International Journal on Engineering Artificial Intelligence Management, Decision Support, and Policies
Vol. 01, Iss. 01, S. No. 004, pp. 29-40, July 2024
ISSN(E): 3048-8788

Enhancing Diabetes Mellitus Prediction: Integrating Hybrid Deep Learning Model with Sampling Techniques

Shivanya Shomir Dutta[1], Aakash Kumar[1], Amutha S[2*] and R Dhanush[3]

[1,2]School of Computer Science and Engineering (SCOPE), Vellore Institute of Technology, Chennai-600127, India.
[3]School of Electronics Engineering (SENSE), Vellore Institute of Technology, Chennai-600127, India

Email address of corresponding author: amutha.s@vit.ac.in

How to cite this paper: Shivanya Shomir Dutta, Aakash Kumar, Amutha S and R Dhansuh, "**Enhancing Diabetes Mellitus Prediction: Integrating Hybrid Deep Learning Model with Sampling Techniques**", *International Journal on Engineering Artificial Intelligence Management, Decision Support, and Policies* , Vol. no. 1, Iss. No 1, S No. 004, pp. 29-40, July 2024.

Received: 07/07/2024
Revised: 15/07/2024
Accepted: 23/07/2024
Published: 31/07/2024

Abstract

Diabetes, characterized by high blood glucose levels, is a leading cause of liver, eye, kidney, and heart diseases. This study evaluates various deep learning models, combined with machine learning classifiers, for predicting diabetes mellitus using the BRFSS dataset. The dataset's imbalance posed a challenge for binary classification, common in medical diagnostics. To address this, different sampling techniques were tested. Hybrid models combining Convolutional Long Short Term Memory (Conv LSTM) networks with traditional classifiers were also explored. The Conv LSTM model combined with Adaboost classifiers achieved the highest accuracy of 89.47% with SMOTE-ENN resampled data. These findings highlight the potential of integrating deep learning and traditional machine learning for effective diabetes prediction, aiding early diagnosis and intervention

Keywords

SMOTE-ENN, Hybrid Deep learning model, BRFSS, Diabetes, Convolutional LSTM.

1. Introduction

Diabetes mellitus is a kind of chronic metabolic disorder characterized by high blood sugar levels and has become one of the significant health issues in the world. The global burden of diabetes was projected at about 9.3% for the 2019 adult population3, totalling around 463 million individuals with diabetes. Projections indicate that the number of people living with the disease may increase to 700 million in 2045. Another report indicates that approximately 422 million people across the world live with diabetes. Most of the people live in low- and middle-income countries. Additionally, diabetes causes approximately 1.5 million deaths annually.

In response to these challenges, researchers and healthcare professionals are turning to advanced computational techniques, particularly deep learning, to enhance our understanding of diabetes and improve patient outcomes. Deep learning (DL), a subset of machine learning, involves training artificial neural networks (ANNs) to learn from vast amounts of data and make predictions or decisions without explicit programming.

DL techniques include a variety of architectures, including recurrent neural networks (RNNs), convolutional neural networks (CNNs), and other neural network variants. These models are designed to capture complex patterns and relationships within datasets, enabling researchers to uncover valuable insights that may inform diabetes prevention, diagnosis, and treatment.

By leveraging these deep learning techniques, researchers aim to develop predictive models capable of identifying individuals at heightened risk of developing diabetes at an early stage. These models have the potential to facilitate proactive interventions, personalized healthcare strategies, and targeted interventions tailored to individual patient needs. Ultimately, the integration of deep learning into diabetes research and clinical practice holds promise for improving outcomes and decreasing diabetes burden on the individual and society.

Despite the progress made with deep learning models, challenges remain in accurately predicting diabetes due to the imbalanced nature of medical datasets and the complex interplay of various risk factors. Traditional models often struggle with these imbalances, leading to biased predictions. This is where hybrid models, which combine multiple learning algorithms, and advanced sampling techniques, such as SMOTE or ADASYN, become crucial. These approaches help to mitigate data imbalance and enhance model robustness, ensuring more accurate and reliable predictions, thereby addressing a critical gap in current diabetes prediction research

2. Related Works

In recent years, diabetes mellitus has emerged as a significant global health challenge, prompting extensive research into effective methodologies for its detection and prediction. Characterized by chronic hyperglycemia, diabetes is associated with severe complications such as cardiovascular diseases, neuropathy, and retinopathy. The World Health Organization estimates that over 422 million people worldwide are living with diabetes, with the prevalence expected to rise dramatically in the coming decades. In response, researchers have increasingly turned to advanced computational techniques to develop predictive models that could facilitate early diagnosis and intervention.

