diagnostics

MDPI

Article

Using Recurrent Neural Networks for Predicting Type-2 Diabetes from Genomic and Tabular Data

Parvathaneni Naga Srinivasu [1,*,†], Jana Shafi [2,†], T Balamurali Krishna [3], Canavoy Narahari Sujatha [4], S Phani Praveen [1] and Muhammad Fazal Ijaz [5,*]

[1] Department of Computer Science and Engineering, Prasad V. Potluri Siddhartha Institute of Technology, Vijayawada 520007, Andhra Pradesh, India
[2] Department of Computer Science, College of Arts and Science, Prince Sattam bin Abdul Aziz University, Wadi Ad-Dawasir 11991, Saudi Arabia
[3] Department of Computer Science and Engineering, Dhanekula Institute of Engineering and Technology, Vijayawada 521139, Andhra Pradesh, India
[4] Department of Electronics and Communication Engineering, Sreenidhi Institute of Science and Technology, Hyderabad 501301, Telangana, India
[5] Department of Intelligent Mechatronics Engineering, Sejong University, Seoul 05006, Republic of Korea
* Correspondence: parvathanenins@gmail.com (P.N.S.); fazal@sejong.ac.kr (M.F.I.)
† These authors contributed equally to this work and are the first co-authors.

Abstract: The development of genomic technology for smart diagnosis and therapies for various diseases has lately been the most demanding area for computer-aided diagnostic and treatment research. Exponential breakthroughs in artificial intelligence and machine intelligence technologies could pave the way for identifying challenges afflicting the healthcare industry. Genomics is paving the way for predicting future illnesses, including cancer, Alzheimer's disease, and diabetes. Machine learning advancements have expedited the pace of biomedical informatics research and inspired new branches of computational biology. Furthermore, knowing gene relationships has resulted in developing more accurate models that can effectively detect patterns in vast volumes of data, making classification models important in various domains. Recurrent Neural Network models have a memory that allows them to quickly remember knowledge from previous cycles and process genetic data. The present work focuses on type 2 diabetes prediction using gene sequences derived from genomic DNA fragments through automated feature selection and feature extraction procedures for matching gene patterns with training data. The suggested model was tested using tabular data to predict type 2 diabetes based on several parameters. The performance of neural networks incorporating Recurrent Neural Network (RNN) components, Long Short-Term Memory (LSTM), and Gated Recurrent Units (GRU) was tested in this research. The model's efficiency is assessed using the evaluation metrics such as Sensitivity, Specificity, Accuracy, F1-Score, and Mathews Correlation Coefficient (MCC). The suggested technique predicted future illnesses with fair Accuracy. Furthermore, our research showed that the suggested model could be used in real-world scenarios and that input risk variables from an end-user Android application could be kept and evaluated on a secure remote server.

Keywords: deep learning; PIMA dataset; Type-2 diabetes; Recurrent Neural Networks; weight optimization

check for updates

Citation: Srinivasu, P.N.; Shafi, J.; Krishna, T.B.; Sujatha, C.N.; Praveen, S.P.; Ijaz, M.F. Using Recurrent Neural Networks for Predicting Type-2 Diabetes from Genomic and Tabular Data. *Diagnostics* **2022**, *12*, 3067. https://doi.org/10.3390/diagnostics12123067

Academic Editors: Wan Azani Mustafa and Hiam Alquran

Received: 4 November 2022
Accepted: 4 December 2022
Published: 6 December 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Diabetes is a metabolic disorder influenced by high blood sugar levels due to insufficient insulin release or synthesis. Diabetes was predicted to affect 285 million people globally in 2010. According to the disease's current development pace, this figure will increase to 552 million by 2030. One in every ten people is projected to have diabetes by 2040 [1]. Diabetes is becoming more prevalent because of individual habits, divergent lifestyles, and living standards. Thus, researching how to effectively and promptly identify and treat diabetes is worthwhile. Diabetes is diagnosed based on genomic patterns. It will

result in a more accurate and precise outcome and assist in adhering to better habits that are less likely to result in diabetes shortly. Very effective identification of an illness allows individuals with future illnesses to slow or postpone the disease's development and enjoy better overall health. Machine learning techniques fall into two categories: screening future illnesses and diagnosing an abnormality [2]. Based on current and prior medical conditions, forward prediction techniques can anticipate diabetes before it occurs [3,4].

Type 2 diabetes (T2D), formerly called non-insulin-dependent diabetes) is a category of metabolic disorders distinguished through hyperglycemia, resulting in abnormalities in insulin production or insulin function. Lifestyle behaviors, including food habits, exercise, and dietary choices, may significantly affect its development. T2D is the type of disease known for decreasing the life span and reducing the standard of living. The illness may be controlled with lifestyle modification and pharmaceutical management. Thus, it is essential to have early diagnosis and treatment of T2D to help patients avoid life-threatening consequences. Many research studies have been conducted on medical diagnoses to predict illness and forecast the future with considerable efficiency accurately. Generally, most diseases are triggered by a combination of two or more gene patterns. Recognition of the combinational gene sequence by rigorous analysis of the reference genes of the healthy person with the samples that are trained genes acquired from the diseased. Deoxyribonucleic Acid (DNA) is significant for cell growth and is generally a hereditary component in each cell of an organism. As stated by A. Arshad and Y. D. Khan [5], DNA is coded through chemical bases adenine (A), guanine (G), cytosine (C), and thymine (T) that form up a cell which is unique for almost all human beings. Analysis of DNA molecular compositions of human genes is immensely used to predict illnesses associated with ancestors. The genomics study would assist in changing the lifestyle of an individual, which results in a lower risk of the disease in the future. DNA analysis can assist in the prediction of a disease that is caused by a mutation of the DNA. Biomedical engineering has recognized an enormous gene data set that could help predict various conditions. The Neural Network approach can identify gene patterns that harm cells of a human body with a high possibility of illness-causing patterns through the proposed mechanism. By incorporating Neural Networks, approaches would have high Accuracy for illness prediction with reasonably acceptable computational latency.

The advancement that has taken place in Genome-Wide Association Studies (GWAS) holds tremendous information related to various gene patterns associated with divergent illnesses that are complex and challenging to perform reductive analysis from a single locus, as stated by Cho Ys [6] and Coron [7]. The evolution of GWAS has focused on integrating data related to multi-locus across the gene that would assist in predicting complex illnesses in advance. Polygenic Risk Scores (PRS) were proposed by Duncan L. et al. [8] for risk analysis using the sum of the weight of each risk-associated locus of genomic sequence obtained from the corresponding evidence. These weights are assessed from the regression coefficient associated with each locus. These combined genetics features and correlation matrices would significantly assist the entire field of genomics study [9]. These studies on analyzing the genomic data and the tabular datasets such as PIMA would largely assist in analyzing the future illness had paved the motivation for the current study, and the role of various neural network components in the performance of the deep learning models are evaluated [10,11].

The current study is primarily motivated by the research challenges in handling genomic data and the pattern recognition for precisely identifying future illnesses. The genomic data comprises more extensive DNA sequences, which requires tremendous computational efforts to assess the disease's probability. Earlier assessment of the future illness would assist the individual in safeguarding themselves from such disease through better living standards. Moreover, the current study has also focused on evaluating the performances of various recurrent neural network models such as RNN, GRU, and LSTM in disease prediction. Performances evaluation metrics such as the confusion matrix

for Sensitivity, Specificity, F1-Score, and Mathews Correlation Coefficient measures are considered in the current study.

The recurrent neural network components such as RNN, GRU, and LSTM are efficient in learning from past experiments and can simultaneously process a sequence of inputs and outputs, which is a kind of sequence-to-sequence network that is exceptionally efficient in handling genomic data. The RNN-based neural network may represent a set of records such that each pattern is thought to rely on preceding ones. The Hidden State, which remembers certain information about a sequence, is the core and most essential aspect of RNN. LSTM feeds genetic sequences into a network and makes assumptions based on the sequence data's discrete time steps. It can learn long-term dependencies, which is notably valuable for sequence prediction issues. GRU uses less memory and is comparatively faster than the RNN and LSTM models. But can effectively work with smaller sequences. GRU employs fewer training parameters, requires less memory, executes quicker, and learns faster than LSTM, although LSTM is much more accurate on more extended sequence datasets.

The main contributions of this work are as follows:

- The reference gene sequence is analyzed against the trained genomic data for possible gene pattern matching. As well, the further correlation between the reference gene and gene pattern associated with diabetes is assessed.
- The probabilistic estimations are performed by the softmax layer towards the future illness based on the gene correlation. Additionally, based on the probabilities, the risk factor outcome is yielded.
- The proposed RNN model is evaluated over the tabular patient data such as PIMA for risk analysis, where the auxiliary memory components such as GRU and LSTM are integrated for better prediction performance.
- The feature selection and weight optimizations are performed over the features of the PIMA dataset for better prediction outcomes.
- The outcome of the present study is being evaluated against conventional classification techniques such as Decision Tree, J48, K Nearest Neighborhood, Logistic Regression, Naive Bayes, Random Forest, and Support Vector Machine.

The entire paper is arranged as follows. The paper's first section introduces the proposed approach and the Genomic domain. Section 2 presents the literature review of existing studies focusing on various disease prediction techniques using genomic and tabular data. Section 3 presents the methodology of the proposed model where various aspects such as background work of the domain that highlights data collection, data preprocessing, feature extraction strategies, and RNN with various memory components are presented. Section 4 presents the result and discussion. Section 5 presents the conclusion and future scope of the proposed model.

2. Literature Review

2.1. ML Models for Smart Diagnosis of Type-2 Diabetes

Machine Learning is the most emerging technology for addressing inevitable problems in various domains. Machine Learning through supervised, semi-supervised approaches, or weakly supervised approaches, is used with data from various sources, including medical records and information obtained from wearable gadgets, to forecast an illness. In either of these approaches, sickness cannot be predicted much earlier, and the patient cannot get rid of the illness by changing his or her lifestyle in a short period. The polygenic scores-based approach is among the most predominantly used strategies for the earlier prediction of an illness. The Polygenic Score approach has been tremendously evaluated before it is used in clinical trials. It is also used in illness screening mechanisms, as in the study of So et al. [12]. The current research and the genomic analysis could change lifestyles and reduce illnesses such as heart attack, cardiovascular diseases, cancer, and Alzheimer's disease. The process of polygenic risk score involves two significant phases: discovery and validation. The Discovery Phase identifies risks through a statistical association test using either linear or

Logistic Regression. The later phase validates approximations performed in the earlier stage for extracting information related to Single Nucleotide Polymorphism (SNP).

Deep learning (DL) [13,14] is the field of Machine Learning that is extensively used in predicting type-2 disease by processing the blood glucose level and spectrogram images generated from the blood glucose levels. Moreover, the DL models could also be used with tabular datasets such as PIMA for the prediction of diabetes. Every layer in the DL model reflects a degree of acquired information. The layer closest to the input layer reflects low-level data elements, whereas the layer closest to the output layer shows a higher degree of discrimination with more concise notions. Deep learning generally needs more data for precise classification and also needs tremendous computational resources for processing [15]. The major limitation of the deep learning models is that the decision mechanism is not interpretable, which limits the trustworthiness of the models.

Clustering is one of the most predominantly performed operations with un-supervisory approaches using dimensionality reduction approaches such as Singular Value Decomposition (SVD) [16], Principle Component Analysis (PCA) stated by Konishi T. et al. [17], Apriori stated by S. Mallik et al. [18]. Dynamic thresholding-based FP-Growth was stated by Mallik S. et al. [19] for treating unusual illnesses and certain types of diseases with unknown variations with different symptoms. However, most of these approaches do not label the output data, as the provided input does not have any labels. The Accuracy of the un-supervisory method is a significant concern as classes are not marked. In some instances, the proposed algorithm might end up misinterpretation. A classification-based illness prediction is a supervisory approach that includes various mechanisms such as Linear and Polynomial Regression, Decision Tree, Random Forest, and many other systems, including the Support Vector Machine (SVM) used by Huang S. et al. [20], K-Nearest Neighbour used by Parry R. et al. [21], and Logistic Regression approach that exhibits better efficiency in terms of accuracy and precision other classification models. Supervisory approaches exhibit optimal performance for known cases. The Accuracy of the prediction outcome is directly proportional to the training set size, which needs many computational efforts. However, in some cases, the approach diverges due to excessive training.

Random Forest is a rapid implementation approach using the Ranger package in the R tool described by Wright and Ziegler [22], which is used to predict future illness from tabular data such as PIMA. Artificial Neural Networks based on illness prediction mechanisms, as discussed by Anifat O. et al. [23] and Mantzaris D. et al. [24] involve a more profound architecture that includes input and output layers alongside multiple hidden layers to process records iteratively, moving data among layers that would minimize the loss function and acquaint weights and biases of each layer. Various ensemble approaches, such as random forest and boosting, have been experimented with as alternatives to machine learning approaches for predicting future illness. Exponential research has been conducted using either of those approaches with real-time and simulated data. The ensemble approaches work faster for classification when compared to the conventional classification models. However, either of the models ends up with non-additive issues. The resultant effect in the forward direction of the layers would determine the predictive analysis, and the backward pass would assess the standard error among the prediction made and the ground facts.

2.2. Deep Learning for Type-2 Diabetes

The Deep Learning (DL) model implements the framework that infers target gene expression obtained from the expression of landmark genes. Utilizing 111,000 individual gene patterns over a Gene expression Omnibus2, Deep Neural Network-based Gene Analysis model (D-GEX) trained a feedforward neural network through three hidden layers. DL models outperform linear Regression in summarizing expression levels of over 21,000 human genes based on a collection of landmark genes with about 1000 sequences. Although the DL model is more accurate than conventional classification models, performance is not adequate in the healthcare domain, where the design of DL models needs to be improved. The deepVariant model outperforms all other recent neural network models [25]. Deep-

Variant generalizes its training samples by utilizing various human genome expressions as train and test datasets.

Additionally, when training with human gene expressions and evaluating with a mouse genomic expressions dataset, DeepVariant obtained Accuracy that outperformed training with mouse data. DeepFIGV is a deep learning algorithm that uses DNA sequences to predict locus-specific signals from epigenetic tests. DeepFIGV quantifies epigenetic variance by employing several investigations with similar cell patterns and experiments [26]. It combines the entire gene sequence to provide a customized genetic line for each person. The Gene Co-Expression model is a differential network analysis model extensively used in gene data analysis to identify gene sequence similarities and topologies [27]. This model considers two classes of the gene through which the classification model is implemented. However, the Gene Co-Expression model has to deal with comparatively larger features than the size of the data and the non-linearity of the network architecture, where dependencies would make it difficult to trust the model's predictions. Reinforcement Learning (RL) based intelligent systems such as Q-Learning, State Action Reward State Action (SARSA), Deep Deterministic Policy Gradient (DDPG), and Deep Q Network (DQN), as stated by Travnik Jaden B. et al. [28] are the most suitable for handling healthcare to better forecast a future illness with minimal training of the algorithm recovers. The underlying technology remains the same with minimal training. The algorithm is mechanized to learn from its previous experiences.

