Design Simulation and Assessment of Prediction of Mortality in Intensive Care Unit Using Intelligent Algorithms

Satish Kumar Alaria[1], Vivek Sharma[2], Ashish Raj[3], Vijay Kumar[4]
[1]Assistant Professor, Department of Computer Science & Engineering, Arya Institute of Engineering & Technology, Jaipur, Rajasthan (India)
[2]Assistant Professor, Department of Computer & Communication Engineering, Manipal University Jaipur, Rajasthan (India)
[3]Assistant Professor, Department of Electrical Engineering, Poornima University, Jaipur, Rajasthan (India)
[4]Professor, Department of Computer Science & Engineering, Arya Institute of Engineering & Technology, Jaipur, Rajasthan (India)
Email: Satish.alaria@gmail.com[1]

Article Info
Page Number: 355 - 367
Publication Issue:
Vol 71 No. 2 (2022)

Abstract
Big data in healthcare refers to vast amounts of data generated by the adoption of digital technology that collect patient records and aid in the management of hospital performance, which would otherwise be too large and complex for traditional technologies. We have chosen Mortality Prediction in Hospital ICU for this paper. In recent years, there has been a determined push in hospitals to implement digital health record systems. Between 2008 and 2014, the number of non-federal acute care hospitals in the United States using basic digital systems climbed from 9.4 percent to 75.5 percent. In the near future, the percentage increase in the hospital's digital information will be enormous, and we will need to apply modern big data techniques to analyse those datasets. In the best sense, the future of medical diagnosis and therapy is a combination of medical judgement and an algorithmic diagnostic tool based on large medical information. For our research, we used the MIMIC III database. In this paper, we will go over why Mortality Prediction in ICU is interesting, as well as the method we used to examine and address the problem.

Article History
Article Received*: 05 January 2022*
Revised*: 20 February 2022*
Accepted*: 04 April 2022*
Publication*: 09 May 2022*

Keywords: Big Data Analytics, Healthcare, Machine Learning, ANN, CNN.

1. INTRODUCTION

Patients hospitalised to an intensive care unit (ICU) often have a critical emergency disease and are at great risk of dying. As a result, they must be treated and evaluated on a regular basis to ensure that they recover. Heart monitors, pulse oximeters, ventilators, catheters, arterial lines, and other gadgets in today's ICUs keep track of a patient's health records on a continuous basis [1]. If there are any unusual fluctuations, the nurse or doctor will be alerted promptly. Data from these devices, as well as data from several patients, will be extremely beneficial in estimating the likelihood of mortality. Estimating the risk of mortality in healthcare can be highly important when it comes to sorting and distributing hospital resources and identifying the proper levels of care required. The doctors can also explain the expected outcomes with the patient's family early on. In this work, we will look through the MIMIC III [2] database for situations where various machine learning models and methodologies have been utilised to determine diagnoses of patients brought to the ICU with cardiovascular illness. The analytical results can be generalised and used to a broader scope of disorders, despite the fact that the study was conducted on a single example.

2. Literature Survey

Several academics have developed several machine-learning techniques to predict death. The use of an ensemble technique called the Super Learner (SL) to improve the performance of mortality prediction was one of these studies. Super Learner estimates the performance of multiple machine-learning models using cross-validation and provides an ideal weighted average. This study's findings obtained a prediction testing accuracy score of roughly 95 percent ([Pirracchio Romain et.al, [3]). A tailored mortality prediction was made in another research investigation by examining similar historical patients. First, a death count among similar patients was recorded, followed by logistic regression and a decision tree. The values of AUROC and AUPRC using logistic regression were 0.830 and 0.474, respectively. The findings of AUROC and AUPRC using a decision tree were 0.753 and 0.347, respectively (Lee, Joon, et.al, [4]). We plan to take a similar strategy, distinguishing between comparable and non-similar patients based on their feature values. Another study used multivariate logistic regression, and the findings were AUCROC = 0.85 and SMR = 1.25 (Fika, Sofia, et.al, [5]). Multiple independent factors are taken into account in multivariate logistic regression to predict the dependent variable. In most cases, severity of sickness (SOI) scores are used to predict mortality. A patient's physiological decompensation (organ system failures) is referred to as SOI (Dybowski R et.al, [9]). On the validation set, SOI scores tend to underperform. A neural network was used to predict death in another research investigation. With an AUCROC score of 0.8445 for 30 days mortality and 0.86 for hospital mortality, the neural network performed better (Zahid, M.A, [6]). Nonparametric techniques, such as neural networks or data mining, have been used to predict hospital mortality in ICU patients in several studies (M. Ghassemi et.al, [10]). Given that software for health care systems should provide these features out of the box, speed, high availability, stability, and scale are essential technological qualities of the technology stack to be used. Spark is one of the most promising big data analytics engines for data scientists that has emerged with those characteristics. Apache Spark's true power and worth comes in its ability to do data science jobs quickly and accurately. The ability to mix ETL, batch analytics, real-time stream analysis, machine learning, graph processing, and visualisations is one of Spark's selling points (Hrayr Harutyunyan et.al, [7]). Python, on the other hand, became the programming language of choice after receiving widespread backing from the scientific community. The language serves as a seamless link between the data processing tool and the machine learning framework of choice. With the right tooling for development, libraries like NumPy, Pandas, and pyspark made the process faster and more straightforward (McKinney, [8]).

The adoption rate and research development in medical field is still hindered by some fundamental problems inherent within the big data paradigm. This work, author discuss some of these major challenges with a focus on three upcoming and promising areas of medical research: image, signal, and genomics based analytics. Recent research which targets utilization of large volumes of medical data while combining multimodal data from disparate sources is discussed. Potential areas of research within this field which have the ability to provide meaningful impact on healthcare delivery are also examined (Ashwin Belle et.al, [12]). This review focus to disseminate some novel approaches to address challenges in medical research. More specifically, approaches ranging from efficient methods of processing large clinical data to predictive models that could generate better predictions from healthcare data are presented (S. Koch et.al, [13]. There are various challenges associated with each step of handling big data which can only be surpassed by using high-end computing solutions for big data analysis. That is why, to provide relevant solutions for improving public health, healthcare providers are required to be fully equipped with appropriate infrastructure to systematically generate and analyze big data. An efficient management, analysis, and interpretation of big data can change the game by opening new avenues for modern healthcare (S. Dash et.al, [14]).