One early effort to address diabetes prediction was the work of Lukmanto and Irwansyah [1], who proposed a pioneering approach using a fuzzy hierarchical model. Their model utilized fuzzy logic to segment variables into input, temporary, and output categories, creating a structured framework for decision-making. By leveraging rules based on critical risk factors such as age and symptoms, their model achieved an accuracy of 87.46%, closely aligning with medical diagnoses. The strength of this model lies in its interpretability and structured approach, which makes it relatively easy to integrate into clinical settings. However, its reliance on a rule-based system introduces limitations, particularly in its ability to capture complex, non-linear relationships between variables. This constraint highlights a significant gap in the need for more flexible models capable of learning directly from data without being confined by predefined rules.

The application of deep learning to diabetes prediction has gained considerable attention due to its capacity to process large and complex datasets. Swapna et al. [2] introduced a classification system that combined support vector machines (SVM) with deep learning layers, focusing on feature extraction from electrocardiogram (ECG) signals. This approach demonstrated high accuracy, showcasing the potential of hybrid models that integrate traditional machine learning with deep learning techniques. However, the model's reliance on ECG signals, which require specialized equipment, limits its applicability to broader populations. Moreover, while the model achieved impressive

accuracy, it did not address the prevalent issue of data imbalance—a critical factor in the performance of models trained on medical datasets. This oversight suggests a need for methodologies that can effectively manage imbalanced data, ensuring that predictive models remain robust and reliable. Authors in [3] conducted a comparative analysis of various machine learning and deep learning algorithms for early-stage diabetes prediction. Their study offered valuable insights into the relative performance of different algorithms, shedding light on their strengths and weaknesses. However, the primary focus on comparing algorithms rather than proposing a novel solution indicates a gap in addressing the specific challenges posed by diabetes prediction, particularly concerning data imbalance and the integration of diverse data types.

Ensemble learning has also been explored as a means to enhance prediction accuracy in diabetes research. Authors in [4] proposed a GA-stacking ensemble learning model, which demonstrated improved performance by combining the strengths of multiple algorithms. Similarly, Prajapati et al. [5] explored ensemble classifier techniques, emphasizing their potential for diabetes detection and prediction. While these studies highlight the advantages of ensemble methods in enhancing model accuracy, they also introduce challenges related to increased computational complexity and reduced interpretability. These limitations can pose barriers to clinical adoption, where transparency and ease of understanding are crucial. Ayon and Islam [6] focused on deep learning methodologies, presenting a robust approach for diabetes prediction through a deep learning framework. Kopitar et al. [7] tackled the early detection of type 2 diabetes mellitus using machine learning-based prediction models, with particular emphasis on random forest algorithms. Both studies demonstrated high accuracy and robustness, but they also underscored the common challenge of dealing with imbalanced datasets. Without proper handling, data imbalance can lead to biased predictions, favoring the majority class and undermining the reliability of the model's predictions.

Additionally, Mahajan et al. [8] and Simaiya et al. [9] contributed to the discourse by exploring supervised machine learning techniques and novel multistage ensemble approaches, respectively. Their research expanded the range of available tools for diabetes prediction, but both studies encountered challenges related to data imbalance and the need for more sophisticated sampling techniques. Addressing these issues is crucial to developing models that are not only accurate but also fair and generalizable across diverse populations. While these studies have collectively advanced the field of diabetes prediction, they also reveal significant gaps that necessitate further exploration. The limitations of existing methods, such as challenges with data imbalance, the need for more flexible and comprehensive models, and the trade-offs between model complexity and interpretability, underscore the importance of continued research. This study aims to address these critical gaps by proposing a novel hybrid approach that integrates advanced sampling methods with state-of-the-art machine learning techniques. This approach seeks to enhance model robustness, mitigate data imbalance, and improve the generalizability of diabetes prediction models, ultimately contributing to more accurate and clinically applicable predictions for diabetes detection and management.

3. Methodology

3.1. Dataset Description

The dataset sourced from Kaggle [10] includes responses from 253,680 individuals from the CDC's BRFSS 2015 survey**.** The dataset includes the following variables: Diabetes_012 (Diabetes status), HighBP (High blood pressure), HighChol (High cholesterol), CholCheck (Cholesterol check), BMI (Body Mass Index), Smoker (Smoking status), Stroke (Stroke occurrence), HeartDiseaseorAttack (Heart disease or heart attack), PhysActivity (Physical activity), Fruits (Fruit consumption), Veggies (Vegetable consumption), HvyAlcoholConsump (Heavy alcohol consumption), AnyHealthcare (Healthcare coverage), NoDocbcCost (No doctor because of cost), GenHlth (General health status), MentHlth (Mental health status), PhysHlth (Physical health status), DiffWalk (Difficulty walking), Sex (Gender), Age (Age), Education (Education level), and Income (Income level). This dataset aids in examining health behaviors and conditions, informing public health strategies.