Various studies have been presented to predict future illness through existing patient data using machine learning algorithms. Predicting future illness has become a demanding topic in healthcare [29]. Several studies have used machine intelligence techniques to analyze the Pima Indian Diabetes Dataset. C. Yue [30] has investigated various hybrid approaches, including Neural Networks, integrated Quantum Particle Swarm Optimization (QPSO), and Weighted Least Square (WLS) Support Vector Machine (SVM) for diabetes prediction, with the WLS-SVM hybrid model showing a classification accuracy of 82.18%. However, the hybridization model needs considerable effort in the evaluation process. In addition, the SVM model is not suitable for working with larger data [31]. Moreover, the SVM model underperforms if the number of attributes for every data point exceeds the training samples. The combinational models for diabetes prediction using Cross-validation and Self-Organizing Maps (SOM) have achieved an accuracy of 78.4% [32,33]. SOM can rely on the associated weights of neurons for precise classification. Inappropriate assignment of initial weights may impact the model's performance. A C4.5 technique [34] has been used to analyze the PIMA dataset, attaining an Accuracy of 71.1%. The model works through the entropy value associated with the feature vector. The conventional classification models exhibit poor performance when working with distinct feature vectors [35].

A fuzzy entropy approach for feature selection for a similarity classifier has been evaluated against various medical datasets, such as Pima–Indian diabetes, exhibiting an accuracy of 75.29% [36]. A fuzzy model primarily depends on the membership evaluation that requires considerable effort. Non-linearity in evaluating the model will limit the model's performance [37]. Genetic Algorithm (GA) with Radial Basis Function Neural Network (RBF NN) has been used in the evaluation process of diabetes data, exhibiting an accuracy of 77.39% over the testing dataset [38]. Moreover, for artificial evolutionary algorithms such as GA, the most prohibitive and restricting element is frequently repeated fitness function assessment for complex gene patterns. Hybridization of models with GA would need more computational efforts than neural networks alone. Various cutting-edge technologies for the classification and prediction of type-2 diabetes are presented in Table 1.

Table 1. Various existing models for diabetes prediction.

Approach	Type of Data	Applicability	Limitations
polygenic scores-based approach [12]	Genomic Data	Used in the evaluation of clinical trials and illness screening mechanisms	The polygenic score approach needs larger samples and tremendous training for considerable Accuracy.
Singular Value Decomposition [13]	Genomic Data Tabular Data The image they are used	They are used in ranking the feature set and compression of the data through the least-square fitting. Gene sequences are ranked based on the probability of illness.	SVD is not an algorithm designed to perform; it is a matrix decomposition mechanism. They are various neural ranking models that perform much better than SVD.
Principle Component Analysis [14]	Genomic Data Tabular Data	PCA technique is extensively used in gene analysis to discover the regional and ethnic patterns of genetic variation.	The independent gene expressions are less interpretable, and information loss is possible if the number of components is carefully chosen.
Gene Co-Expression model [27]	Genomic Data	The Gene Co-Expression model analyzes the genomic data's insights through similarity assessment of expressions and topologies.	The Gene Co-Expression model may not deal with larger features than the data size and non-linearity in the network architecture.
Reinforcement approaches (SARSA, DDPG, DQN) [28]	Genomic Data Tabular Data Image Data	The reinforcement learning models are widely used in studies where the states in the problem are deterministic and in situations where control over the environment is needed. RL models are proven to exhibit better non-linearity in gene analysis.	Adding excessive amounts of reinforcement learning may result in an overflow of states, which might reduce the effectiveness of the findings. As well, RL models are data-hungry.
Decision Tree [39]	Tabular Data Image Data	Using Decision Trees, the efforts to preprocess data can be reduced as normalization and scaling are not required, and missing values will not influence the model's outcome.	DT models consume more time to train the model, and more effort is desired.
J48 [40]	Tabular Data Image Data	J48 is a decision tree that can handle outliers effectively and robustly in non-linear problems.	J48 model is less stable, and noisy data compromises the efficiency of the data.
K Nearest Neighbor [41]	Tabular Data Image Data	The K Nearest Neighbor model does not need prior training for classifying the class data. It requires lesser computational efforts and a faster resultant outcome.	The KNN model fails to work with a larger dataset and high-dimensional data. The feature scaling phase is crucial for an optimal classification level, which requires considerable effort.
Logistic Regression [42]	Tabular Data Image Data	Logistic Regression is the very predominantly used classification technique. The model efficiently classifies the data based on the likelihood and the association among the data items. The model can sustain the overfitting and underfitting issues.	The challenging part of the Logistic Regression is linear separatable and often leads to overfitting when observations are fewer concerning the feature set size.
Naive Bayes [43]	Tabular Data Image Data	Naive Bayes algorithms perform well for multi-class classification models with minimal training.	NB assumes all the feature vectors as mutually independent components in the classification process. NB may not perform better in evaluating the problems with the interdependent feature set.
Random Forest [44]	Tabular Data Image Data	Random Forest models perform bagging for classification. RF models efficiently reduce the over-fitting issue and can handle the missing effectively. Moreover, the feature scaling task need not be performed.	RF models need tremendous training, and frequent hyperparameter tuning is required for considerable Accuracy.

Table 1. *Cont.*

Approach	Type of Data	Applicability	Limitations
Support Vector Machine [45]	Tabular Data Image Data	Support Vector Machine is efficient in handling thigh-dimensional and efficient memory handling capability.	SVM is inappropriate for working with a larger dataset with a larger feature set. The outcome of the SVM model is largely dependent on the objective function. Too many support vectors will be generated when choosing a larger kernel, which might impact the model's training process.
Genetic Algorithm [46]	Genomic Data Tabular DataImage Data	A genetic algorithm is an evolutionary algorithm that uses probabilistic transaction rules, and non-linearity in the searching process would yield better model accuracy. As well, can effectively handle the larger search space.	The genetic algorithm has susceptible to local maxima and minima and similarly to global maxima and minima. That might result in poor prediction performances.

All the mentioned models rely on tabular datasets such as PIMA and ECG signals [47] in classifying the records with possible diabetic illnesses. The current study considers that genomic data yields a better patient-centric outcome than tabular data.

2.3. Genomics for Type 2 Diabetes

Many research studies have been carried out on genetic-based illness prediction. Incorporating machine learning approaches with genetic-based illness prediction could result in an accurate outcome. This has intensified the role of Artificial Intelligence (AI) in healthcare. It has been estimated that approximately $36 billion will be invested in AI by 2025 [48]. Deep genomics through machine learning approaches has outperformed accuracy in predicting and diagnosing illnesses such as cancer with minimal inclusion of radiologists. It is desired to have sufficient biological knowledge to understand how genetics can help us predict various conditions and analyze each chromosome to identify the disease-causing gene. Pre-existing research studies have focused on genomics and gene interaction patterns of various persistent illnesses such as Alzheimer's, multiple cancers, and Parkinson's.

Many aspects need to be considered in the predictive analysis of an illness, as a gene mutation might lead to two or three diseases. The main challenge when handling genomic data for illness prediction is that the prototypical microarray image consists of fewer records. In contrast, the number of fields concerning genres could result in a few lakhs that might misinterpret the data with a significant false-positive ratio. Enhanced Gene-Set analysis can be deployed to extract and analyze genes resulting in soaring throughput on molecular assessments. Gene-Set analysis, as stated by Mooney M. A. and Wilmot B. [49] and Mathur R. et al. [50], is also referred to as pathway analysis, is meticulous in aggregating gene-sets with identical properties or sequences per the reference's gene trained or presented in the disease's knowledge base. Genome-wide association studies (GWAS) have demonstrated that many disease-causing genes are related to human diseases. GWAS has also provided polygenic characteristics of diseases. Figure 1 presents a block of GWAS in disease prediction. There are many steps during a gene-set analysis. They are shown below as Steps 1 through Step 6:

Step 1: Preliminary genome-wide analysis and data preprocessing;
Step 2: Identifying gene-set definitions whose patterns have to be recognized;
Step 3: Processing genomic data such as filtering and identifying gene patterns;
Step 4: Identify gene set analysis models, such as identifying the statistical hypothesis;
Step 5: Assessing the statistical magnitude;
Step 6: Report summarization and visualization.

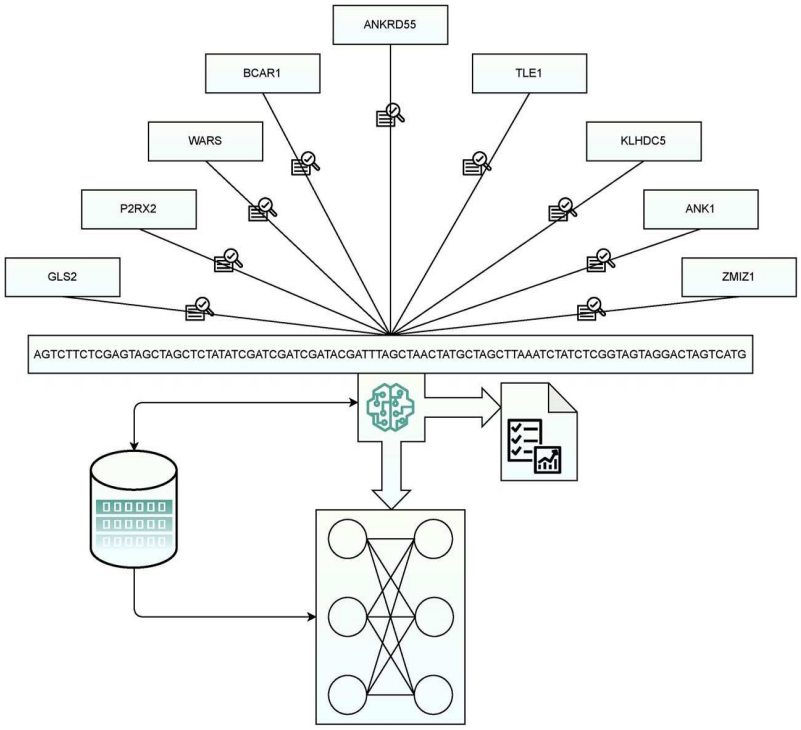

Figure 1. Gene analysis-based disease perdition framework.

Gene data include metadata about the information associated with type 2 diabetes, consisting of alleles, MegaBase, and Single nucleotide polymorphisms. An allele is a word that denotes a particular gene sequence copy associated with a specific context. A mutation might be considered one of two or more varieties of a particular gene. Most individuals have SNPs. However, some variants are more prevalent than others in particular populations. A single DNA-building unit, the nucleotide, is found at tens of thousands of different sites on the human genome. In genetics, a MegaBase is a unit of length representing a genomic region's length. MegaBase is used to determine the distance between two genes. Values of these gene features mentioned above are considered when evaluating the possibility of feature disease.

A highly dense genotyping collection is considered for coverage throughout the whole genome, such as covering common and uncommon variations in the genome. These gene sequences contain many single-nucleotide polymorphisms (SNP) that can significantly improve the capture of low-frequency variations, which is advantageous to users of other genome-wide collections. Gene sequences that hold a higher possibility of T2D in the future are listed in Table 2. Disease-corresponding gene sequences are cross-validated against individual data for forecasting the likelihood of the disease.

Table 2. Genomic information associated with Type 2 diabetes.

Gene Data	Type 2 Diabetes	Fasting Glucose	Alleles	SNP	Megabase
GLS2		✔	G/A	rs2657879	55.2
P2RX2		✔	A/G	rs10747083	131.6
WARS		✔	G/T	rs3783347	99.9
BCAR1	✔		T/G	rs7202877	73.8
ANKRD55	✔		G/A	rs459193	55.8
TLE1	✔		G/A	rs2796441	83.5
KLHDC5	✔		C/T	rs10842994	27.9
ANK1	✔		C/T	rs516946	41.6
ZMIZ1	✔		A/G	rs12571751	80.6

3. Methodology

This study is focused on predicting future illnesses such as type-2 diabetes from genomic and tabular data. Genomic data are analyzed for possible gene expression highly likely to be affected by type-2 diabetes. Tabular data from the PIMA dataset with various features are also explored through the proposed RNN model by identifying the feature vector's pivotal features. The proposed model relies on the Deep Neural Networks (DNN) framework for analyzing the genomic data, making the precise assessment of possible future illnesses with better Accuracy than the conventional pattern-matching techniques. DNN is a probabilistic measure that would summarize the possible illness outcome that would better assist in decision-making by the physicians. The working procedure and implementation details are discussed in the current section. The models are trained from the available gene base from scratch initially, and at the later stages, the model learns from the experimental outcomes.

3.1. Recurrent Neural Network Model for Type 2 Diabetes Forecasting Based on Genomic Data

Predictions of future illness can be performed through Convolutional Neural Networks (CNN), as stated by Leevy J.L. et al. [51] and Yadav S.S. and Jadhav S. M. [52] using Recurrent Neural Network (RNN) module-based architecture described by SivaSai J.G. et al. [53]. CNN model consists of many intermediate nodes connected. Each node is significant in delivering the output following the anticipated outcome. RNN is robust in handling variable-length input sequences with the help of internal auxiliary memory modules [54]. The detailed architecture along with the implementation procedure for the proposed approach, is presented in this section.

With the proposed approach, gene patterns are analyzed against pre-trained sequences of genes that cause the disease. For the effective implementation of illness prediction, the recurrent neural network component is incorporated with gene set analysis, which could minimize the false positive ratio. The Recurrent neural network model is a layered architecture approach where each layer works independently. The output of the previous phase is fed as the input for the next phase. Recurrent neural networks can transform individualistic components into contingent components by adjusting each layer's weight and bias by minimizing the number of parameters to be considered and reducing the complexity of memorizing the previous layer's output. The responsibility of each layer is presented in this section, along with the working procedure of the proposed model.

3.1.1. Data Collection and Processing

Gene-related data were acquired from the open-access comprehensive miRbase-18.0 R dataset with human gene sequences of 10,094 records labeled and annotated [55,56]. In the present experimental study, 303 samples were considered for the training and validation of the model at 70:30 proportion, respectively. Generally, gene sequences are 84 nucleotides in length, ranging from 43 nucleotides to 154 nucleotides.

