**ML And DL Based Brain Stroke Prediction By Analysing Structured Data**

***Abstract -* Brain Stroke continues to be a major global health issue, as it is a contributing factor to high rates of morbidity and mortality. Early detection and prevention of stroke are made possible by the potential intersection of deep learning (DL) and machine learning (ML) in the healthcare industry. The performance constraints of current methods for early stroke detection highlight the need for creative fixes. Here, we introduce a brand-new web-based tool that evaluates a person's risk of stroke based on important factors like gender and BMI. Our interface makes use of cutting-edge predictive modelling techniques to offer a user-friendly platform that overcomes the shortcomings of current approaches. After extensive testing, our suggested combination of Logistic Regression (Grid searched), SMOTE and Standard Scaler obtains a 36.7% F1 score for the minority class (stroke) and that of 96.4% for the majority class (no stroke) in our unbalanced dataset. By providing a workable solution for stroke risk assessment and bridging the gap between intricate predictive models and public accessibility, this study advances the field of health informatics. With potential applications for both individuals looking to assess their stroke risk and healthcare professionals involved in preventive care initiatives, the proposed interface holds promise for improving public health outcomes by enabling early detection and prevention of stroke.**

***Index Terms -*** Deep Learning, Early Stroke Detection, Machine Learning, Predictive Modelling, Public Health, Stroke Risk Assessment, Web-based Interface

1. **Introduction**

Stroke is a leading cause of death and disability worldwide. According to a statement by Dr PK Singh, WHO regional director for South-East Asia, 11 million strokes occur in low-and middle-income countries. This causes 4 million deaths annually and leaves approximately 30% of survivors seriously disabled. For the 70% of survivors who recover, the likelihood of suffering further strokes is greatly increased [1]. Early detection and intervention can significantly reduce the severity of outcomes and improve patient prognosis. Stroke (also called cerebrovascular accident), is the damage to the brain because of interruption of its blood supply. A stroke occurs when a blood vessel in the brain becomes blocked or narrowed, or when a blood vessel bursts and spills blood into the brain [2]. The two main types of brain strokes are Ischemic Stroke and Hemorrhagic Stroke [3]. Since there is no effective solution for stroke, early detection can prevent various deaths [4]. If the stroke is detected or diagnosed early, the loss of death and severe brain damage can be prevented in 85% of cases [5]. The stroke starts with a ministroke which is known as a transient ischemic attack (TIA). It is the condition which reveals that the person will face a stroke within a couple of days from the occurrence of the ministroke [6]. With the advent of machine learning and deep learning, predictive models have been developed to assess the risk of stroke based on various health parameters such as BMI, Smoking Status etc. However, these models are often complex and require technical expertise to use, limiting their accessibility to the general public.

In response to this, we have developed a web-based interface that leverages machine learning to assess an individual’s vulnerability to brain stroke. This interface is designed to serve as a pre-check or primary measure for people to determine their potential risk. It is based on a model trained on a dataset from Kaggle, which was selected as the best performing model after training and testing 11 machine learning models (including grid-searched models of the traditional ML models) and 5 deep learning models across 3 types of scalers and 3 types of oversamplers each.

The aim of this paper is to present the development and evaluation of this web-based interface, and to provide a comparative analysis of the different predictive models used. We also discuss the limitations of our interface, particularly the fact that the output does not guarantee medical accuracy due to the unverified nature of the dataset. Despite this, we believe that our interface can serve as a valuable resource for individuals to assess their stroke risk and seek further medical consultation if necessary. We hope that our work will contribute to the field of health informatics and inspire further research in this area.

1. **Related works**

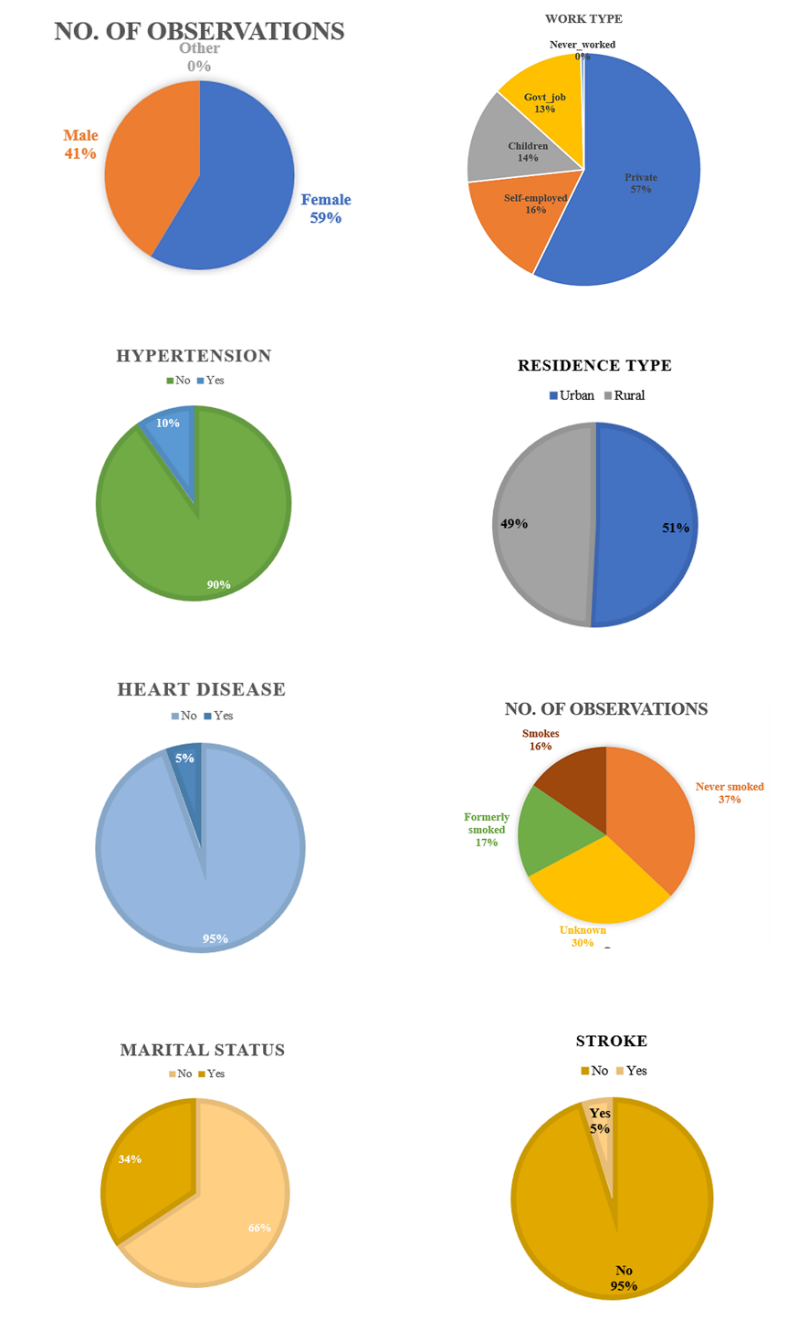
Shoily discussed the detection of stroke disease using machine learning algorithms, which is a challenging task given the complexity of the disease and the variety of risk factors involved [7]. Chang et al. explored the application of machine learning for facial stroke detection, focusing on expressional asymmetry and mouth askew as key facial features of stroke [8]. Kashi et al. proposed a machine-learning model for automatic detection of movement compensations in stroke patients, which can be adapted for use in in-clinic and at-home exercise programs [9]. Al-Mekhlafi et al. discussed the use of deep learning and machine learning for early detection of stroke and haemorrhage, with a focus on the use of the AlexNet model and a hybrid technique between deep learning and machine learning on the Magnetic Resonance Imaging (MRI) dataset for cerebral haemorrhage [10]. Malini et al. presented a cloud-based stroke prediction system to identify approaching stroke using a machine learning approach, with the major goal to accurately detect a stroke and alert the doctor or carer as soon as possible [11]. Tusher et al. developed a system through which brain stroke can be predicted earlier and very firstly, using some classification algorithms such as Logistic Regression, Classification and Regression Tree, K-Nearest Neighbor and Support Vector Machine [12]. Raja et al. discussed the prognosis of stroke using machine learning algorithms, with the main objective to develop an effective stroke detection health care service to predict stroke in better performance [13]. Together, the studies covered above point out a number of shortcomings in the state of stroke detection with machine learning algorithms today. These drawbacks include the need for manual model adaptation for use in clinical and at-home settings, difficulties in achieving early detection using MRI datasets, and reliance on particular facial features for detection, which may not capture all cases of stroke. Concerns exist over the prompt notification of medical personnel or carers as well as the general efficacy of predictive models in correctly identifying stroke cases.