3.2. Dataset preprocessing

3.2.1 Balancing the dataset

The dataset used was imbalanced, hence several resampling methods were used. The resampling methods are described in detail below:

SMOTE-ENN:
The SMOTE-ENN [11]method was chosen due to its ability to balance datasets while simultaneously reducing noise. SMOTE (Synthetic Minority Oversampling Technique) generates synthetic instances for the minority class to address class imbalance. However, oversampling can introduce noise by generating overlapping or redundant instances. ENN (Edited Nearest Neighbors) counters this by removing noisy or misclassified samples from the majority class. This combination helps in improving model performance by creating a more refined dataset. SMOTE-ENN was applied by first generating synthetic instances for the minority class and then applying ENN to remove potentially noisy samples from the dataset. This method was implemented using the imblearn library in Python.

SMOTE:
SMOTE was selected because it is a widely used and straightforward technique to address class imbalance by synthetically generating instances for the minority class. This helps in reducing the bias towards the majority class, leading to more balanced model training. The SMOTE algorithm was applied by generating synthetic samples for the minority class until the desired balance between classes was achieved. The imblearn library was used to implement SMOTE in this study.

SMOTE-Tomek:
SMOTE-Tomek [12] was chosen because it not only balances the dataset by generating synthetic samples but also refines the data by removing Tomek links, which are pairs of instances from opposite classes that are close to each other. This method helps in cleaning the dataset by eliminating borderline instances, which may contribute to classification errors. SMOTE was first applied to generate synthetic instances, followed by the application of Tomek links to remove borderline examples. The implementation was carried out using the imblearn library.

ADASYN:
ADASYN (Adaptive Synthetic Sampling) [13] was chosen due to its adaptive nature, where more synthetic data is generated for minority class instances that are harder to classify. This targeted approach ensures that the model focuses more on challenging examples, potentially improving classification performance. ADASYN was applied by generating synthetic instances for the minority class, with more emphasis on harder-to-classify examples, based on their distribution in the feature space. The implementation was done using the imblearn library.

Random Sampling:
Random undersampling [14] was chosen as a baseline method to reduce the size of the majority class, ensuring a balanced dataset. This method is straightforward but can be effective in preventing the model from becoming biased towards the majority class. The majority class was randomly reduced by removing instances until the dataset achieved the desired balance. This technique was implemented using basic Python data manipulation libraries.

3.2.2. Normalizing the dataset

For normalizing the dataset Standard Scaler was used before feeding the data top the models. Standard Scaler is a preprocessing technique in machine learning used to standardize features by removing the mean and scaling to unit variance. It transforms data to have a mean of 0 and a standard deviation of 1, ensuring all features have the same scale, which can improve the performance of certain algorithms.

3.3. Deep learning models

3.3.1 LSTM

The LSTM (Long Short-Term Memory) is a specialized variant of RNNs designed to process sequential data and avoid the vanishing gradient problem in conventional RNNs. The structure can be seen to be comprised of a cell alongside an input gate, output gate, and a forget gate. The cell stores information for a long period, and the gates control its flow. What is insignificant in the previous state of information is determined by the forget gate, which represents the information available for the next input is represented by the input gate, and the degree to which the current state is made available is represented by the output gate. These functions are controlled by computational equations (1-6).

$$i_t = \sigma(x_t U^i + h_{t-1} W^i) \quad \text{-(1)}$$

$$f_t = \sigma(x_t U^f + h_{t-1} W^f) \quad \text{-(2)}$$

$$o_t = \sigma(x_t U^o + h_{t-1} W^o) \quad \text{-(3)}$$

$$C = tanh(x_t U^g + h_{t-1} W^g) \quad \text{-(4)}$$

$$C_t = \sigma(f_t * C_{t-1} + i_t * C_t\tilde{}) \quad \text{-(5)}$$

$$h_t = tanh(C_t) * o_t t \quad \text{-(6)}$$

3.3.2 Convolutional LSTM

ConvLSTM [15] is a special form of RNN that integrates convolutional layers in the input-to-state and state-to-state transitions; it does very well in spatiotemporal forecasting. ConvLSTM predicts the future state of one cell from the grid by considering the inputs and previous states of the neighbouring cells. This is facilitated through the use of convolution operators in transitioning between states. The core equations of ConvLSTM involve convolutions and element-wise multiplication, enabling it to capture complex spatio-temporal dependencies effectively.
The modified functional aspects of ConvLSTM are governed by computational equations (7-11).

$$i_t = \sigma(W_{ii} \cdot X_t + W_{hi} \cdot H_{t-1} + W_{ci} \circ C_{t-1} + b_i) \quad \text{- (7)}$$

$$f_t = \sigma(W_{xf} \cdot X_t + W_{hf} \cdot H_{t-1} + W_{cf} \circ C_{t-1} + b_f) \quad \text{- (8)}$$

$$C_t = f_t \circ C_{t-1} + i_t \circ tanh(W_{xc} \cdot X_t + W_{hc} \cdot H_{t-1} + b_c) \quad \text{- (9)}$$

$$o_t = \sigma(W_{xo} \cdot X_t + W_{ho} \cdot H_{t-1} + W_{co} \circ C_t + b_o) \quad \text{- (10)}$$

$$H_t = o_t \odot tanh(C_t) \quad \text{- (11)}$$

In practical terms, ConvLSTM models with larger transitional kernels capture faster motions, while smaller kernels excel at slower motions. Padding is applied before convolution operations to maintain consistency in dimensions between states and inputs, extending hidden states to boundary points and incorporating surrounding information. Typically, all LSTM states are initialized to zero before the first input, establishing a baseline for future state transitions and predictions.