The data acquired from online repositories must be processed following the model's outcome. The information is organized in tables to be further refined to predict gene sequence better, including aligning the region of interest in genomic patterns. Gene sequences could be expressed as a grid in which each location corresponds to a single-hot vector containing letters A, C, G, and T. Gene expression is indeed a matrix containing absolute values, each such element representing the pattern that is an integral part of the gene in a particular environment, such as a cell. Spatial information is often described as a three-dimensional array, with two dimensions representing the entity's actual location and a third dimension representing colors or genes. Typically, texts are defined as a one-hot matrix for each token entering a stable database. While most cells have the same genome, individual genes are expressed at highly variable amounts in variable tissues and cells in response to various treatments and settings. Such degrees of gene expression could be quantified by measuring levels of mRNA transcripts. In such context, comparison of gene

expression of patients with the illness to that in healthy cohorts (without the disease of interest) and different link genes with the diseases underlying biological systems.

3.1.2. Feature Selection

When analyzing gene data for illness prediction, features are significant in obtaining an accurate and precise outcome. Feature selection is one of the vital phases of the proposed approach. The feature selection process performed during the training step would have a noticeable contribution to the dimensionality reduction of gene data, including discarding irrelevant data and recognizing vital records in the dataset. The proposed approach's performance depends on the feature selection mechanism in the present work. It is significant in identifying the diseased gene from the extracted genomic information for the human body. Minimum Redundancy Maximum Relevance (mRMR), as stated by Zena M. Hira and Duncan F. Gillies [57] and M. B. Shirzad and M. R. Keyvanpour [58], was used for feature selection and extraction of microarray data in the current study.

The minimum Redundancy Maximum Relevance (mRMR) approach maximizes the relevancy of components concerning the genomic information and minimizes the number of corresponding classes. mRMR-based feature selection technique that favors features that have a strong correlation with class but a low correlation among themselves. In the feature extraction process, divergent statistical metrics are considered, including Mutual Information (MI), which assesses the entropy of a random variable concerning other variables in the corresponding class. The mRMR approach can also be used with both continuous and discrete variables. The amount of MI among features is used to calculate redundancy. If the value of MI is substantial, it indicates a significant degree of data redundancy among the two characteristics, i.e., redundancies. A lower redundancy measure value suggests more effective feature selection criteria. The purpose of redundancy is to locate the feature with the lowest MI value among all features. According to the premise that the lower the value of information redundancy across features, the more helpful it is to activity categorization, which may be stated by decreasing MI among features [59]. The following equation determines the gene that is not redundant for a set of features $\beta(x \in \{1, 2, \ldots f\})$

$$M_r = \frac{1}{|f|^2} \sum_{\alpha,\beta \in C} MI(\alpha, \beta) \quad (1)$$

In the above Equation (1), the discrete variable MI is the Mutual Information, variables α and β represent genes, and $|f|$ represents the number of features in class C. The maximum relevance concerning the target class is determined through the following equation:

$$R = \frac{1}{|f|} \sum_{\alpha \in C} MI(\gamma, \alpha) \quad (2)$$

In the above Equation (2), the variable γ is the class label for discrete variables. F-Statistics for assessing the mean of two classes are significantly divergent for determining the maximum relevance among corresponding genes and the class label. The minimal redundancy that approximates the correlation of the complementary gene pairs in the class is approximated as shown in Equations (3) and (4)

$$R = \frac{1}{|f|} \sum_{\alpha \in C} F_s(\alpha, \gamma) \quad (3)$$

$$M_r = \frac{1}{|f|^2} \sum_{\alpha,\beta \in C} x(\alpha, \beta) \quad (4)$$

The mutual information among the two gene sequences let them be p and q, and if some gene-sequence of p is there in the gene sequence q. The MI for the gene sequences is assessed using the generic formula for mutual information, as shown in Equation (5).

$$MI(p;q) = \sum_{p,q} f(p,q) log \frac{f(p,q)}{f(p), f(q)} \quad (5)$$

The Pymrmre package helps work with the mRMR method by employing an ensemble mechanism to further investigate the feature map and construct a more robust feature set. Figure 2 represents the feature selection mechanism for selecting the optimal features for gene-data analysis.

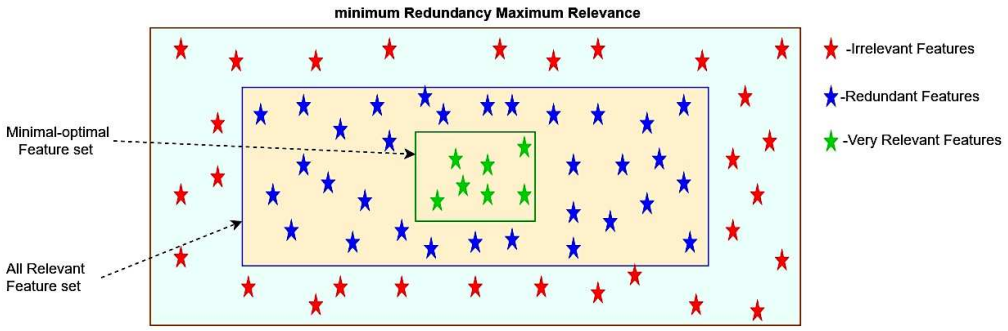

Figure 2. Diagram representing the mRMR Features selection technique.

3.1.3. Layered Architecture of RNN-Based Prediction Model

There are multiple layers in the proposed RNN model. Each plays a vital role in performing the predictive analysis of the illness, according to Carrara, F., Elias [60], and Che et al. [61]. The RNN model has kernels that work on inputs to create a feature map to detect referee patterns in the corresponding input sequence. The outermost layers would be the input and output layers. There are many other intermediate layers, including the Convolutional, max-pooling, Flattening, fully connected, and softmax layers. The outermost layer captures the gene sequences that must be validated against the training set. The inner Convolutional layers are used to handle complex patterns. Each of those Convolutional layers also decomposes gene sequences.

The pooling layer acts as the interface between two convolution layers. Its focus is on minimizing the number of parameters required for processing the data, thereby handling overfitting. The pooling layer is responsible for reducing the spatial size of the model so that the model is computationally feasible. The max-pooling layer would result in the statistical outcome of decomposing the input to the minor extent possible and performing components' filtering. Members that hold the maximum values are processed to the further stage. The rest of the components are left unprocessed. To flatten the layer associated with the conversion process of the data obtained from the previous layer, it is necessary to create a one-dimensional array of gene data that contain data to be fed to the next layer. The convolutional result is flattened to compress the outcome of convolutional layers into a single lengthy feature vector. The final classification model is termed a fully connected layer. It is linked to the output. The fully connected layer does have connections to all nodes in the layer. It is feasible to learn all nonlinear combinations of various complex patterns. The reasonability of this layer is to obtain the probability of the gene causing the abnormality. The fully connected layers comprise two significant layers. The first fully connected layer gets the input data from parameter analysis and labels the input GENE sequence for accurate prediction through weights. The fully connected output layer approximates probabilities of illness-causing genes from gene sequences [62].

In addition, the Softmax layer expands the concept into something similar to a multi-class environment. Specifically, in a multi-class classification issue such as disease predictions, Softmax gives decimal probability to each class. The sum of all such probabilities

associated with each category is equivalent to 1.0 in the long run when dealing with decimal probability. Figure 3 presents the layered approach of RNN used in the prediction model.

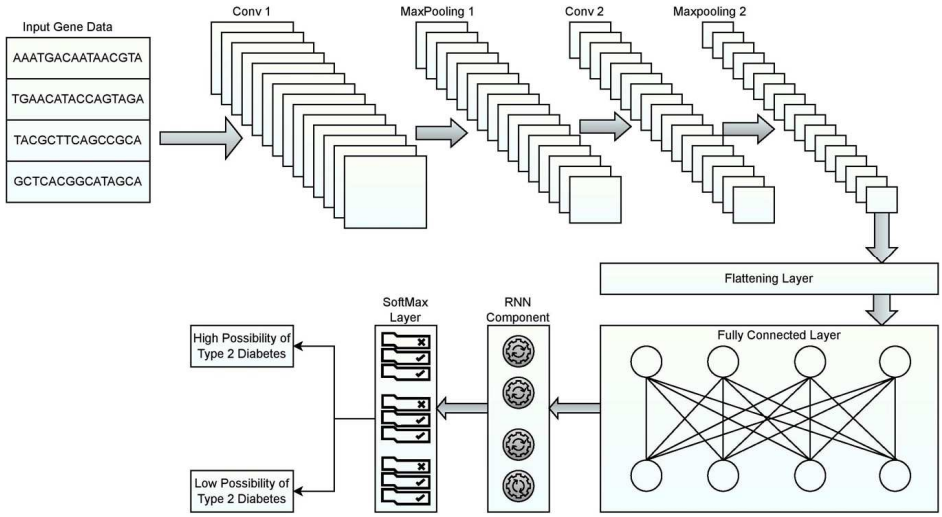

Figure 3. Layered architecture diagram of the RNN model.

Convolution is an operation that transforms a function into an output component; it is a technique that follows a certain sequence and involves intertwining two different information sources. Every single Convolutional Neural Network starts with a Convolutional Layer as its very first layer. The input is subjected to a convolutional operation in convolutional layers, and the output is then passed on to the next layer. A convolution reduces the values of all the pixels included within its receptive field to one. ReLU activation function is used with the Convolution layer. When using ReLU, all of the negative pixels are converted to 0 via an element-wise procedure. The result is a corrected feature map, which adds non-linearity to the network.

- After the Convolutional layer, the pooling layer is often applied. The pooling layer's purpose is to minimize the volume of the input matrix for subsequent layers. In the current study, the MaxPooling function is used in the current study.
- A flattening operation transforms data into a one-dimensional array to be used in a subsequent layer. This is conducted so that CNN's output may be sent to a fully connected network.
- A neural network is a collection of non-linear, mutually dependent functions. Neurons are the building blocks of every single function (or a perceptron). The neuron uses a weights matrix as a fully connected layer to apply a transformation matrix to the input vector. The result is then subjected to a non-linear transformation via a non-linear input signal s as shown in Equation (6).

$$f_c = f\left(\sum_{i=1}^{p} \omega_{ck} a_i + \omega_{c0}\right) \quad (6)$$

- One way to represent a set of numbers as probabilities are to use the Softmax mathematical function, which multiplies all the values in a set by the scale at which they appear in the vector. The likelihood of belonging to each class is calculated using the outcome of the softmax algorithm.

3.1.4. RNN Component Structure

A recurrent Neural Network, also known as a back-feeding neural network, is a more robust alternative to conventional feedforward neural networks as it does not need an internal auxiliary memory. As the outcome of a current input relies on the previous calculation, RNN is recurrent. After the outcome has been produced, copying and sending the output into the recurrent network is known as "back-feeding." The decision-making

process analyses what it has learned from the prior information and applies it to the present situation. Using the gene patterns present in the sequence, RNN may extract the correlated patterns that result in type-2 diabetes. The same could be employed in analyzing variable-length gene data for the probability of being affected by type 2 diabetes.

An RNN can evaluate any sequences, irrespective of length, iteratively through its transition function over the state vector O_i. At iteration i, state activation may be calculated as a function of the input sequence character Z_i and the prior state vector N_{i-1} transformed into the R_i in the current state cell. The tan h is the activation function associated with each cell. In RNN, the vanishing gradient issue is considered the most crucial challenge. More extensive sequences need an activation function such as tan h with a high second derivative that can maintain the gradient over iterations. Mathematical notations for each RNN module are presented in Equations (7)–(11).

$$N_i = \sigma_N(x_i) \tag{7}$$

$$\sigma_N(x_i) = \sigma_N(\alpha_N z_i + \beta_N R_{i-1} + b_N) \tag{8}$$

$$g_i = \sigma_g(x_t) \tag{9}$$

$$\sigma_g(x_t) = \sigma_g(W_g N_i + b_N) \tag{10}$$

$$O_i = \tan h\,(W_g O_{i-1} + W_{g-1} Z_i) \tag{11}$$

In Equations (7)–(11), the variable R_i denotes the input vector of size $(1 \times x)$, the variable Z_i. The input for the RNN cell denotes the input vector of length $(1 \times x)$. Variables α and β denote the parameter matrix associated with pivotal features. The bias is represented by b_N. Variables σ_N and σ_g denote the activation function in the RNN cell. Variable W_g and W_{g-1} denote weights associated with the cell in the current and previous state. The RNN cell structure is presented in Figure 4, where the output of the previous component is fed as the input for the upcoming component in the RNN cell.

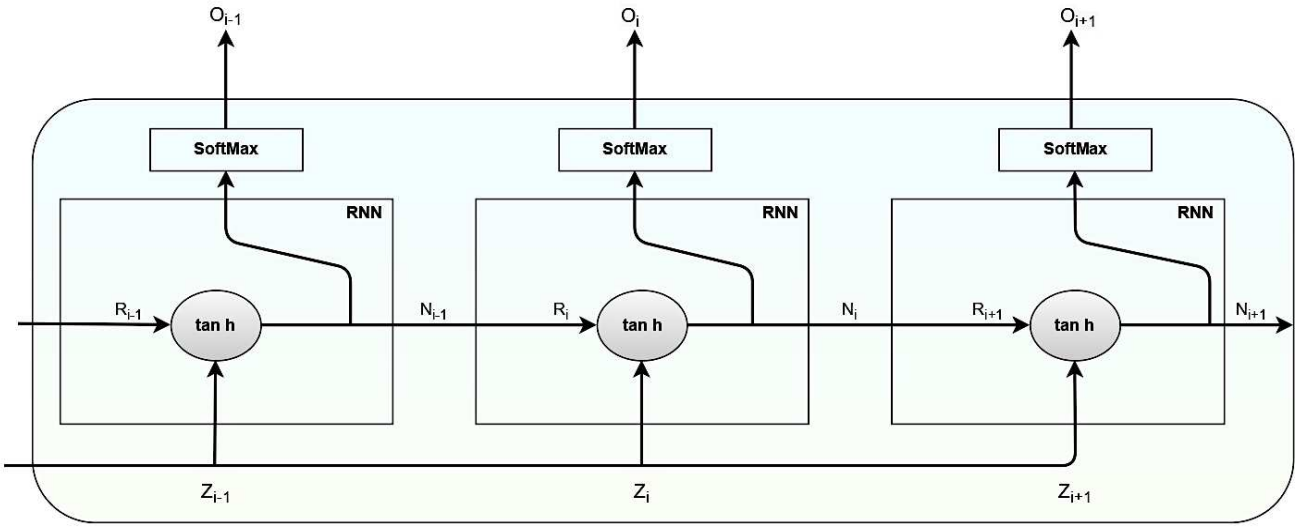

Figure 4. Image representing the RNN cell structure.