Most common method of detecting stroke are reading CT scans and MRI scans [14-20]. The main reason for which the models can read brain scans is because of the colour differences, that is different tissues of the brain are shown with a different shade [15]. From diagnosis to prevention and treatment of stroke, ML & DL algorithms have helped us a lot. Available algorithms can both detect and classify the stroke. Different models show different accuracies in different types of scans [14, 16-19]. So, we can conclude that no fixed algorithm can be expected to give high-end results for every scan. We carefully read and analysed the types of algorithms, the type of preprocessing, and the type of datasets that these papers use to find the find solution. For this various 3D image reading algorithms have helped us to find the best solution and hence the most accurate results [14, 20].

1. **Methodology**

As an overview, our system consists of multiple important steps. These include using basic data preprocessing techniques like data imputation and data splitting in conjunction with the application of one-hot encoding to multiple features. The cleaned data is then fed into multiple combinations of scaling strategies, oversampling strategies, and machine learning (ML) and deep learning (DL) models. Based on the F1 Score metrics for the minority class, these strategies are evaluated. The best performing model is then chosen and used to build and implement a web-based interface with Streamlit.

The data chosen for this study was taken from Kaggle and it had several lifestyle variables which can be vital for the early prediction of Stroke such as gender, age, BMI, smoking status etc. Relevant patient information is provided by each row in the data. There were a total of 5110 observations in the raw dataset with 12 attributes before preprocessing.



*Figure 1: Distribution of features (a) gender (b) hypertension (c) heart disease (d) marital status (e) work type (f) residence type (g) smoke status (h) outcome of stroke, in the raw data.*

1. **Data preparation**

The initial step in our data preparation process was to eliminate the “id” column from the dataset. This was due to the absence of any additional data that could be mapped to this column, and to avoid potential confusion during model training.

Subsequently, we conducted a thorough examination of the dataset to further enhance its usability. As part of this process, we removed a single entry in the “other” gender category. This decision was made to streamline the data preprocessing phase, and given that this single entry was unlikely to significantly impact the overall dataset of 5110 observations.

Following this, we transformed the categorical values in the dataset into numerical data. This was achieved using the “label\_encoder” from the scikit-learn Python library. The labels which were transformed for the given attributes, are listed in Table 1.

*Table 1: One-hot encoding for the listed attributes.*

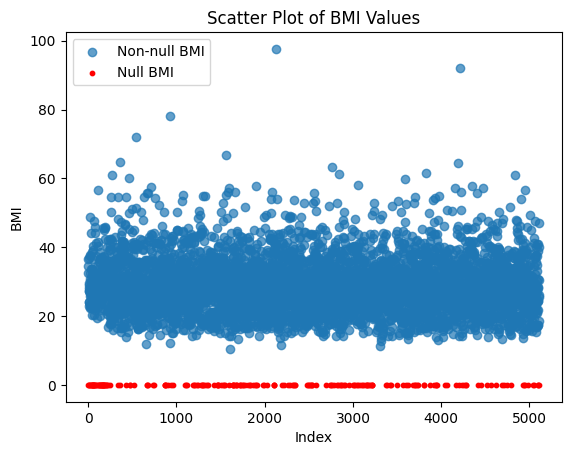
|  |  |  |
| --- | --- | --- |
| **Attribute** | **Original** | **Changed** |
| Gender | Male | 1 |
| Female | 0 |
| Ever Married | Yes | 1 |
| No | 0 |
| Residence Type | Urban | 1 |
| Rural | 0 |

In the data preparation process, we further enhanced the dataset by applying one-hot encoding to the “work\_type” and “smoking\_status” columns. This was accomplished using the “get\_dummies” function from the pandas library in Python. One-hot encoding is a process of converting categorical data variables so they can be provided to machine learning algorithms to improve predictions. With one-hot, we convert each categorical value into a new categorical value and assign a binary value of 1 or 0. Each integer value is represented as a binary vector. This step was crucial in ensuring that our machine learning models could process the data effectively. All the binary vectors are orthogonal to each other. This process makes the representation of categorical data more expressive and easier for the algorithms to understand. As said earlier, the “get\_dummies” method is used for this purpose. This method converts the categorical variable into dummy/indicator variables.

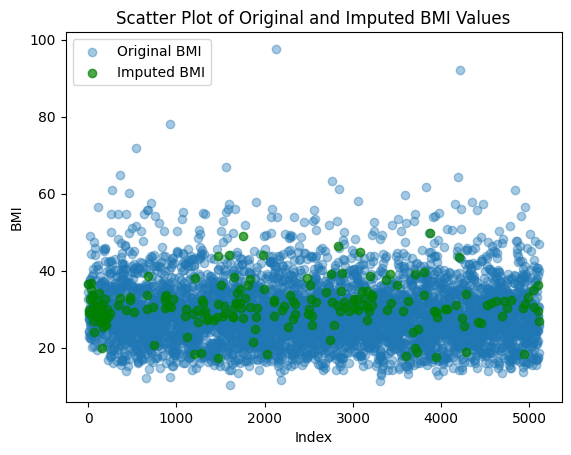
1. **Data imputation**

During our detailed examination of the dataset, we noticed that the "bmi" column contained approximately 4% null values. To handle these missing values, we employed a Random Forest Regressor for imputation. This decision was based on the distribution of the null values, which were scattered throughout the dataset rather than being concentrated in a specific region.

We have included scatter plots to visually represent this process. The first plot displays the original "bmi" values, with null values marked in red and existing values in blue. The second plot shows the dataset after imputation, with the newly imputed "bmi" values indicated in green. These plots provide a clear visual representation of the imputation process and its impact on the "bmi" column in our dataset.



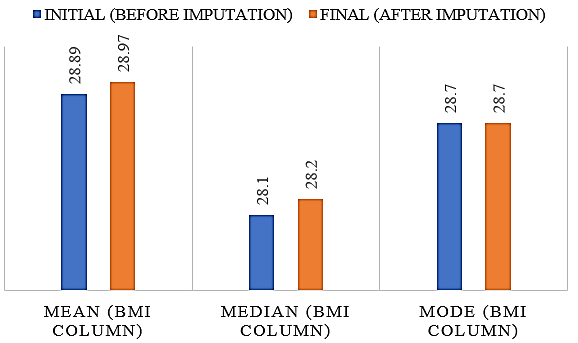
(a)



(b)

*Figure 2: (a) Plot of BMI values in original dataset (b) Plot of BMI values after imputation.*

In addition to handling null values, we also conducted a thorough statistical analysis of the "bmi" column both before and after the imputation process. This involved calculating the mean, median, and mode of the "bmi" values. Our analysis confirmed that the imputation process did not significantly alter these statistical measures, indicating that our imputation method was effective and did not introduce any significant bias into the dataset. The specific values for the mean, median, and mode before and after the imputation process are provided in the plot further. This rigorous statistical analysis ensures the integrity of our data preparation process and the reliability of the subsequent predictive models.

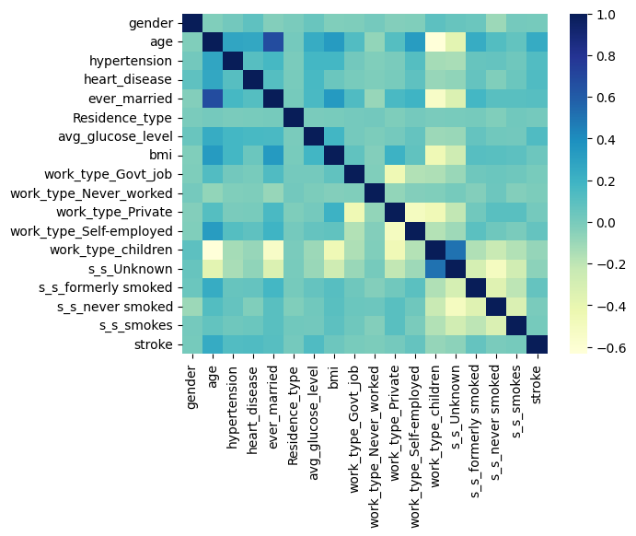


*Figure 3: Mean, median and mode values before and after the imputation*.

1. **Correlation analysis**

To further understand the relationships among the attributes in our dataset, we conducted a correlation analysis. This was visualized using a heatmap, which provides a color-coded representation of the correlation coefficients between each pair of attributes.

However, our analysis did not reveal any significant correlations between any two or three attributes in the dataset. Given this, we decided to retain all the columns in the dataset and did not drop any additional columns. The heatmap illustrating these correlations is provided in Figure 4. This comprehensive analysis ensures that our models are trained on a dataset that accurately represents the complexity and diversity of the factors contributing to stroke risk.