The concept of Big Data and associated analytics are to be taken seriously when approaching the use of vast volumes of both structured and unstructured data in science and health-care. Future exploration of issues surrounding data privacy, confidentiality, and education are needed. A greater focus on data from social media, the quantified self-movement, and the application of analytics to “small data” would also be useful (M. M. Hansen et.al, [15]). This research first introduced the evolution of cognitive computing from four aspects, i.e., knowledge discovery, cognitive science, big data and cognitive computing. Then, the cognitive computing system architecture is proposed which consists of three parts, i.e., IoT, big data analysis and cloud computing. Furthermore, we introduce the enabling technologies in cognitive computing including reinforcement learning and deep learning. Finally, the representative applications of cognitive computing are illustrated from three scenarios, i.e., robot technology, emotion communication system and medical cognitive system (M. Chen et.al, [16]). This review study unveils that there is a paucity of information on evidence of real-world use of Big Data analytics in healthcare. This is because, the usability studies have considered only qualitative approach which describes potential benefits but does not take into account the quantitative study. Also, majority of the studies were from developed countries which brings out the need for promotion of research on Healthcare Big Data analytics in developing countries (Nishita Mehta et.al, [17]). The main outcomes of the SLR include proposal on future research direction, challenges faced by researchers, capabilities and the impact of cognitive computing on healthcare outcome and a conceptual model, showcasing the better utilization of cognitive computing in healthcare domain (Rajat Kumar Behera et.al, [18]). Big data analytics in medicine and healthcare is very promising process of integrating, exploring and analysing of large amount complex heterogeneous data with different nature: biomedical data, experimental data, electronic health records data and social media data. Integration of such diverse data makes big data analytics to intertwine several fields, such as bioinformatics, medical imaging, sensor informatics, medical informatics, health informatics and computational biomedicine. As a further work, the big data characteristics provide very appropriate basis to use promising software platforms for development of applications that can handle big data in medicine and healthcare. One such platform is the open-source distributed data processing platform Apache Hadoop MapReduce that use massive parallel processing (MPP). These applications should enable applying data mining techniques to these heterogeneous and complex data to reveal hidden patterns and novel knowledge from the data (Blagoj Ristevski et.al, [19]).

3. PROPOSED APPROACH

The majority of the models in the literature survey were developed to give real-time or retrospective patient death prediction at least 24 hours or 48 hours following ICU admission. We present a two-phase model framework in this study to handle the challenge of predicting mortality and death hours in the early stages of an ICU stay. In the second phase, if a patient is anticipated to die in the first phase, our algorithms will also provide an estimate of death hours since ICU admission. Our goal is to identify patients who are at high risk of dying within hours or days of being admitted to the ICU. During each ICU stay, data was pre-processed and extracted from the MIMIC-III database for the 24 hour period since ICU admission. For the time period provided, multiple models were trained on the retrieved attributes of the study population. In this study, the model results are contrasted and explained in detail.

4. DATA STRUCTURE

The original dataset in the MIMIC-III database consists of 61,532 distinct ICU stays of 46,520 unique patients. To form our study population, we have included all ICU stays to analyze the data of unusual short stays and only consider all patients. The final study population covered 49,632 ICU stays of 36,343 patients. Multiple machine learning models in this study have been trained and evaluated based on this study population. Table 1 provides summary statistics of the study population.

Table 1: Summary Statistics of the Study Population

S. No.	Variables	Statistics
1	Age	Mean 62.61
2	Gender	Male 57.79%
3	Ethnicity	White 71%
4	Admission Type	Emergency 82.31%
5	Number of ICU Stays	Mean 1.37
6	In-Hospital Mortality Ratio	11.62%

Among 49,633 ICU stays, there are 5,766 in-hospital mortality. After filtering out the ICU stays with negative death time since ICU admission (which is likely an administrative error resulting in an incorrect ICU admission or incorrect death time), 5,718 in-hospital mortality were resulted. The average death time since ICU admission is 9.57 days, maximum death time is 206.38 days and minimum death time is 0 day. Below line and clustered chart (Age Distribution by Count for Dead Patients) shows perfect distribution of deaths by age and as per the trends in the graph, most of the deaths happen between ages 55-85. Age groups 20-40 has less death rate but still the number is in the range of 100-200.

Table 2: ICD9 Codes and Death Percentage

S. No.	Category	Code Ranges	No. of Codes	Death %
1	Metabolic Disorder	240-279	23927	11.3
2	Pulmonary Disease	460-519	18694	8.8
3	Other form of Heart Disease	420-429	13203	6.3
4	Digestive disease	520-579	13181	6.2
5	Supplementary Classification of Factors Influencing Health status and Contact with Health Service	V01-V86	12801	6.1
6	Ischemic Heart Disease	410-414	12794	6.1
7	Renal Inefficiency	580-629	11262	5.3
8	Hypertensive Disease	401-405	10970	5.2
9	Symptoms, Signs and Ill-Defined Conditions	780-799	8777	4.2
10	Disease of the Blood and Blood Forming Organs	280-289	8744	4.1
11	Trauma	800-959	7975	3.8
12	Heart Failure	428	7470	3.5
13	Infection Disease	001-139	7388	3.5
14	Mental Disorders	290-319	7337	3.5
15	Supplementary Classification of Causes of Injury and Poisoning	ES00-E999	6632	3.1
16	Arteries and Veins	440-459	5828	2.8
17	Neoplasms	140-239	5403	2.6
18	Neurology Disease	320-389	5140	2.4
19	Disease of Musculoskeletal System & Connective Tissue	710-739	3632	1.7
20	Other Complications of Procedure NEC	998	2836	1.4
21	Cerebrovascular Disease	430-438	2849	1.4
22	Disease of the Skin and Subcutaneous	680-709	2799	1.5
23	Complications Affecting Specified Body Systems- Not Elsewhere Classified	997	2551	1.2
24	Complications Peculiar to Certain Specified Procedure	996	2403	1.1
25	Other and Unspecified Effects of External Causes	990-995	1941	0.9
26	Chromic Rheumatic Heart Disease	393-398	1438	0.7
27	Disease of Pulmonary Circulation	415-417	1233	0.6
28	Congenital Anomalies	740-759	673	0.3
29	Complication of Pregnancy, Childbirth and Puerperium	630-677	627	0.3
30	Poisoning	960-989	590	0.3
31	Complication of Medical Care, Not Elsewhere Classified	999	213	0.1
32	Acute Rheumatic Fever	390-392	5	0.0

ICD9_CODE is one of the major feature for finding the mortality of the patients and it contains the actual code corresponding to the diagnosis assigned to the patient. Hence, we did more analysis of those codes and extracted the data from MIMIC III database on death percentage among the various ICD-9 codes. We extracted only dead patients from and their related ICD-9 codes and calculated the percentage death among them. The below figure 1 and figure 2 describes top 15 ICD-9 codes by percentage of death for more visibility. After closely observing the graph we can clearly tell that metabolic disorder, pulmonary disease, heart disease, and digestive diseases are the common diseases for

mortality resulting into death. Table 2 describes the overall ICD-9 codes resulting into death.

