
**A Synopsis on**

# "ANTICIPATING MENSTRUAL MIGRAINE USING DEEP LEARNING"

**Submitted in partial fulfilment for the award of the degree of**

## BACHELOR OF TECHNOLOGY

**IN**

## COMPUTER SCIENCE AND ENGINEERING

**(ARTIFICIAL INTELLIGENCE)**

**Submitted by**

**Pranav Polavarapu**
**19BTRCR008**

**Sushil Bokade**
**19BTRCR017**

**Sai Keerthi Chelluri**
**19BTRCR036**

**Sai Sharanya Y**
**19BTRCR043**

**Under the guidance of**
**Prof. Mohammed Zabeeulla A N**

**(HOD)**

**Faculty of Engineering & Technology**

**Jain (Deemed-To-Be University)**

**Department of Computer Science & Engineering**

Jain Global Campus, KanakapuraTaluk - 562112

Ramanagara District, Karnataka, India

2020-2021

## 1. Introduction

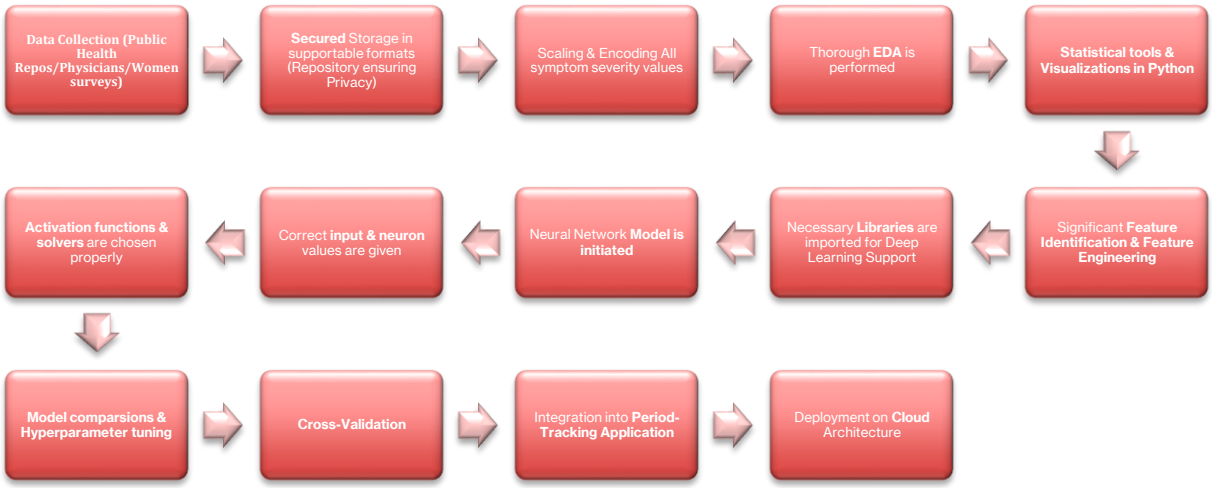
This project aims to build an automated system that identifies, classifies, and anticipates an occurrence of Menstrual Migraine in women. This is based on the severity of symptoms that a woman experiences – like throbbing, periods, Nausea, visual disturbances, intensity, & many more. The data collected from a woman is cleaned & modelled into the Deep Learning Classification system that identifies the type of migraine occurring (Menstrual | non-Menstrual | Others). This classification system can be integrated with period-tracking applications and cloud-based systems to scale up the impact created on a woman's health. The project enables women to consider prior precautions & medication in cases of menstrual-related migraines.