In the RNN model, the tan h denotes the Activation function, which implements a non-linearity that negates maximum activation values, creating a negative activation -1. The softmax layer deliberates probabilities that will assist in determining the possibility of future illness from the provided input genomic data.

3.1.5. GRU Component Structure

GRU's component in neural networks is used to address the degradation issue and create a feasible deeper layout for better Accuracy that can retain lengthy semantic patterns

without calibrating model parameters [63]. The GRU component consists of the update gate and the reset gate. The update gate regulates the inflow of data to the memory component. The reset gate regulates data flowing out of the memory component, GRU. The gating unit controls the data flow inside rather than having a separate memory component to perform the task. The unit consists of two activation functions: σ and $\tan h$. The output of the current units is identified by cs_t becomes the input for next unit as cs_{t-1} over the time t. The variable α_t is assumed as the input training data and β_t is the corresponding output generated by the activation functions Γ_r and Γ_u that denotes the reset gate and the update gate, respectively. The value of Γ_u lies in between 0 and 1. When its values are close to 0, more data from the previous states are retained. The range of the variable Γ_r lies in between -1 and 1. When the value is close to -1, it implies that more previous data are ignored. The GRU can be shown mathematically through Equations (12)–(15).

$$\Gamma_u = \sigma(\omega_u[cs_{t-1}, \alpha_t] + bias_u) \tag{12}$$

$$\Gamma_r = \sigma(\omega_r[cs_{t-1}, \alpha_t] + bias_r) \tag{13}$$

$$\hat{cs}_t = \tan h(\omega_{cs}[\Gamma_r \times cs_{t-1},\ \alpha_t] + bias_{cs}) \tag{14}$$

$$cs_t = (1 - \Gamma_u) \times cs_{t-1} + \Gamma_u \times \hat{cs}_t \tag{15}$$

From Equations (12)–(15), variables ω_u, ω_r, and ω_{cs} designate weights associated with training the update gate, reset gate, and candidate activation, respectively. Similarly, variables $bias_u$, $bias_r$, and $bias_{cs}$ designate the bias associated with the update gate, reset gate, and candidate activation, respectively. Figure 5 presents the architecture of the GRU module.

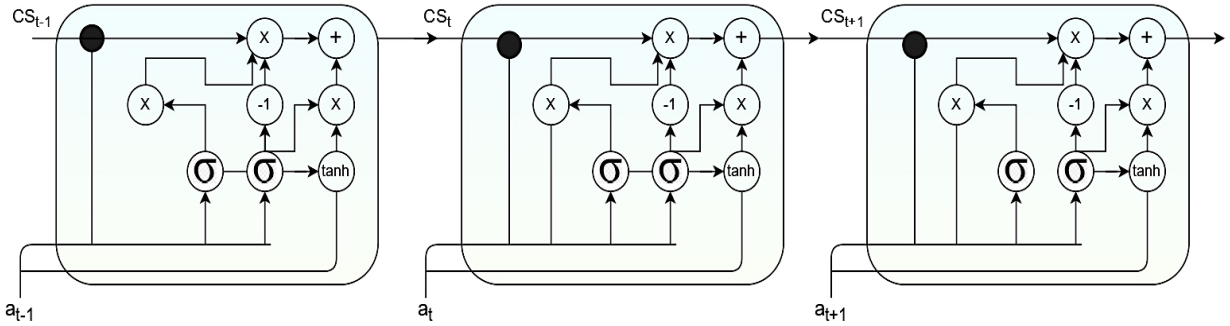

Figure 5. Image representing the GRU cell structure.

3.1.6. LSTM Component Structure

LSTM component is often used in recurrent neural network designs for pattern estimation issues in the sequential data over divergent time scales. Memory cells handle memory components in an abstract LSTM layer module, including an input and output gate, a forgetting gate, and a window connection [64,65]. Associated weights are comparable to those that change during a model's training process to regulate input and hidden states. Activation functions for the LSTM component are explained with Equations (16)–(20). States are identified through variable S_t with a hidden state vector identified by ϑ_t concerning the time t over the input i^t.

$$Input\ Gate\ (\rho_t) = \sigma\left(i^t\omega_{i\rho} + \gamma_{t-1}\omega_{\gamma\rho 0} + ps_{t-1}\omega_{ps\rho} + bias_\rho\right) \tag{16}$$

$$Output\ Gate\ (o_t) = \sigma\left(i^t\omega_{io} + \gamma_{t-1}\omega_{\gamma o} + ps_t\omega_{pso} + bias_o\right) \tag{17}$$

$$Forget\ Gate\ (\chi_t) = \sigma\left(i^t\omega_{i\chi} + \gamma_{t-1}\omega_{\gamma\chi} + ps_t\omega_{ps\chi} + bias_\chi\right) \tag{18}$$

$$Cell\ State\ Gate\ (ps_t) = \chi_t \cdot ps_{t-1} + \rho_t \cdot \tan\gamma\left(i^t\omega_{ips} + \gamma_{t-1}\omega_{\gamma ps} + bias_{ps}\right) \tag{19}$$

$$LSTM\ Output(\gamma_t) = o_t \cdot \tan\gamma\ (ps_{t-1}) \tag{20}$$

From Equations (16)–(20), variables $\omega_{i\rho}$, ω_{io}, $\omega_{i\chi}$, and ω_{ips} designate weights associated with the input, output, forget, and cell state gates, respectively. In addition, $\omega_{\gamma\rho}$, $\omega_{\gamma o}$, and $\omega_{\gamma\chi}$ designate weights associated with the hidden layer. Similarly, variables $bias_{\rho}$, $bias_{o}$, $bias_{\chi}$, and $bias_{ps}$ designate the bias component associated with the input gate, output gate, forget gate, and cell state gate, respectively. The architecture of the LSTM component is presented in Figure 6.

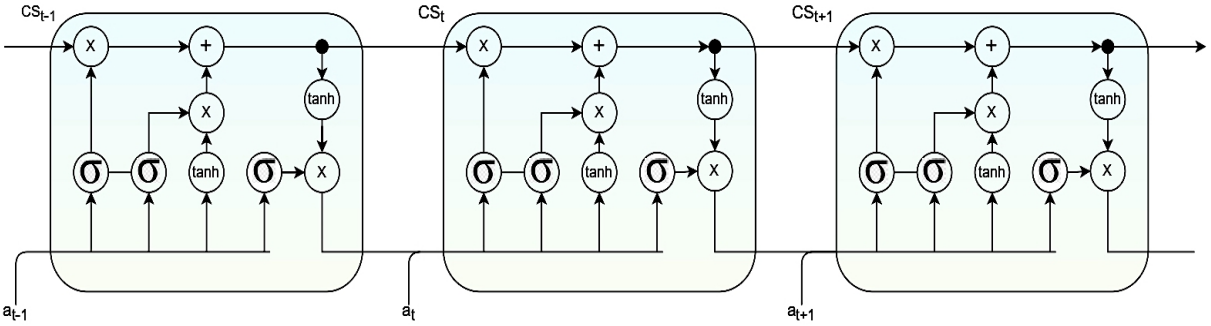

Figure 6. Image representing the LSTM cell structure.

3.1.7. Working Procedure of the Proposed Approach

The working procedure presents the sequence of tasks performed for future illness prediction, including tasks ranging from initial data acquisition to final assumptions of the future illness.

Step 1: Acquire gene data from the annotated miRbase data set;
Step 2: Data are preprocessed to remove the outlier data and fill out acquired data gaps;
Step 3: Data is converted into 1D data, followed by aligning of genomic patterns;
Step 4: Data is categorized into a training set (80% of the data) and a testing set (20% of the data);
Step 5: Patterns are labeled based on sequence patterns of various illnesses. Moreover, weights are assigned in the later phases according to the correlation between the input sequence and the trained gene pattern;
Step 6: When a new GENE sequence is fed as input for testing the algorithm, features are extracted through the mRMR approach that is pivotal in the prediction process;
Step 7: The cumulative weight is evaluated from assigned weights based on the correlation of gene sequences between the input and the trained set;
Step 8: Based on the approximated weight of the gene sequence, the probability of a future illness is assessed;
Step 9: Final assumptions are made based on probabilistic approximations.

3.2. RNN Model for Illness Prediction from Tabular Data (PIMA Dataset)

The possibility of being affected by a chronic disease such as type-2 diabetes is analyzed from the tabular data with various features such as Stabilized Glucose, age, High-Density Lipoproteins (HDL) Ratio, Total Cholesterol, First Systolic Blood Pressure, Second Diastolic Blood Pressure, body mass, the height of individual, gender, and many other things. Significant features are selected, weights are adjusted in favor of pivotal features, and prediction is performed based on the feature vector. The significance of these features in the evaluation process has been discussed in earlier studies on a similar feature vector for type 2 diabetes [66–68]. Ranks associated with each of these features are presented in Table 3, shown below.

Table 3. Feature set associated with Type 2 diabetes.

Feature	Data_Type	Min_Value	Max_Value	Information Gain	Mean Rank
Glucose (mg/dL)	Integer	0	199	0.2497	3
Pregnancies	Integer	0	17	~	~
Age	Integer	21	81	0.0761	3.17
Heart Rate	Integer				7.67
Waist	Integer	~	~	0.0356	9.5
Pulse Pressure	Integer			~	12.33
Insulin (mm U/mL)	Integer	0	846	~	13.33
Hypertension (Blood Pressure) (mm Hg)	Integer	0	122	0.0304 (bp1), 0 (bp2)	15
BMI (weight) (kg/m^2)	Real	0	67.1	~	~
Diabetes Pedigree Function	Real	0.08	2.42	~	~
Skin thickness (mm)	Real	0	99	~	~

3.2.1. Feature Weight Initialization

Features are essentially important for analyzing the possibility of diabetes disease. Weights associated with features and corresponding layers are updated over iterations [69]. When the feature is significant, it will cascade forward via hidden nodes, showing greater influence over output nodes. Thus, weighting such a significant feature is important. The feature that contributes more to the prediction process will be given considerable weightage for further processing. After training, the feature weight is obtained from the trained neural network, as shown in Equation (21).

$$I_w = \sum_{i=0}^{p-1} \sum_{j=0}^{q-1} \left| \omega_{i,j} \times \omega_{j,k} \right| \tag{21}$$

From Equation (21), the variable I_w is the initial weight assigned to the feature vector, $\omega_{i,j}$ denotes the network weight between the input node i through hidden node j. Similarly, the variable $\omega_{j,k}$ denotes weights of the hidden node j through output node k. The summation covers all potential forwarding routes between input node i and output nodes. For a few less significant features in the evaluation process, their weights are adjusted so that the sum of approximated weights of less significant features is equivalent to the total number of features. Associated weights for less significant features are given through Equation (22).

$$\omega_{lsf} = \frac{1}{n} \sum_{i=0}^{n-1} \omega_{lsf} \tag{22}$$

From Equation (22), the variable ω_{lsf} designates less significant weights, and the variable n designates the number of features in the considered problem. In the current context, the value of n is 8 from the Pima dataset.

3.2.2. Weight Optimization

Weights associated with features must be optimized regularly for better performance of the model. These weights are optimized concerning the loss function and model parameters associated with each parameter in the training dataset [70,71]. In the current study, the input-target pair (i, j) and the $\left\{ \left(i_p, j_p \right), 0 \le p \le n \right\}$ denote the training set. The validation set is associated with the model for fine-tuning the model's performance using the set $\left\{ i'_p, j'_p \right), 0 \le p \le m \}$, where the size of m is much smaller than the size of records in n. The RNN model is denoted by $\Re(p, \theta)$. The associated loss function will be $L(j', j)$ which is desired to be minimal, where $j' = \Re(i, \theta)$. The expected loss associated with the training set is determined through the variable T_l as shown in Equation (23).

$$T_l = \frac{1}{n} \sum_{p=0}^{n-1} L(j', j) \tag{23}$$

$$T_l = \frac{1}{n}\sum_{p=0}^{n-1} f_p(\theta) \text{ where } L(j', j) = f_p(\theta) \quad (24)$$

In Equation (24), the function $f_p(\theta)$ is the loss function concerning data i_p. Weights associated with parameters are optimized to minimize the weighted loss through Equation (25).

$$O(\theta)_w = \theta' \sum_{p=0}^{n-1} \omega_p f_p(\theta) \quad (25)$$

The value of the variable ω_p is not known at the initial iteration. The value $\{\omega_p\}_{p=0}^{n-1}$ is tuned by training hyperparameters. The validation dataset could result in fine-tuning the value of ω to reduce the weighted loss of the prediction model, as shown in Equation (26).

$$\omega' = \min_{\omega, \omega'} \frac{1}{m} \sum_{p=0}^{m-1} f'_p(\theta \times \omega) \quad (26)$$

To reduce negative training loss that could result in an unstable model, The value associated with the weight $\omega \geq 0$ for all parameters p.

3.3. Dataset Description

The Pima Indian Dataset is used in the current study to predict Type-2 diabetes. It is part of the UCI machine learning repository maintained by the National Institute of Diabetes, Digestive, and Kidney Diseases. The dataset consists of eight columns representing parameters of Pregnancy, Glucose, Blood Pressure, Skin Thickness, Insulin, Body mass index (BMI), Diabetes Pedigree, and age. The PID dataset consists of a single output class with a binary value indicating whether or not an individual has diabetes. The dataset consists of 768 cases (500 non-diabetics and 268 diabetics) [72,73]. The Pima dataset is considered in the current study as it is widely used for comparing the performances of techniques. The dataset is partitioned as training and testing in a ratio of 70:30, with an initial learning rate of 0.0002, and it is observed that the model has

3.4. Implementation Environments

The computer is equipped with an Intel(R) Core i7(11th Gen) 4.70 GHz processor and 16 GB of main memory running over a 64-bit Windows 10 environment. The proposed RNN model for gene analysis is implemented over Kaggle, an online platform for executing such frameworks [74]. Python version 3.6.6, also widely known as the anaconda, is used in the implementation. Tensor Flow version 2.4.1, along with various libraries such as NumPy, pandas, matplotlib, seaborn, and sklearn, are used in the implementation process of the proposed model.

