | 0.379  | 0.378     | 0.369    | 0.591     | 0.023                 |
| DT      | 0.421  | 0.460  | 0.421  | 0.417     | 0.411    | 0.620     | 0.014                 |

### TABLE IV: Performance of the ML algorithms on age prediction. Best values are shown in bold.

| ML Algo | Acc.   | Prec.  | Recall | F1 weigh. | F1 Macro | ROC - AUC | Train. runtime (sec.) |
|---------|--------|--------|--------|-----------|----------|-----------|-----------------------|
| SVM     | 0.917  | 0.931  | 0.917  | 0.916     | 0.903    | 0.940     | 0.014                 |
| NB      | 0.736  | 0.770  | 0.736  | 0.730     | 0.722    | 0.827     | 0.015                 |
| MLP     | 0.825  | 0.838  | 0.825  | 0.825     | 0.825    | 0.948     | 0.227                 |
| KNN     | 0.843  | 0.836  | 0.843  | 0.825     | 0.752    | 0.867     | 0.016                 |
| RF      | 0.911  | 0.923  | 0.911  | 0.907     | 0.882    | 0.933     | 0.020                 |
| LR      | 0.857  | 0.856  | 0.857  | 0.853     | 0.810    | 0.891     | 0.009                 |

2) **Application Prediction:** Table III shows a summary of the results computed using different ML algorithms for social media mobile application prediction (best values are shown in bold). MLP performed the best on this task. We realized that the mobile applications used in our study (Facebook, Twitter, and Instagram) produce similar clicking patterns. As a result, there are small differences in information gain values for these applications (see Figure 6). Hence, it is difficult for classifiers to distinguish between applications.

3) **Age Prediction:** Table IV shows a summary of results computed using different ML algorithms for age prediction (best values are shown in bold). MLP performed the best on this task. Figure 7 illustrates different features contribution towards the prediction of age. We can see that since more features have high IG values, therefore, we are getting a higher predictive accuracy, as shown in Table IV. It means that there is a correlation between the number of features having higher IG values and the classification performance for different ML classifiers.
B. UCI Repository Data Analysis

To further analyze our results, we use our trained ML models on the UCI Repository data (for testing purposes) [26] to discover how similar their activity signals (Walking, Walking Upstairs, Walking Downstairs, Sitting, Standing, Laying) are to our collected data. Their dataset had a total of 30 Subjects. They used the Accelerometer and Gyroscope sensors to collect the data. They had a total of 2947 Entries. Despite the fact that the labeled 'activities' collected in their experiment varied from those collected in ours, it is still possible that the users were walking, laying, sitting, etc., when they were using our mobile application to collect the data. In this case, the collected signals from UCI data and our data should have some correlations. Therefore, we used all of our data, 112 samples, for training the ML models and all of their data, 2947 samples of UCI repository data, for testing. Because their labels are different from the ones used in our experiment, the classifiers we trained will predict the labels that it already detected during the training phase (the labels that we have in our data). Due to the label differences, we cannot compute accuracy, precision, etc., for this experiment. Instead, we analyzed the patterns using a contingency table as shown in Table V. Table VI, Table VII, and Table VIII for social media applications, gender, age, and hands (using the best performing classifiers only). We then used the information gain to see the contribution of features toward the prediction of class labels. As previously stated, if more features have high information gain values, the predicted labels are more likely to be correct (because accuracies will be higher, as can be seen from Table I to Table IV). Since we can see the higher information gain values in some cases, this indeed means that there is some undiscovered correlation between their data and our data, as shown in Figure 8 for Age, Application, and Hand labels, respectively (for only the best classifiers from Table II, Table III, and Table IV).

