Classifying Depression in Imbalanced Datasets using an Autoencoder-Based Anomaly Detection Approach*

Walter Gerych, Emmanuel Agu, Elke Rundensteiner
Data Science and Computer Science Programs,
Worcester Polytechnic Institute, Worcester, MA 01609, USA
wgerych@wpi.edu, emmanuel@wpi.edu, rundenst@wpi.edu

***Abstract*— Depression is the most prevalent mental health ailment in the United States, affecting 15% of the population. Untreated depression can significantly decrease quality of life, physical health, and has significant economic and societal costs. The traditional method of diagnosing depression requires the patient to respond to medical questionnaires and is subjective. Passive methods to autonomously detect depression are desirable. Prior work on smartphone sensing of depression has utilized machine learning classification of smartphone sensor data. However, as with many ailments, the percentage of afflicted users in most populations is small compared with those unaffected, leading to severe class imbalance. In this work, we explore anomaly detection methods as a method for mitigating class imbalance for depression detection. Our approach adopts a multi-stage machine learning pipeline. First, using autoencoders, we project the mobility features of the majority class (undepressed users). Thereafter, the trained autoencoder then classifies a test set of users as either depressed (anomalous) or not depressed (inliers) using a One Class SVM algorithm. Our method, when applied to the real-world StudentLife data set shows that even with an extremely imbalanced dataset, our method is able to detect individuals with depression symptoms with an AUC-ROC of 0.92, significantly outperforming traditional machine learning classification approaches.**

I. INTRODUCTION

A. Background

Depression is the most prevalent mental illness in the United States, where each year, 7% of the population will suffer from at least one major depressive episode. Depression is also the cause of two-thirds of suicides and is estimated to cost over $70 billion dollars annually in the US. [1].

B. The Problem

Despite the fact that 80% of patients who are treated for depression show an improvement of symptoms within 4 to 6 weeks, nearly two thirds of depressed adults do not seek treatment. Reasons for not seeking treatment include the high cost of doctor visits, the stigma associated with mental health treatment, and individuals not recognizing the signs of depression [1]. Consequently, passive screening methods for depression that do not require active user involvement or an evaluation by a professional are desirable.

*The work presented in this paper was funded by DARPA WASH program HR001117S0032

C. State-of-the-Art and its Limitations

Several machine-learning methods to automatically classify depression have been proposed [6] [5]. These methods treat depression detection as a classification problem where they infer users depression levels (e.g. their PHQ-9 scores). Such machine learning classification methods face a challenging dataset imbalance problem that arises from the fact that depression occurs at a low frequency in the general population, yielding imbalanced datasets.

D. Our Approach

Machine learning classification on significantly imbalanced datasets can be reformulated as an anomaly detection problem, where instances of the minority class are considered anomalies. Treating a binary classification problem in this way can be advantageous due to the fact that the range of possible signals that can be indicative of the minority class is not likely to be well represented in any given training set. Alternatively, anomaly detection methods need only to learn patterns indicative of the majority class, and classify inputs that deviate from this as anomalous. Due to the severe imbalance, binary depression classification is suitable to be reformulated as an anomaly detection problem.

Using autoencoders, we projected mobility features and characteristics of undepressed users. Our autoencoder was then able to classify a test set of users as either depressed or not depressed using a One Class SVM algorithm. Our results demonstrate that even with an extremely imbalanced dataset, our method is able to detect individuals with depression symptoms with an AUC-ROC of 0.92, outperforming all standard classification algorithms we compared against.

E. Our Contributions

1) Recognition of the extreme imbalance in depression occurrence, which we quantify in the StudentLife dataset, and leverage this to reformulate depression classification as anomaly detection
2) Conducting experimental evaluation comparing our proposed depression detection method against several baseline machine learning classification algorithms as well as the state-of-the-art in depression detection techniques in the literature – yielding very encouraging results superior to all alternate methods.

II. Background and Related Work

Smartphones can be used to gather behavioral data that indicates the health status of the user. A wide array of work has sensed user behavior and contexts using smartphone sensor data [2], including detecting depressed smartphone users [5]. Prior smartphone depression sensing has typically utilized a supervised machine learning classification approach.

In this paper, we focus on predicting the depression scores of smartphone users from their location traces. Saeb inferred smartphone users' PHQ-9 scores by classifying mobility features extracted from their GPS locations, which was demonstrated on the StudentLife dataset. Saeb also demonstrated the efficacy of their approach on real patients [5]. Canzian *et al* conducted a study in which they gathered sensor data from 184 smartphone users [6]. The subjects completed PHQ-8 surveys periodically throughout the study. Analysis showed that changes in mobility patterns were predictive of depressive state.

LiKamWa *et al* classified the mood of smartphone users using features extracted from location traces, application logs, calls, SMS, and web usage data [7].