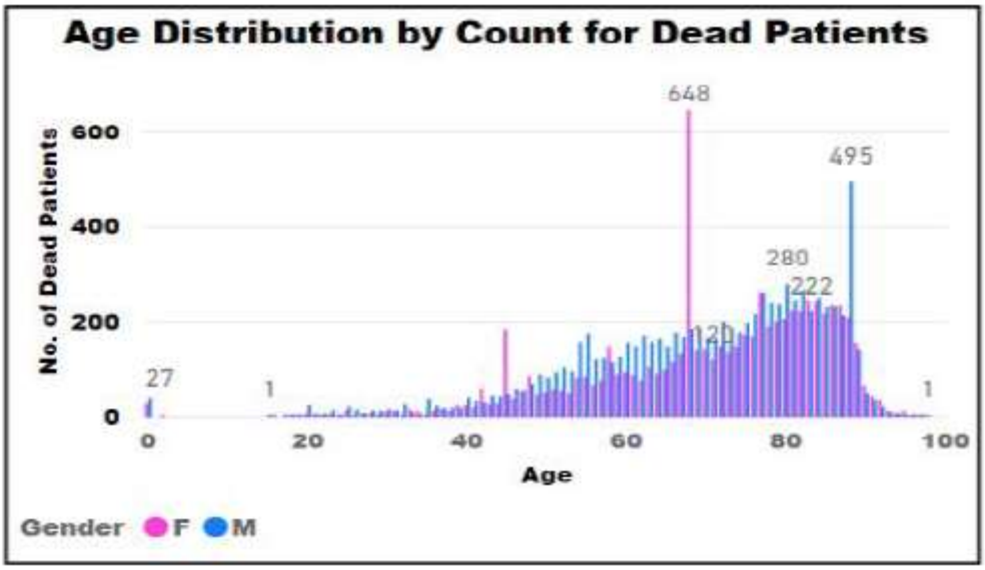

Figure 1: Age Distribution by Count for Dead Patients

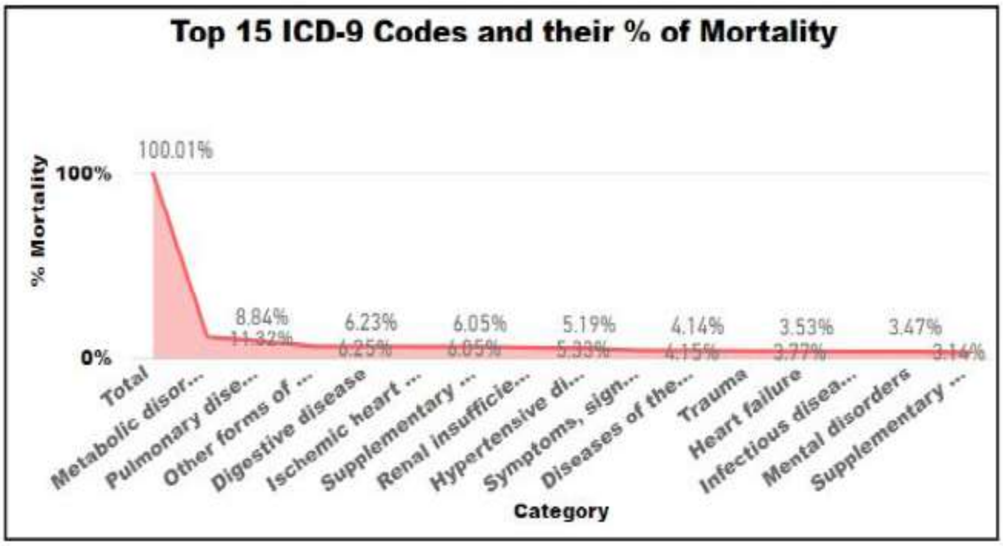

Figure 2: Top 15 ICD-9 Codes and their % of Mortality

5. DATA PRE-PROCESSING

We have a total of 26 datasets downloaded from the MIMIC III database [2]. These datasets have a mixture of both categorical and continuous variables. Since the total size of the datasets is around 43.3 GB with over a million records, we used hadoop spark to automate the process of data extraction. Our algorithm was inspired by the homework assignments and the benchmarking techniques used in the Multitask Learning and Benchmarking with Clinical Time Series Data research paper [9]. Despite using spark, the overall data extraction process was computationally very expensive. Each patient in the database are referred to as subjects and are given a unique subject ID. We created folders for each subject ID and in each folder, we stored specific patient information that will aid in predicting the mortality. Subject's hospital stays, diagnosis, and events information was stored in each of those folders. The below figure 3 shows the attributes from each of the CSV files used and its descriptions. The attributes in the below figure are extracted from several datasets. Please click on this excel link to learn about from which specific datasets the below attributes are derived from. If you notice, the attributes are a mixture of demographic information (such as age, date of birth, etc.) and technical information (such as mortality, the name of the illness, etc.) about the illness that the patient was suffering from. Eventually, this will aid in our analysis to answer questions like, is there a specific risk for people belonging to a certain age or ethnicity to suffer from a particular illness? Does frequent visit to the hospitals or longer hospital stays indicates that the patient's health is deteriorating and may increase the chance of mortality?

