Application of Machine Learning algorithms for Stress level Recognition and Comparison Analysis based on Sleeping gestures

Tarique Mahmud*[a], Dr. Jabed Tomal[b] , Dr. Piper Jackson[c]

[a]*Department of Science, Masters in Data Science, Thompson Rivers University, Kamloops, 805 TRU Way, V2C 0C8, Canada*
[b]***Research Supervisor:*** *Masters in Science, Data Science, Thompson Rivers University, Kamloops, 805 TRU Way, V2C 0C8, Canada*
[c]***Research Supervisor:*** *Masters in Science, Data Science, Thompson Rivers University, Kamloops, 805 TRU Way, V2C 0C8, Canada*

Abstract—

Sleep is an important activity which directly impacts human depression, stress, and happiness. Also, it plays a concrete influence on stress analysis. Around the world, there are a plenty of people who suffers in stress and depression which can take them to unexpected endings like suicide, social distress, and drug addictions. Considering today's way of life, individuals just keep overlooking the benefits which sleep gives to the human body. Stress can be cured if it is identified at early stages. For this analysis, Smart-Yoga Pillow (SaYoPillow) has proposed to assist in understanding the relationship between stress and sleep and to completely materialize the thought of "Smart-Sleeping" by proposing an edge gadget. To analyze the data, many Machine Learning approaches have been considered and compared to each other, the best model(s) has been suggested. After this research, the identification of stress or depression can be precise, and the path of further analysis scopes will be more transparent. [19]

Keywords: Smart Healthcare, Smart Home, Stress, Internet-of-Medical-Things (IoMT), Machine Learning algorithms, Privacy Assurance, Stress Sleep, Stress Detection, Sleeping Habits.

---

* Tarique Mahmud. Tel.: 1-250-879-5554.
E-mail address: mahmudt20@mytru.ca ID: T00685251

1. INTRODUCTION

Concurring to the World Wellbeing Organization (WHO), more than 350 million individuals of all relatively objective assessment. As the number of depressed patients increases, early-stage diagnosis and re-assessments for tracking treatment effects are often limited and time consuming. [1] Therefore, machine learning/deep learning-based automatic potential stress risk detection or stress recognition is expected to facilitate objective and fast diagnosis to ensure excellent clinical care quality and fundamentally reduce potential harm in real life. Under the influence of stress, sleeping disorder-based signals for stress recognition are increasingly extensive. This work focuses on using facial expressions to recognize patients with potential stress risk through the sleeping pattern. The research based on stress with sleeping gesture data which was captured. [2]

An edge processor with a demonstrate analyzing the physiological changes that occur during sleep at the side the sleeping habits is proposed. The secure exchange of the analyzed stress data at the side the average physiological changes to the IoT cloud for capacity is implemented. With the captured data, the stress level has been monitored and detected by this research.[3]

"One of the most common reasons for not seeking help is fear and shame. People recognize the negative stigma and discrimination associated with having a stressed and related issues and don't want to be labelled 'mentally ill' or 'crazy."

- if people can get analysed by their sleeping pattern, computer, or algorithms, it will help to avoid social anxiety and thus people will be able to know about their stress state as well.

● Also it will help to take precautions and early treatment, if people can get to know their stress level anytime they feel like.

Recently, machine learning techniques have been privileged to classify the decisions. Here almost all the classification have been used to classify and compare. The main approach was to create a model for a small dataset which will work for big data as well. Other Machine Learning techniques can identify the features and classify the depression level. However, with a large number of multivariate data, the different types of Machine Learning algorithms have observed to gather precise outcomes. Some tend to provide overfitting and few have given underfitting outcomes as well. Therefore, our main contributions in this research can be summarized as follows:

1. Data has been collected from IOT based pillow devices and based on their feature metadata the analysis will take place.
2. Data sorting and elimination has been done before feeding the models.
3. Initially feature correlation analysis was done and then used three approaches for comparing the models with better detection efficiency.
4. There will be an attempt to integrate networks that can achieve the fusion of static and dynamic features, which can improve recognition performance. [19]

After the analysis the best fitted model will be chosen based on accuracy. Finally, summarizing the research and discussion of future scopes will take place.