4. Results and Discussion

The proposed model has been evaluated on genomic data and the tabular data by using the same feature engineering mechanism and the layered approach for predicting the type-2 diabetes. The proposed RNN-based type-2 diabetes is evaluated against genomic and tabular data from the PIMA Indian dataset independently and the evaluations are presented independently in the current section. The model was evaluated against two datasets concerning various evaluation metrics such as sensitivity, specificity, Accuracy, and F1 score. The classification efficiency of the proposed model was assessed using true positive (TuP, the number of times that the model accurately predicted the gene with a high possibility of diabetes correctly), true negative (TuN, identifying the gene with less possibility of diabetes precisely), false positive (FsP, misinterpreting the gene with the high possibility of diabetes as low possibility of diabetes), and false negative (FsN, misinterpreting the low diabetes gene as a high possibility of illness). The sensitivity metric determines the ratio of how many were accurately recognized as positive samples out of how many were truly positive samples in the complete dataset. The specificity measure determines the ratio of how many were recognized as negative samples out

of how many among the samples are truly negative from the complete dataset. The Accuracy measures the correctly predicted True positives and Negative samples against the overall sample in the complete dataset. The harmonic mean of sensitivity and specificity measures are determined as the F1 score. MCC is the best single-value classification score for summarizing the confusion matrix. The formulas for the aforementioned metrics are presented through Equations (27)–(32) [75].

$$senstivity(recall) = \frac{TuP}{(TuP + FsN)} \tag{27}$$

$$Specificity = \frac{TuN}{(TuN + FsP)} \tag{28}$$

$$Accuracy = \frac{TuP + TuN}{(TuP + FsP + TuN + FsN)} \tag{29}$$

$$Precision = \frac{Tup}{(TuP + FsP)} \tag{30}$$

$$F1\text{-}score = 2 \times \frac{(precision \times recall)}{(precision + recall)} \tag{31}$$

$$mcc = \frac{(TuP \times TuN) - (FsP \times FsN)}{\sqrt{(TuP + FsP)(TuP + FsN)(TuN + FsP)(TuN + FsN)}} \tag{32}$$

It is a far more appropriate statistical rate that yields a good score only if the prediction performed well among all the assumptions in the confusion matrix. The current section presents results about the experimental outcome of both genomic and tabular data with adequate analysis concerning existing models.

4.1. Experimental Outcome of Genomic Data

The performance of the proposed RNN model for predicting type 2 diabetes was analyzed using performance evaluation metrics such as sensitivity, specificity, F1 score, Mathews correlation Coefficient, and accuracy measures [76]. The above-discussed metrics are assessed through true positive, true negative, false positive, and false negative values approximating experimental outcomes. The dataset is split into a training set and a validation set at a ratio of 70:30. In the following graph, as shown in Figure 7, it is clear that data values are skewed toward data instances, indicating that no diabetes exists. The percentage of available data records of non-diabetic patients (or those who do not have diabetes) is almost double that of diabetic patients.

Correlation coefficients among data points as input gene data are analyzed using linear bivariate Pearson correlation coefficient (PCC). The correlation coefficient between two samples of gene expression is expressed as PCC. Correlation coefficient with a common confidence interval and covariance, the relationship among them is the ratio of the covariance of two variables and the product of their standard deviations. This gives a numeric representation of the covariance with an outcome between -1 and 1. Only a linear correlation between variables can be considered, even using the metric. Also, the metric does not represent several relationships or correlations. Figure 8 shows a two-dimensional heat map of data records.

Training and validation performances of the proposed model were evaluated using hyperparameters such as train and testing scores. The training score determined how perfectly the algorithm could generalize across its training samples. The testing score determined how well the model could accurately correlate the known gene sequence among individual records. An exceptionally high training score combined with a low-test result indicates overfitting. When the training score is quite low, and the test score is low, it indicates an underfitting. Performances of the proposed model concerning hyperparameters are presented in Figure 9.

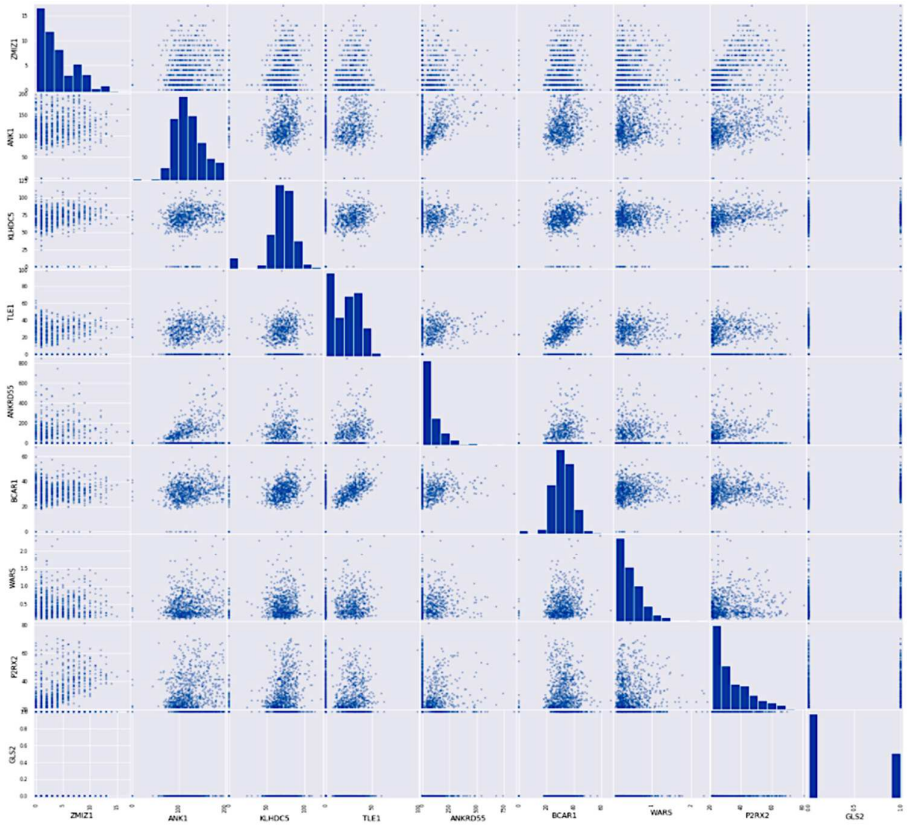

Figure 7. A scatter plot shows relationships among data points in input records.

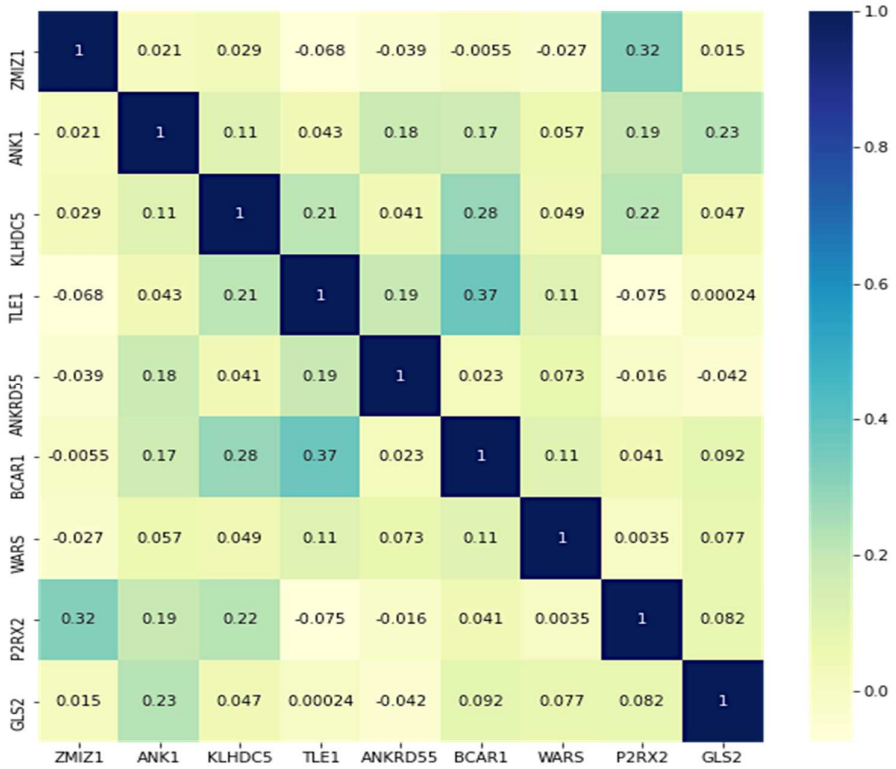

Figure 8. A heat map was generated from the gene dataset.

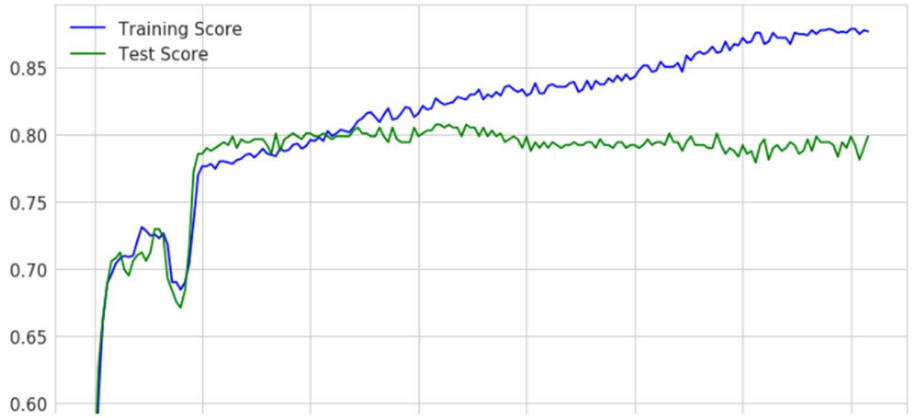

Figure 9. A graph showing training and testing scores of the proposed model.

From graphs on the training and test scores, it can be depicted that the performance is reasonably fair in making the classifications precisely as there is no considerable deviation among either of the scores. In the current context, the gene expressions are classified as sequences with a high possibility of affecting type 2 diabetes and sequences with a low possibility of type 2 diabetes. The Decision Boundary is shown in a Scatter Plot, with every data point visualized on the data scatter plot and characteristics represented by x- and y-axes. The Decision Boundary forms a boundary for dividing data points into regions and their classes. Categories of gene sequences with high and low possibilities of developing diabetes are shown in Figure 10.

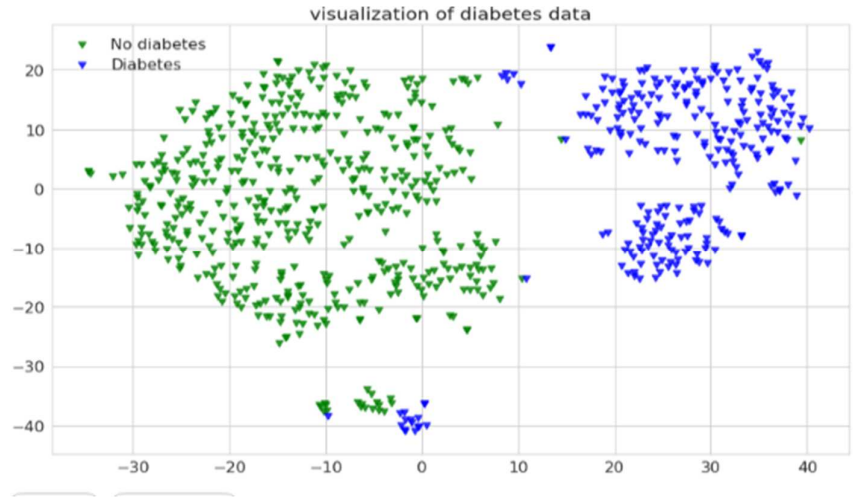

Figure 10. A graph showing the decision boundaries of two classes of records.

The confusion matrix would assist in analyzing the performance of the proposed model in analyzing future illness. The evaluated samples, i.e., TuP, TuN, FsP and FnP are shown in the confusion matrix in Figure 11, and the corresponding performance evaluation metrics are shown in Table 4.

Table 4. Performance evaluation metric and estimated values.

Metric	Estimated Value
Sensitivity	83.66
Specificity	49.38
Precision	75.73
Accuracy	71.79
Mathew's correlation Coefficient	35.09

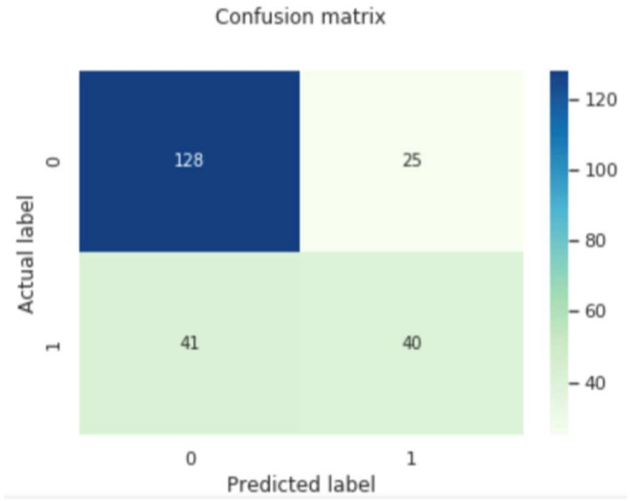

Figure 11. Image representing the confusion matrix for the proposed RNN model for future illness prediction.

As shown in Table 4, estimated values clearly demonstrated that the model made predictions reasonably with few records. However, the model's performance could be further improvised when more data records could be used. The Receiver Operating Characteristic (RoC) Curve of the proposed model is presented in Figure 12, and it is depicted that the model has outperformed with reasonable accuracy in precisely classifying the genomic data. The RoC curve estimates how well the proposed approach can differentiate the two-class records that include gene expression with a higher or lower possibility of being affected with type-2 diabetes. An accurate model can tell the difference between the two. An improper model will find it difficult to tell the difference between the two sets of records.