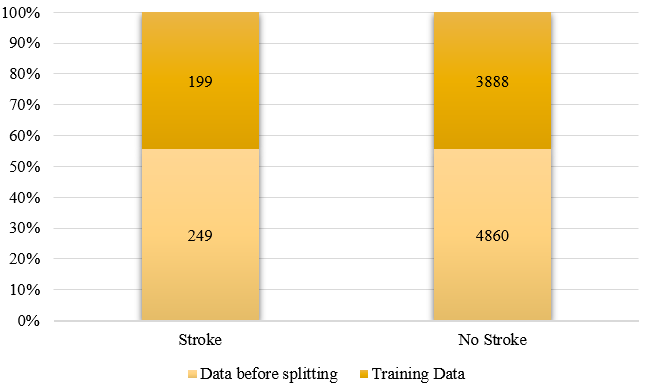


*Figure 4: Correlation heatmap in different data columns*

1. **Data splitting**

Following the data preparation, we segregated the attributes into features and target variables in the code. We then divided the data into training and testing sets, allocating 80% of the data for training and 20% for testing. This was accomplished using the "train\_test\_split" function from the scikit-learn Python library, with "random\_state" set to 42 and "stratify" set to y.

The "stratify" parameter is particularly important in our context. When we set `stratify` to y, we ensure that the distribution of labels (i.e., the target variable) remains consistent in both the training and testing sets. This is crucial when dealing with imbalanced datasets such as ours in this case, where certain labels might be underrepresented. We had only 249 values of “1” in our stroke (target) attribute before splitting, among the total of 5109 observations, making it the minority class. By preserving the original distribution of labels, we can improve the generalizability of our models and prevent them from being biased towards the majority class. This approach enhances the reliability of our subsequent analyses and findings.



*Figure 5: Data samples for each class before and after splitting*

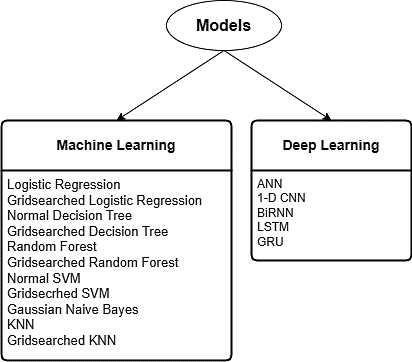
From the figure above, it can be observed that the ratio of the 1 stroke values and the 0 stroke values among the data before splitting and the data used for the training is coherent.

1. **Models and techniques**

Our comparative analysis incorporated a diverse range of models, spanning both traditional Machine Learning algorithms and several Deep Learning architectures.

The choice of these particular models was driven by their widespread use and proven performance in various machine-learning tasks. The traditional Machine Learning models were selected for their interpretability and robustness, while the Deep Learning models were chosen for their ability to model complex, non-linear relationships. Furthermore, we employed grid search techniques

to try and optimize the hyperparameters of the Machine Learning models, in order to extract the maximum predictive power from these algorithms.



*Figure 6: A figure depicting Different types of machine learning and deep learning models used in this study.*

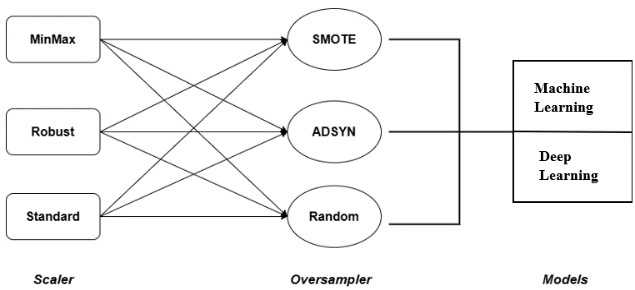
In addition to their individual strengths, the combination of these models allowed us to explore a broad spectrum of the machine learning landscape. This comprehensive approach ensured that our analysis was not limited to a specific type of model or learning algorithm, thereby enhancing the robustness and generalizability of our findings. The selection of these models is detailed in the Table 2.

*Table 2: A table depicting different types of models and the parameters on their initialisation used in this study.*

|  |  |
| --- | --- |
| **Model Name** | **Parameters on initialisation** |
| Logistic Regression (Normal) | random\_state = 42,  class\_weight=‘balanced’ |
| Logistic Regression (Gridsearched) | random\_state = 42, class\_weight=‘balanced’  param\_grid\_lr = {‘penalty’: [‘l1’, ‘l2’],  ‘C’: [0.001, 0.01, 0.1, 1, 10],  ‘solver’: [‘liblinear’, ‘saga’],  ‘max\_iter’: [100, 200, 300]}  cv = 5 |
| Decision Trees (Normal) | random\_state = 42, class\_weight=‘balanced’ |
| Decision Trees (Gridsearched) | random\_state = 42, class\_weight=‘balanced’  param\_grid\_dt = {‘criterion’: [‘gini’, ‘entropy’],  ‘max\_depth’: [None, 10, 20, 30, 40],  ‘min\_samples\_split’: [2, 5, 10],  ‘min\_samples\_leaf’: [1, 2, 4]}  cv = 5 |
| Random Forest (Normal) | random\_state = 42, class\_weight=‘balanced’ |
| Random Forest (Gridsearched) | random\_state = 42, class\_weight=‘balanced’  param\_grid\_rf = {‘n\_estimators’: [50, 100, 200],  ‘max\_depth’: [None, 10, 20],  ‘min\_samples\_split’: [2, 5, 10],  ‘min\_samples\_leaf’: [1, 2]}  cv = 5 |
| SVM (Support Vector Machines) (Normal) | random\_state = 42, class\_weight=‘balanced’ |
| SVM (Support Vector Machines) (Gridsearched) | random\_state = 42, class\_weight=‘balanced’  param\_grid\_svm = {‘C’: [0.1, 1, 10],  ‘kernel’: [‘linear’, ‘rbf’, ‘poly’],  ‘gamma’: [‘scale’, ‘auto’]}  cv = 5 |
| Gaussian Naïve Bayes | Nil |
| K-Nearest Neighbours (Normal) | Nil |
| K-Nearest Neighbours (Gridsearched) | param\_grid\_knn = {‘n\_neighbors’: [3, 5, 7, 9],  ‘weights’: [‘uniform’, ‘distance’],  ‘p’: [1, 2]}  cv = 5 |
| Artificial Neural Network | ann\_model = Sequential([  Dense(128, activation=‘relu’, input\_dim=X\_train\_scaled.shape[1]),  Dropout(0.5),  Dense(64, activation=‘relu’),  Dense(1, activation=‘sigmoid’)  ])  Compiled with: optimizer=‘adam’, loss=‘binary\_crossentropy’, metrics=[‘accuracy’] |
| 1D – Convolutional Neural Network | Reshaping the input data.  cnn\_model = Sequential([  Conv1D(32, 3, activation=‘relu’, input\_shape=X\_train\_cnn.shape[1:]),  MaxPooling1D(2),  Flatten(),  Dense(128, activation=‘relu’),  Dropout(0.5),  Dense(1, activation=‘sigmoid’)  ])  Compiled with: optimizer=‘adam’, loss=‘binary\_crossentropy’, metrics=[‘accuracy’] |
| Bi-directional Recurrent Neural Network | birnn\_model = Sequential([  Bidirectional(LSTM(64, activation=‘relu’, return\_sequences=True), input\_shape=X\_train\_rnn.shape[1:]),  Dropout(0.5),  Bidirectional(LSTM(32, activation=‘relu’)),  Dropout(0.5),  Dense(1, activation=‘sigmoid’)  ])  Compiled with: optimizer=‘adam’, loss=‘binary\_crossentropy’, metrics=[‘accuracy’] |
| Long Short Term Memory (LSTM) | lstm\_model = Sequential([  LSTM(64, activation=‘relu’, input\_shape=X\_train\_rnn.shape[1:], return\_sequences=True),  Dropout(0.5),  LSTM(32, activation=‘relu’),  Dropout(0.5),  Dense(1, activation=‘sigmoid’)  ])  Compiled with: optimizer=‘adam’, loss=‘binary\_crossentropy’, metrics=[‘accuracy’] |
| Gated Recurrent Unit (GRU) | gru\_model = Sequential([  GRU(64, activation=‘relu’, input\_shape=X\_train\_rnn.shape[1:], return\_sequences=True),  Dropout(0.5),  GRU(32, activation=‘relu’),  Dropout(0.5),  Dense(1, activation=‘sigmoid’)  ])  Compiled with: optimizer=‘adam’, loss=‘binary\_crossentropy’, metrics=[‘accuracy’] |

For all the Machine Learning models listed in Table2, we set the ‘scoring’ parameter to either ‘accuracy’ or ‘precision’ wherever applicable. The Deep Learning models, on the other hand, were trained for 25 epochs each with a batch size of 32.