Nomenclature

fEMG facial electromyography
AFEA-CV automatic facial expression analysis by computer-vision.
BNN Bayesian Neural Network

1.1 Related Prior Works And Research Gap

The better the quality of sleep, the lower the stress levels [2]. A way to monitor and control physiological stress is presented

L. Rachakonda is with the Dept. of Computer Science and Engineering, University of North Texas, E-mail: RachakondaLaavanya@my.unt.edu.
A. K. Bapatla is with the the Dept. of Computer Science and Engineering, University of North Texas, E-mail: AnandKumarBapatla@my.unt.edu.
S. P. Mohanty is with the Dept. of Computer Science and Engineering, University of North Texas, E-mail: saraju.mohanty@unt.edu.
E. Kougianos is with the Dept. of Electrical Engineering, University of North Texas, E-mail: elias.kougianos@unt.edu.

In the context of IoMT driven smart healthcare, protecting patient data is an important element in smart healthcare. It is important as it requires collection, storage and use of large amounts of sensitive personal information. There were 15 million and 32 million patient records which were compromised with 503 breaches in 2018 and 2019, respectively [5]. Thus an attempt to mitigate the lack of security is also been addressed in this work. To insert images in Word, position the cursor at the insertion point and either use Insert | Picture | From File or copy the image to the Windows clipboard and then Edit | Paste Special | Picture (with “float over text” unchecked).

Data analysis

2.1. Feature analysis

At first the data were observed carefully and while the analysis, the data has been observed of sleeping features. The data captured by SaYoPillow has an outstanding range of data concentrating on various aspects and can be used in multiple research and decision analysis.

	Count	mean	std	min	25%	50%	75%	max
sr2	**630.0**	**71.6**	**19.372833**	**45.0**	**52.50**	**70.0**	**91.25**	**100.0**
rr	630.0	21.8	3.966111	16.0	18.50	21.0	25.00	30.0
t	630.0	92.8	3.529690	85.0	90.50	93.0	95.50	99.0
lm	630.0	11.7	4.299629	4.0	8.50	11.0	15.75	19.0
bo	630.0	90.9	3.902483	82.0	88.50	91.0	94.25	97.0
rem	630.0	88.5	11.893747	60.0	81.25	90.0	98.75	105.0
sr	630.0	3.7	3.054572	0.0	0.50	3.5	6.50	9.0
hr	630.0	64.5	9.915277	50.0	56.25	62.5	72.50	85.0

sl	630.0	2.0	1.415337	0.0	1.00	2.0	3.00	4.0

2.2 Multivariate data plot:
Here it is seen the variation in the each of the data features by plotting each sample value. (Figure – 2.2)

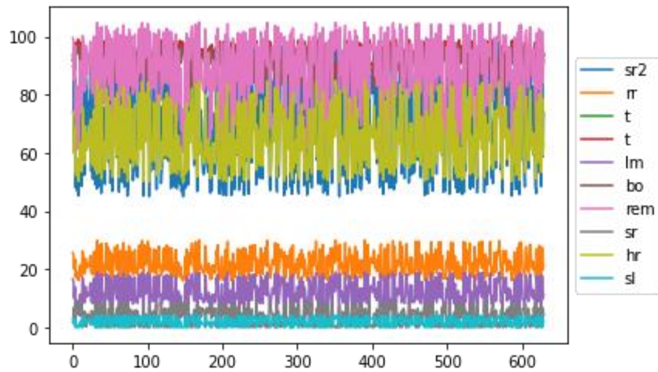

Figure 2.2: Data Plot

2.3 Data Standardization:
For the analysis, data need to be standardized and to check that data is standardized or not.

sr2	19.357452
rr	3.962962
t	3.526887
lm	4.296215
bo	3.899385
rem	11.884303
sr	3.052147
hr	9.907405
sl	1.414214

After analysing the SD of the data as sr2 and rem value is higher than others. [6]

PCA has been done and before that standardization and scaled the data using the StandardScaler library.

	sr2	rr	t	lm	bo	rem	sr	hr	sl
0	1.146845	0.979066	-0.272195	1.140539	-0.271838	0.934005	-0.609407	0.979066	0.707107
1	1.035260	0.833720	-0.353853	0.972949	-0.345696	0.873421	-0.703767	0.833720	0.707107

PCA table

The table has been created till PC6.

