Emergency Department Triage Prediction of Clinical Outcomes Using Deep Learning Approach

Chaltu Mulatu Ejeta

A Thesis Submitted to the Department of Computer Science and Engineering
School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Computer Science and Engineering

Office of Graduate Studies
Adama Science and Technology University

June 2023
Adama, Ethiopia

Emergency Department Triage Prediction of Clinical Outcomes Using Deep Learning

Chaltu Mulatu Ejeta

Advisor: Dr. Sintayehu Hirpassa.

A Thesis Submitted to the Department of Computer Science and Engineering
School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Computer Science and Engineering

Office of Graduate Studies
Adama Science and Technology University

June 2023

Adama, Ethiopia

Declaration

I, hereby declare that this Master Thesis entitled "**Emergency Department Triage Prediction of Clinical Outcomes Using Deep Learning"** is my own work and has not been submitted to any university for a similar purpose. The references used in this proposal are duly recognized by proper citations.

<u>Chaltu Mulatu Ejete</u>	________________________	________________
Name of student	Signature	Date

Recommendation of Advisors/ Supervisors

I, the major advisor of this research thesis, hereby certify that I have closely advised the student while developing this thesis and read the draft thesis entitled "**Emergency Department Triage Prediction of Clinical Outcomes Using Deep Learning**" prepared under my guidance by Chaltu Mulatu Ejeta. Therefore, I recommend the submission of the thesis to the department for further review and evaluation.

Dr. Sintayehu Hirpassa	____________________	______________
Major Advisor/Supervisor	Signature	Date
______________________	____________________	______________
Co-advisor/Co-supervisor	Signature	Date

APPROVAL PAGE OF M.SC. THESIS

I/we, the advisors of the thesis entitled "**Emergency Department Triage Prediction of Clinical Outcomes Using Deep Learning**" developed by Chaltu Mulatu Ejeta, hereby certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

Dr. Sintayehu Hirpassa	____________________	____________________
Major Advisor	Signature	Date
____________________	____________________	____________________
Co-advisor Signature	Date	Date

We, the undersigned, members of the Board of Examiners of the thesis by Chaltu Mulatu Ejeta have read and evaluated the thesis entitled "**Emergency Department Triage Prediction of Clinical Outcomes Using Deep Learning**" and examined the candidate during open defense. This is, therefore, to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of Master of Science in Computer Science and Engineering.

____________________	____________________	____________________
Chairperson	Signature	Date
____________________	____________________	____________________
Internal Examiner	Signature	Date
____________________	____________________	____________________
External Examiner	Signature	Date

Finally, approval and acceptance of the thesis is contingent upon the submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

____________________	____________________	____________________
Department Head	Signature	Date
____________________	____________________	____________________
School Dean	Signature	Date
____________________	____________________	____________________
Office of Postgraduate Studies, Dean	Signature	Date

ACKNOWLEDGMENT

First and foremost, thank the Almighty God for blessing me with full health, being always beside me throughout my life and my activities. Next, I would like to convey my profound gratitude to Dr. Sintayehu Hirpassa my research adviser, for his tremendous guidance, support, and supervision throughout my research study. Working and studying under his guidance was a wonderful honor and privilege. Thank you always!!

Next, I would like to express my deepest gratitude to my Husband Dereje Ayana for his support for the success of this thesis by providing a comfortable environment for me, and also to my best friend Leti Wakuma for her guidance and experience sharing from the beginning to the end.

Additionally, I would like to thank my family, who inspired and encouraged me, and A special thanks to my brother Tilahun Mulatu for his endless Advice and encouragement. Last but not least, I want to thank everyone who was with me and my friends for sharing their wisdom with me. They are the source of guidance for me.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ACSMC	Adama Comprehensive Specialized Medical College
AI	Artificial intelligence
ANN	Artificial Neural Network
BiGRU	Bidirectional Long Short-Term Memory Network
BiLSTM	Bidirectional Long Short-Term Memory Network
CNN	Convolutional Neural Networks
CSV	Common Separated Value
CHAS	Canadian triage acuity scale
CPU	Central Processing Unit
DL	Deep Learning
EC	Emergency center
E.C	Ethiopian calendar
ED	Emergency Department
ER	Emergency room

GRU	Gated Recurrent Unit
LSTM	Long Short-Term Memory Networks
ML	Machine Learning
MTS	Manchester triaging system
NMEI	New Medical Education Initiatives
SATS	South African triaging system
START	Simple triage and Rapid Treatment
TEWS	Triage Early Warning score
ICU	Intensive Care unit
USA	United States of America

Abstract

Triage systems development in hospitals is essential for Emergency Department (ED) to accurately differentiate and prioritize critically ill from stable patients. It is to give treatment priority to the most urgent patients. Adama Hospital is one of the largest referral hospitals in east Showa, many patients, especially those with very critical conditions are admitted daily to hospitals. The hospital Emergency Department (ED) is still working on the manual procedure to triage the patients who are even under critical conditions, this makes the hospital difficult to save lives and inefficient in resource mobilization. In this study, we have planned to automate this process by building a deep learning-based model, which can reduce complexity and boost the quality of service. We have developed a dataset for model development by gathering real data from ACSMC. Our dataset comprises 5000 instances with Twelve variables, one of which is the target variable. We have been applying multiple data preprocessing and preparation techniques like cleaning and feature scaling. Then we trained distinct modes, by using these deep learning algorithms such as Long-short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Convolutional Neural Network (CNN), for comparison of their performance of prediction of the triage outcome of patients. We have investigated the effect of algorithms hybridization such as CNN with LSTM, in this case, we used CNN for extracting optimized features directly from the dataset. In addition, we also analyzed bidirectional encodings such as Bi-LSTM and Bi-GRU. With performance evaluation parameters like accuracy and f1-score, the performance of the models is assessed under stratified 10-fold cross-validation. The performance evaluation findings showed that the LSTM model outperformed other models for prediction. It achieved an f1-score of 83.5 % and an accuracy of 83.7 %. At the end, we recommend a further experiment to improve the accuracy of the work itself by gathering ample data that is efficient for the deep learning model.