Each model was trained separately on three different datasets, each corresponding to a different oversampling technique: SMOTE, ADASYN, and Random Oversampling. Following oversampling, each dataset was scaled using three different scalers: MinMax, Robust, and Standard Scaler. This process was undertaken to identify the optimal combination of scaler, oversampler, and model to achieve the best performance on the test dataset. This comprehensive comparative study truly explored the breadth and depth of possibilities in our modelling approach.



*Figure 7: Flowchart showing the combination of scalers, oversamplers and models used.*

1. **Experimental Results & discussion**
2. **Experimental Setup & Dataset**

The raw dataset chosen for our study was the Stroke Prediction dataset from Kaggle. Using input criteria including id, gender, age, various anomalies, marital status, work type, residence type, average glucose level, BMI and smoking status, this dataset is used to predict a patient’s risk of having a stroke. Relevant patient information is provided by each row in the data. There were a total of 5110 observations in the raw dataset with 12 attributes originally. These factors were all relevant in stroke prediction [21-30]. This dataset was pre-processed and used to train various models after dividing it into training and testing data.

The constitution of the raw data is shown in Fig. 1.

1. **Results and Discussion**

Our study involved a comprehensive comparative analysis of 144 cases, comprising 99 cases using 11 Machine Learning (ML) models and 45 cases using 5 Deep Learning (DL) models. Each model was trained and tested across three different oversampling techniques (SMOTE, ADASYN, and Random Oversampling) and scaled using three different scalers (MinMax, Robust, and Standard Scaler).

The results of our analysis revealed significant variations in the performance of the different models, oversampling techniques, and scalers.

Table 3: Comparison of accuracy for machine learning models

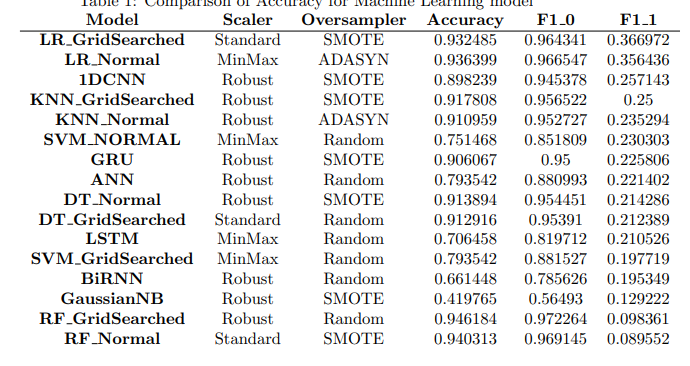


Table 4: Comparison of accuracy for deep learning models

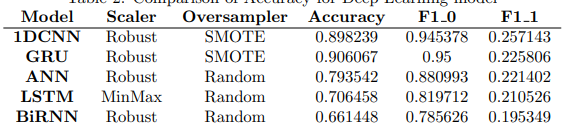
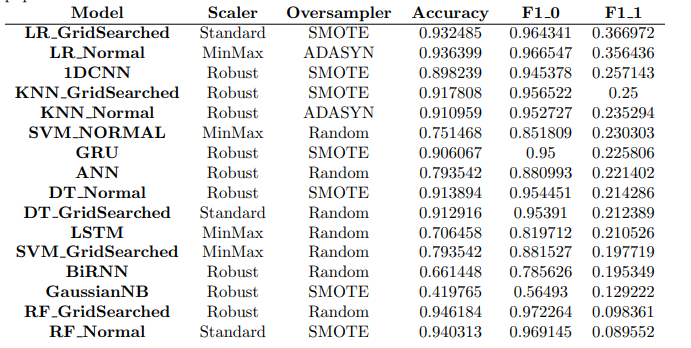
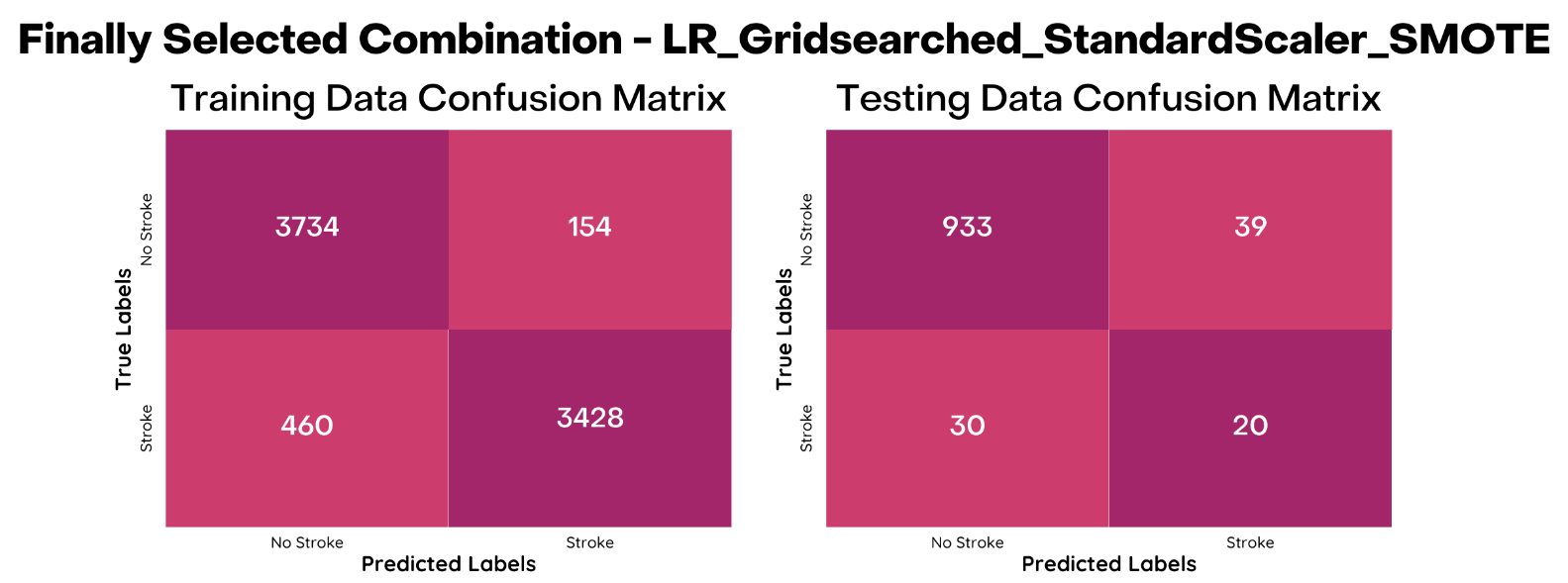


Table 5: Comparison of Accuracy for all the best performing models from our paper



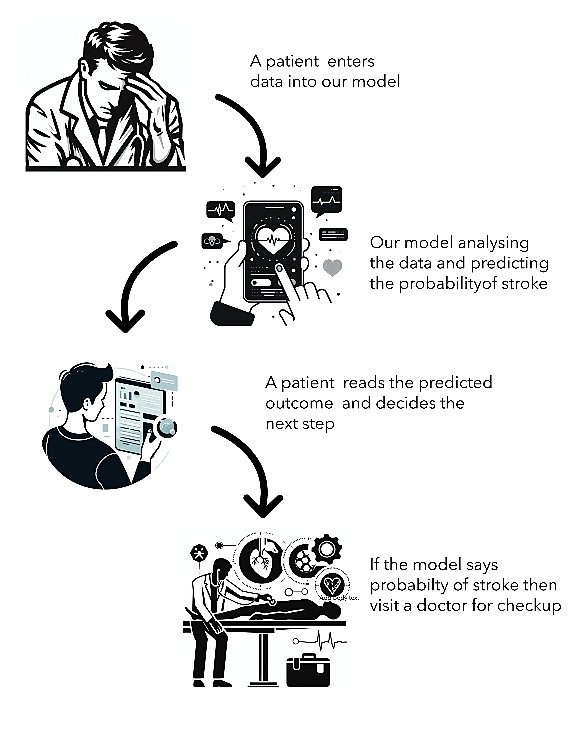
Each combination of model, oversampler, and scaler produced unique results, underscoring the complexity of stroke prediction and the importance of selecting the appropriate techniques for each specific task.

The diversity of results obtained in this study highlights the importance of conducting a comprehensive comparative analysis when developing predictive models. By testing a wide range of models, oversampling techniques, and scalers, we were able to identify the combinations that performed best on our test dataset.



*Figure 8: Confusion matrix for the final model, on the final Scaler and Oversampler used in our web tool*

Our dataset had an inherent class imbalance between instances of strokes (1) and non-strokes (0), so we prioritised the F1 Score for the positive class (F1\_1) as a crucial metric when selecting the model to use for our web application's development. The combination that yielded the highest F1\_1 (on the minority class) score of 0.36697 was the grid-searched Logistic Regression (LR\_Gridsearched) model, used in conjunction with the Standard Scaler and SMOTE oversampling technique. Consequently, this combination was selected for implementation in our web application, which serves as a preliminary check for individuals seeking to assess their susceptibility to stroke.