variable	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9
sr2	-0.34	-0.20	-0.00	-0.24	0.88	-0.00	-0.02	-0.11	-0.00
rr	-0.34	-0.30	0.36	-0.13	-0.25	-0.24	-0.10	-0.15	-0.71
t	0.33	-0.49	-0.21	-0.25	-0.03	0.08	-0.64	0.35	0.00
lm	-0.34	-0.29	0.09	0.16	-0.12	0.83	0.17	0.20	0.00
bo	0.33	-0.48	-0.20	-0.37	-0.08	-0.06	0.68	-0.12	-0.00
rem	-0.33	-0.29	-0.69	0.48	-0.09	-0.20	-0.05	-0.22	0.00
sr	0.33	-0.35	0.41	0.66	0.26	-0.20	0.12	0.20	-0.00
hr	-0.34	-0.30	0.36	-0.13	-0.25	-0.24	-0.10	-0.15	0.71
sl	-0.34	0.08	-0.12	-0.09	-0.03	-0.33	0.25	0.82	-0.00

Table: PCA analysis

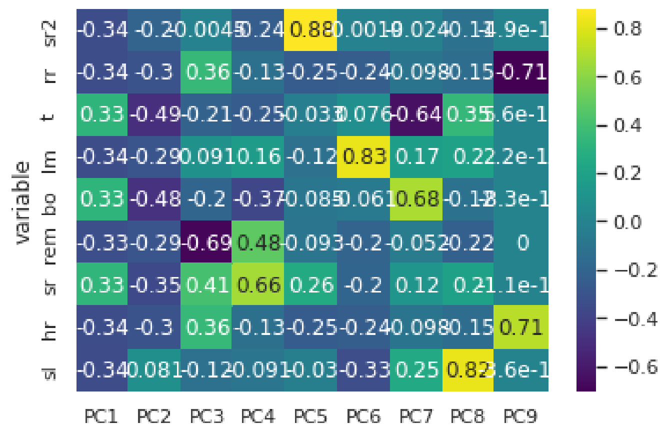

Figure 2.3 : PCA in heatmap

2.4 Data visualization with seaborn and matplotlib

Data heatmap of the features (Figure – 2.4.1):

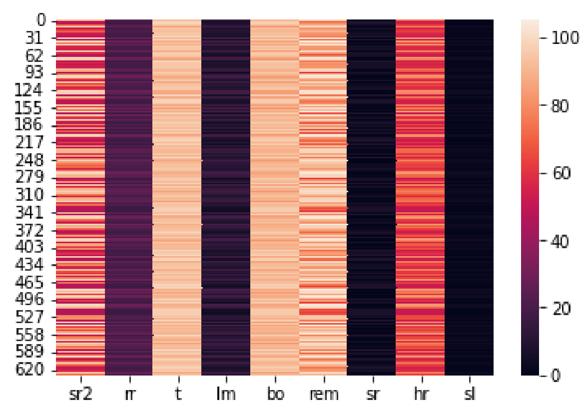

Figure – 2.4.1 Heatmap analysis

And the histogram plotting (Figure – 2.4.2)

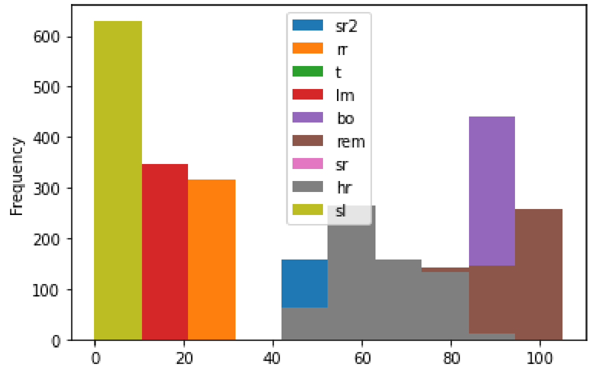

Figure – 2.4.1 Histogram analysis

Here, is visible that the more of sl and bo feature data is available.

2.5 Correlation analysis:

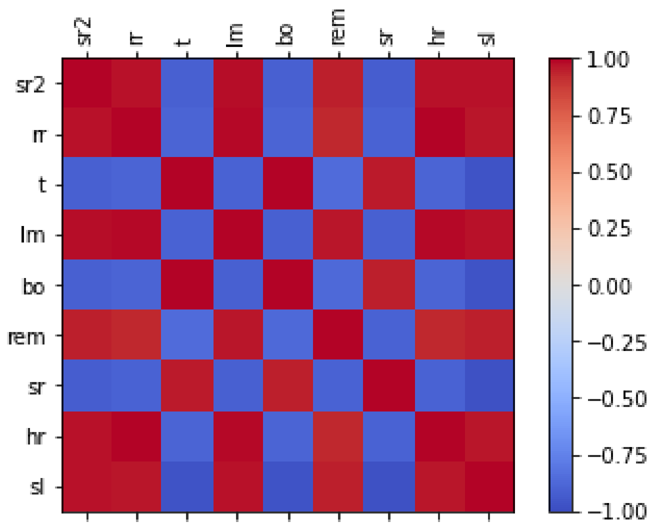

Figure – 2.5 Correlation

Here its visible that sl has clear correlation with the other values whether its positive or negative independent variables. (Figure – 2.5)