## 2. Literature Survey/Review

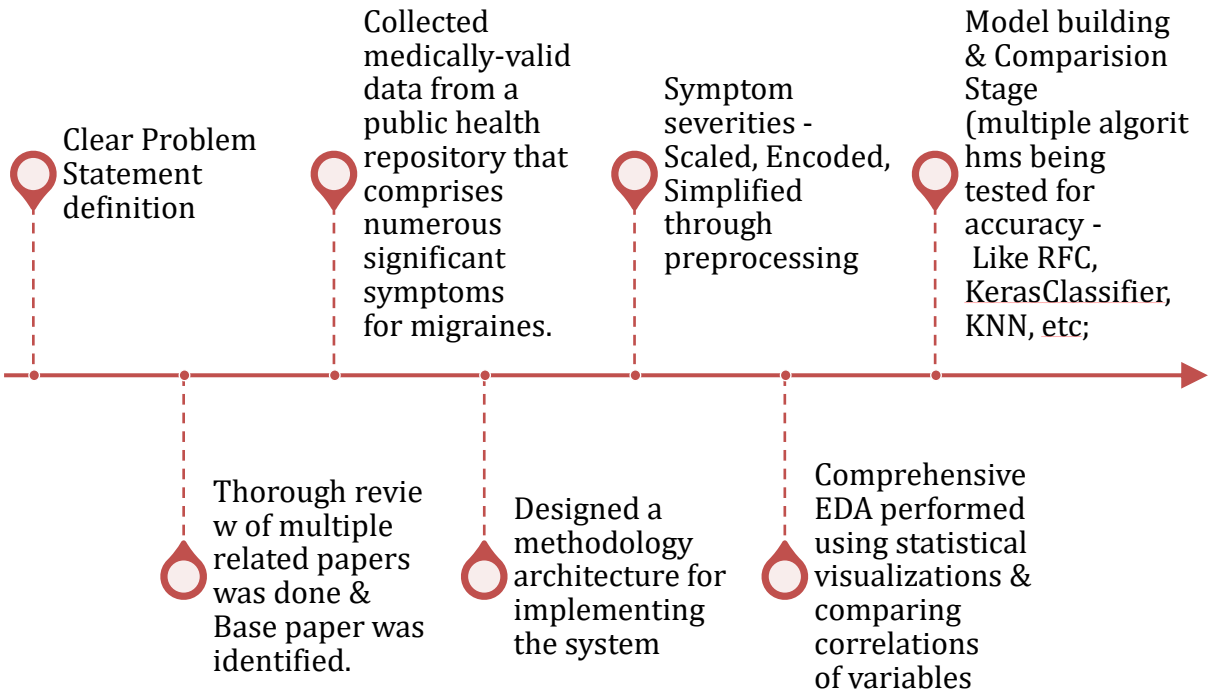
| No. (Author) | Name | Methodology | Drawbacks |
|---|---|---|---|
| **1** (Alkan A) | Use of K-means clustering in migraine detection by using EEG records under flash stimulation | EEG signals are pre-processed & analysed using Histograms. Features are set to K-means algorithm for detecting migraine. | Based upon analysing brain wave signals. Requires resources and is inefficient. |
| **2** (Subasi A, Tuncel D) | Analysis of repetitive flash stimulation frequencies and record periods to detect migraine using ANN | Flash stimulation is based on frequency analysis in EEGs. Frequency ranges in Hz are used to determine migraine at beta-bands | Complex process of analysing frequencies over time. Requires long duration and equipment to just detect migraine. |
| **3** (Akben SB, Subasi A, Kiymik MK) | Comparison of artificial neural network and support vector machine classification methods in diagnosis of migraine by using EEG | Frequency spectrums are classified using ANN & SVMs. Detecting migraine among headache symptoms, using triggering factors. | Similar to the previous ones |
| **4** (Andrew Charles) | The Migraine Aura | Basic mechanisms of migraine aura and its clinical significance based upon evidence from human studies and animal models. | Only intended to identify migraine with Aura (unrelated to non-aura migraines) |
| **5** (Catherine, Nathan) | Migraine classification using magnetic resonance imaging resting | Used ML techniques to develop biomarkers for rs-fMRI data | Just provided insights upon pain circuits in the brain that acted as biomarkers to identify migraine patients |