*Figure 9:* *A diagram illustrating the user-end process*.

The web interface was developed using the Python library, Streamlit, which enabled us to design a user-friendly interface. We initiated the process by generating a template from the ‘streamlit-hello’ app, as per the official Streamlit documentation. This template was then customized on GitHub Codespaces to create our web interface.

Our interface provides an engaging and straightforward interface, akin to filling out an online form. Users can input their age, BMI, work type, and other relevant information via dropdown menus in the Graphical User Interface (GUI). To respect user privacy, all code is executed in real-time without any backend database, ensuring that no user data is collected or stored.

The user’s input is processed through the same scaler used during model training, and the scaled inputs are then fed into the selected model. The model’s output is presented to the user in a clear and comprehensible manner, enabling them to take appropriate action based on the results.

Our web interface is hosted and deployed via Streamlit’s Share platform, also known as the Streamlit Community Cloud. Any cookies associated with the interface are managed by the platform, and their data privacy policy applies. The authors do not store any user data in any form from the interface.

A diagram illustrating the user-end process is provided in Fig 9. The web interface can be accessed via <https://strokedetection.streamlit.app/>.

1. **Conclusion**

In conclusion, this study has successfully developed and evaluated a web-based interface that allows individuals to assess their vulnerability to brain stroke, a leading global cause of death. The interface is grounded on a model trained on a Kaggle dataset, with rigorous testing and training conducted on 11 machine learning models and 5 deep learning models, each with different types of scalers and oversamplers. The best-performing model was ultimately selected for the interface. This research not only provides a valuable tool for early detection and prevention of brain stroke but also contributes to the field of health informatics by offering a comparative analysis of various predictive models. It underscores the potential and effectiveness of machine learning in developing practical health interfaces, paving the way for future advancements in this field. The hope is that this work will inspire further research and development in the application of machine learning for health informatics, ultimately leading to more effective and accessible health care solutions.

**References**

[1] Prevent brain stroke (2019) World Health Organization. Available at: https://www.who.int/southeastasia/news/detail/29-10-2016-prevent-brain-stroke

[2] Stroke. (n.d.). National Institute of Neurological Disorders and Stroke. https://www.ninds.nih.gov/health-information/disorders/stroke#toc-who-is-more-likely-to-have-a-stroke-

[3] Louis R. Caplan, Roger P. Simon, Sara Hassani, Chapter 27 - Cerebrovascular disease—stroke∗, Neurobiology of Brain Disorders (Second Edition), Academic Press, 2023, Pages 457-476, ISBN 9780323856546, https://doi.org/10.1016/B978-0-323-85654-6.00044-7.

[4] B. Kim, N. Schweighofer, J. P. Haldar, R. M. Leahy, and C. J. Winstein, “Corticospinal tract microstructure predicts distal arm motor improvements in chronic stroke,” Journal of Neurologic Physical Therapy, vol. 45, no. 4, pp. 273–281, 2021.

[5] M. Lee, J. Ryu, and D. Kim, “Automated epileptic seizure waveform detection method based on the feature of the mean slope of wavelet coefficient counts using a hidden Markov model and EEG signals,” ETRI Journal, vol. 42, no. 2, pp. 217–229, 2020.

[6] Kaur, M., Sakhare, S. R., Wanjale, K., & Akter, F. (2022, April 11). Early Stroke Prediction Methods for Prevention of Strokes. Behavioural Neurology; Hindawi Publishing Corporation.

[7] Shoily, T. I., Islam, T., Jannat, S., Tanna, S. A., Alif, T. M., & Ema, R. R. (2019, July). Detection of stroke disease using machine learning algorithms. In 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-6). IEEE.

[8] Chang, C. Y., Cheng, M. J., & Ma, M. H. M. (2018, November). Application of machine learning for facial stroke detection. In 2018 IEEE 23rd International Conference on Digital Signal Processing (DSP) (pp. 1-5). IEEE.

[9] Kashi, S., Polak, R. F., Lerner, B., Rokach, L., & Levy-Tzedek, S. (2020). A machine-learning model for automatic detection of movement compensations in stroke patients. IEEE Transactions on Emerging Topics in Computing, 9(3), 1234-1247.

[10] Al-Mekhlafi, Z. G., Senan, E. M., Rassem, T. H., Mohammed, B. A., Makbol, N. M., Alanazi, A. A., ... & Ghaleb, F. A. (2022). Deep learning and machine learning for early detection of stroke and haemorrhage. Computers, Materials and Continua, 72(1), 775-796.

[11] Malini, T., Deepalakshmi, M., Dhivyaa, B., Karthikeswari, P., & Kavipriya, N. (2022, June). Advanced Stroke Detection and Alert System using Machine Learning. In 2022 7th International Conference on Communication and Electronics Systems (ICCES) (pp. 1084-1089). IEEE.

[12] Tusher, A. N., Sadik, M. S., & Islam, M. T. (2022, December). Early Brain Stroke Prediction Using Machine Learning. In 2022 11th International Conference on System Modeling & Advancement in Research Trends (SMART) (pp. 1280-1284). IEEE.

[13] K. S. R. S, B. Chandra, K. Kausalya, C. RM and G. R. V, "Prognosis of Stroke using Machine Learning Algorithms," 2023 7th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2023, pp. 1-6, doi: 10.1109/ICCMC56507.2023.10084158.

[14] Singh, S. P., Wang, L., Gupta, S., Goli, H., Padmanabhan, P., & Gulyás, B. (2019). 3D Deep Learning on Medical Images: A Review. Sensors, 20(18), 5097

[15] M. Chawla, S. Sharma, J. Sivaswamy and L. T. Kishore, "A method for automatic detection and classification of stroke from brain CT images," 2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Minneapolis, MN, USA, 2009, pp. 3581-3584, doi: 10.1109/IEMBS.2009.5335289.

[16] Manisha Sanjay Sirsat, Eduardo Fermé, Joana Câmara “Machine Learning for Brain Stroke: A Review”, Journal of Stroke and Cerebrovascular Diseases, Volume 29, Issue 10, 2020, 105162, ISSN 1052-3057, https://doi.org/10.1016/j.jstrokecerebrovasdis.2020.105162.

[17] A.M. Barrett, Olga Boukrina, Soha Saleh, “Ventral attention and motor network connectivity is relevant to functional impairment in spatial neglect after right brain stroke”, Brain and Cognition, Volume 129, 2019, Pages 16-24, ISSN 0278-2626, https://doi.org/10.1016/j.bandc.2018.11.013.

[18] David A. Brenner, Rich M. Zweifler, Camilo R. Gomez, Brett M. Kissela, Deborah Levine, George Howard, Bruce Coull, Virginia J. Howard, “Awareness, Treatment, and Control of Vascular Risk Factors among Stroke Survivors”, Journal of Stroke and Cerebrovascular Diseases, Volume 19, Issue 4, 2010, Pages 311-320, ISSN 1052-3057, https://doi.org/10.1016/j.jstrokecerebrovasdis.2009.07.001.

[19] Stephen Bacchi, Toby Zerner, Luke Oakden-Rayner, Timothy Kleinig, Sandy Patel, Jim Jannes, “Deep Learning in the Prediction of Ischaemic Stroke Thrombolysis Functional Outcomes: A Pilot Study”, Academic Radiology, Volume 27, Issue 2, 2020, Pages e19-e23, ISSN 1076-6332, https://doi.org/10.1016/j.acra.2019.03.015.

[20] Puttagunta, M., Ravi, S. Medical image analysis based on deep learning approach. Multimed Interfaces Appl 80, 24365–24398 (2021), https://doi.org/10.1007/s11042-021-10707-4.

[21] Haley, Michael J., and Catherine B. Lawrence. "Obesity and stroke: Can we translate from rodents to patients?." Journal of Cerebral Blood Flow & Metabolism 36.12 (2016): 2007-2021.

[22] Miller, J., Kinni, H., Lewandowski, C., Nowak, R., & Levy, P. (2014). Management of hypertension in stroke. Annals of emergency medicine, 64(3), 248-255.

[23] Cipolla, Marilyn J., David S. Liebeskind, and Siu-Lung Chan. "The importance of comorbidities in ischemic stroke: Impact of hypertension on the cerebral circulation." Journal of Cerebral Blood Flow & Metabolism 38.12 (2018): 2129-2149.

[24] Faraco, Giuseppe, and Costantino Iadecola. "Hypertension: a harbinger of stroke and dementia." Hypertension 62.5 (2013): 810-817.