Also, with the pair plot the correlation influence is also visible clearly. (Figure – 2.5.1)

2.6 Null data checking:
No null value or missing value has been observed.

```
sr2     0
rr      0
t       0
lm      0
bo      0
rem     0
sr      0
hr      0
sl      0
```

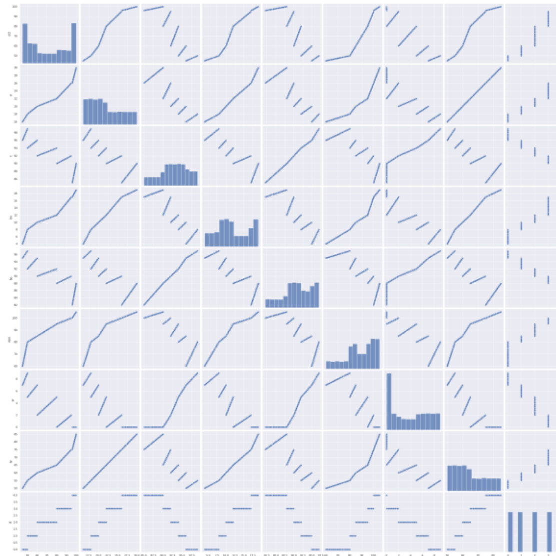

Figure – 2.5.1 Pair plot

Designing the model

Model designing is with 5 fold of cross validation. Then the train test accuracy has been observed and found 86.72% and 84.52% respectively.

```
Fitting 5 folds for each of 9
candidates, totalling 45 fits
Train Accuracy : 1.0
Test Accuracy : 1.0
Fitting 5 folds for each of 9
candidates, totalling 45 fits
Train Accuracy : 1.0
Test Accuracy : 1.0
Fitting 5 folds for each of 9
candidates, totalling 45 fits
Train Accuracy : 0.8677248677248677
Test Accuracy : 0.8452380952380952
Fitting 5 folds for each of 9
candidates, totalling 45 fits
Train Accuracy : 1.0
Test Accuracy : 1.0
Fitting 5 folds for each of 9
candidates, totalling 45 fits
Train Accuracy : 1.0
Test Accuracy : 1.0
```

Here in the heatmap of feature co-efficient With the feature coefficients for all the various stress levels. This analysis is useful for understanding feature and stress level relationship. [7]

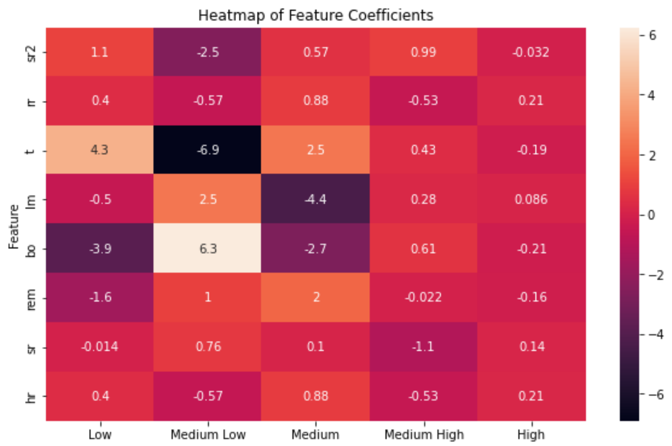

Figure – 3.1 Heatmap of feature coefficients

Renaming the data features with the full mapped names.

$$error +/- z\sqrt{\frac{(error.(1-error)}{n}}$$

where n is the sample size, error is the confidence error and z is a critical value from the normal distribution. The detailed alternate approach used for performing this performance analyses for all the metrics including precision, recall, average precision and F-1 score is presented in our detailed version. [20]