3.3.3 Convolutional LSTM and Machine learning models

By combining ConvLSTM with ML algorithms such as Decision trees, or Naive bayes , the strengths of both approaches can be leveraged. ConvLSTM networks extract hierarchical features from input sequences, which are then fed into ML models for further processing and classification which is shown in Figure.1.
Some more details of the classifiers used in combination Convolutional LSTM are:

1. K-Nearest Neighbors (KNN):
K-Nearest Neighbors (KNN) [16] is a simple and intuitive classification algorithm that operates on the principle of similarity. It assigns a data point to the majority class among its k nearest neighbors in the feature space.
2. Naive Bayes:
Naïve Bayes [17], a straightforward learning algorithm, employs Bayes' rule and assumes strong conditional independence among attributes given the class.

3. XGBoost (Extreme Gradient Boosting):
XGBoost [18] employs second-order Taylor expansion for distributed training and CPU multithreading. XGBoost employs diverse strategies to counter overfitting..

4. LightGBM (Light Gradient Boosting Machine):
LightGBM [19] accelerates training by employing a leaf-wise strategy. This approach constructs regression trees iteratively, using residuals as targets for subsequent trees.

5. AdaBoost (Adaptive Boosting):
The AdaBoost algorithm generates a series of weak learners by assigning, at each round, a set of weights to the training data. It increases the weights of those examples misclassified after every iteration and decreases the weights of those classified correctly.

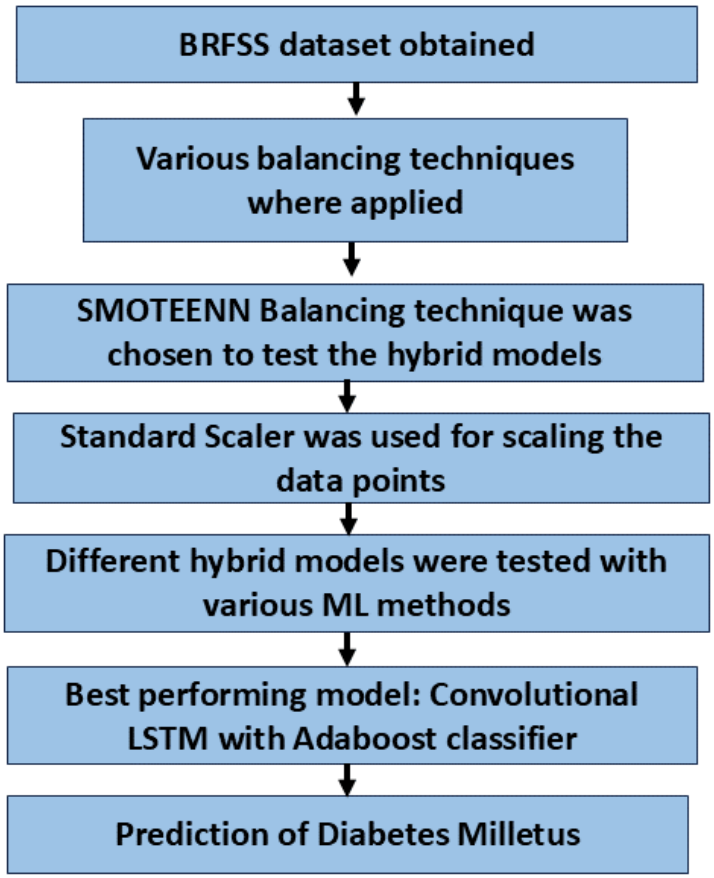

Figure 1: Flowchart depicting methodology

3.4. Proposed model architecture

The data was obtained from the BRFSS dataset, and underwent pre-processing techniques to fill in the NAN values. It's an imbalanced dataset, with the number of healthy (Diabetic) data points outnumbering unhealthy and (non-Diabetic) datapoints by 10 times. To address this issue, dataset sampling technique is used to maintain the balance of both healthy and unhealthy data points of the dataset. In this study, we present a hybrid DL architecture that leverages both CNNs and LSTM networks for feature extraction, followed by various classical machine learning classifiers for final classification. The architecture is specifically designed to handle sequential data, which is first scaled and reshaped to be suitable for the Conv1D and LSTM layers.