Stays
SUBJECT_ID: patient Unique ID
HADM_ID: refers to a unique hospital admission
ICUSTAY_ID: corresponds to a single HADM_ID and a single SUBJECT_ID
LAST_CAREUNIT: contain, respectively, the first and last ICU type in which the patient was cared for
DBSOURCE: contains the original ICU database the data was sourced from
INTIME: provides the date and time the patient was transferred into the ICU
OUTTIME: provides the date and time the patient was transferred out of the ICU
LOS: is the length of stay for the patient for the given ICU stay
ADMITTIME: provides the date and time the patient was admitted to the hospital
DISCHTIME: provides the date and time the patient was discharged from the hospital
DEATHTIME: Time of Death
ETHNICITY: Racial identity of the patient
DIAGNOSIS: provides a preliminary, free text diagnosis for the patient on hospital admission
GENDER: Male or Female
DOB: Date of birth of the patient
DOD: Date of death of the patient
AGE: Age
MORTALITY_INUNIT: Patient Mortality Information
MORTALITY: Patient Mortality Information
MORTALITY_INHOSPITAL: Patient Mortality Information

Diagnosis
SUBJECT_ID: patient Unique ID
HADM_ID: refers to a unique hospital admission
ICUSTAY_ID: corresponds to a single HADM_ID and a single SUBJECT_ID
ICD9_CODE: contains the actual code corresponding to the diagnosis assigned to the patient for the given row
SHORT_TITLE: The title fields provide a brief definition for the given diagnosis code
LONG_TITLE: The title fields provide a brief definition for the given diagnosis code
SEQ_NUM: provides the order in which the ICD diagnoses relate to the patient
HCUP_CCS_2015: Name of the illness (ex: Pneumonia)
USE_IN_BENCHMARK: used in benchmark task

Events
SUBJECT_ID: patient Unique ID
HADM_ID: refers to a unique hospital admission
ICUSTAY_ID: corresponds to a single HADM_ID and a single SUBJECT_ID
CHARTTIME: records the time at which an observation was charted, and is usually the closest proxy to the time the data was actually measured
ITEMID: Identifier for a single measurement type in the database
VALUE: contains the value measured for the concept identified by the ITEMID

Figure 3: Stays, Diagnosis and Events in Hospitals

The number of events prior to removing missing information was 253,116,833. Events with missing HADM_ID (5,162,703) and ICUSTAY_ID (15,735,688) information in the three CSV files were removed and the total number of events after this step turned out to be 232,218,442. Every time a subject is admitted to the ICU, their ICU stay is represented with an ICU stay ID. An individual may have several ICU stay IDs, indicating that they have been hospitalised to the ICU more than once. Medical data from an ICU stay, such as capillary refill rate, diastolic blood pressure, and percent inspired oxygen, is collected. glascow coma scale eye opening, glascow coma scale motor response, glascow coma scale total, glascow coma scale verbal response, glucose heart rate, height, mean blood pressure, oxygen saturation, respiratory rate, systolic blood pressure, temperature, glascow coma scale total, glascow coma scale verbal response, glucose heart rate, height, mean blood pressure, oxygen saturation, respiratory rate, sysweight, and pH, was extracted for every ICU stay ID in different csv files (based on the number of ICU visits) for every subject ID.

All the subject level ICU stay information csv files were distributed to three folders namely, Train (60% ~14,681), Validation (20% ~ 3,222), and Test (20% ~ 3,236). In this process, three separate csv files were generated namely, train_listfile, val_listfile, and test_listfile. These 3 csv files contains the names of all the ICU stay csv files in the three folders and their corresponding mortality information represented by 0 (dead) and 1 (alive). We have ran our machine learning models on the full data set.

5.1 Model Architecture

In phase 1 and phase 2, we trained and experimented with feature rich MIMIC III data sets using a combination of classification and deep learning keras models namely: Logistic Regression, Random Forest, Artificial Neural Network

(ANN), Convolutional Neural Network (CNN), and Long Short-Term Memory Network (LSTM). We finalized two keras deep learning frameworks in Python for our machine learning model evaluation in Phase 2. Keras is an open source machine learning API framework that can run on top of Theano, Tensorflow, or CNTK framework. Karas APIs are simple to use and to understand the machine learning problem and in addition, we get the full ability of its underlying powerful deep learning architecture. We chose to run Keras model on top of Tensorflow. The figure 2.b shows the popularity of Keras API. Keras Model has built in multi - GPU Parallelism support. We noticed that LSTM model training took 30 hours to complete in CPU. For the ANN/CNN Keras models, we tried 'binary_crossentropy' and 'mean_squired_error' as two loss functions, and 'adam', and sgd as two optimizers. We noticed that 'sgd' optimizer, and 'mean_squired_error' loss function yielded expected AUC/ROC, F1 score, and Accuracy, and we finalized our model based on this finding. We trained our models using train data size of 14681* 714, validated using 3222*714, and tested using test set of size 3236*714. The training dataset was of reasonable size to run on a local laptop MAC OS environment, therefore we chose to train model in Spyder/Anaconda Python 3.6 environment, and capture the results log in HTML form for further review. Figure 4 shows deep learning frameworks ranking computed by Jeff Hale, based on 11 data sources across 7 categories.