III. Our Approach

In contrast to prior work that utilized a supervised machine learning classification approach to infer the depression levels of smartphone users, we investigate an alternate anomaly detection approach to mitigate severe class imbalance in depression datasets. We train an autoencoder using the location traces of undepressed users (majority inlier class), which is then able to detect depressed subjects as anomalies.

To demonstrate the efficacy of our approach, we utilize the StudentLife dataset [13], which contains smartphone data continuously gathered from 48 Dartmouth College students over a 10-week semester, along with their corresponding PHQ-9 scores [3].

We transform depression detection into a binary classification problem by classifying all PHQ-9 scores greater than 10 as outliers, and scores less than 10 as inliers. Inspired by prior work that established that depression can be detected from mobility traces, we also extracted various mobility features. We then use an autoencoder to project these high dimensional location features into a lower dimensional space. Finally, we use the One Class Support Vector Machines anomaly detection algorithm to perform binary classification of user depression levels.

A. Dataset

Participants in the $StudentLife$ study installed a smartphone app that passively gathered sensor data, and the study participants also responded to various mental health questionnaires at the beginning and end of the study, including the PHQ-9 questionnaire. Our work focuses on the subjects' PHQ-9 scores as the measure of depression, which our technique tries to infer from their smartphone sensor data.

B. PHQ-9 Depression Measure

The PHQ-9 questionnaire is a clinically-validated 9-question depression questionnaire which requires subjects to respond on a 0-3 scale to 9 psychophysical questions which are indicative of depression. Studies have shown that high PHQ-9 scores are indeed strongly correlated with clinical depression [4].

Subjects with scores in the 10-19 range are considered to have mild depression, while subjects with scores 20 and above are considered to be severely depressed. In order to use an anomaly detection framework, we had to convert depression detection to a binary classification problem. Thus, we had to select a threshold PHQ-9 score such that subjects whose PHQ-9 scores were higher than that threshold were considered depressed (anomalies).

We selected a PHQ-9 score of 10 as our threshold, which is supported by results of our analysis of the frequency of occurrence of the PHQ-9 scores of subjects in the StudentLife dataset. Fig. 1 is a plot of the frequency of PHQ-9 scores generated from the survey taken at the completion of the StudentLife study. Scores greater than 10 make up the tail of this distribution, indicating that this is a good cutoff value for distinguishing between the inlier and outlier class.

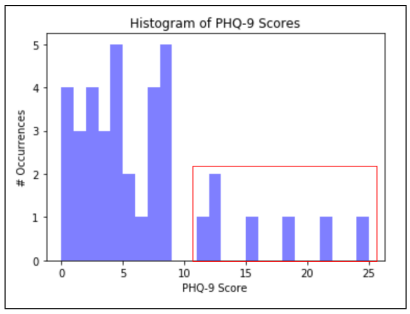

Fig. 1. Frequency of post-study PHQ-9 scores in StudentLife dataset

C. Feature Extraction

We extracted mobility features that prior work have previously shown to be predictive of the smartphone user's depressive state [5]. Specifically, we extracted GPS location features that Saeb *et al* found to be correlated with PHQ-9 scores, including:

1) Location variance:

$$\text{Location variance} = \log(\sigma_{lat}^2 + \sigma_{long}^2),$$

where σ_{lat}^2 and σ_{long}^2 are the variance of the user's latitude and longitude.

2) Speed mean:

$$\text{Speed mean} = \frac{1}{n}\sum_{i}^{n}\sqrt{(\frac{\text{lat}_i - \text{lat}_{i-1}}{t_i - t_{i-1}})^2 + (\frac{\text{lon}_i - \text{lon}_{i-1}}{t_i - t_{i-1}})^2},$$

where lat_i and lon_i are the latitude and longitude at time i, and there are n time-stamps for the user.

3) Speed variance: The variance of the instantaneous values of the users' speeds.

4) Total distance: The total distance traveled by a given participant during the study.

$$\text{Total distance} = \sum_{i}^{n} \sqrt{(\text{lat}_i - \text{lat}_{i-1})^2 + (\text{lon}_i - \text{lon}_{i-1})^2}$$

5) Number of Clusters: First, latitude and longitude points are clustered using an adaptive k-means clustering algorithn, developed by Saeb *et al* [5]. First, the k-means clustering algorithm is run for several iterations, increasing the value of k by one during each iteration, starting with k = 0. This is done until the farthest distance between any point and the center of its nearest cluster is 500 meters. This feature is then computed as the number of clusters found by this method.

6) Entropy:

$$\text{Entropy} = -\sum_{i=1}^{N} p_i \log(p_i),$$

where p_i is the percentage of time that the subject spent at the ith cluster, out of a total of N clusters.