| | | | |
|---|---|---|---|
| **6** (Wei-Ta Chen, Cing-Yan) | Migraine classification by ML with functional near-infrared spectroscopy during the mental arithmetic task | Subjects' haemoglobin in the brain was measured using fNIRS during mental attacks | fNIRS & haemoglobin analysis is not quite the easy or standard way to quickly identify the types of migraine. |
| **7** (Yolanda Garcia-Chimeno, Juan Carlos) | Automatic migraine classification via feature selection committee and ML techniques over imaging and questionnaire data | Diffusion tensor Images & questionnaires like IQ were used to enable feature selection & Multiple ML techniques. | DTI images & IQ questionnaires are a complex process to create classification models. Lot of scope for optimization |
| **8** (Konrad Jackowski) | Migraine Diagnosis Support System Based on Classifier Ensemble | Ensemble pool of elementary classifiers are combined to increase accuracy in identifying migraine from regular headaches | Does not provide any insights into types of migraines based on aura, and null on menstrual-related migraines |
| **9** (Paola Sanchez) | Automatic migraine classification using artificial neural networks | Designs & tests an early classification system to classify 7 different types of migraines using ANN | No significant EDA performed. Increased dimensions and incomprehensible correlations among the variables. Leaves scope for efficient system to be built over neural networks (other than old MLP Classifier) |
| **10** (Stephen D Silverstein) | Diagnosis and Treatment of the Menstrual Migraine Patient | Provides medical diagnostic criteria for women having migraine without Aura. | Provides a perspective into post-identification procedures & treatments for women with menstrual migraine. |