[25] Shuai Zhang, Wei Zuo, Xiao-Feng Guo, Wen-Bin He, Nai-Hong Chen, Cerebral glucose transporter: The possible therapeutic target for ischemic stroke, Neurochemistry International, Volume 70, 2014, Pages 22-29, ISSN 0197-0186, https://doi.org/10.1016/j.neuint.2014.03.007.

[26] H Jørgensen, H Nakayama, H O Raaschou, T S Olsen. “Stroke in patients with diabetes. The Copenhagen Stroke Study.” 1994, Stroke,1977-1984 doi:10.1161/01.STR.25.10.1977

[27] Sifat, A.E.; Nozohouri, S.; Archie, S.R.; Chowdhury, E.A.; Abbruscato, T.J. Brain Energy Metabolism in Ischemic Stroke: Effects of Smoking and Diabetes. Int. J. Mol. Sci. 2022, 23, 8512. https://doi.org/10.3390/ijms23158512

[28] Hawkins, Brian T., Rachel C. Brown, and Thomas P. Davis. "Smoking and ischemic stroke: a role for nicotine?." Trends in pharmacological sciences 23.2 (2002): 78-82.

[29] Jinghua Wang, Xianjia Ning, Li Yang, Jun Tu, Hongfei Gu, Changqing Zhan, Wenjuan Zhang, Ta-Chen Su. “Sex Differences in Trends of Incidence and Mortality of First-Ever Stroke in Rural Tianjin, China, From 1992 to 2012”, 2014, Stroke, Vol. 45, 6 doi:10.1161/STROKEAHA.113.003899

[30] Northern Manhattan Stroke Study Collaborators, Ralph L. Sacco, Bernadette Boden-Albala, Robert Gan, Xun Chen, Douglas E. Kargman, Steven Shea, Myunghee C. Paik, W. Allen Hauser, Stroke Incidence among White, Black, and Hispanic Residents of an Urban Community: The Northern Manhattan Stroke Study, American Journal of Epidemiology, Volume 147, Issue 3, 1 February 1998, Pages 259–268