Table V: Contingency table for MLP classifier for Application label.

| Position       | Twitter | Facebook | Instagram | Whatsapp | Total |
|----------------|---------|----------|-----------|----------|-------|
| Laying         | 55      | 348      | 32        | 104      | 537   |
| Sitting        | 36      | 326      | 36        | 93       | 491   |
| Standing       | 54      | 321      | 19        | 138      | 532   |
| Walking        | 21      | 169      | 15        | 291      | 496   |
| Walking Upstairs | 26    | 144      | 11        | 239      | 420   |
| Total          | 216     | 1526     | 129       | 1076     | 2947  |

Table VI: Contingency table for SVM classifier for Gender label.

| Position     | Male    | Total |
|--------------|---------|-------|
| Laying       | 537     | 537   |
| Sitting      | 491     | 491   |
| Standing     | 532     | 532   |
| Walking      | 496     | 496   |
| Walking Downstairs | 420 | 420   |
| Total        | 2947    | 2947  |

C. t-distributed stochastic neighborhood embedding (t-SNE)

The t-SNE [27] is an approach used to get a 2-dimensional representation of the data while preserving the pairwise distance between the data points. This method is ideal for visual data analysis because it allows us to see if the data has any natural clustering visually. We use t-SNE to get the 2-D representation of the UCI data and use the predicted labels (from our trained classifiers) to represent the data points by different colors. We can see that there are some patterns in the data, e.g., in Figure 9d, we can notice that those who prefer to use the phone with their left hand (green dots) form separate clusters and are different from people who prefer to use the phone with their right hand or using both hands. The contingency table (in Table VIII) can also be used to validate this behavior. This behavior shows that most of the subjects involved in collecting the UCI data were left-handed, information that was originally not provided in that data repository.

VI. Conclusion

In summary, we show that using statistical features and standard machine learning models, we can predict the patterns

Fig. 8: Importance of the features (accelerometer and gyroscope) for Age, Application, and Hand attributes, respectively (y-axis shows the IG values). We did not show the information gain plot for gender as the best classifier predicted all Males. Since UCI repository data uses only accelerometer and gyroscope sensors, we report IG values for these two sensors only.
of people, which can be used by researchers and relevant authorities to understand their behavior. Moreover, we show that using information gain, we can observe some correlation between the features in the data and the predictive performance of the ML models. Using this correlation-based analysis, we can predict some missing attributes about users, which were not known to the ML models initially. This could be a potential privacy concern because of this, authorities need to take more steps to ensure the privacy of people’s data. In the future, we will collect more people’s data to further analyze the robustness of ML models. Collecting data for applications other than social media could also help us to understand some more hidden features of humans. Similarly, using deep learning for human activity recognition could also be a potential future work.

REFERENCES

[1] M. Shoaih, S. Bosch, O. D. Incel, H. Scholten, and P. J. Havenga, “Fusion of smartphone motion sensors for physical activity recognition,” Sensors, vol. 14, no. 6, pp. 10 146–10 176, 2014.

[2] O. D. Lara and M. A. Labrador, “A survey on human activity recognition using wearable sensors,” IEEE communications surveys & tutorials, vol. 15, no. 3, pp. 1192–1209, 2012.

[3] M. Vrigkas, C. Nikolou, and I. A. Kakadiaris, “A review of human activity recognition methods,” Frontiers in Robotics and AI, vol. 2, p. 28, 2015.

[4] E. Kim, S. Helal, and D. Cook, “Human activity recognition and pattern discovery,” IEEE pervasive computing, vol. 9, no. 1, pp. 48–53, 2009.

[5] S. Kaghyan and H. Sarukhanyan, “Activity recognition using k-nearest neighbor algorithm on smartphone with tri-axial accelerometer,” IJIMA, vol. 1, pp. 146–156, 2012.

[6] M. Derawi and P. Bours, “Gait and activity recognition using commercial phones,” computers & security, vol. 39, pp. 137–144, 2013.

[7] S. K. Polu and S. Polu, “Human activity recognition on smartphones using machine learning algorithms,” International Journal for Innovative Research in Science & Technology, vol. 5, no. 6, pp. 31–37, 2018.