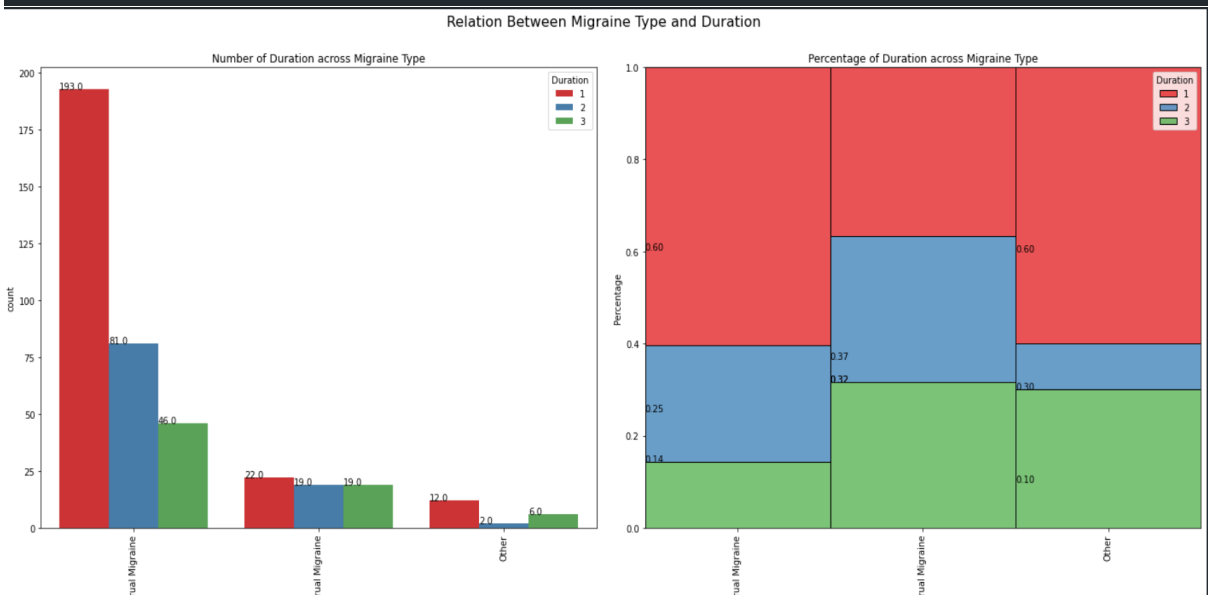
## 3. Proposed Methodology

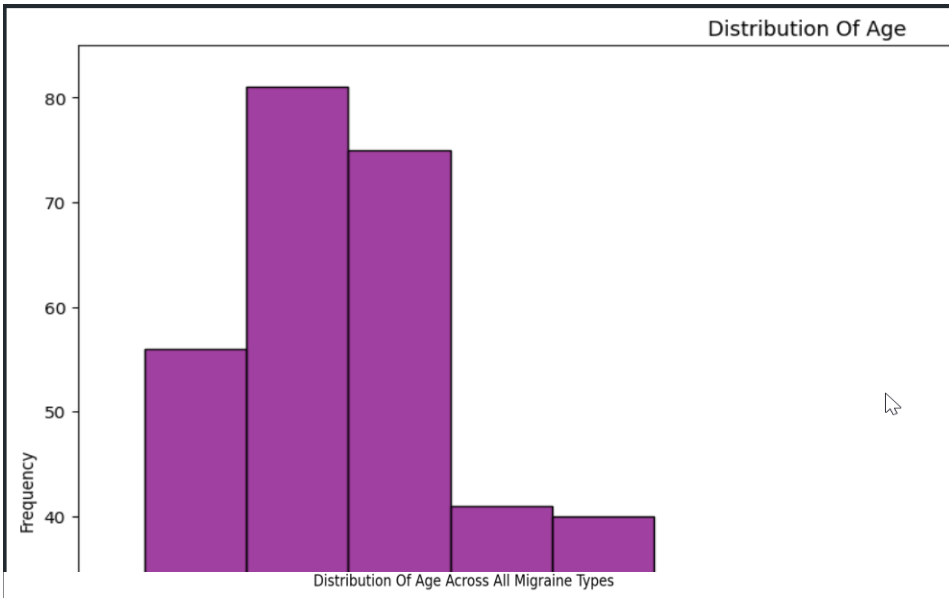


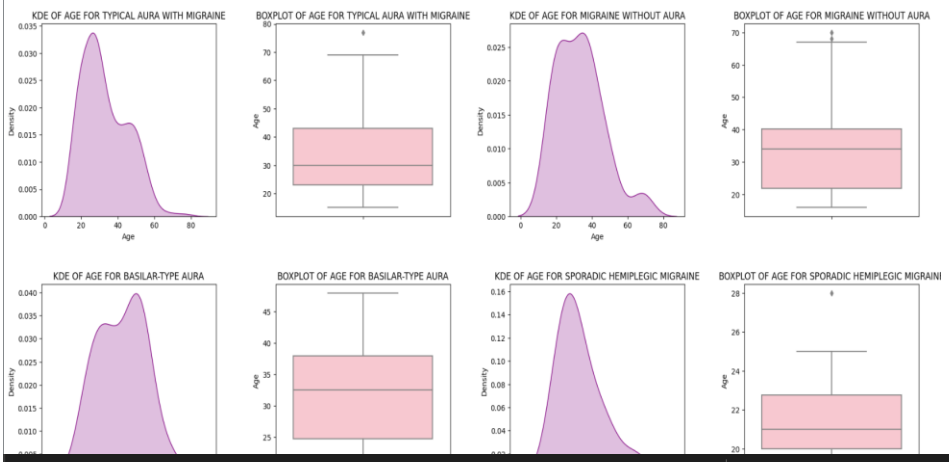

We propose an efficient methodology to accurately classify the headaches into 3 major types. Our primary focus would be on identifying menstrual migraines from regular migraines so that one can anticipate its occurrence for prior treatment properly. Data shall be collected from public health repositories, physicians, and women who are experiencing a variety of symptoms relating to headaches or migraines. This data is pre-processed and maintained neatly in a comprehensible format. The data is comprised of numerous quantifiable and simple but important symptoms that a woman would experience. Ex – Nausea, Visual aura, Intensity level, Frequency, Period status, etc. All the symptoms and severities are scaled and encoded into simpler values for efficient processing by the models. Data undergoes thorough Exploratory Analysis to identify correlations, impacts and insignificances of the variables present at hand. This is done in a visual manner for better understanding of the data. Finally, the important features are identified, and they are ready to be fed into a Deep Learning- based Neural network Classifier. The classifier system is built using Neural Networks for categorizing multiple classes effectively. Each layer is added in a sequential format, with the correct parameters relating to the shape of the input & output data. Right activations and loss functions are chosen for productive multi-class classification. Final set of models and its hyperparameters are tuned using methods like cross validation. This system is intended to be integrated into a period-tracking system for efficient User-Interface and easy healthcare.