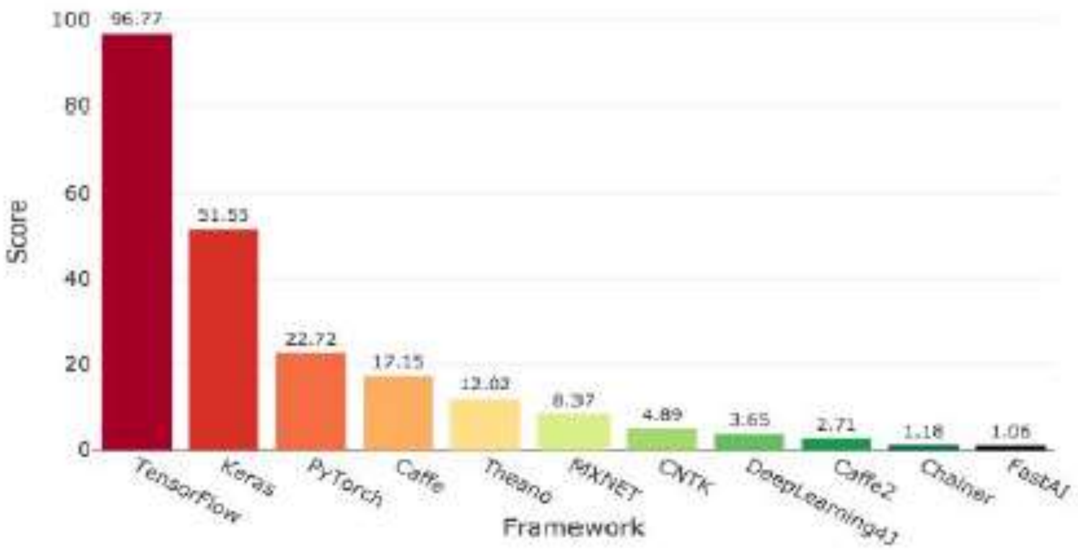

Figure 4: Deep Learning Power Scores

5.2 System Specification

Our research is divided into two stages: I feature engineering with Hadoop HDFS and PySpark on a local Docker MAC OS environment (1 master node and 2 worker nodes, each with 900 GB of storage, 14 GB of RAM, and 4 processors), and (ii) machine learning with Python 3.6 on a local cluster (500 GB space, 16GB RAM, 4 GB CPU and 4 processors). MIMIC-decompressed III's dataset takes up about 50 GB of disc space. Because the dataset is rather enormous, we picked big data tools such as Apache Hadoop/HDFS and PySpark to perform data pretreatment and feature engineering. The first stage's data is then used as feature input for the second stage's model training. For efficient model testing, hyperparameter tuning, and model evaluation, we used Python and libraries like Keras, Pandas, and Scikit-learn. In the conda environment specification added to the root of the work, all dependencies for the execution of this work have been defined.

6. RESULT ANALYSIS

6.1 ANN Loss, Accuracy, and ROC Curve

ANN model is trained and validated on the dataset for 10 epochs to give a quick view of the model's working. Figure 5 ANN LC states the training loss and validation loss which starts from ~0.13. The robustness of model increases along with the decrease of the value of loss function. Training Loss gradually decreases and stabilizes at ~0.04. Validation Loss also reduces with training loss. Loss curve states that the model is neither over fitting nor under fitting.

Accuracy Curves helps in notifying the score accuracy of prediction. Figure 6 of ANN accuracy denotes that the validation accuracy is between 0.882-0.884 i.e. model can predict even for the new unseen data with most accuracy. Figure 7 ANN ROC gives a smooth AUC curve which covers an area of about 0.85. It gives the notion of a positive learning model.

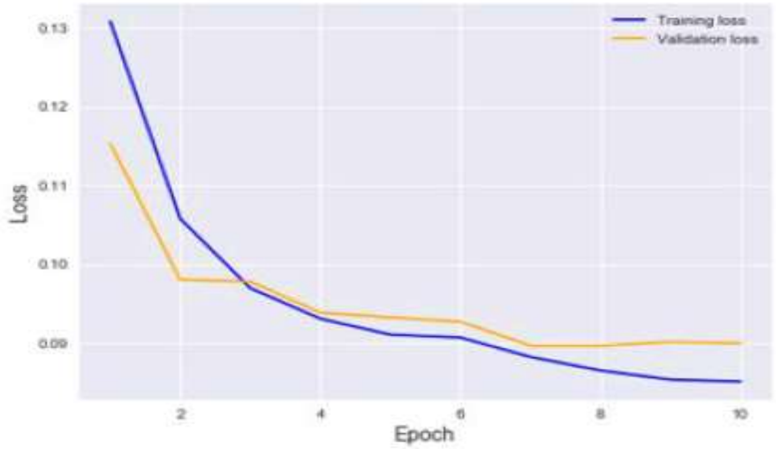

Figure 5: ANN Loss Curve

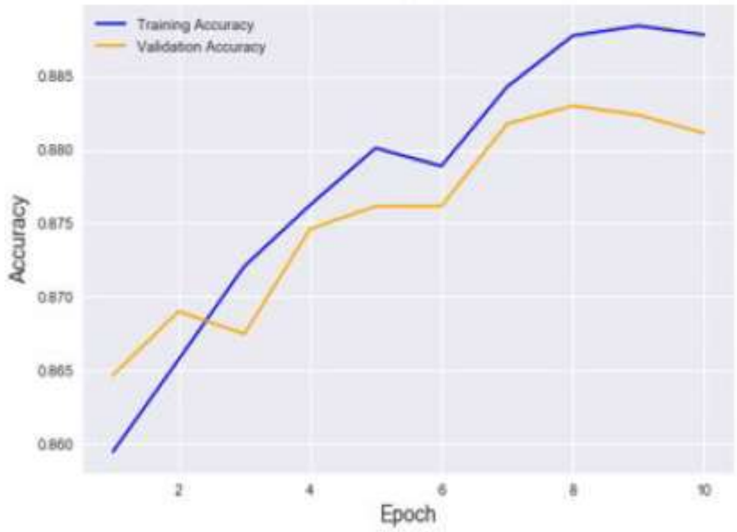

Figure 6: ANN Accuracy Curve

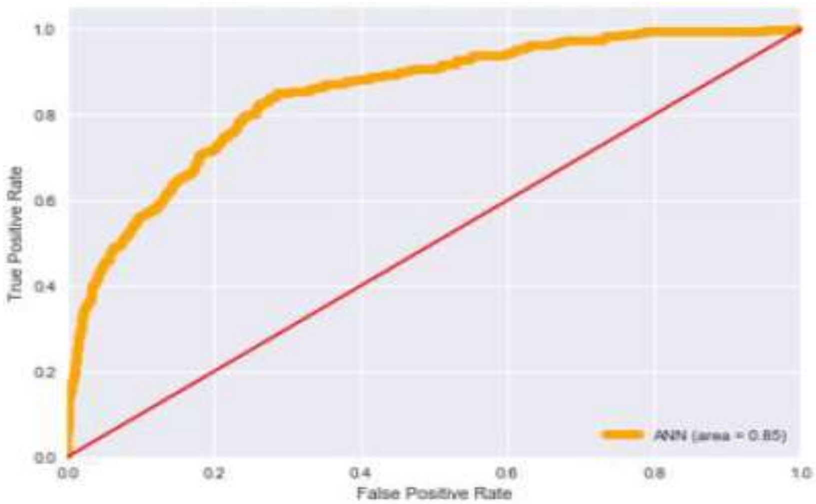

Figure 7: ANN ROC Curve

6.2 ANN Confusion Matrix

True positive gives 0.95 and false positive is 0.05 for the confusion matrix for ANN which explains that the training is appropriate for predicting positive data. Even if false negative is greater than the true negative, the difference in prediction is much lower. F1-score was 0.49 which considers not even positive but also negative prediction values.