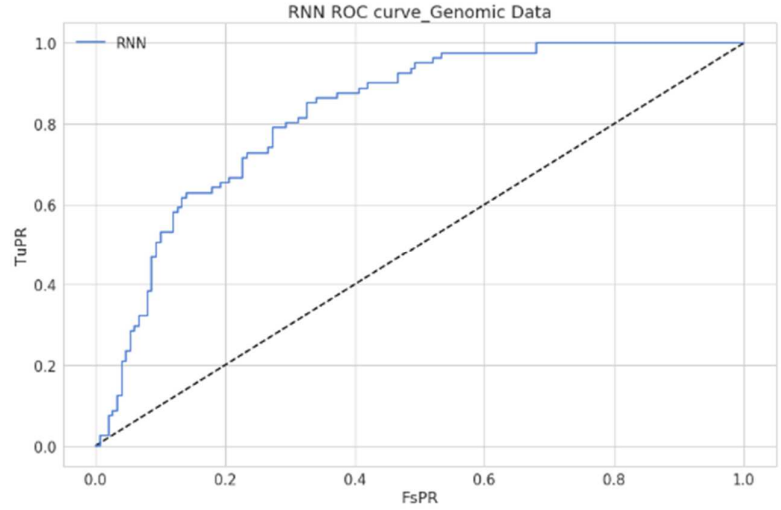

Figure 12. Graph presenting the ROC curve of the proposed model.

4.2. Experimental Outcome with Tabular Data (PIMA Dataset)

The Pima Indian dataset consists of eight features that help predict the possibility of affecting type-2 diabetes. The model's performance was evaluated using various evaluation metrics. The heat map represents the association of multiple parameters in determining a future illness. Figure 13 illustrates the heat map of features in the PIMA Indian dataset.

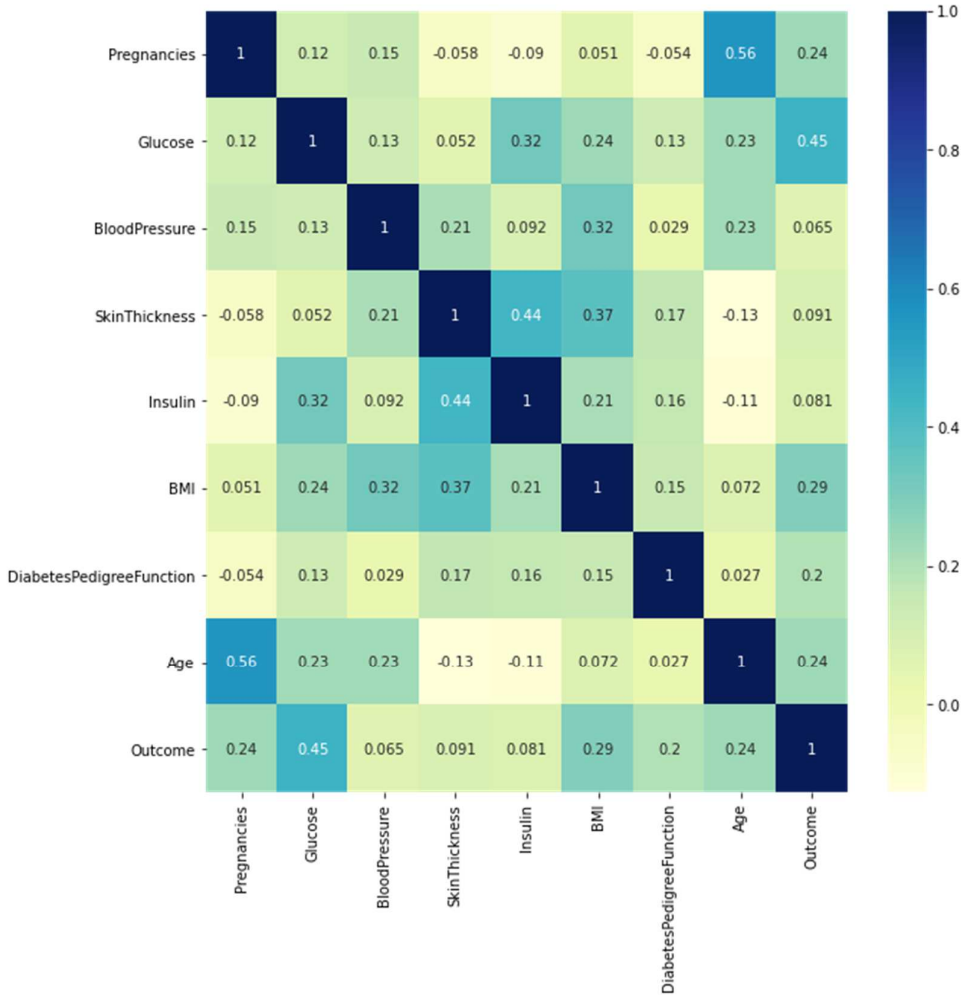

Figure 13. Heat map generated with the PIMA dataset.

The experimentation was performed with the PIMA data by optimizing initial weights assigned to parameters. The proposed model exhibited better accuracy in optimizing weights. Figure 14 presents the resultant confusion matrix obtained over data with and without weight optimization for all three recurrent neural network components. Among these, the LSTM-based architecture has outperformed in terms of classification accuracy. Correctly identifying the individual record as a diabetic patient was assumed as a True Positive (TuPR). The correctly predicting the non-diabetic patient was assumed as a True Negative (TuNR). When the model misinterpreted normal cases as diabetic cases, False Positive (FsPR) was considered. When diabetic cases were recognized as normal cases, False Negative (FsNR) was considered.

The percentage of true positives that are accurately recognized is what sensitivity analyzes. Specificity, often known as the real negative rate, is a measurement that determines the percentage of actual negative instances that are accurately classified as such. The ratio of the number of instances properly categorized to the total number of instances is called Accuracy. The F1 score is a statistic calculated by taking the harmonic mean of a classifier's accuracy and recall values and combining them into a single value. A low number of false positives and false negatives gives you an excellent F1 score. The Matthews correlation coefficient, or MCC, is a correlation coefficient that compares predicted values to actual values, which is mostly used in binary class classification problems. The weight optimization process could help evaluate the dataset more precisely as features with more significance would be considered in the evaluation process. Weights are optimized over the iteration. Resultantly, more significant features are involved in the evaluation process.

The weight optimization could yield considerable Accuracy over a conventional model. The experimental outcome presented in Table 5 shows the outcome of the proposed model concerning optimized weights.

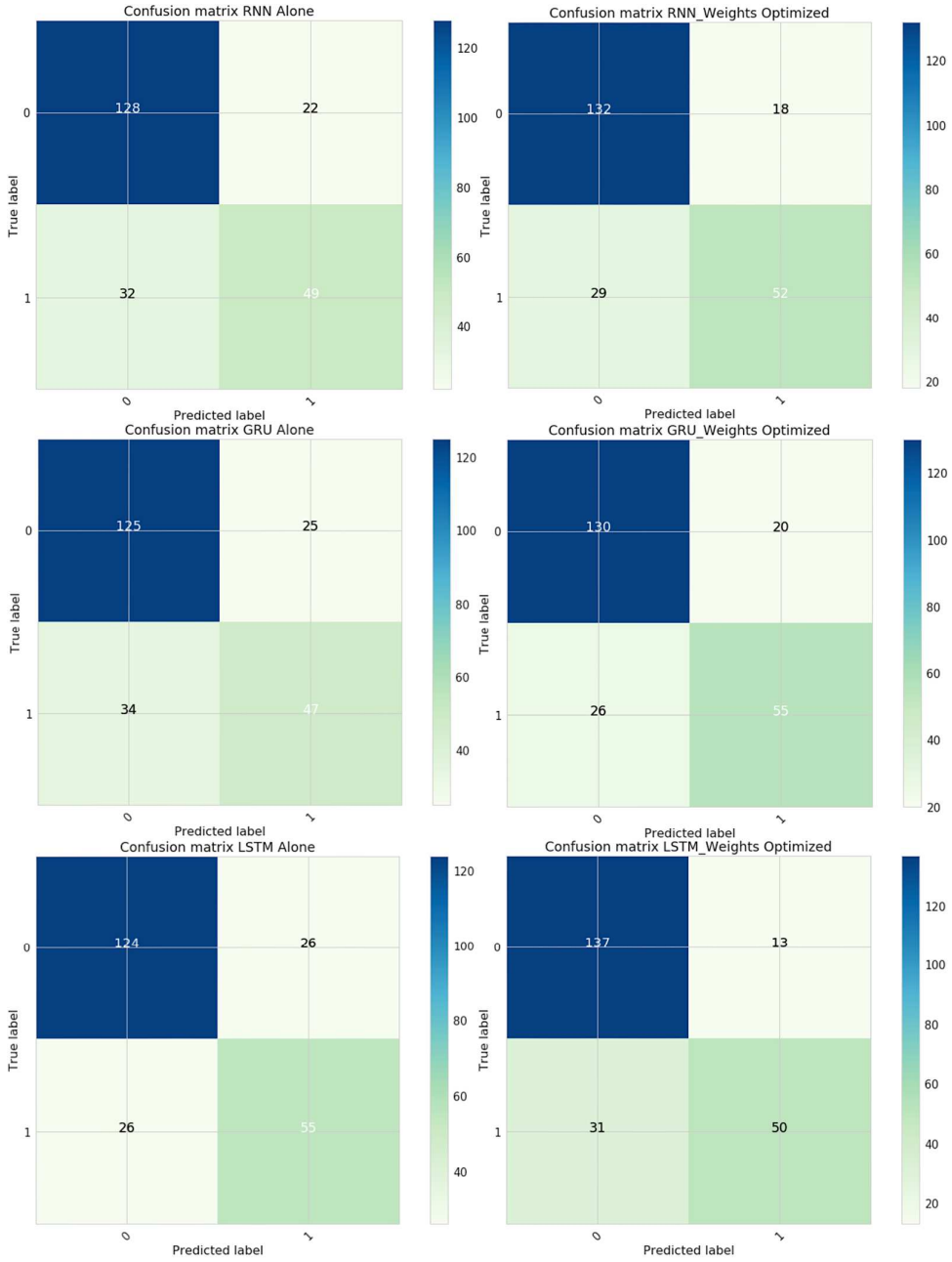

Figure 14. Confusion matrix of the proposed model.

Table 5. Performance of the proposed model with weight optimization.

	Sensitivity	Specificity	Accuracy	F1-Score	MCC
RNN Model	0.800	0.690	0.753	0.825	0.473
RNN + GRU	0.786	0.652	0.744	0.809	0.426
RNN + LSTM	0.826	0.679	0.774	0.823	0.505
RNN Model (WO)	0.819	0.742	0.796	0.848	0.541
RNN + GRU(WO)	0.833	0.733	0.800	0.849	0.558
RNN + LSTM(WO)	0.815	0.793	0.810	0.856	0.568

The classification efficiency assessment of the proposed model was compared with various existing studies concerning evaluation parameters such as sensitivity, specificity, Accuracy, and F1 score. Table 6 presents experimental values obtained by the proposed model over other existing models such as Naive Bayes, J48, Logistic Regression, K Nearest Neighbor, Random Forest, Decision Tree, REPTree, Sequential Minimal Optimization (SMO) and BayesNet. Experimental outcomes of the current model are evaluated against the outcomes of other existing models using similar datasets [77,78].

Table 6. Performance analysis of the proposed model with existing studies.

	Sensitivity	Specificity	Accuracy	F1-Score	MCC
Decision Tree	0.781	0.561	0.697	0.762	0.349
J48	0.688	0.695	0.691	0.754	0.383
K Nearest Neighbour	0.748	0.603	0.708	0.787	0.331
Logistic Regression	0.775	0.666	0.744	0.813	0.416
Naive Bayes	0.820	0.687	0.689	0.830	0.502
Random Forest	0.789	0.661	0.750	0.813	0.436
Support Vector Machine	0.775	0.666	0.744	0.813	0.416
REPTree	0.530		0.744	0.590	
SMO	0.280		0.724	0.410	
BayesNet	0.570		0.738	0.600	
RNN model	0.837	0.774	0.818	0.864	0.591

A resampling technique for evaluating the machine learning approaches is known as cross-validation, where a small data sample is considered for evaluation. The technique includes a single parameter, k which specifies how many groups are provided with sample data. The k-fold cross-validation describes the number of groups associated with the evaluation. When $k = 2$ means the model reference to 2-fold cross-validation. The formula for the cross-validation over kf folds concerning to the mean square error (MSE) is shown in Equation (33). In the current study, the accuracies of the RNN model with different auxiliary memory components are evaluated against divergent K-Values, as presented in Table 7.

$$cross\_validation_{k_f} = \frac{1}{k_f} \sum_{x=1}^{k_f} mse_x \quad (33)$$

Table 7. The Accuracy of the RNN model with auxiliary memory components against divergent K- Values.

Value of K	RNN Model	RNN + GRU	RNN + LSTM	RNN Model (WO)	RNN + GRU (WO)	RNN + LSTM (WO)
2	0.716	0.704	0.723	0.752	0.771	0.789
5	0.745	0.739	0.770	0.791	0.799	0.812
10	0.774	0.762	0.798	0.810	0.821	0.824

The ROC curve of the proposed model states the trade-off between the True Positive assumption and the False Positive assumption of the proposed model concerning predictions made with the Pima diabetic dataset. Figure 15 presents the ROC of the proposed model with optimized weights using tabular data. The proposed model has shown the classification's desired performance concerns.

The present model was trained with limited genomic data or PIMA diabetic dataset. Either of these datasets consisted of approximately 700 records. Of them, 30% of the overall data were meant for testing, resulting in training the model with inadequate records that could impact its performance. RNN-based models in various applications have exhibited noticeable accuracies. However, the neural network model's Accuracy can vary depending on the ratio of the training sample to the testing sample.

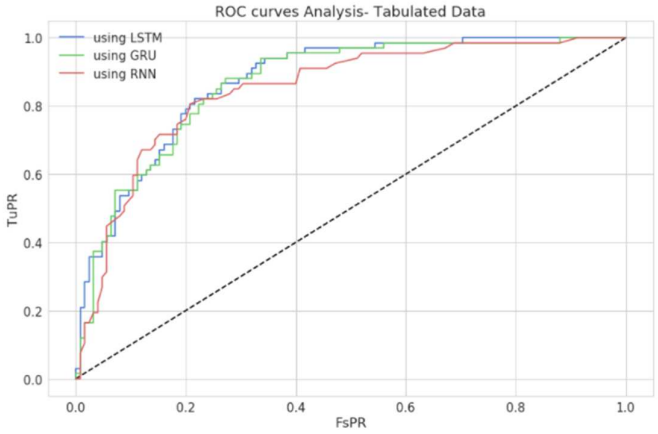

Figure 15. ROC curve of the proposed model using PIMA data.

4.3. Practical Implications

The proposed technique for forecasting type-2 diabetes can be implemented over a mobile framework with a front-end module. Patients and practitioners can perform the initial assessment of the illness. Users can provide details such as glucose levels, pregnancies, insulin, hypertension, BMI, Diabetes Pedigree, skin thickness, and heart rate. Based on the provided input and the trained data, the model can analyze the input with the trained data for predicting the illness. The model can be implemented in the iOS platform to the back-end Kaggle using the back-end service such as the MBaaS component. A secured socket layer (SSL) and two-factor authentication can ensure the security of the model [79,80].