Analyzing the data and classification using Machine learning algorithms

Table – 3.1

index	snoring range	respiration rate	body temperature	limb movement	blood oxygen	eye movement	snoring rate	heart rate	stress level
0	93.8	25.68	91.84	16.6	89.84	99.6	1.84	74.2	3
1	91.64	25.104	91.552	15.88	89.552	98.88	1.552	72.76	3
2	60	20	96	10	95	85	7	60	1
3	85.76	23.536	90.768	13.92	88.768	96.92	0.768	68.84	3
4	48.12	17.248	97.872	6.496	96.248	72.48	8.248	53.12	0

Math

After the data is fed to the model, the data goes through all the hidden layers where the weighed inputs to that layer are calculated using:

$$\sum_{i=1}^{N}(\omega i\, Xi + \omega 0)$$

where X = x1; x2; _ _ _ ; xn is the n-dimensional input, z is the response of the neuron, !i are the weights for each input and !o is a constant bias.

Metrics are used to monitor and measure the performance of a model. Some of the most important metrics for any model's performance evaluation are accuracy, precision vs recall curve and loss. As SaYoPillow uses sklearn for its training model, the metrics of the model can be defined by support functions.

$$\alpha = \left(\frac{TP + TN}{TP + TN + TF + FN}\right) * 100$$

In this section, the machine learning library "lazypredict" has been used to fit the data with the predictor features and target features to classify the level of stress.

Here also splitting the train test with 80/20 ratio of the data. The Shuffle state has been enabled. After fitting in this way, the library will fit the data in approximately 27 ML models and compares. (Table – 4.1)

With the fitted features, it is observed that below models are providing 100% accuracy LinearSVC, KNeighborsClassifier, PassiveAggressiveClassifier, NuSVC, NearestCentroid, LogisticRegression, LinearDiscriminantAnalysis, LabelSpreading, LabelPropagation, GaussianNB, ExtraTreesClassifier, ExtraTreeClassifier, SGDClassifier SVC, CalibratedClassifierCV and QuadraticDiscriminantAnalysis. Here it can be understand that these models are overfitting the data.

On the other hand, BaggingClassifier, RidgeClassifierCV, RandomForestClassifier, XGBClassifier, LGBMClassifier, DecisionTreeClassifier, RidgeClassifier, Perceptron are giving very little error rate which means a great F1 score has been observed.

Finally if the models are observed of the below models

Time taken:

All the model fitting has taken less than 1 second but XGBClassifier and LGBMClassifier have taken more than 1 second to provide the classification.

Decision

Based on the algorithm F1 score results and time has been taken to provide the output, it is observed that the below models as worked well to classify the stress level by analyzing the sleep gestures.

Some models has provided overfitting results. It because the original data has been generated with AI.

Dummy classifier has provided very low accuracy.

Future scopes and limitations:

After observing the values and results, we found out which approach works better for sleeping gesture data or small or big amounts of data. There were some limitations faced during the work such as time shortage, system configurations, data collection involvement and sample size. These limitations forced this research to narrow it down, made it focused and left with a lot of future scopes.

There is timestamps data which can be used to analyze the time series and also the scaling of the parameters based on it. Additionally, it can be taken a larger number of samples to check if the model works better or not. This have more features which are labeled with other posture that can be used to make a more accurate model. Another way is to based on the stress scale of 1-5 it can be suggested the remedies accordingly. As the current model is providing better results, hence there are plenty of scopes available to work on it in future.

Conclusion

The proposed SaYoPillow not only collects and analyzes eight different physiological signal data to predict stress, but also educates the user about the benefits of smart-sleeping.Throughout this report, the problems that arise with stress can be detected with sleeping gesture data at the early stages. [20]

Decision Table 4.1:

Model	Accuracy	Balanced Accuracy	F1 Score	Time Taken
LinearSVC	1	1	1	0.083152533
KNeighborsClassifier	1	1	1	0.033973217
PassiveAggressiveClassifier	1	1	1	0.041487455
NuSVC	1	1	1	0.056794405
NearestCentroid	1	1	1	0.023270369
LogisticRegression	1	1	1	0.083914518
LinearDiscriminantAnalysis	1	1	1	0.047007084
LabelSpreading	1	1	1	0.109766006
LabelPropagation	1	1	1	0.060281754
GaussianNB	1	1	1	0.0246346
ExtraTreesClassifier	1	1	1	0.136325121
ExtraTreeClassifier	1	1	1	0.015447855
SGDClassifier	1	1	1	0.034374952
SVC	1	1	1	0.05759716
CalibratedClassifierCV	1	1	1	0.165123224
QuadraticDiscriminantAnalysis	1	1	1	0.026119709
BaggingClassifier	0.992063492	0.992	0.99205738	0.04248929
RidgeClassifierCV	0.976190476	0.976307692	0.976174963	0.101676702
RandomForestClassifier	0.976190476	0.976	0.976184364	0.474361897
XGBClassifier	0.976190476	0.976	0.976184364	1.255365849
LGBMClassifier	0.976190476	0.976	0.976184364	1.296026707
DecisionTreeClassifier	0.968253968	0.968307692	0.968397249	0.015122652
RidgeClassifier	0.968253968	0.968	0.968049485	0.058660507
Perceptron	0.896825397	0.896	0.898250412	0.036023855
BernoulliNB	0.595238095	0.6	0.461222501	0.01554513
AdaBoostClassifier	0.579365079	0.584	0.485043763	0.148079157
DummyClassifier	0.158730159	0.159076923	0.156261872	0.017313242

References

A. References

References need not be cited in text. When they are, they appear on the line, in square brackets, inside the punctuation.

[1]Reddy, M. S. (2012). Depression - the global crisis. Indian J. Psychol. Med. 34, 201–203. doi: 10.4103/0253-7176.106011

[2] Kroenke, K., and Spitzer, R. L. (2002). The phq-9: a new depression diagnostic and severity measure. Psychiatr. Ann. 32, 509–521. doi: 10.3928/0048-5713-20020901-06

[3] McPherson, A., and Martin, C. R. (2010). A narrative review of the beck depression inventory (bdi) and implications for its use in an alcohol-dependent population. J. Psychiatr. Ment. Health Nurs. 17, 19–30. doi: 10.1111/j.1365-2850.2009.01469.x

[4] Ooi, K. E. B., Lech, M., and Allen, N. B. (2013). Multichannel weighted speech classification system for prediction of major depression in adolescents. IEEE Trans. Biomed. Eng. 60, 497–506. doi: 10.1109/TBME.2012.2228646

[5] Schwartz, G. E., Fair, P. L., Salt, P., and Klerman, G. L. (1976). Facial expression and imagery in depression: an electromyographic study. Psychosom. Med. 38, 337–347. doi: 10.1097/00006842-197609000-00006

[6] Alghowinem, S. (2015). Multimodal analysis of verbal and nonverbal behaviour on the example of clinical depression (Ph.D. thesis). The Australian National University Canberra, ACT.

[7] Alghowinem, S., Goecke, R., Wagner, M., Epps, J., Hyett, M.,Parker, G., et al. (2018). Multimodal depression detection: fusion analysis of paralinguistic, head pose and eye gaze behaviors. IEEE Trans. Affect. Comput. 9, 478–490. doi: 10.1109/TAFFC.2016.26 34527

[8] Alghowinem, S., Goecke, R., Wagner, M., Parker, G., and Breakspear, M. (2013). "Eye movement analysis for depression detection," in 2013 IEEE International Conference on Image Processing (Melbourne, VIC), 4220–4224.

[9] Winograd-Gurvich, C., Georgiou-Karistianis, N., Fitzgerald, P., Millist, L., and White, O. (2006). Ocular motor differences between melancholic and non-melancholic depression. J. Affect. Disord. 93, 193–203. doi: 10.1016/j.jad.2006.03.018

[10] Gong, B., Wang, Y., Liu, J., and Tang, X. (2009). "Automatic facial expression recognition on a single 3d face by exploring shape deformation," in Proceedings of the 17th ACM International Conference on Multimedia, MM '09 (New York, NY: Association for Computing Machinery), 569–572. doi: 10.3969/j.issn.1000-6729.2011.01.011

[11] Gong, X., Huang, Y., Wang, Y., and Luo, Y. (2011). Revision of the chinese facial affective picture system. Chinese Ment. Health J. 25, 40–46.