6.3 CNN Loss, Accuracy, and ROC Curve

Figure 8 CNN LC plots the Training Loss and Validation Loss. Model trains quickly with the training data as the training loss touches 0.08 before 40 epochs and goes on decreasing further. Validation loss goes down towards 0.09 and below. This gives a very fine-tuned binary classifier for mortality prediction. Training accuracy in figure 9 CNN AC reaches 0.895 before 40 epochs. Validation accuracy continuously increases and attains 0.885 which states the correct positive prediction. Area under the AUC/ROC Curve determines the working of the model. CNN model built with specific configurations which gives the area of 0.85 and curve almost touching the top-left corner of the graph.

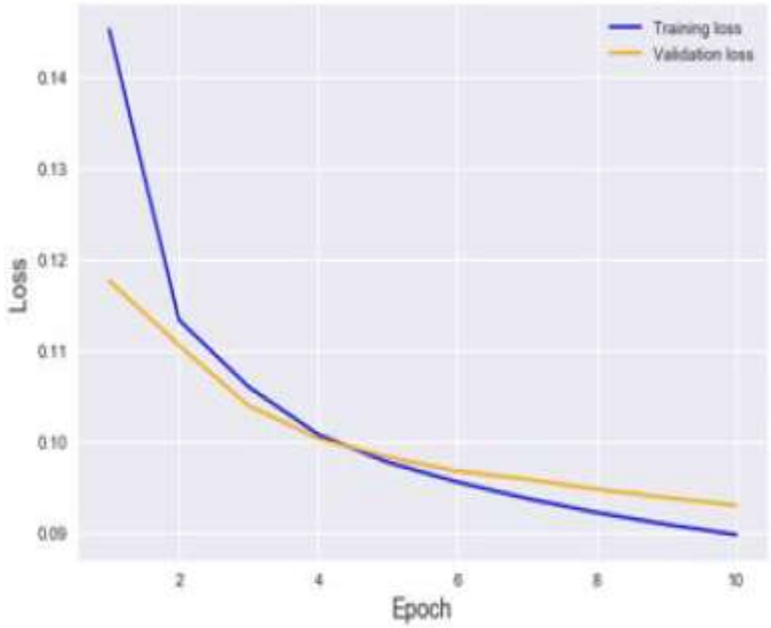

Figure 8: CNN Loss Curve

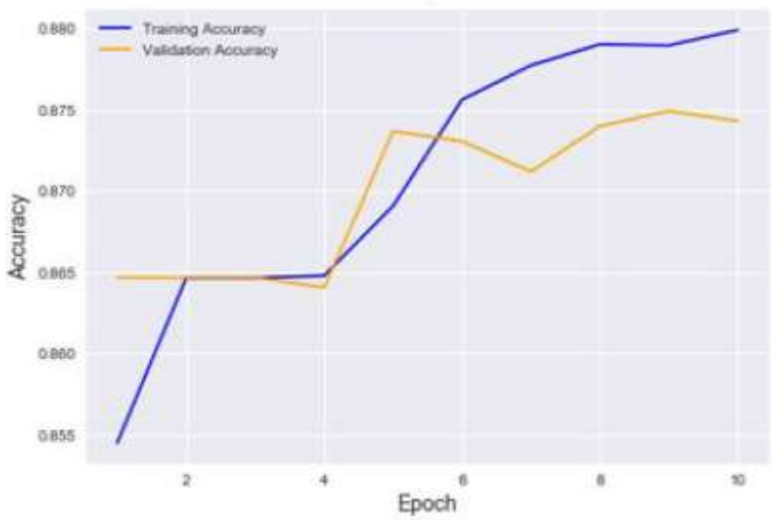

Figure 9: CNN Accuracy Curve

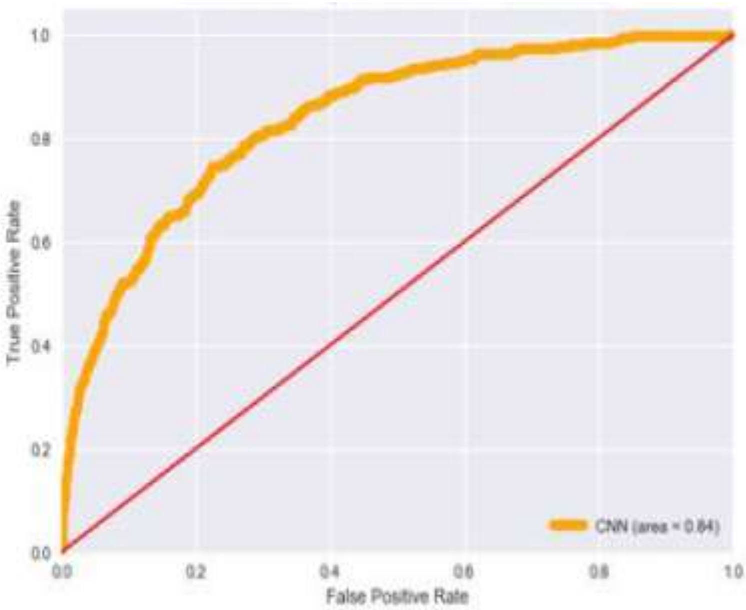

Figure 10: CNN ROC Curve

6.4 CNN Confusion Matrix

True positive scores 0.96 and false positive 0.04 for the confusion matrix. It means the training works well on positive data. False Negative is greater than True Negative which explains that still the Negative prediction is not enough good and needs further tuning.