Images of the user interface of the application model are presented in Figure 16. The leftmost image represents the registration page of the application. The middle image shows the user information page, followed by the resultant prediction screen of the model. The model makes the task of predicting a future illness more convenient. The model can be improved by incorporating the genomic module for accepting gene data and evaluating the illness based on gene information.

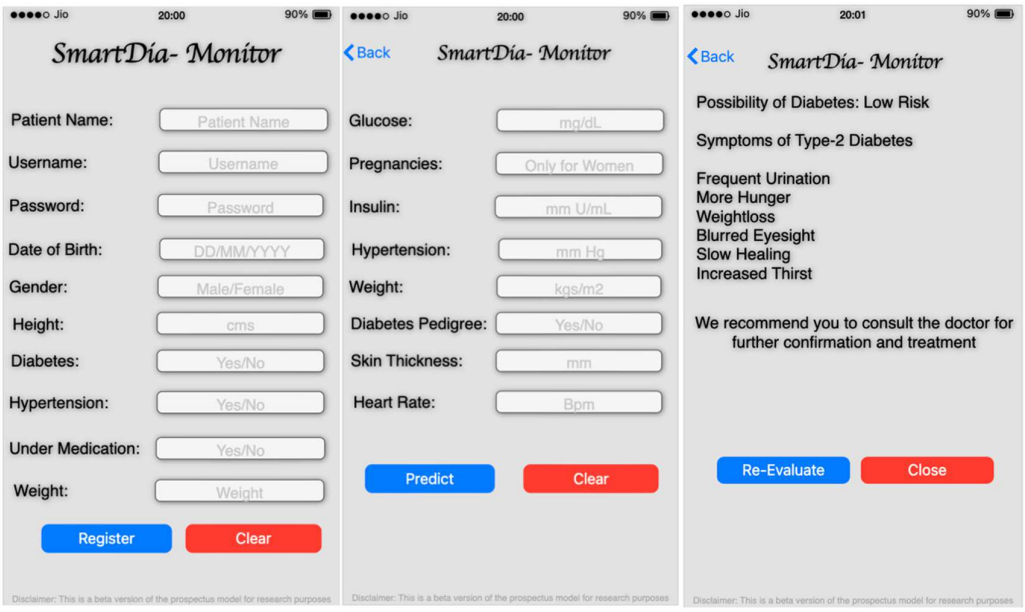

Figure 16. Images presenting the mobile interface of the future implication model.

The future implication model was inspired by a non-invasive way of assessing future illness more precisely. The assessment could help physicians and individuals better

evaluation of health conditions. The prediction could assist individuals in adopting better life standards and living habits to avoid or prolong the chance of being affected by the illness [81]. The computationally efficient approaches like the MobileNet V2 and MobileNet V3 architectures would assist better in deploying the models in lightweight computational devices. The RNN models need tremendous computation efforts despite providing highly accurate performances.

5. Conclusions

The genomic-based future illness prediction is a path-breaking approach for precisely assessing future illness. Genomic-based data can be conveniently analyzed through supervised-based approaches such as Neural Network models. The sequence of GENE can be downsampled and analyzed based on the weight concerning the diseased sequence. The approach decomposes a large GENE sequence into smaller GENE strings, raising the chances of accurate matching with diseased sequences and resulting in a precise prediction. Although there might be a considerable burden on the machine to decompose the large genomic patterns, decomposing them to a certain predetermined extent has better Accuracy than conventional approaches. When the proposed model was evaluated over the PIMA diabetic dataset, it exhibited a reasonable performance in predicting type-2 diabetes. The PIMA dataset consists of 768 records, of which only 537 are used for training. The Accuracy would be much better when more records are for training purposes. Statistical analysis for disease progression [82] has exhibited better performance than the existing models when adequate data are available. Users may access the suggested model's prediction result via the Android application. As a result, it is desired to give an effective method for determining the possibility of being affected by diabetes at an early stage. However, it is exceedingly challenging to work with larger sequence gene data, the quantum and federated learning techniques would effectively handle such a larger sequence data. On the other side, when dealing with tabular data, the ensemble classification models would yield almost identical performance with minimal computation to the suggested RNN models.

Although the proposed approach showed promising results, it was challenging when decomposing to a more significant extent. In such situations, incorporating Long Short-Term Memory (LSTM) can make the approach more robust with considerably lesser computational latency. For handling an unusual illness, self-Learning Based algorithms and the use of cognitive technology would be appropriate to minimize the steps needed for training the algorithm [83]. The proposed approach based on Genomics with Self-Learning algorithms might result in better results than supervisory approaches alone. In future work, comparison with other smart diagnosis techniques and assessment of other clinical datasets need to be performed. Once the model validation is performed with more datasets, other risk factors affecting diabetes can be revealed. The future dimensions of the research include the deep learning-driven pattern recognition models for analyzing the gene sequences for identifying the possible future illness and developing mobile applications that can generalize the information from the genomic data. However, there is great demand for explainable Artificial Intelligence models that are interpretable in decision-making.

Author Contributions: The authors contributions are as follows, Conceptualization of the study, P.N.S., M.F.I. and J.S.; Formal analysis of the study, T.B.K., J.S. and S.P.P.; Formal Investigation, S.P.P., C.N.S. and P.N.S.; Methodology, M.F.I., S.P.P. and T.B.K.; Project administration, C.N.S., P.N.S. and M.F.I.; Resources acquisition, P.N.S., C.N.S. and M.F.I.; Software, J.S., C.N.S., T.B.K. and S.P.P.; Writing—original draft, P.N.S., J.S. and M.F.I.; and Writing—review and editing, M.F.I.; All authors have checked and approved the final version of the work. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: Parvathaneni Naga Srinivasu would like to thank the management of Prasad V Potluri Siddhartha Institute of technology for their support in carrying out the current study. Jana Shafi would like to thank the Deanship of Scientific Research, Prince Sattam bin Abdul Aziz University, for supporting this work.

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

References

1. Zou, Q.; Qu, K.; Luo, Y.; Yin, D.; Ju, Y.; Tang, H. Predicting Diabetes Mellitus with Machine Learning Techniques. *Front. Genet.* **2018**, *9*, 515. [CrossRef] [PubMed]
2. Hemu, A.A.; Mim, R.B.; Ali, M.; Nayer, M.; Ahmed, K.; Bui, F.M. Identification of Significant Risk Factors and Impact for ASD Prediction among Children Using Machine Learning Approach. In Proceedings of the 2022 Second International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), Bhilai, India, 21–22 April 2022; pp. 1–6. [CrossRef]
3. Ravaut, M.; Sadeghi, H.; Leung, K.K.; Volkovs, M.; Rosella, L.C. Diabetes mellitus forecasting using population health data in Ontario, Canada. *arXiv* **2019**, arXiv:1904.04137.
4. Deberneh, H.; Kim, I. Prediction of Type 2 Diabetes Based on Machine Learning Algorithm. *Int. J. Environ. Res. Public Health* **2021**, *18*, 3317. [CrossRef] [PubMed]
5. Arshad, A.; Khan, Y.D. DNA Computing: A Survey. In Proceedings of the 2019 International Conference on Innovative Computing (ICIC), Lahore, Pakistan, 1–2 November 2019; pp. 1–5. [CrossRef]
6. Cho, Y.S.; Chen, C.-H.; Hu, C.; Long, J.; Ong, R.T.H.; Sim, X.; Takeuchi, F.; Wu, Y.; Go, M.J.; et al.; DIAGRAM Consortium. Meta-analysis of genome-wide association studies identifies eight new loci for type 2 diabetes in east Asians. *Nat. Genet.* **2011**, *44*, 67–72. [CrossRef] [PubMed]
7. The Coronary Artery Disease (C4D) Genetics Consortium. A genome-wide association study in Europeans and South Asians identifies five new loci for coronary artery disease. *Nat. Genet.* **2011**, *43*, 339–344. [CrossRef]
8. Duncan, L.; Shen, H.; Gelaye, B.; Meijsen, J.; Ressler, K.; Feldman, M.; Peterson, R.; Domingue, B. Analysis of polygenic risk score usage and performance in diverse human populations. *Nat. Commun.* **2019**, *10*, 3328. [CrossRef]
9. Jordan, D.M.; Do, R. Using Full Genomic Information to Predict Disease: Breaking Down the Barriers Be-tween Complex and Mendelian Diseases. *Annu. Rev. Genom. Hum. Genet.* **2018**, *19*, 289–301. [CrossRef]
10. Rahaman, A.; Ali, M.; Ahmed, K.; Bui, F.M.; Mahmud, S.M.H. Performance Analysis between YOLOv5s and YOLOv5m Model to Detect and Count Blood Cells: Deep Learning Approach. In Proceedings of the 2nd International Conference on Computing Advancements (ICCA'22). Association for Computing Machinery, Dhaka, Bangladesh, 10–12 March 2022; pp. 316–322. [CrossRef]
11. Ontor, Z.H.; Ali, M.; Ahmed, K.; Bui, F.M.; Al-Zahrani, F.A.; Mahmud, S.M.H.; Azam, S. Early-Stage Cervical Cancerous Cell Detection from Cervix Images Using YOLOv5. *Comput. Mater. Contin.* **2023**, *74*, 3727–3741. [CrossRef]
12. So, H.-C.; Sham, P.C. Exploring the predictive power of polygenic scores derived from genome-wide association studies: A study of 10 complex traits. *Bioinformatics* **2016**, *33*, 886–892. [CrossRef]
13. Sarra, R.R.; Dinar, A.M.; Mohammed, M.A.; Ghani, M.K.A.; Albahar, M.A. A Robust Framework for Data Generative and Heart Disease Prediction Based on Efficient Deep Learning Models. *Diagnostics* **2022**, *12*, 2899. [CrossRef]
14. Ali, M.; Ahmed, K.; Bui, F.M.; Paul, B.K.; Ibrahim, S.M.; Quinn, J.M.; Moni, M.A. Machine learning-based statistical analysis for early stage detection of cervical cancer. *Comput. Biol. Med.* **2021**, *139*, 104985. [CrossRef]
15. Ali, M.M.; Paul, B.K.; Ahmed, K.; Bui, F.M.; Quinn, J.M.W.; Moni, M.A. Heart disease prediction using supervised machine learning algorithms: Performance analysis and comparison. *Comput. Biol. Med.* **2021**, *136*, 104672. [CrossRef]
16. Bell, C.G.; Teschendorff, A.E.; Rakyan, V.K.; Maxwell, A.P.; Beck, S.; Savage, D.A. Genome-wide DNA methylation analysis for diabetic nephropathy in type 1 diabetes mellitus. *BMC Med. Genom.* **2010**, *3*, 33. [CrossRef]
17. Konishi, T.; Matsukuma, S.; Fuji, H.; Nakamura, D.; Satou, N.; Okano, K. Principal Component Analysis applied directly to Sequence Matrix. *Sci. Rep.* **2019**, *9*, 19297. [CrossRef]
18. Mallik, S.; Mukhopadhyay, A.; Maulik, U.; Bandyopadhyay, S. Integrated analysis of gene expression and genome-wide DNA methylation for tumor prediction: An association rule mining-based approach. In Proceedings of the 2013 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), Singapore, 16–19 April 2013; pp. 120–127. [CrossRef]
19. Mallik, S.; Bhadra, T.; Mukherji, A. DTFP-Growth: Dynamic Threshold-Based FP-Growth Rule Mining Algorithm Through Integrating Gene Expression, Methylation, and Protein–Protein Interaction Profiles. *IEEE Trans. NanoBiosci.* **2018**, *17*, 117–125. [CrossRef]
20. Huang, S.; Cai, N.; Pacheco, P.P.; Narandes, S.; Wang, Y.; Xu, W. Applications of Support Vector Machine (SVM) Learning in Cancer Genomics. *Cancer Genom.-Proteom.* **2018**, *15*, 41–51. [CrossRef]
21. Parry, R.M.; Jones, W.; Stokes, T.H.; Phan, J.H.; Moffitt, R.; Fang, H.; Shi, L.; Oberthuer, A.; Fischer, M.; Tong, W.; et al. k-Nearest neighbor models for microarray gene expression analysis and clinical outcome prediction. *Pharmacogenom. J.* **2010**, *10*, 292–309. [CrossRef]