[12] Guo W, Yang H, Liu Z, Xu Y and Hu B (2021) Deep Neural Networks for Depression Recognition Based on 2D and 3D Facial Expressions Under Emotional Stimulus Tasks. Front. Neurosci. 15:609760. doi: 10.3389/fnins.2021.609760

[13] https://towardsdatascience.com/light-on-math-machine-learning-intuitive-guide-to-understanding-kl-divergence-2b382ca2b2a8

[14] https://analyticsindiamag.com/hands-on-guide-to-bayesian-neural-network-in-classification/

[15] iMotion program guide, 2018

[16] Github link: https://github.com/TariqueMahmud/Application-of-Bayesian-Deep-Neural-Networks-for-Depression-Recognition

[17] https://jacobbuckman.com/2020-01-17-a-sober-look-at-bayesian-neural-networks/

[18] https://info.cambridgespark.com/latest/neural-networks-in-python

[19]https://www.kaggle.com/datasets/laavanya/human-stress-detection-in-and-through-sleep

[20] SaYoPillow: Blockchain-Integrated Privacy-Assured IoMT Framework for Stress Management Considering Sleeping Habits.
Laavanya Rachakonda, Student Member, IEEE, Anand K. Bapatla, Student Member, IEEE, Saraju P. Mohanty, Senior Member, IEEE and Elias Kougianos, Senior Member, IEEE.