Additionally, we resampled the data using several sampling techniques, as mentioned in section 3.2.1. Among these techniques, SMOTEEN (Synthetic Minority Over-sampling Technique and Edited Nearest Neighbours) resampling provided the best performance when the extracted features were finally classified using the machine learning classifiers. It is hugely important that the input features are normalized to have a mean of zero and a standard deviation of one by pre-processing the input data using StandardScaler, as explained in section 3.2.2. The scaled data is then reshaped to fit the input requirements of the Conv1D layer, with each feature being considered a timestep. Reshaping the BRFSS dataset which is a non-sequential data, which comprises 21 columns, into a sequential format for a Conv1D-LSTM model is a strategic approach that can enhance the model's performance. By converting the data to timesteps, the Conv1D layer can effectively capture local dependencies and interactions between the features, which might reveal underlying patterns not immediately apparent in the raw data. This transformation allows the LSTM layer to process the data as a sequence, enabling it to learn and retain long-term dependencies and relationships among features and rows. This approach is particularly beneficial in capturing complex interactions within the dataset, thus leveraging the strengths of the Conv1D-LSTM architecture to provide a more nuanced and comprehensive analysis. Ultimately, this reshaping facilitates the extraction of richer, more informative feature representations, leading to improved predictive accuracy and deeper insights into the dataset.

The model begins with a Conv1D layer that applies 128 filters of size 3 with a sigmoid activation function. This layer is responsible for detecting local patterns in the input sequences. The output of the Conv1D layer is then fed into two stacked LSTM layers, each with 64 units. The first LSTM layer returns sequences, allowing the second LSTM layer to capture more complex temporal dependencies. A dense layer with 32 units follows the LSTM layers to further process the extracted features. A dropout layer with a dropout rate of 0.5 is added to prevent overfitting. Finally, a dense layer with a single unit is used for the final output, tailored for classification tasks. It uses the Adam optimizer with binary cross-entropy as the loss function and accuracy as the metric to be evaluated. The model is trained for 300 epochs with a batch size of 32, and 20 steps per epoch, using 20% of the training data for validation. After training, the LSTM-encoded features are extracted from both the training and testing datasets for subsequent classification tasks. The overall architecture is shown in Figure 2.

To evaluate the effectiveness of the extracted features, we employed several classical machine learning classifiers: Decision Tree, Gaussian Naive Bayes, XGBoost, LightGBM and KNN. These classifiers were trained on the LSTM-encoded features and evaluated on the test data. As mentioned earlier as Conv1D-LSTM trained with SMOTEEN resampled data worked the best with ML classifiers, in Table 1 we have provided the results of only SMOTEEN resampled data used with Conv1D-LSTM and ML classifiers. Also the results of Conv1D-LSTM trained with other sampling techniques are tabulated. Conv1D-LSTM was trained with imbalanced data and also passed through ML classifiers, those results are also depicted in Table 1.