6.5 Random Forest Curve

We produced AUC/ROC curve, learning curve, confusion matrix, accuracy, and performance metrics tables, and precision recall curves. It was interesting to notice that in python, producing these visualization was computationally time consuming when we tuned different parameters particularly for this model due its increased complexity as compared to logistic regression. Model gives the varying differences in metrics while training and testing the random forest model with specific parameters. In model accuracy results table, accuracies are almost 90%, which shows even for fewer parameters this model works great. An another reason for such a high accuracy score is due to the model's inherent ability to not overfit. AUC/ROC curve for the sample entry from the above table where the area is calculated as 0.60. Every test done gives a Confusion Matrix like the one in figure 4.3 which helps in calculating precision and recall metrics. For this Confusion Matrix, we have precision as 0.9846 and recall as 0.9099. In Performance Metrics result table, for varying parameters the Precision range was 0.55-0.61 while recall ranges from 0.34-0.38. Precision-Recall curve shows a good steady descent where AUC=0.494 and F1=0.344. Training score curve gradually decreases but the cross-validation score is almost steady which ensures the model is highly biased. It needs increasing the complexity of the model by engineering new and more relevant features to deliver the greatest impact and we will work on those enhancements in the next phase.

6.6 Team's Reflection on Experimental Results

We have shown that by combining static features such as patient demographic information and dynamic features such as physiological variables measured in ICU, we could train an effective model to predict in-hospital mortality in the early stage of ICU stay. The predictive performance of the model trained ICU data has competitive performance (AUROC 0.88) with the same model trained on 12-hour or 24-hour ICU data (AUROC 0.90 and 0.92 respectively). We then showed that the death time multiclass classifier trained different ICU stay data offers an effective base (micro-average AUROC 0.77) to provide a rough estimate of death hours since ICU admission. The result is competitive to the same model trained on 12-hour or 24-hour ICU data (micro-average AUROC 0.79 and 0.82 respectively). A natural question is that can we use regression instead of classifier to more precisely predict death time in Phase 2. In our proof-of-concept stage, we have fitted a random forest regressor using the same set for extracted features. However, the regressor overfitted and only achieved R = 0.20 on the test set. R is the coefficient of determination, which is a regression metric to measure how good the model explains the data. Best possible R is 1. A constant model that always predicts the expected value of target would get a R of 0. Due to the poor predictive performance, the trained regressor is not very useful in the ICU setting. Hence, it is better to formulate the death time prediction problem as a multiclass classification problem instead of a regression problem. Apart from this, our model framework provides a base for potential improvement. One possible way to enhance the performance is to fit the model with time-series data instead of aggregation of dynamic features. In our study, we have aggregated dynamic features over a specified timeframe using mean, maximum, minimum or sum. In fact, some features such as heart rate are highly temporal. Methods such as CNN and ANN, are types of Recurrent Neural Network, which is designed to handle sequence dependencies for time-series prediction problem, could be applied to handle temporal data. In our proof-of-concept stage, we have fitted ANNs and CNNs to dynamic features such as heart rate, blood pressure and respiratory rate, and ensembled them with random forest classifiers trained on other features. The initial result is in fact quite satisfactory. Fitting ANNs and random forest classifier without any tuning achieved AUC score of 0.8. It is possible that we could improve the model performance if we could find an effective algorithm to interpolate missing time-series data, construct a deeper CNN and tune the model. However, due to time and resource constraint, we have focused our study on constructing a single random forest classifier in phase 1 and ANN and CNN models in phase 2.

6.7 Comparison

Now, using the table 3 below, a comparative analysis is carried out. The number of correct predictions divided by the total number of input samples is known as accuracy. When compared to the others, Random Forest has the highest accuracy (0.92). The accuracy of ANN and CNN is nearly identical. A classifier's AUC is the likelihood that a randomly chosen positive example will be ranked higher than a randomly picked negative example. The area under the curve of a plot of True Positive Rate (Sensitivity) versus False Positive Rate is known as the AUC (Specificity). The AUC for Logistic Regression is 0.88, indicating that it works effectively for a positive binary classifier outcome. The F1 Score is used to assess the correctness of a test. The higher the F1 Score, the better our model's performance. The number of correct positive results divided by the number of positive results predicted by the classifier equals precision. While recall is calculated by dividing the number of correct positive outcomes by the total number of relevant (real positive) samples. Only the F1 score incorporates both precision and recall, which aids in determining how precise (how many occasions it properly classifies) as well as how resilient your classifier is (it does not miss a significant number of instances).Comparing F1 Score of all the experimented ML models, it turns out that ANN has the highest score i.e. 0.489. Even if RF has 0.45 F1 Score, it classifies highly for one category than other. Hence, ANN model is the most robust and more accurate classifier for mortality prediction on given dataset. Apart from fine-tuning the model, more than one classifier can be adjoined for this mortality prediction problem to further increase the accuracy.

Table 3: Model Performance Matrix

Category	Accuracy	AUC	Precision	Recall	F1-score
Logistic Regression	0.882150136	0.88	0.68	0.66	0.37
Random Forest	0.921508268	0.67	0.58	0.38	0.45
Artificial Neural Network (ANN)	0.892150803	0.698454205	0.540453074	0.44652406	0.489019034
Convolutional Neural Network (CNN)	0.896168109	0.627507035	0.611764706	0.27807487	0.382352941

6.8 Challenges

We spent a lot of time in reading numerous Literature Survey papers to understand the previous related work done by other researchers and in analyzing MIMIC III dataset to understand the tables and its underlying data structures. We did research on the Azure HDInsight as well as AWS EMR Big Data cloud services for identifying a cost effective way of loading, pre-processing and transforming our raw data set into valid training, validation and test set. We finally decided to use our local Hadoop Docker machine instance to run pre-processing job in PySpark, it took 48 hours to complete a run, that we could’ve avoided if we utilized GT GPU cluster instances. Another challenge that we faced is in tuning model hyper parameters. We spent good amount of time in fine tuning hyper parameter on LR/RF/CNN/ANN models, and analysing ROC/Loss curves. We didn't get enough time to fix Keras LSTM model issues. We used Keras deep learning models for the first time. This project was a great learning experience for our team to see a full life cycle of a Machine Learning problem implementation.

7. CONCLUSION

This work divided into two phases namely, phase 1 and phase 2. This final report is dedicated mainly for phase 2.The entire large dataset was processed using big data tools such as hadoop, spark, sql and cohorts based mainly on ICU related technical information for mortality prediction are created. This was the most time-consuming portion of phase 2. To understand the structure and the behavior of the datasets, three advanced neural network models namely: Artificial Neural Network, Convolutional Neural Network, and Recurrent Neural Network (LSTM) are used. Inspired by the homework assignments, an environment.yml file is created with a list of all python libraries and its specific version numbers and the experiments in Docker and Spyder IDE were conducted.