[8] N. Al-Naffakh, N. Clarke, P. Dowland, and F. Li, “Activity recognition using wearable computing,” in 2010 11th International Conference for Internet Technology and Secured Transactions (ICITST). IEEE, 2016, pp. 189–195.

[9] P. M. Scholl and K. Van Laerhoven, “A feasibility study of wrist-worn accelerometer based detection of smoking habits,” in 2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing. IEEE, 2012, pp. 886–891.

[10] J. P. Varkey, D. Pomppili, and T. A. Walls, “Human motion recognition using a wireless sensor-based wearable system,” Personal and Ubiquitous Computing, vol. 16, no. 7, pp. 897–910, 2012.

[11] S. Ali, M. H. Shakeel, I. Khan, S. Faizullah, and M. A. Khan, “Predicting attributes of nodes using network structure,” ACM Transactions on Intelligent Systems and Technology, vol. 12, no. 2, pp. 1–23, 2021.

[12] S. Ali, H. Mansoor, N. Arshad, and I. Khan, “Short term load forecasting using smart meter data,” in International Conference on Future Energy Systems, 2019, pp. 419–421.

[13] S. Ali, H. Mansoor, I. Khan, N. Arshad, M. A. Khan, and S. Faizullah, “Short-term load forecasting using ami data,” arXiv preprint arXiv:1912.12479, 2019.

[14] A. Ullah, S. Ali, I. Khan, M. A. Khan, and S. Faizullah, “Effect of analysis window and feature selection on classification of hand movements using emg signal,” in SAI Intelligent Systems Conference (IntelliSys), 2020, pp. 400–415.

[15] S. Ali, M. K. Alvi, S. Faizullah, M. A. Khan, A. Alshanqiti, and I. Khan, “Detecting ddos attack on sdn due to vulnerabilities in openflow,” in International Conference on Advances in the Emerging Computing Technologies (AECT). IEEE, 2020, pp. 1–6.

[16] C. Leslie, E. Eskin, J. Weston, and W. Noble, “Mismatch string kernels for svm protein classification,” in Advances in neural information processing systems (NeurIPS), 2003, pp. 1441–1448.

[17] P. Kuksa, I. Khan, and V. Pavlovic, “Generalized similarity kernels for efficient sequence classification,” in SIAM International Conference on Data Mining (SDM), 2012, pp. 873–882.

[18] S. Ali, T. E. Ali, M. A. Khan, I. Khan, and M. Patterson, “Effective and scalable clustering of sars-cov-2 sequences,” in International Conference on Big Data Research (ICBDIR), 2021, pp. 42–49.

[19] S. Ali, S. Ciccolella, L. Lucarella, G. D. Vedova, and M. Patterson, “Simpler and faster development of tumor phylogeny pipelines,” Journal of Computational Biology, vol. 28, no. 11, pp. 1142–1155, 2021.

[20] S. Ali and M. Patterson, “Spike2vec: An efficient and scalable embedding approach for covid-19 spike sequences,” in IEEE International Conference on Big Data (Big Data), 2021, pp. 1533–1540.

[21] H. Ismail Fawaz, B. Lucas, G. Forestier, C. Pelletier, D. F. Schmidt, J. Weber, G. I. Webb, L. Imdoumghar, 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.

[22] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[23] Z. Wang, W. Yan, and T. Oates, “Time series classification from scratch with deep neural networks: A strong baseline,” in IJCNN, 2017, pp. 1578–1585.

[24] J. Chang, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” arXiv preprint arXiv:1412.3555, 2014.

[25] G. Marcus, “Deep learning: A critical appraisal,” arXiv preprint arXiv:1801.00631, 2018.

[26] UCI Machine Learning Repository: Human Activity Recognition Using Smartphones Data Set, https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones, 2021, [Online; accessed 01-March-2022].

[27] L. Van der M. and G. Hinton, “Visualizing data using t-sne.”