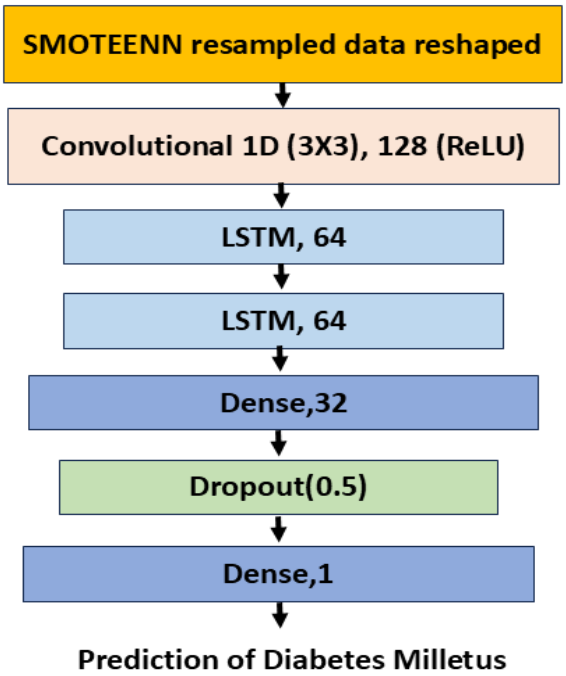

Figure 2: Proposed Model Architecture

4. Results and Discussions

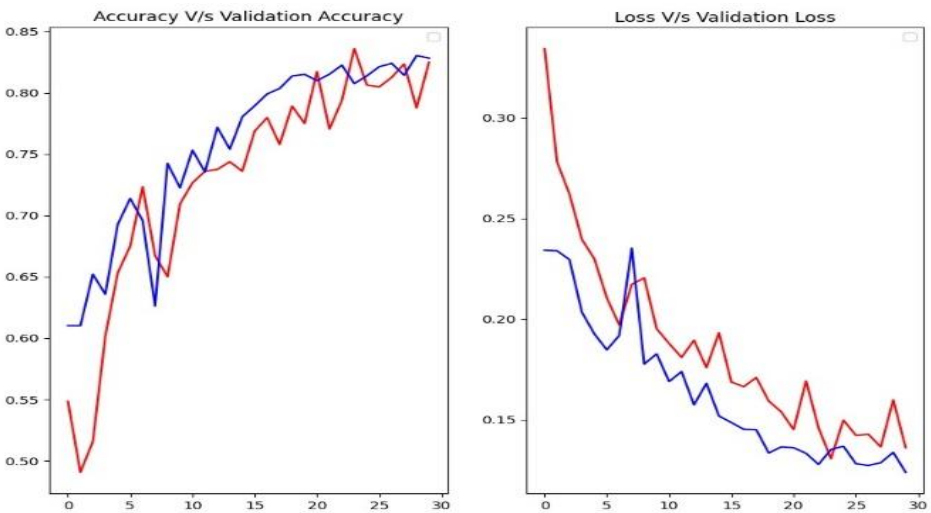

Figure 3 a: Accuracy plot for Conv1D-LSTM and Figure 3 b:Loss plot for Conv1D-LSTM

The training and validation plots for the Conv1D-LSTM model trained on SMOTE-ENN resampled data demonstrate the model's robust performance and effective learning capability. The accuracy plot (Figure 3 a) illustrates a consistent increase in both training and validation accuracy, stabilizing around 80% after 30 epochs, indicating the strong ability of

the model to generalize to unseen data. Concurrently, the loss plot (Figure 3 b) shows a steady decrease in both training and validation loss, highlighting the model's effective minimization of prediction errors. The convergence of training and validation metrics suggests that the model is not overfitting, benefitting significantly from the SMOTE-ENN resampling technique. This balanced approach of addressing class imbalance has evidently contributed to the enhanced performance and stability of the Conv1D-LSTM model in predicting outcomes accurately from sequential diabetic data.

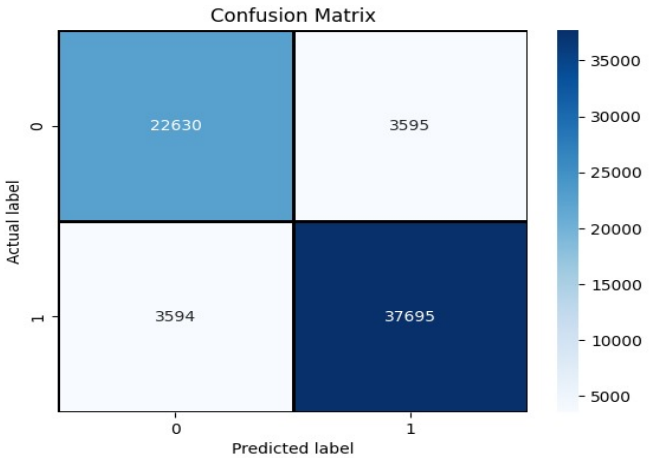

Figure 4: Confusion matrix of the proposed model- Conv1D-LSTM trained with SMOTE-ENN resampled data and Adaboost classifier.

The confusion matrix for the proposed Conv1D-LSTM model (Figure 4), trained with SMOTE-ENN resampled data and classified using AdaBoost, highlights its high performance and accuracy. The model correctly classifies both classes with 22,630 true negatives and 37,695 true positives. It has relatively low false positives (3,595) and false negatives (3,594), demonstrating strong precision and recall. This balance indicates the model's robustness and reliability in handling imbalanced datasets, with SMOTE-ENN resampling and AdaBoost integration enhancing overall performance.

Table 1. Evaluation metrics

Model	Accuracy	F1	Recall	Precision
LSTM	84.7	24	17	40
Conv LSTM	85.88	0	0	0
ConvLSTM+ Naive Beyes	84.9	16.9	10.8	37.9
ConvLSTM+ KNN	83.9	14.2	9.45	28.7
Conv LSTM+DT	78.9	21.6	20.5	22.8
RandomSampling ConvLSTM	73.8	75.9	82.8	70.
SMOTEEN Conv LSTM	82.8	85.9	86.2	85.7
SMOTETomek Conv LSTM	84.2	84.7	87.9	81.7
ADASYN ConvLSTM	82.1	81.1	77.7	84.7
SMOTE Conv LSTM	81.9	81.9	68.2	93.9
SMOTEEN ConvLSTM+ KNN	88.2	90.4	90.8	90
SMOTEEN ConvLSTM+ naivebayes	89.4	91.4	91.9	90.9
SMOTEEN ConvLSTM + XgBoost	89.4	**91.4**	**92.2**	90.6
SMOTEEN ConvLSTM+ LGBM	89.4	91.42	91.9	90.8
SMOTEEN ConvLSTM+ Adaboost	**89.4**	91.42	91.8	**91.0**