22. Wright, M.N.; Ziegler, A. Ranger: A fast implementation of random forests for high dimensional data in C++ and R. *J. Stat. Softw.* **2017**, *77*, 1–17. [CrossRef]
23. Nagaraj, P.; Deepalakshmi, P.; Ijaz, M.F. Optimized adaptive tree seed Kalman filter for a diabetes recommen-dation system—Bilevel performance improvement strategy for healthcare applications. In *Intelligent Data-Centric Systems, Cognitive and Soft Computing Techniques for the Analysis of Healthcare Data*; Academic Press: Cambridge, MA, USA, 2022; pp. 191–202.
24. Mantzaris, D.H.; Anastassopoulos, G.C.; Lymberopoulos, D.K. Medical disease prediction using Artificial Neural Networks. In Proceedings of the 8th IEEE International Conference on BioInformatics and BioEngineering, Athens, Greece, 8–10 October 2008; pp. 1–6. [CrossRef]
25. Huang, P.-J.; Chang, J.-H.; Lin, H.-H.; Li, Y.-X.; Lee, C.-C.; Su, C.-T.; Li, Y.-L.; Chang, M.-T.; Weng, S.; Cheng, W.-H.; et al. DeepVariant-on-Spark: Small-Scale Genome Analysis Using a Cloud-Based Computing Framework. *Comput. Math. Methods Med.* **2020**, *2020*, 7231205. [CrossRef]
26. Koumakis, L. Deep learning models in genomics; are we there yet? *Comput. Struct. Biotechnol. J.* **2020**, *18*, 1466–1473. [CrossRef]
27. Van Dam, S.; Võsa, U.; Van Der Graaf, A.; Franke, L.; De Magalhães, J.P. Gene co-expression analysis for functional classification and gene–disease predictions. *Brief. Bioinform.* **2018**, *19*, 575–592. [CrossRef] [PubMed]
28. Travnik, J.B.; Mathewson, K.W.; Sutton, R.S.; Pilarski, P.M. Reactive Reinforcement Learning in Asynchronous Environments. *Front. Robot. AI* **2018**, *5*, 79. [CrossRef] [PubMed]
29. Battineni, G.; Sagaro, G.G.; Chinatalapudi, N.; Amenta, F. Applications of Machine Learning Predictive Models in the Chronic Disease Diagnosis. *J. Pers. Med.* **2020**, *10*, 21. [CrossRef] [PubMed]
30. Yue, C.; Xin, L.; Kewen, X.; Chang, S. An Intelligent Diagnosis to Type 2 Diabetes Based on QPSO Algorithm and WLS-SVM. In Proceedings of the 2008 International Symposium on Intelligent Information Technology Application Workshops, Shanghai, China, 21–22 December 2008; pp. 117–121. [CrossRef]
31. Srinivasu, P.N.; Rao, T.S.; Dicu, A.M.; Mnerie, C.A.; Olariu, I. A comparative review of optimisation techniques in segmentation of brain MR images. *J. Intell. Fuzzy Syst.* **2020**, *38*, 6031–6043. [CrossRef]
32. Nadesh, R.K.; Arivuselvan, K. Type 2: Diabetes mellitus prediction using Deep Neural Networks classifier. *Int. J. Cogn. Comput. Eng.* **2020**, *1*, 55–61. [CrossRef]
33. Abedini, M.; Bijari, A.; Banirostam, T. Classification of Pima Indian Diabetes Dataset using Ensemble of Decision Tree, Logistic Regression and Neural Network. *Int. J. Adv. Res. Comput. Commun. Eng.* **2020**, *9*, 1–4. [CrossRef]
34. Kundu, N.; Rani, G.; Dhaka, V.S.; Gupta, K.; Nayak, S.C.; Verma, S.; Ijaz, M.F.; Woźniak, M. IoT and Interpretable Machine Learning Based Framework for Disease Prediction in Pearl Millet. *Sensors* **2021**, *21*, 5386. [CrossRef]
35. Reddy, G.S.; Chittineni, S. Entropy based C4.5-SHO algorithm with information gain optimization in data mining. *PeerJ Comput. Sci.* **2021**, *7*, e424. [CrossRef]
36. Luukka, P. Feature selection using fuzzy entropy measures with similarity classifier. *Expert Syst. Appl.* **2011**, *38*, 4600–4607. [CrossRef]
37. Szmidt, E.; Kacprzyk, J. Some Problems with Entropy Measures for the Atanassov Intuitionistic Fuzzy Sets. In *Applications of Fuzzy Theory, Proceedings of the WILF 2007, Camogli, Italy, 7–10 July 2007*; Lecture Notes in Computer Science; Masulli, F., Mitra, S., Pasi, G., Eds.; Springer: Berlin/Heidelberg, Germany, 2007; Volume 4578. [CrossRef]
38. Choubey, D.K.; Paul, S. GA_RBF NN: A classification system for diabetes. *Int. J. Biomed. Eng. Technol.* **2017**, *23*, 71–93. [CrossRef]
39. Jackins, V.; Vimal, S.; Kaliappan, M.; Lee, M.Y. AI-based smart prediction of clinical disease using random forest classifier and Naive Bayes. *J. Supercomput.* **2020**, *77*, 5198–5219. [CrossRef]
40. Almustafa, K.M. Prediction of heart disease and classifiers' sensitivity analysis. *BMC Bioinform.* **2020**, *21*, 278. [CrossRef] [PubMed]
41. Tayeb, S.; Pirouz, M.; Sun, J.; Hall, K.; Chang, A.; Li, J.; Song, C.; Chauhan, A.; Ferra, M.; Sager, T.; et al. Toward predicting medical conditions using k-nearest neighbors. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 11–14 December 2017; pp. 3897–3903. [CrossRef]
42. Xu, W.; Zhao, Y.; Nian, S.; Feng, L.; Bai, X.; Luo, X.; Luo, F. Differential analysis of disease risk assessment using binary logistic regression with different analysis strategies. *J. Int. Med. Res.* **2018**, *46*, 3656–3664. [CrossRef]
43. Wei, W.; Visweswaran, S.; Cooper, G.F. The application of naive Bayes model averaging to predict Alzheimer's disease from genome-wide data. *J. Am. Med. Inform. Assoc.* **2011**, *18*, 370–375. [CrossRef]
44. Benbelkacem, S.; Atmani, B. Random Forests for Diabetes Diagnosis. In Proceedings of the 2019 International Conference on Computer and Information Sciences (ICCIS), Sakaka, Saudi Arabia, 10–11 April 2019; pp. 1–4. [CrossRef]
45. Son, Y.-J.; Kim, H.-G.; Kim, E.-H.; Choi, S.; Lee, S.-K. Application of Support Vector Machine for Prediction of Medication Adherence in Heart Failure Patients. *Health Inform. Res.* **2010**, *16*, 253–259. [CrossRef]
46. Ghaheri, A.; Shoar, S.; Naderan, M.; Hoseini, S.S. The Applications of Genetic Algorithms in Medicine. *Oman Med. J.* **2015**, *30*, 406–416. [CrossRef]
47. Swapna, G.; Soman, K.P.; Vinayakumar, R. Diabetes Detection Using ECG Signals: An Overview. In *Deep Learning Techniques for Biomedical and Health Informatics*; Studies in Big Data; Dash, S., Acharya, B., Mittal, M., Abraham, A., Kelemen, A., Eds.; Springer: Cham, Switzerland, 2019; Volume 68. [CrossRef]

48. Available online: https://www.forrester.com/webinar/AI+Software+Market+Sizing+Understand+Forresters+Four+Segments+To+Invest+Wisely/-/E-WEB32605?utm_source=prnewswire&utm_medium=pr&utm_campaign=cio20 (accessed on 7 October 2021).
49. Mooney, M.A.; Wilmot, B. Gene set analysis: A step-by-step guide. *Am. J. Med. Genet. Part B Neuropsychiatr. Genet.* **2015**, *168*, 517–527. [CrossRef]
50. Mathur, R.; Rotroff, D.; Ma, J.; Shojaie, A.; Motsinger-Reif, A. Gene set analysis methods: A systematic comparison. *BioData Min.* **2018**, *11*, 8. [CrossRef]
51. Leevy, J.L.; Khoshgoftaar, T.M.; Villanustre, F. Survey on RNN and CRF models for de-identification of medical free text. *J. Big Data* **2020**, *7*, 73. [CrossRef]
52. Yadav, S.S.; Jadhav, S.M. Deep convolutional neural network based medical image classification for disease diagnosis. *J. Big Data* **2019**, *6*, 113. [CrossRef]
53. SivaSai, J.G.; Srinivasu, P.N.; Sindhuri, M.N.; Rohitha, K.; Deepika, S. An Automated Segmentation of Brain MR Image through Fuzzy Recurrent Neural Network. In *Bio-Inspired Neurocomputing*; Studies in Computational Intelligence; Bhoi, A., Mallick, P., Liu, C.M., Balas, V., Eds.; Springer: Singapore, 2020; Volume 903. [CrossRef]
54. Ahmed, S.; Srinivasu, P.N.; Alhumam, A.; Alarfaj, M. AAL and Internet of Medical Things for Monitoring Type-2 Diabetic Patients. *Diagnostics* **2022**, *12*, 2739. [CrossRef] [PubMed]
55. Kozomara, A.; Birgaoanu, M.; Griffiths-Jones, S. miRBase: From microRNA sequences to function. *Nucleic Acids Res.* **2019**, *47*, D155–D162. [CrossRef] [PubMed]
56. Guan, Z.-X.; Li, S.-H.; Zhang, Z.-M.; Zhang, D.; Yang, H.; Ding, H. A Brief Survey for MicroRNA Precursor Identification Using Machine Learning Methods. *Curr. Genom.* **2020**, *21*, 11–25. [CrossRef] [PubMed]
57. Hira, Z.M.; Gillies, D.F. A Review of feature selection and future extraction methods applied on microarray data. *Adv. Bioinform.* **2015**, *2015*, 198363. [CrossRef] [PubMed]
58. Shirzad, M.B.; Keyvanpour, M.R. A feature selection method based on minimum redundancy maximum relevance for learning to rank. In Proceedings of the 2015 AI & Robotics (IRANOPEN), Qazvin, Iran, 12–12 April 2015; pp. 1–5. [CrossRef]
59. Fang, H.; Tang, P.; Si, H. Feature Selections Using Minimal Redundancy Maximal Relevance Algorithm for Human Activity Recognition in Smart Home Environments. *J. Health Eng.* **2020**, *2020*, 8876782. [CrossRef]
60. Carrara, F.; Elias, P.; Sedmidubsky, J.; Zezula, P. LSTM-based real-time action detection and prediction in human motion streams. *Multimed. Tools Appl.* **2019**, *78*, 27309–27331. [CrossRef]
61. Che, Z.; Purushotham, S.; Cho, K.; Sontag, D.; Liu, Y. Recurrent Neural Networks for Multivariate Time Series with Missing Values. *Sci. Rep.* **2018**, *8*, 6085. [CrossRef]
62. Srinivasu, P.N.; JayaLakshmi, G.; Jhaveri, R.H.; Praveen, S.P. Ambient Assistive Living for Monitoring the Physical Activity of Diabetic Adults through Body Area Networks. *Mob. Inf. Syst.* **2022**, *2022*, 3169927. [CrossRef]
63. Wu, L.; Kong, C.; Hao, X.; Chen, W. A Short-Term Load Forecasting Method Based on GRU-CNN Hybrid Neural Network Model. *Math. Probl. Eng.* **2020**, *2020*, 1428104. [CrossRef]
64. Swapna, G.; Soman, K.; Vinayakumar, R. Automated detection of diabetes using CNN and CNN-LSTM network and heart rate signals. *Procedia Comput. Sci.* **2018**, *132*, 1253–1262.
65. Srinivasu, P.N.; SivaSai, J.G.; Ijaz, M.F.; Bhoi, A.K.; Kim, W.; Kang, J.J. Classification of skin disease using deep learning neural networks with MobileNet V2 and LSTM. *Sensors* **2021**, *21*, 2852. [CrossRef]
66. Ijaz, M.F.; Alfian, G.; Syafrudin, M.; Rhee, J. Hybrid Prediction Model for Type 2 Diabetes and Hypertension Using DBSCAN-Based Outlier Detection, Synthetic Minority Over Sampling Technique (SMOTE), and Random Forest. *Appl. Sci.* **2018**, *8*, 1325. [CrossRef]
67. Zhang, L.; Wang, Y.; Niu, M.; Wang, C.; Wang, Z. Machine learning for characterizing risk of type 2 diabetes mellitus in a rural Chinese population: The Henan Rural Cohort Study. *Sci. Rep.* **2020**, *10*, 4406. [CrossRef]
68. Naz, H.; Ahuja, S. Deep learning approach for diabetes prediction using PIMA Indian dataset. *J. Diabetes Metab. Disord.* **2020**, *19*, 391–403. [CrossRef]
69. Hertzog, M.I.; Correa, U.B.; Araujo, R.M. SpreadOut: A Kernel Weight Initializer for Convolutional Neural Networks. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; pp. 1–7. [CrossRef]
70. Yang, B.; Urtasun, R. Learning to Reweight Examples for Robust Deep Learning. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018.
71. Mitra, A.; Mohanty, D.; Ijaz, M.F.; Rana, A.u.H.S. Deep Learning Approach for Object Features Detection. In *Advances in Communication, Devices, and Networking*; Lecture Notes in Electrical Engineering; Dhar, S., Mukhopadhyay, S.C., Sur, S.N., Liu, C.M., Eds.; Springer: Singapore, 2022; Volume 776.
72. Pranto, B.; Mehnaz, S.M.; Mahid, E.B.; Sadman, I.M.; Rahman, A.; Momen, S. Evaluating Machine Learning Methods for Predicting Diabetes among Female Patients in Bangladesh. *Information* **2020**, *11*, 374. [CrossRef]
73. Lai, H.; Huang, H.; Keshavjee, K.; Guergachi, A.; Gao, X. Predictive models for diabetes mellitus using machine learning techniques. *BMC Endocr. Disord.* **2019**, *19*, 101. [CrossRef]
74. Web-Based Data-Science Environment. Available online: https://www.kaggle.com/ (accessed on 8 January 2022).

75. Ontor, Z.H.; Ali, M.; Hossain, S.S.; Nayer., M.; Ahmed, K.; Bui, F.M. YOLO_CC: Deep Learning based Approach for Early Stage Detection of Cervical Cancer from Cervix Images Using YOLOv5s Model. In Proceedings of the 2022 Second International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), Bhilai, India, 21–22 April 2022; pp. 1–5. [CrossRef]
76. Srinivasu, P.N.; Rao, T.S.; Balas, V.E. A systematic approach for identification of tumor regions in the human brain through HARIS algorithm. In *Deep Learning Techniques for Biomedical and Health Informatics*; Academic Press: Cambridge, MA, USA, 2020; pp. 97–118. [CrossRef]
77. Tigga, N.P.; Garg, S. Prediction of Type 2 Diabetes using Machine Learning Classification Methods. *Procedia Comput. Sci.* **2020**, *167*, 706–716. [CrossRef]
78. Larabi-Marie-Sainte, S.; Aburahmah, L.; Almohaini, R.; Saba, T. Current Techniques for Diabetes Prediction: Review and Case Study. *Appl. Sci.* **2019**, *9*, 4604. [CrossRef]
79. Ijaz, M.F.; Attique, M.; Son, Y. Data-Driven Cervical Cancer Prediction Model with Outlier Detection and Over-Sampling Methods. *Sensors* **2020**, *20*, 2809. [CrossRef]
80. Vulli, A.; Srinivasu, P.N.; Sashank, M.S.K.; Shafi, J.; Choi, J.; Ijaz, M.F. Fine-Tuned DenseNet-169 for Breast Cancer Metastasis Prediction Using FastAI and 1-Cycle Policy. *Sensors* **2022**, *22*, 2988. [CrossRef]
81. Chae, S.; Kwon, S.; Lee, D. Predicting Infectious Disease Using Deep Learning and Big Data. *Int. J. Environ. Res. Public Health* **2018**, *15*, 1596. [CrossRef] [PubMed]
82. Pinto, M.F.; Oliveira, H.; Batista, S.; Cruz, L.; Pinto, M.; Correia, I.; Martins, P.; Teixeira, C. Prediction of disease progression and outcomes in multiple sclerosis with machine learning. *Sci. Rep.* **2020**, *10*, 21038. [CrossRef] [PubMed]
83. Srinivasu, P.N.; Balas, V.E. Self-Learning Network-based segmentation for real-time brain M.R. images through HARIS. *PeerJ Comput. Sci.* **2021**, *7*, e654. [CrossRef]