| | Age | Duration | Frequency | Location | Character | Intensity | Nausea | Vomit | Phonophobia | Photophobia | ... | Hypoacusis | Diplopia | Visual_defect |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 30 | 1 | 5 | 1 | 1 | 2 | 1 | 0 | 1 | 1 | ... | 0 | 0 | 0 |
| 1 | 50 | 3 | 5 | 1 | 1 | 3 | 1 | 1 | 1 | 1 | ... | 0 | 0 | 0 |
| 2 | 53 | 2 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | ... | 0 | 0 | 0 |
| 3 | 45 | 3 | 5 | 1 | 1 | 3 | 1 | 0 | 1 | 1 | ... | 0 | 0 | 0 |
| 4 | 53 | 1 | 1 | 1 | 1 | 2 | 1 | 0 | 1 | 1 | ... | 0 | 0 | 0 |






```
[ ]
    #Defining our Base Model
    def baseline_model():
        # Create model here
        model = Sequential()
        model.add(Dense(14, input_dim = 24, activation = 'relu')) # Rectified Linear Unit Activation Function
        model.add(Dense(14, activation = 'relu'))
        model.add(Dense(3, activation = 'softmax')) # Softmax for multi-class classification
        # Compile model here
        model.compile(loss = 'categorical_crossentropy', optimizer = 'adam', metrics = ['accuracy'])
        return model
```

```
[ ]  #Create Keras Classifier to use predefined base model
    estimator = KerasClassifier(build_fn = baseline_model, epochs = 100, batch_size = 10, verbose = 0)
```

```
    /usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: DeprecationWarning: KerasClassifier is depreca
```

```
[ ]  # KFold Cross Validation
    kfold = KFold(n_splits = 5, shuffle = True, random_state = seed)
```

```
[ ]  # Object to describe the result
    results = cross_val_score(estimator, x, y, cv = kfold)
    # Result
    print("Result: %.2f%% (%.2f%%)" % (results.mean()*100, results.std()*100))
```

```
    Result: 99.75% (0.50%)
```

```
[ ]  # Create a model
     rfc = RandomForestClassifier(max_depth=23,
                                  max_features='sqrt',
                                  min_samples_split=10,
                                  n_estimators=40)

     # Fit the model
     rfc.fit(X_train, Y_train)
```

```
     RandomForestClassifier(max_depth=23, max_features='sqrt', min_samples_split=10,
                            n_estimators=40)
```

```
     # Print metrics of the model
     model_metrics(rfc)
```

```
     ***** Model Fit *****

     FIT ON TRAIN DATA    --> 1.0
     FIT ON TEST DATA     --> 0.9916666666666667
```

```
]  # Adding the input layer and the first hidden layer...
   classifier.add(Dense(units=13, kernel_initializer='uniform', activation='relu', input_dim=24))
```

```
]  # Adding the second hidden layer...
   classifier.add(Dense(units=13, kernel_initializer='uniform', activation='relu'))
```

```
]  # Adding the last and final output layer...
   classifier.add(Dense(units=1, kernel_initializer='uniform', activation='softmax'))
```

```
]  # Compiling the ANN...
   classifier.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

```
   # Fitting the ANN to the Training set...
   classifier.fit(X_train, y_train, batch_size=10, epochs=100)
```