Table 1 further shows that hybrid models combining Convolutional LSTM (Conv LSTM) with ML classifiers show varying accuracies. For instance, SMOTEEN Conv LSTM + Adaboost achieve the highest accuracy of 89.47%, suggesting that the

combination of DL and ML techniques enhances predictive performance. Hybrid models combining Conv LSTM with ML classifiers, particularly those utilizing tree based algorithms achieve the highest overall performance in terms of accuracy, F1 score, recall, and precision. This highlights the benefits of integrating DL and ML techniques for improved predictive modeling. Models incorporating under-sampling techniques such as Random Sampling generally exhibit lower performance compared to oversampling techniques and hybrid approaches. This suggests that removing instances from the majority class may lead to loss of valuable information and decreased model performance.

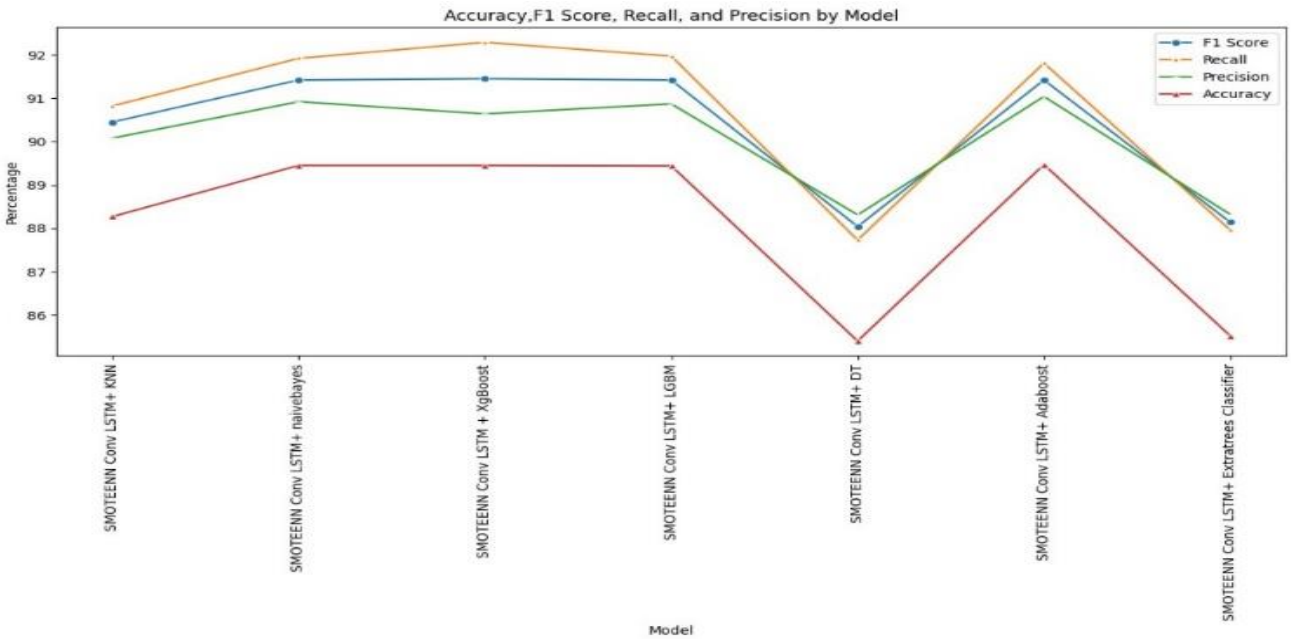

Figure 5: Accuracy, F1 Score, Recall and Precision of Conv1D-LSTM models with ML classifiers trained on SMOTEENN resampled data

Figure 5 illustrates the performance metrics—accuracy, F1 score, recall, and precision—of Conv1D-LSTM models combined with various machine learning classifiers trained on SMOTEENN resampled data. Among the models evaluated, the Conv1D-LSTM combined with Adaboost classifiers stands out, achieving the highest accuracy, recall, and F1 score, indicating its superior predictive capability for diabetes mellitus detection. This model consistently outperforms others in precision as well, demonstrating its robustness and effectiveness in handling the imbalanced dataset. The Conv1D-LSTM with Adaboost exhibits a well-rounded performance, making it the proposed model for accurate and reliable diabetes prediction.

5. Conclusion

In conclusion, this paper underscores the significance of employing a diverse range of deep learning and machine learning techniques for predicting diabetes mellitus, especially when faced with imbalanced datasets. By evaluating various sampling methods and hybrid model approaches, we have demonstrated the effectiveness of integrating Convolutional Long Short-Term Memory (Conv LSTM) networks with traditional machine learning classifiers. Notably, our findings reveal that the hybrid model, particularly the combination of Conv LSTM with Adaboost Classifiers, achieves a remarkable accuracy of 89.47% with SMOTEENN resampled data. These results offer valuable insights into developing robust predictive models for early detection of diabetes mellitus, thereby facilitating proactive healthcare interventions and improved patient outcomes.