7) Normalized Entropy:

$$\text{Normalized Entropy} = -\frac{Entropy}{log(N)},$$

8) Raw Entropy: After placing location points into 10 bins, the entropy is calculated as the previous entropy value, where now N = 10 and p_i is the number of data points in the ith bin.

9) Percent of Time at Home: Home is determined to be the cluster where users spend the majority of their nights.

10) Transition Time: Percentage of time each user spends in transit.

D. Feature Projection with Autoencoder

The StudentLife dataset contains data for only 48 students over 10 weeks of the study. Our mobility features are calculated from the data over the entire study period, which means each student is represented by a single 10-dimensional vector. Such a small dataset with a relatively high number of features is very susceptible to the well-known curse of dimensionality [8], which led us to seek a lower dimensional representation for our data. We perform this dimension reduction using an autoencoder.

An autoencoder [9] is a type of Artificial Neural Network (ANN) that is typically trained to minimize the error between input data and the output reconstructed by the neural network. The middle layer of the network encodes the input data in a lower dimensional latent space. For dimension reduction, we simply take this compressed representation as the lower dimension projection of the input data.

The structure of our autoencoder is shown in Fig. 2. We train the network in a method similar to leave-one-out cross validation. That is, we train on all but one inlier instance (undepressed subject). We then run the remaining inliers through the network and extract the output of the latent space as the lower dimensional representation of this data point. We repeat this process until the process has been repeated for all inlier data. We then fit the autoencoder on all the inlier data points, and then run the outlier points through the network, again extracting the latent representation of each of these points. At the point, we have determined the low dimensional representation of every individual.

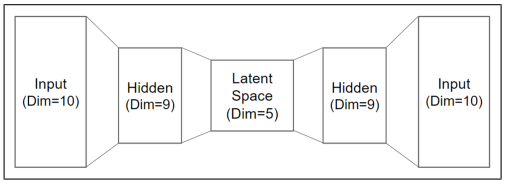

Fig. 2. Structure of our autoencoder

E. Anomaly Detection using One Class SVM

After feature projection, we perform anomaly detection on our data. While several anomaly detection algorithms have been proposed [10], we selected One class Support Vector Machine (OC-SVM) [11] as our anomaly detection method.

OC-SVMs find a decision boundary that separates one class (the inliers) from a single example of a second class (the origin) with maximum margin by using a kernel projection to project the inlier data into a higher dimensional space. This boundary then encapsulates the space in which inliers lie, with the maximum probability. Points that lie on the other side of this boundary are classified as outliers.

We fit a OC-SVM classifier on all but one of the inlier points. We then repeated this process, leaving out a different inlier each time. Then, we fit an OC-SVM on all of the inlier data and tested it on the outlier points. This way, every inlier and outlier receives an anomaly score from an OC-SVM that was not trained on that point.

F. Comparison of our Anomaly Detection Method to Classification Algorithms

We compared our method with several standard machine learning algorithms: RandomForest, Gradient Boosted Classifier, SVM, and Multilayer Perceptron. Additionally, we compared our method to standard anomaly detection methods: IsolationForest, OneClassSVM, Local Outlier Factor, and Elliptic Envelope. All methods were implimented with scikit-learn use the default parameters.

IV. Results

To the best of our knowledge, there have been no published results for the binary classification of the PHQ-9 scores of subjects in the StudentLife dataset. Thus, we were not able to meaningfully compare our results to previously published results. Instead we compare our method to traditional classification and standard anomaly detection methods.

A. Experimental Methodology

Each anomaly detection method is trained as the OneClassSVM was trained, as explained in the methodology section. The standard machine learning classifiers are trained in a similar way to the anomaly detection methods. However, these methods are trained on both inlier and outlier data,

as opposed to the anomaly detection methods that are only trained on inliers. This is of course necessary as standard classifiers should have examples of both classes in the training set. Results from these methods are from leave-one-out cross validation.