```
Epoch 33/100
30/30 [==============================] - 0s 1ms/step - loss: 0.0000e+00 - accuracy: 0.1533
Epoch 34/100
30/30 [==============================] - 0s 1ms/step - loss: 0.0000e+00 - accuracy: 0.1533
Epoch 35/100
30/30 [==============================] - 0s 1ms/step - loss: 0.0000e+00 - accuracy: 0.1533
Epoch 36/100
30/30 [==============================] - 0s 2ms/step - loss: 0.0000e+00 - accuracy: 0.1533
Epoch 37/100
30/30 [==============================] - 0s 1ms/step - loss: 0.0000e+00 - accuracy: 0.1533
```

## 4. Future Scopes and Application

***Integration into period tracking applications:***

This project serves as a feature which can be added to the period trackers and can be used for other period tracking applications. The integration of this project to the period tracking applications will open other more possibilities to the tracking and would also help the user to get a clear view about a women's migraine symptoms.

***Building a simple UI for surveying women in real-time:***

A simple user-interface gives a user more freedom to explore it and use it and also the system to get most out of the user and hence the project is intended to create a simple UI in order to get surveys done as to get to know about the menstrual migraine and its symptoms which could really help researchers for their further study.

***Employing cloud applications for scaling up operations:***

Cloud applications are used now in a large scale and are the most handy and comfortable way one can work on with data and no doubt the future truly relies on it so taking all this into consideration this project to could be deployed on cloud for more smooth and handy usage, configuration and maintenance.

## 5. References


1. Ahmet, Alkan, and Batuhan Akben Selahaddin. "Use of k-means clustering in migraine detection by using EEG records under flash stimulation." *International Journal of Physical Sciences* 6, no. 4 (2011): 641-650.
2. Akben, Selahaddin Batuhan, Abdulhamit Subasi, and Deniz Tuncel. "Analysis of repetitive flash stimulation frequencies and record periods to detect migraine using artificial neural network." *Journal of medical systems* 36, no. 2 (2012): 925-931.
3. Akben, Selahaddin Batuhan, Abdülhamit Subaşı, and Mahmut Kemal Kıymık. "Comparison of artificial neural network and support vector machine classification methods in diagnosis of migraine by using EEG." In *2010 IEEE 18th Signal Processing and Communications Applications Conference*, pp. 637-640. IEEE, 2010.
4. Charles, Andrew. "The migraine aura." *Continuum: Lifelong Learning in Neurology* 24, no. 4 (2018): 1009-1022.
5. Chong, Catherine D., Nathan Gaw, Yinlin Fu, Jing Li, Teresa Wu, and Todd J. Schwedt. "Migraine classification using magnetic resonance imaging resting-state functional connectivity data." *Cephalalgia* 37, no. 9 (2017): 828-844.
6. Ang, Kai Keng, Cuntai Guan, Kerry Lee, Jie Qi Lee, Shoko Nioka, and Britton Chance. "A brain-computer interface for mental arithmetic task from single-trial near-infrared spectroscopy brain signals." In *2010 20th International Conference on Pattern Recognition*, pp. 3764-3767. IEEE, 2010.
7. Garcia-Chimeno, Yolanda, Begonya Garcia-Zapirain, Marian Gomez-Beldarrain, Begonya Fernandez-Ruanova, and Juan Carlos Garcia-Monco. "Automatic migraine classification via feature selection committee and machine learning techniques over imaging and questionnaire data." *BMC medical informatics and decision making* 17, no. 1 (2017): 1-10.
8. Jackowski, Konrad, Dariusz Jankowski, Dragan Simić, and Svetlana Simić. "Migraine diagnosis support system based on classifier ensemble." In *International conference on ICT innovations*, pp. 329-339. Springer, Cham, 2014.
9. Sanchez-Sanchez, Paola A., José Rafael García-González, and Juan Manuel Rúa Ascar. "Automatic migraine classification using artificial neural networks." *F1000Research* 9 (2020).
10. Silberstein, Stephen D., and Susan L. Hutchinson. "Diagnosis and treatment of the menstrual migraine patient." *Headache: The Journal of Head and Face Pain* 48 (2008): S115-S123.