Comparison of best F1\_1 scores of all DL Models



Comparison of best F1\_1 scores of all ML Models



**Appendix**

The following table depicts the results of each of the ML combination on the test data:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Scaler** | **Oversampler** | **Accuracy** | **Precision\_0** | **Precision\_1** | **Recall\_0** | **Recall\_1** | **F1\_0** | **F1\_1** |
| LR\_GridSearched | Standard | SMOTE | 0.932485 | 0.968847 | 0.338983 | 0.959877 | 0.4 | 0.964341 | 0.366972 |
| LR\_Normal | MinMax | ADASYN | 0.936399 | 0.967044 | 0.352941 | 0.966049 | 0.36 | 0.966547 | 0.356436 |
| LR\_GridSearched | Robust | ADASYN | 0.935421 | 0.96701 | 0.346154 | 0.965021 | 0.36 | 0.966014 | 0.352941 |
| LR\_Normal | Robust | SMOTE | 0.931507 | 0.967842 | 0.327586 | 0.959877 | 0.38 | 0.963843 | 0.351852 |
| LR\_Normal | Standard | ADASYN | 0.92955 | 0.967775 | 0.316667 | 0.957819 | 0.38 | 0.962771 | 0.345455 |
| LR\_GridSearched | Robust | SMOTE | 0.932485 | 0.966908 | 0.327273 | 0.961934 | 0.36 | 0.964415 | 0.342857 |
| LR\_Normal | Standard | SMOTE | 0.931507 | 0.966874 | 0.321429 | 0.960905 | 0.36 | 0.96388 | 0.339623 |
| LR\_Normal | MinMax | SMOTE | 0.932485 | 0.965944 | 0.320755 | 0.962963 | 0.34 | 0.964451 | 0.330097 |
| LR\_GridSearched | Standard | ADASYN | 0.931507 | 0.965909 | 0.314815 | 0.961934 | 0.34 | 0.963918 | 0.326923 |
| LR\_Normal | Robust | ADASYN | 0.930528 | 0.965874 | 0.309091 | 0.960905 | 0.34 | 0.963383 | 0.32381 |
| LR\_GridSearched | MinMax | SMOTE | 0.933464 | 0.965021 | 0.32 | 0.965021 | 0.32 | 0.965021 | 0.32 |
| LR\_GridSearched | MinMax | ADASYN | 0.933464 | 0.964066 | 0.3125 | 0.966049 | 0.3 | 0.965057 | 0.306122 |
| KNN\_GridSearched | Robust | SMOTE | 0.917808 | 0.9625 | 0.225806 | 0.950617 | 0.28 | 0.956522 | 0.25 |
| KNN\_Normal | Robust | ADASYN | 0.910959 | 0.962225 | 0.202899 | 0.943416 | 0.28 | 0.952727 | 0.235294 |
| LR\_GridSearched | MinMax | Random | 0.740705 | 0.986245 | 0.135593 | 0.737654 | 0.8 | 0.844026 | 0.231884 |
| SVM\_NORMAL | MinMax | Random | 0.751468 | 0.983827 | 0.135714 | 0.751029 | 0.76 | 0.851809 | 0.230303 |
| LR\_Normal | MinMax | Random | 0.734834 | 0.98613 | 0.13289 | 0.731481 | 0.8 | 0.839929 | 0.22792 |
| LR\_GridSearched | Robust | Random | 0.733855 | 0.986111 | 0.13245 | 0.730453 | 0.8 | 0.839243 | 0.227273 |
| LR\_Normal | Standard | Random | 0.732877 | 0.986092 | 0.132013 | 0.729424 | 0.8 | 0.838557 | 0.226629 |
| LR\_GridSearched | Standard | Random | 0.732877 | 0.986092 | 0.132013 | 0.729424 | 0.8 | 0.838557 | 0.226629 |
| LR\_Normal | Robust | Random | 0.73092 | 0.986053 | 0.131148 | 0.727366 | 0.8 | 0.837182 | 0.225352 |
| KNN\_GridSearched | Robust | ADASYN | 0.911937 | 0.961297 | 0.19697 | 0.945473 | 0.26 | 0.95332 | 0.224138 |
| SVM\_NORMAL | Robust | Random | 0.772016 | 0.976774 | 0.129555 | 0.778807 | 0.64 | 0.866629 | 0.215488 |
| DT\_Normal | Robust | SMOTE | 0.913894 | 0.960417 | 0.193548 | 0.94856 | 0.24 | 0.954451 | 0.214286 |
| DT\_GridSearched | Standard | Random | 0.912916 | 0.960375 | 0.190476 | 0.947531 | 0.24 | 0.95391 | 0.212389 |
| DT\_Normal | Robust | ADASYN | 0.89726 | 0.96068 | 0.160494 | 0.930041 | 0.26 | 0.945112 | 0.198473 |
| SVM\_GridSearched | MinMax | Random | 0.793542 | 0.970334 | 0.122066 | 0.807613 | 0.52 | 0.881527 | 0.197719 |
| KNN\_Normal | Robust | SMOTE | 0.910959 | 0.959333 | 0.174603 | 0.946502 | 0.22 | 0.952874 | 0.19469 |
| SVM\_NORMAL | Standard | Random | 0.761252 | 0.972727 | 0.115079 | 0.770576 | 0.58 | 0.859931 | 0.192053 |
| SVM\_GridSearched | Robust | Random | 0.825832 | 0.965962 | 0.123529 | 0.846708 | 0.42 | 0.902412 | 0.190909 |
| SVM\_GridSearched | Standard | Random | 0.819961 | 0.964623 | 0.114943 | 0.841564 | 0.4 | 0.898901 | 0.178571 |
| KNN\_Normal | MinMax | SMOTE | 0.927593 | 0.95723 | 0.2 | 0.967078 | 0.16 | 0.962129 | 0.177778 |
| KNN\_Normal | MinMax | Random | 0.863992 | 0.96124 | 0.12605 | 0.893004 | 0.3 | 0.925867 | 0.177515 |
| DT\_GridSearched | Robust | Random | 0.925636 | 0.957143 | 0.190476 | 0.965021 | 0.16 | 0.961066 | 0.173913 |
| KNN\_Normal | Robust | Random | 0.865949 | 0.960309 | 0.121739 | 0.896091 | 0.28 | 0.927089 | 0.169697 |
| DT\_Normal | Standard | SMOTE | 0.90411 | 0.958071 | 0.147059 | 0.940329 | 0.2 | 0.949117 | 0.169492 |
| DT\_Normal | MinMax | SMOTE | 0.911937 | 0.957469 | 0.155172 | 0.949588 | 0.18 | 0.953512 | 0.166667 |
| DT\_GridSearched | MinMax | Random | 0.911937 | 0.957469 | 0.155172 | 0.949588 | 0.18 | 0.953512 | 0.166667 |
| KNN\_GridSearched | Robust | Random | 0.902153 | 0.957983 | 0.142857 | 0.938272 | 0.2 | 0.948025 | 0.166667 |
| DT\_Normal | MinMax | ADASYN | 0.899217 | 0.95785 | 0.136986 | 0.935185 | 0.2 | 0.946382 | 0.162602 |
| DT\_Normal | Standard | ADASYN | 0.908023 | 0.957292 | 0.145161 | 0.945473 | 0.18 | 0.951346 | 0.160714 |
| KNN\_Normal | Standard | SMOTE | 0.91683 | 0.956746 | 0.156863 | 0.955761 | 0.16 | 0.956253 | 0.158416 |
| KNN\_GridSearched | Standard | Random | 0.895303 | 0.957672 | 0.12987 | 0.93107 | 0.2 | 0.944184 | 0.15748 |
| KNN\_Normal | MinMax | ADASYN | 0.924658 | 0.956167 | 0.170732 | 0.965021 | 0.14 | 0.960573 | 0.153846 |
| DT\_GridSearched | Standard | SMOTE | 0.912916 | 0.956567 | 0.145455 | 0.951646 | 0.16 | 0.9541 | 0.152381 |
| KNN\_Normal | Standard | ADASYN | 0.912916 | 0.956567 | 0.145455 | 0.951646 | 0.16 | 0.9541 | 0.152381 |
| SVM\_NORMAL | Robust | SMOTE | 0.934442 | 0.95569 | 0.206897 | 0.976337 | 0.12 | 0.965903 | 0.151899 |
| KNN\_GridSearched | MinMax | Random | 0.89726 | 0.956797 | 0.123288 | 0.934156 | 0.18 | 0.945341 | 0.146341 |
| DT\_GridSearched | MinMax | ADASYN | 0.917808 | 0.955852 | 0.145833 | 0.957819 | 0.14 | 0.956835 | 0.142857 |
| KNN\_Normal | Standard | Random | 0.856164 | 0.957825 | 0.099174 | 0.88786 | 0.24 | 0.921516 | 0.140351 |
| DT\_GridSearched | Robust | ADASYN | 0.915851 | 0.955761 | 0.14 | 0.955761 | 0.14 | 0.955761 | 0.14 |
| DT\_GridSearched | Robust | SMOTE | 0.913894 | 0.95567 | 0.134615 | 0.953704 | 0.14 | 0.954686 | 0.137255 |
| SVM\_NORMAL | Robust | ADASYN | 0.938356 | 0.954955 | 0.217391 | 0.981481 | 0.1 | 0.968037 | 0.136986 |
| GaussianNB | Robust | SMOTE | 0.419765 | 0.984655 | 0.069731 | 0.396091 | 0.88 | 0.56493 | 0.129222 |
| KNN\_GridSearched | Standard | ADASYN | 0.907045 | 0.955348 | 0.118644 | 0.946502 | 0.14 | 0.950904 | 0.12844 |
| GaussianNB | MinMax | SMOTE | 0.406067 | 0.984085 | 0.068217 | 0.381687 | 0.88 | 0.550037 | 0.126619 |
| GaussianNB | Robust | ADASYN | 0.403131 | 0.983957 | 0.067901 | 0.378601 | 0.88 | 0.546805 | 0.126074 |
| GaussianNB | MinMax | ADASYN | 0.402153 | 0.983914 | 0.067797 | 0.377572 | 0.88 | 0.545725 | 0.125894 |
| DT\_Normal | Standard | Random | 0.903131 | 0.955162 | 0.111111 | 0.942387 | 0.14 | 0.948731 | 0.123894 |
| GaussianNB | Robust | Random | 0.309198 | 0.996269 | 0.064987 | 0.274691 | 0.98 | 0.430645 | 0.121891 |
| GaussianNB | MinMax | Random | 0.278865 | 0.995781 | 0.06242 | 0.242798 | 0.98 | 0.390405 | 0.117365 |
| GaussianNB | Standard | SMOTE | 0.350294 | 0.98125 | 0.062678 | 0.323045 | 0.88 | 0.486068 | 0.117021 |
| GaussianNB | Standard | ADASYN | 0.341487 | 0.980707 | 0.061885 | 0.313786 | 0.88 | 0.475448 | 0.115637 |
| KNN\_GridSearched | MinMax | SMOTE | 0.922701 | 0.954222 | 0.128205 | 0.965021 | 0.1 | 0.959591 | 0.11236 |
| KNN\_GridSearched | MinMax | ADASYN | 0.921722 | 0.954175 | 0.125 | 0.963992 | 0.1 | 0.959058 | 0.111111 |
| DT\_GridSearched | Standard | ADASYN | 0.920744 | 0.954128 | 0.121951 | 0.962963 | 0.1 | 0.958525 | 0.10989 |
| GaussianNB | Standard | Random | 0.221135 | 0.994382 | 0.058057 | 0.182099 | 0.98 | 0.307826 | 0.10962 |
| SVM\_NORMAL | Standard | ADASYN | 0.936399 | 0.953954 | 0.173913 | 0.980453 | 0.08 | 0.967022 | 0.109589 |
| DT\_Normal | Robust | Random | 0.915851 | 0.953893 | 0.108696 | 0.957819 | 0.1 | 0.955852 | 0.104167 |
| DT\_Normal | MinMax | Random | 0.914873 | 0.953846 | 0.106383 | 0.95679 | 0.1 | 0.955316 | 0.103093 |
| SVM\_NORMAL | Standard | SMOTE | 0.931507 | 0.953722 | 0.142857 | 0.975309 | 0.08 | 0.964395 | 0.102564 |
| KNN\_GridSearched | Standard | SMOTE | 0.912916 | 0.953751 | 0.102041 | 0.954733 | 0.1 | 0.954242 | 0.10101 |
| RF\_GridSearched | Robust | Random | 0.946184 | 0.953511 | 0.272727 | 0.99177 | 0.06 | 0.972264 | 0.098361 |
| SVM\_NORMAL | MinMax | SMOTE | 0.940313 | 0.953234 | 0.176471 | 0.985597 | 0.06 | 0.969145 | 0.089552 |
| RF\_Normal | Standard | SMOTE | 0.940313 | 0.953234 | 0.176471 | 0.985597 | 0.06 | 0.969145 | 0.089552 |
| DT\_GridSearched | MinMax | SMOTE | 0.908023 | 0.952577 | 0.076923 | 0.950617 | 0.08 | 0.951596 | 0.078431 |
| RF\_Normal | Robust | Random | 0.945205 | 0.952569 | 0.2 | 0.99177 | 0.04 | 0.971774 | 0.066667 |
| RF\_Normal | MinMax | Random | 0.944227 | 0.952522 | 0.181818 | 0.990741 | 0.04 | 0.971256 | 0.065574 |
| RF\_Normal | MinMax | ADASYN | 0.943249 | 0.952475 | 0.166667 | 0.989712 | 0.04 | 0.970737 | 0.064516 |
| RF\_GridSearched | MinMax | Random | 0.943249 | 0.952475 | 0.166667 | 0.989712 | 0.04 | 0.970737 | 0.064516 |
| RF\_GridSearched | Standard | ADASYN | 0.94227 | 0.952428 | 0.153846 | 0.988683 | 0.04 | 0.970217 | 0.063492 |
| RF\_GridSearched | Standard | Random | 0.94227 | 0.952428 | 0.153846 | 0.988683 | 0.04 | 0.970217 | 0.063492 |
| RF\_Normal | Standard | Random | 0.941292 | 0.952381 | 0.142857 | 0.987654 | 0.04 | 0.969697 | 0.0625 |
| RF\_Normal | Robust | ADASYN | 0.938356 | 0.952239 | 0.117647 | 0.984568 | 0.04 | 0.968134 | 0.059701 |
| RF\_GridSearched | MinMax | SMOTE | 0.937378 | 0.952191 | 0.111111 | 0.983539 | 0.04 | 0.967611 | 0.058824 |
| RF\_GridSearched | Robust | SMOTE | 0.94227 | 0.951533 | 0.090909 | 0.989712 | 0.02 | 0.970247 | 0.032787 |
| RF\_GridSearched | MinMax | ADASYN | 0.941292 | 0.951485 | 0.083333 | 0.988683 | 0.02 | 0.969728 | 0.032258 |
| RF\_GridSearched | Robust | ADASYN | 0.941292 | 0.951485 | 0.083333 | 0.988683 | 0.02 | 0.969728 | 0.032258 |
| RF\_GridSearched | Standard | SMOTE | 0.941292 | 0.951485 | 0.083333 | 0.988683 | 0.02 | 0.969728 | 0.032258 |
| RF\_Normal | Standard | ADASYN | 0.941292 | 0.951485 | 0.083333 | 0.988683 | 0.02 | 0.969728 | 0.032258 |
| RF\_Normal | Robust | SMOTE | 0.940313 | 0.951437 | 0.076923 | 0.987654 | 0.02 | 0.969207 | 0.031746 |
| RF\_Normal | MinMax | SMOTE | 0.938356 | 0.951341 | 0.066667 | 0.985597 | 0.02 | 0.968166 | 0.030769 |
| SVM\_NORMAL | MinMax | ADASYN | 0.938356 | 0.951341 | 0.066667 | 0.985597 | 0.02 | 0.968166 | 0.030769 |
| SVM\_GridSearched | MinMax | SMOTE | 0.951076 | 0.951076 | 0 | 1 | 0 | 0.974925 | 0 |
| SVM\_GridSearched | MinMax | ADASYN | 0.951076 | 0.951076 | 0 | 1 | 0 | 0.974925 | 0 |
| SVM\_GridSearched | Robust | SMOTE | 0.951076 | 0.951076 | 0 | 1 | 0 | 0.974925 | 0 |
| SVM\_GridSearched | Robust | ADASYN | 0.951076 | 0.951076 | 0 | 1 | 0 | 0.974925 | 0 |
| SVM\_GridSearched | Standard | SMOTE | 0.951076 | 0.951076 | 0 | 1 | 0 | 0.974925 | 0 |
| SVM\_GridSearched | Standard | ADASYN | 0.951076 | 0.951076 | 0 | 1 | 0 | 0.974925 | 0 |