Appendix: Code base

```
# -*- coding: utf-8 -*-
"""Final Machine Learningf project.ipynb

Automatically generated by Colaboratory.

Original file is located at

https://colab.research.google.com/drive/1zoJY9jW3z_SPIIN-uW1auZpQUCVxQEUr
"""

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import csv
import sklearn
from sklearn.linear_model import LinearRegression
import statsmodels.api as sm
import seaborn as sns
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import MinMaxScaler

from google.colab import files
import io

files = files.upload()

df = pd.read_csv(io.StringIO(files['dataset.csv'].decode('latin-1')))

df.tail()

df.columns

"""1. (i)
###-Dataset is a numerical data
###-We can use classification
### Column labeling:
snoring range of the user: sr2,
respiration rate: rr,
body temperature: t,
limb movement rate: lm,
blood oxygen levels: bo,
eye movement: rem,
number of hours of sleep: sr,
heart rate: hr and
Stress Levels: sl (0- low/normal, 1 – medium low, 2- medium, 3-medium high, 4 -high)

Descriptive analysis:
"""

df.describe().T

"""Multivariate data plot
"""

ax = df[["sr2","rr","t","t","lm","bo", "rem", "sr", "hr", "sl"]].plot()
ax.legend(loc='center left', bbox_to_anchor=(1, 0.5));

"""Here we can see the variation in the each of the data deatures by ploting the each sample value."""

df.apply(np.std)

"""Here we can see that data need to be standerized after analysing the SD of the data as sr2 and rem value is higher than others.

(vi) Data visualization with seaborn and matplotlib
"""

sns.heatmap(df)

df.plot(kind="hist")

"""(ii) Correlation"""

corr = df.corr()
fig = plt.figure()
ax = fig.add_subplot(111)
cax = ax.matshow(corr,cmap='coolwarm', vmin=-1, vmax=1)
fig.colorbar(cax)
ticks = np.arange(0,len(df.columns),1)
ax.set_xticks(ticks)
plt.xticks(rotation=90)
ax.set_yticks(ticks)
ax.set_xticklabels(df.columns)
ax.set_yticklabels(df.columns)
plt.show()

"""Here its visible that sl has clear corelation with the other values whether its positive or negative independent variables."""

sns.set()
cols = ['sr2', 'rr', 't', 'lm','bo', 'rem', 'sr', 'hr', 'sl']
sns.pairplot(df[cols], size = 3.5)
plt.show()

corr_mat = df.corr(method='spearman')
cg = sns.clustermap(corr_mat, cmap="PuBuGn", linewidths=0.1);
plt.setp(cg.ax_heatmap.yaxis.get_majorticklabels(), rotation=0)
cg

"""(v) PCA analysis with visualiztion

- Data Standardiztion and PCA analysis
"""

df_st = StandardScaler().fit_transform(df)
pd.DataFrame(df_st, columns = df.columns ).head(2)
```

```
pca_out = PCA().fit(df_st) #getting the component varienne
# Proportion of the vatince we can see.

pca_out.explained_variance_ratio_

np.cumsum(pca_out.explained_variance_ratio_)

loadings = pca_out.components_
num_pc = pca_out.n_features_

pc_list = ["PC"+str(i) for i in list(range(1, num_pc+1))]
loadings_df = pd.DataFrame.from_dict(dict(zip(pc_list, loadings)))
loadings_df['variable'] = df.columns.values
loadings_df = loadings_df.set_index('variable')
loadings_df

ax = sns.heatmap(loadings_df, annot=True, cmap='viridis')
plt.show()

"""b. data preparation"""

df.isnull().sum()   #Checking the null values

"""c. designing the test"""

from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

#a linear model, Ridge Classifier.

scaler = StandardScaler()

from sklearn.linear_model import RidgeClassifier

Stress_Levels = ['Low','Medium Low','Medium','Medium High','High']
Feature_Coefficients = pd.DataFrame()
for i in range(0,5):
   SleepStress_Pred = df.copy()
   SleepStress_Pred['sl'] = SleepStress_Pred['sl'].apply(lambda x: 1 if x==i else 0)
   X = SleepStress_Pred.drop('sl',axis=1)
   y = SleepStress_Pred['sl']
   X_train, X_test, y_train, y_test = train_test_split(X,y,train_size=0.6,random_state=100)

   X_train = scaler.fit_transform(X_train)
   X_train = pd.DataFrame(X_train,columns=X.columns)

   X_test = scaler.transform(X_test)
   X_test = pd.DataFrame(X_test,columns=X.columns)

   Model = RidgeClassifier(random_state=100)

   params = {'alpha':[10000,1000,100, 10, 1.0, 0.1, 0.01,0.001,0.0001]}

   grid_search = GridSearchCV(estimator=Model,param_grid=params,n_jobs=-1,verbose=1,scoring='accuracy')
   grid_search.fit(X_train,y_train)

   Model_best = grid_search.best_estimator_

   y_train_pred = Model_best.predict(X_train)
   y_test_pred = Model_best.predict(X_test)

   print('Train Accuracy :',accuracy_score(y_train,y_train_pred))
   print('Test Accuracy :',accuracy_score(y_test,y_test_pred))

   Feature_Coefficients['Feature'] = X_train.columns
   Feature_Coefficients[Stress_Levels[i]] = Model_best.coef_[0]

Feature_Coefficients.set_index('Feature',inplace=True)
Feature_Coefficients.head(10)

plt.figure(figsize=(10,6))
sns.heatmap(Feature_Coefficients,annot=True)
plt.title('Heatmap of Feature Coefficients')
plt.show()

"""#With the feature coefficients for all the various stress levels. This analysis is useful for understanding feature and stress level relationship. """

from scipy import stats
df[["sr2","rr","t","lm"]].describe()
ttest,pval1 = stats.ttest_rel(df['sr2'], df['rr'], df['t'], df['lm'])
print(pval1)
if pval1<0.05:
   print("reject null hypothesis")
else:
   print("accept null hypothesis")

df[["sr2","rr","t","lm"]].describe()
ttest,pval2 = stats.ttest_rel(df['bo'], df['rem'], df['sr'], df['hr'])
print(pval2)
if pval2<0.05:
   print("reject null hypothesis")
else:
   print("accept null hypothesis")

df.drop_duplicates(inplace=True)
df.shape

df.rename(columns = {'sr2' : 'snoring range' ,'sr':'snoring rate', 'rr':'respiration rate',
                  't':'body temperature', 'lm':'limb movement',
                  'bo':'blood oxygen', 'rem':'eye movement',
                  'sr.1':'sleeping hours','hr':'heart rate',
                  'sl':'stress level'}, inplace = True)
df.head()

!pip install lazypredict
```

```
pip uninstall scikit-learn -y

pip install scikit-learn==0.23.1

from lazypredict.Supervised import LazyClassifier

y = df.pop('stress level')
X = df

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split( X, y, test_size=0.2, random_state=42,shuffle=True, stratify=y)

clf = LazyClassifier(verbose=0,predictions=True)
models,predictions = clf.fit(X_train, X_test, y_train, y_test)
models

predictions.head()

from sklearn.metrics import classification_report
for i in predictions.columns.tolist():
    print('\t',i,'\n')
    print(classification_report(y_test, predictions[i]),'\n')

"""Most of the models are giving as overfitting however Decision Tree classifier and ridge classifier are giving acceptable accuracy."""
```

Github link:

https://github.com/TariqueMahmud/Application-of-Machine-Learning-algorithms-for-Stress-level-Recognition

Google Colab link:

https://colab.research.google.com/drive/1IyCYB08NiQHWqxm-uEardBG9-Keox19B?usp=sharing