Logistic Regression (LR) is a mathematical model used in statistics to estimate the probability of an event occurring having been given some previous data. Random Forest (RF) formulates some set of rules that can be used to perform predictions. LR gives the function value which classifies the data while RF branches to the specific decisive sub-trees which categorizes the input for multi-class classification. Artificial Neural Networks (ANN) are based on the working of a Human- brain. It learns from the input with self-adjusting weights and stores knowledge in the form of frozen weights. Convolutional Neural Network (CNN) is Neural Network with the refinement of Deep Learning. Deep Learning is achieved by performing Optical Character Recognition (OCR), pooling and down-sampling. ANN and CNN both learns from the train data by storing weights and then classifying the input based on the tuned model. Only difference in CNN is that it processes more efficiently on even image and sound (represented as spectrogram) inputs. NN solves all types of ML problems i.e. clustering, classification and regression.

REFERENCES

[1.] http://www.ncmedicaljournal.com/content/77/2/112.full
[2.] https://mimic.physionet.org/
[3.] Pirracchio Romain, Petersen. Maya L. "Mortality prediction in the ICU: can we do better? Results from the Super ICU Learner Algorithm (SICULA) project, a population-based study" NCBI, 24 Nov. 2014, ncbi.nlm.nih.gov/pmc/articles/PMC4321691/
[4.] Lee, Joon, et al. "Personalized Mortality Prediction Driven by Electronic Medical Data and a Patient Similarity Metric." PLOS ONE, PublicLibrary of Science, 2015, journals.plos.org/plosone/article?id=10.1371%2Fjournal.pone.0127428.
[5.] Fika, Sofia, et al. "A Novel Mortality Prediction Model for the Current Population in an Adult Intensive Care Unit." Heart and Lung, Jan. 2018,www.heartandlung.org/article/S0147-9563(17)30151-6/fulltext.
[6.] Zahid, M.A. "Mortality Prediction with Self Normalizing Neural Networks in Intensive Care Unit Patients." IEE Xplore, Wiley-IEEE Press, 4Mar. 2018, ieeexplore.ieee.org/abstract/document/8333410.
[7.] Hrayr Harutyunyan, Hrant Khachatrian, David C. Kale, Greg Ver Steeg, and Aram Galstyan. Multitask Learning and Benchmarking withClinical Time Series Data. arXiv:1703.07771
[8.] McKinney, Wes. Python for Data Analysis, 2nd Edition. O'Reilly Media, Inc. 2017
[9.] Dybowski R, Weller P, Chang R, Gant V (1996) Prediction of outcome in critically ill patients using artificial neural network synthesized by genetic algorithm. Lancet 347(9009):1146–1150. https://linkinghub.elsevier.com/retrieve/pii/S0140673696906091
[10.] M. Ghassemi, T. Naumann, F. Doshi-Velez, N. Brimmer, R. Joshi, A. Rumshisky, and P. Szolovits. Unfolding Physiological State: Mortality Modelling in Intensive Care Units. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 14, pages 7584, New York, NY, USA, 2014. ACM.
[11.] https://towardsdatascience.com/deep-learning-framework-power-scores-2018-23607ddf297a
[12.] Ashwin Belle, Raghuram Thiagarajan, S. M. Reza Soroushmehr, Fatemeh Navidi, Daniel A. Beard, Kayvan Najarian, "Big DataAnalyticsinHealthcare", *BioMedResearchInternational*,vol.2015,ArticleID 370194, 16 pages, 2015. https://doi.org/10.1 155/2015/370194.
[13.] S. S. -L. Tan, G. Gao, S. Koch, "Big Data and Analytics in Healthcare", Methods Inf Med 2015; 54(06): 546-547, DOI: 10.3414/ME15-06-1001.

[14.] Dash, S., Shakyawar, S.K., Sharma, M. *et al.* Big data in healthcare: management, analysis and future prospects. *J Big Data* **6,** 54 (2019). https://doi.org/10.1186/s40537-019-0217-0.

[15.] M. M. Hansen, T. Miron-Shatz, A. Y. S. Lau, C. Paton, "Big Data in Science and Healthcare: A Review of Recent Literature and Perspectives", Yearb Med Inform. 2014; 9(1): 21–26. Published online 2014 Aug 15. doi: 10.15265/IY-2014-0004, PMCID: PMC4287084

[16.] M. Chen, F. Herrera and K. Hwang, "Cognitive Computing: Architecture, Technologies and Intelligent Applications," in IEEE Access, vol. 6, pp. 19774-19783, 2018, doi: 10.1109/ACCESS.2018.2791469.

[17.] Nishita Mehta and Anil Pandit, "Concurrence of big data analytics and healthcare: A systematic review", International Journal of Medical Informatics, Volume 114, June 2018, Pages 57-65, https://doi.org/10.1016/j.ijmedinf.2018.03.013

[18.] Rajat Kumar Behera , Pradip Kumar Bala and Amandeep Dhir, "The emerging role of cognitive computing in healthcare: A systematic literature review", International Journal of Bio-Medical Computing, Volume 129, September 2019, Pages 154-166, https://doi.org/10.1016/j.ijmedinf.2019.04.024.

[19.] Blagoj Ristevski, Ming Chen, "Big Data Analytics in Medicine and Healthcare"J Integr Bioinform. 2018 Sep; 15(3): 20170030. Published online 2018 May 10. doi: 10.1515/jib-2017-0030, PMCID: PMC6340124.

[20.] Jadaun, Abha, Satish Kumar Alaria, and Yashika Saini. 2021. "Comparative Study and Design Light Weight Data Security System for Secure Data Transmission in Internet of Things". *International Journal on Recent and Innovation Trends in Computing and Communication* 9 (3):28-32. https://doi.org/10.17762/ijritcc.v9i3.5476.

[21.] Vijay, Aditi, Ashutosh Gupta, Ashwani Pal, B. Sriswathi, Geetika Mathur, and Satish Kumar Alaria. "IoT SOCIAL DISTANCING & MONITORING ROBOT FOR QUEUE." International Journal of Engineering Trends and Applications, Vo-8, Issue-4, 2021, PP-20-25.