The following table depicts the results of each of the DL combination on the test data:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Scaler** | **Oversampler** | **Accuracy** | **Precision\_0** | **Precision\_1** | **Recall\_0** | **Recall\_1** | **F1\_0** | **F1\_1** |
| 1DCNN | Robust | SMOTE | 0.898239 | 0.965665 | 0.2 | 0.925926 | 0.36 | 0.945378 | 0.257143 |
| 1DCNN | MinMax | SMOTE | 0.891389 | 0.966414 | 0.191919 | 0.917695 | 0.38 | 0.941425 | 0.255034 |
| 1DCNN | Robust | ADASYN | 0.912916 | 0.962304 | 0.208955 | 0.945473 | 0.28 | 0.953814 | 0.239316 |
| 1DCNN | MinMax | ADASYN | 0.908023 | 0.962105 | 0.194444 | 0.940329 | 0.28 | 0.951093 | 0.229508 |
| 1DCNN | Robust | Random | 0.795499 | 0.97628 | 0.140271 | 0.804527 | 0.62 | 0.882121 | 0.228782 |
| GRU | Robust | SMOTE | 0.906067 | 0.962025 | 0.189189 | 0.938272 | 0.28 | 0.95 | 0.225806 |
| 1DCNN | MinMax | Random | 0.774951 | 0.978093 | 0.134146 | 0.780864 | 0.66 | 0.868421 | 0.222973 |
| ANN | Robust | Random | 0.793542 | 0.975031 | 0.135747 | 0.803498 | 0.6 | 0.880993 | 0.221402 |
| ANN | MinMax | SMOTE | 0.934442 | 0.95846 | 0.257143 | 0.973251 | 0.18 | 0.965799 | 0.211765 |
| GRU | Standard | Random | 0.722114 | 0.983146 | 0.122581 | 0.720165 | 0.76 | 0.831354 | 0.211111 |
| LSTM | MinMax | Random | 0.706458 | 0.985549 | 0.121212 | 0.701646 | 0.8 | 0.819712 | 0.210526 |
| GRU | Standard | SMOTE | 0.933464 | 0.958418 | 0.25 | 0.972222 | 0.18 | 0.965271 | 0.209302 |
| ANN | Standard | Random | 0.81409 | 0.969952 | 0.131579 | 0.830247 | 0.5 | 0.894678 | 0.208333 |
| 1DCNN | Standard | SMOTE | 0.917808 | 0.959627 | 0.196429 | 0.953704 | 0.22 | 0.956656 | 0.207547 |
| 1DCNN | Standard | Random | 0.816047 | 0.9689 | 0.129032 | 0.833333 | 0.48 | 0.896018 | 0.20339 |
| BiRNN | Robust | Random | 0.661448 | 0.987539 | 0.110526 | 0.652263 | 0.84 | 0.785626 | 0.195349 |
| ANN | MinMax | Random | 0.755382 | 0.973753 | 0.115385 | 0.763374 | 0.6 | 0.855825 | 0.193548 |
| ANN | MinMax | ADASYN | 0.932485 | 0.957447 | 0.228571 | 0.972222 | 0.16 | 0.964778 | 0.188235 |
| GRU | MinMax | Random | 0.650685 | 0.985782 | 0.105398 | 0.641975 | 0.82 | 0.77757 | 0.186788 |
| LSTM | Robust | Random | 0.64775 | 0.985714 | 0.104592 | 0.638889 | 0.82 | 0.775281 | 0.18552 |
| BiRNN | Standard | Random | 0.664384 | 0.983103 | 0.105121 | 0.658436 | 0.78 | 0.788663 | 0.185273 |
| GRU | Robust | Random | 0.645793 | 0.985669 | 0.104061 | 0.636831 | 0.82 | 0.77375 | 0.184685 |
| ANN | Standard | ADASYN | 0.92955 | 0.957317 | 0.210526 | 0.969136 | 0.16 | 0.96319 | 0.181818 |
| 1DCNN | Standard | ADASYN | 0.907045 | 0.958203 | 0.153846 | 0.943416 | 0.2 | 0.950752 | 0.173913 |
| LSTM | Robust | SMOTE | 0.932485 | 0.956522 | 0.212121 | 0.973251 | 0.14 | 0.964814 | 0.168675 |
| ANN | Robust | SMOTE | 0.930528 | 0.956434 | 0.2 | 0.971193 | 0.14 | 0.963757 | 0.164706 |
| BiRNN | MinMax | Random | 0.57045 | 0.985428 | 0.088795 | 0.556584 | 0.84 | 0.711374 | 0.160612 |
| LSTM | Standard | Random | 0.562622 | 0.985213 | 0.087318 | 0.548354 | 0.84 | 0.70456 | 0.158192 |
| BiRNN | MinMax | SMOTE | 0.893346 | 0.957582 | 0.126582 | 0.929012 | 0.2 | 0.943081 | 0.155039 |
| LSTM | Robust | ADASYN | 0.899217 | 0.956887 | 0.126761 | 0.936214 | 0.18 | 0.946438 | 0.14876 |
| ANN | Robust | ADASYN | 0.928571 | 0.95542 | 0.171429 | 0.970165 | 0.12 | 0.962736 | 0.141176 |
| BiRNN | Robust | ADASYN | 0.862035 | 0.957096 | 0.097345 | 0.895062 | 0.22 | 0.92504 | 0.134969 |
| ANN | Standard | SMOTE | 0.930528 | 0.954591 | 0.16129 | 0.973251 | 0.1 | 0.963831 | 0.123457 |
| GRU | Standard | ADASYN | 0.937378 | 0.954 | 0.181818 | 0.981481 | 0.08 | 0.967546 | 0.111111 |
| BiRNN | Robust | SMOTE | 0.949119 | 0.953649 | 0.375 | 0.994856 | 0.06 | 0.973817 | 0.103448 |
| BiRNN | Standard | SMOTE | 0.946184 | 0.952616 | 0.222222 | 0.992798 | 0.04 | 0.972292 | 0.067797 |
| GRU | Robust | ADASYN | 0.937378 | 0.952191 | 0.111111 | 0.983539 | 0.04 | 0.967611 | 0.058824 |
| LSTM | Standard | SMOTE | 0.934442 | 0.952048 | 0.095238 | 0.980453 | 0.04 | 0.966042 | 0.056338 |
| LSTM | MinMax | SMOTE | 0.932485 | 0.951952 | 0.086957 | 0.978395 | 0.04 | 0.964992 | 0.054795 |
| BiRNN | Standard | ADASYN | 0.952055 | 0.952008 | 1 | 1 | 0.02 | 0.975414 | 0.039216 |
| GRU | MinMax | SMOTE | 0.939335 | 0.951389 | 0.071429 | 0.986626 | 0.02 | 0.968687 | 0.03125 |
| LSTM | Standard | ADASYN | 0.938356 | 0.951341 | 0.066667 | 0.985597 | 0.02 | 0.968166 | 0.030769 |
| BiRNN | MinMax | ADASYN | 0.951076 | 0.951076 | 0 | 1 | 0 | 0.974925 | 0 |
| LSTM | MinMax | ADASYN | 0.935421 | 0.950298 | 0 | 0.983539 | 0 | 0.966633 | 0 |
| GRU | MinMax | ADASYN | 0.949119 | 0.95098 | 0 | 0.997942 | 0 | 0.973896 | 0 |