Funding: "This research received no external funding"

Conflicts of Interest: "The authors declare no conflict of interest."

Acknowledgements

All authors would like to thank Vellore Institute of Technology, Chennai for the motivation and for providing the resources to complete this paper.

References

1. Lukmanto, R. B., & Irwansyah, E. J. P. C. S. (2015). The early detection of diabetes mellitus (DM) using fuzzy hierarchical model. Procedia Computer Science, 59, 312-319. 5.
2. Swapna, G., Vinayakumar, R., & Soman, K. P. (2018). Diabetes detection using deep learning algorithms. ICT express, 4(4), 243-246.Refat, M. A. R., Al Amin, M., Kaushal, C., Yeasmin, M. N., & Islam, M. K. (2021, October). A comparative analysis of early stage diabetes prediction using machine learning and deep learning approach. In 2021 6th International Conference on Signal Processing, Computing and Control (ISPCC) (pp. 654-659). IEEE.
3. Larabi-Marie-Sainte, S., Aburahmah, L., Almohaini, R., & Saba, T. (2019). Current techniques for diabetes prediction: review and case study. Applied Sciences, 9(21), 4604.
4. Tan, Y., Chen, H., Zhang, J., Tang, R., & Liu, P. (2022). Early risk prediction of diabetes based on GA-Stacking. Applied Sciences, 12(2), 632.
5. Prajapati, Y. R., Hihoriya, D. G., & Verma, S. (2023, July). Early Detection and Prediction of Diabetes Using Ensemble Classifier. In 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT) (pp. 1-6). IEEE.
6. Ayon, S. I., & Islam, M. M. (2019). Diabetes prediction: a deep learning approach. International Journal of Information Engineering and Electronic Business, 13(2), 21.
7. Kopitar, L., Kocbek, P., Cilar, L., Sheikh, A., & Stiglic, G. (2020). Early detection of type 2 diabetes mellitus using machine learning-based prediction models. Scientific reports, 10(1), 11981.
8. Mahajan, S., Sarangi, P. K., Sahoo, A. K., & Rohra, M. (2023, May). Diabetes Mellitus Prediction using Supervised Machine Learning Techniques. In *2023 International Conference on Advancement in Computation & Computer Technologies (InCACCT)* (pp. 587-592). IEEE.
9. Simaiya, S., Kaur, R., Sandhu, J. K., Alsafyani, M., Alroobaea, R., Margala, M., & Chakrabarti, P. (2022). A novel multistage ensemble approach for prediction and classification of diabetes. *Frontiers in Physiology*, *13*, 1085240.
10. Teboul, A. (n.d.). Diabetes Health Indicators Dataset. Retrieved from https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset?select=diabetes_binary_health_indicators_BRFSS2015.csv
11. Lamari, M., Azizi, N., Hammami, N. E., Boukhamla, A., Cheriguene, S., Dendani, N., & Benzebouchi, N. E. (2021). SMOTE–ENN-based data sampling and improved dynamic ensemble selection for imbalanced medical data classification. In *Advances on Smart and Soft Computing: Proceedings of ICACIn 2020* (pp. 37-49). Springer Singapore.
12. He, H., Bai, Y., Garcia, E. A., & Li, S. (2008, June). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)* (pp. 1322-1328). Ieee.
13. Prusa, J., Khoshgoftaar, T. M., Dittman, D. J., & Napolitano, A. (2015, August). Using random undersampling to alleviate class imbalance on tweet sentiment data. In *2015 IEEE international conference on information reuse and integration* (pp. 197-202). IEEE.
14. Shi, X., Chen, Z., Wang, H., Yeung, D. Y., Wong, W. K., & Woo, W. C. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, *28*.
15. Guo, G., Wang, H., Bell, D., Bi, Y., & Greer, K. (2003). KNN model-based approach in classification. In *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings* (pp. 986-996). Springer Berlin Heidelberg.
16. Webb, G. I., Keogh, E., & Miikkulainen, R. (2010). Naïve Bayes. *Encyclopedia of machine learning*, *15*(1), 713-714.
17. Li, W., Yin, Y., Quan, X., & Zhang, H. (2019). Gene expression value prediction based on XGBoost algorithm. *Frontiers in genetics*, *10*, 484931.

18. Ju, Y., Sun, G., Chen, Q., Zhang, M., Zhu, H., & Rehman, M. U. (2019). A model combining convolutional neural network and LightGBM algorithm for ultra-short-term wind power forecasting. *Ieee Access*, *7*, 28309-28318.
19. Wang, R. (2012). AdaBoost for feature selection, classification and its relation with SVM, a review. Physics Procedia, 25, 800-807.