B. Final Results

TABLE I
AUC ROC AND WEIGHTED F1 FOR A VARIETY OF CLASSIFIERS

Method	AUC-ROC	Weighted F1
RandomForest (without Autoencoder for feature projection)	0.59	0.85
RandomForest (with Autoencoder for feature projection)	0.58	0.78
GradientBoostedClassifier (without Autoencoder for feature projection)	0.54	0.86
GradientBoostedClassifier (with Autoencoder for feature projection)	0.60	0.79
SVM (without Autoencoder for feature projection)	0.80	0.88
SVM (with Autoencoder for feature projection)	0.88	0.90
MLPClassifier (without Autoencoder for feature projection)	0.53	0.75
MLPClassifier (with Autoencoder for feature projection)	0.59	0.74
OC-SVM (without Autoencoder for feature projection) (Anomaly Detection Method)	0.62	0.75
OC-SVM (with Autoencoder for feature projection) (Anomaly Detection Method)	**0.92**	**0.91**
IsolationForest (without Autoencoder for feature projection) (Anomaly Detection Method)	0.55	0.75
IsolationForest (with Autoencoder for feature projection) (Anomaly Detection Method)	0.66	0.76
EllipticEnvelope (without Autoencoder for feature projection) (Anomaly Detection Method)	0.70	0.76
EllipticEnvelope (with Autoencoder for feature projection) (Anomaly Detection Method)	0.78	0.89
LocalOutlierFactor (without Autoencoder for feature projection) (Anomaly Detection Method)	0.65	0.84
LocalOutlierFactor (with Autoencoder for feature projection) (Anomaly Detection Method)	0.60	0.77

As can be seen in Table I, the autoencoder + OC-SVM method outperforms the other classification methods.

Additionally, Table I shows that in general the outlier detection methods performed better than the standard classification methods, other than SVM. Our finding that outlier methods performed better than standard classification methods is expected, due to the imbalanced nature of the dataset.

V. Conclusion

Motivated by the extreme class imbalance of depression in most populations, we have reframed depression classification as an anomaly detection problem. We evaluated this formulation on the StudentLife dataset. Motivated by prior work that demonstrated that mobility features are predictive of depression, we created standard mobility features from the StudentLife dataset as proposed by Saeb [5]. Additionally, we have proposed to use an autoencoder as a dimension-reduction preprocessing step for the task of anomaly detection. Finally, we compare our method with baseline methods, including standard machine learning classifiers and our proposed anomaly detection method without dimension reduction. Our proposed method outperformed all other algorithms considered with an AUC-ROC of 0.92 and a Weighted F1-score of 0.91. This suggests that treating depression detection using an anomaly detection approach is viable.

VI. Future Work

In future work, we plan to go beyond mobility features and explore detecting depression using other smartphone-sensed data types and modalities including voice, social interactions, smartphone communication patterns, and browsing patterns. We also plan to apply our approach to smartphone sensing of other ailments such as Traumatic Brain Injury (TBI or concussions) and infectious diseases. Since subjects in the StudentLife dataset are mostly students, we plan to explore the robustness of our approach by applying it to depression data gathered from other user populations.

VII. Acknowledgements

We would like to thank Abdulaziz Alajaji, Hamid Mansoor, Luke Buquicchio, and Kavan Chandrasekaran of WPI's WASH research group for their feedback on this research.

References

[1] Morin, Amy, and Lcsw. How Many People Are Actually Affected by Depression Every Year? Verywell Mind, 22 Aug. 2018, www.verywellmind.com/depression-statistics-everyone-should-know-4159056.
[2] Yrr, zgr, et al. "Context-awareness for mobile sensing: A survey and future directions." IEEE Comm Surveys Tutorials 18.1 (2016): 68-93.
[3] Wang, Rui, Fanglin Chen, Zhenyu Chen, Tianxing Li, Gabriella Harari, Stefanie Tignor, Xia Zhou, Dror Ben-Zeev, and Andrew T. Campbell. "StudentLife: Assessing Mental Health, Academic Performance and Behavioral Trends of College Students using Smartphones." In Proceedings of the ACM Conference on Ubiquitous Computing. 2014.
[4] Kroenke, Kurt, Robert L. Spitzer, and Janet BW Williams. "The PHQ9: validity of a brief depression severity measure." Journal of general internal medicine 16.9 (2001): 606-613.
[5] Saeb, Sohrab, et al. "Mobile phone sensor correlates of depressive symptom severity in daily-life behavior: an exploratory study." Journal of medical Internet research 17.7 (2015).
[6] Canzian, Luca, and Mirco Musolesi. "Trajectories of depression: unobtrusive monitoring of depressive states by means of smartphone mobility traces analysis." Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing. ACM, 2015.
[7] LiKamWa, Robert, et al. "Moodscope: Building a mood sensor from smartphone usage patterns." in Proc. ACM MobiSys, 2013.
[8] Richard Ernest Bellman; Rand Corporation. Dynamic programming. Princeton University Press (1957)
[9] Ballard, Dana. "Modular Learning in Neural Networks." AAAI. 1987.
[10] Chandola, Varun, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey." ACM Comp surveys (CSUR) 41.3 (2009): 15.
[11] Scholkopf, J.C. Platt, J.Shawe-Taylor, A.J. Smola, and R.C. Williamson. "Estimating the support of a high-dimensional distribution", Technical report, Microsoft Research, MSR-TR-99-87, 1999.
[12] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011
[13] The StudentLife dataset: http://studentlife.cs.dartmouth.edu/dataset.html