Keywords*: Triage, Emergency Department, overcrowding, TEWS-score, Deep learning, mortality, triage levels.*

CHAPTER ONE

1. INTRODUCTION

1.1. Background of the study

A healthcare system is how all health services are provided. According to the World Health Organization, a health system consists of all organizations, institutions, resources and people, and actions whose primary intent is to promote, restore or maintain health. In most countries, Its recognized to include public, private, and informal sectors (White, Stallones, and Last 2013). Health systems are about more than patient care: they attend to why people become ill in the first place, foster health-promoting environments, and sound preventive practices. Ethiopia’s health service is structured into a three-tier system: primary, secondary, and tertiary levels of care (Manyazewal 2017),(Organization 2017)(Organization 2017). According to Ethiopia’s Ministry of Health, maternal and child health are two of the most serious issues in Ethiopia(Health 2015).

Adama Hospital Medical College (AHMC)was one of the very early medical colleges that were established by missionaries from abroad in 1938 E.C. It was among the 1st non-governmental hospitals among a great commitment because it is the nation’s leading service provider. To become a known institution, the hospital developed its facilities, technology, service standards, and personnel. It was upgraded to Medical College in 2003 E.C. because of its location, patient load, and staff strength. The College is led by the board and has three major wings (Academic and research, Health service, and Development and Administration). Still, the hospital has undergone different reforms (BPR, BSC, HMIS, HCF, EHRIG, and CASH). Currently, the hospital is giving service to more than 6 million population who are from five regions (Oromia, Amhara, Afar, Somali, and Dire-Dawa). The hospital has a capacity of 232 beds for hospitalization and gives a service to an average of 1000 patients per day at six medical case groups (OPDs) and other specialty services.

One of the common practices of activity used in healthcare facilities is the emergency department triage system. The triage system is a sorting mechanism that classifies patients according to the severity of their disease level and the availability of resources in the department. No appointment is required to access care within the emergency department(ED); members of the public have the right to have to present at any time of the day without notice and access treatment. This process can result in many people with undifferentiated conditions presenting to the ED for treatment at the same time(Burgess, Kynoch, and Hines 2019). When resources are insufficient to serve all patients at once, triage is used to increase efficiency and prioritize those who need the most urgent care(Abdelwahab, Yang, and Teka 2017a). In an effective emergency center (EC) triage system, trauma, and non-trauma patients should be segregated according to the level of acuity(Abdelwahab, Yang, and Teka 2017a).

The term "triage" comes from the French word "trier" which refers to the procedures of grouping and organizing according to the degree of damage (Ericksen-Pereira, Roman, and Swart 2018). Triage has come a long way. Surgeon Dr. Dominique Jean Larrey introduced the idea of triage to medicine during the Napoleonic War. Documentation from the early 18th century demonstrates that triage was invented in the military allowing field doctors to quickly assess and treat wounded soldiers in battle. The triage technique was first used in hospitals in 1964. The trauma centers in the United States of America (USA) received the initial credit for triaging in 1986(Iserson and Moskop 2007a). the Australian College of Emergency Physicians adopted a formal triage protocol in response to this(FitzGerald et al. 2010). The Manchester triaging system (MTS) was launched in the United Kingdom in 1996 as well(Van Niekerk and Goris 1990). Canada developed the Canadian triage acuity scale (CTAS) in 1999 (Jiménez et al., 2003). In 2003, the United States adopts and creates the Emergency Severity Index (ESI) (Gilboy et al. 2012). In 2006, South Africa created the South African Triaging System (SATS) (Iserson and Moskop 2007b). These are all rated on a five-category scale based on the main complaint, vital signs, and triage staff's experience.

There is a great application of AI to contribute to the modernization of the environment of health centers to give health services effectively. Deep learning is machine learning that teaches a computer to do what comes naturally to humans: learn by example It is an interesting field of health research to help health workers to determine the outcome of health service. Deep learning can achieve state–of–the–art accuracy, that can exceed human performance. Models are trained using a large dataset of labeled data and neural networks with many layers. Deep learning can analyze data to make better predictions. Patients need effective, accurate, and timely treatment; therefore, with deep learning health workers better treat and save their life. Our study will work on labeled data to predict the outcomes, deep neural network algorithm will be used.

In ACSMC, the emergency department triage system is one of the very important departments. And they use a color code mechanism to give service. These color codes are indicated by 4 colors (red, orange, yellow, and green) to identify the priority of the patient to be treated fastest, and this is done by calculating the modified Early Warning score (MEWS) values of a patient. The MEWS include mobility, HR, Spo2, Temp, Trauma, SBP, pain score, CNS/AVPU, and RR. We used the guideline which is in use currently in the hospital for the emergency department triage system to create a dataset that fits the prediction model. This study aims to develop a best-performing model to predict the severity level of the case and accurate assignment of the patients to related emergency classes by applying a deep learning approach. To achieve this different neural networks were compared, and the one with the highest accuracy was chosen. As we tried to review different works of literature, this study is the first to predict the emergency department triage outcome in Ethiopia.

1.2. Motivation for the Study

Every day in our country, many patients who come to the emergency room irregularly suffer from various diseases of unknown severity, urgency, and unclear diagnosis. But these patients do not receive effective treatment and many of them die before they have been treated. This happens because of different cases: one is a mismatch of the patient number and the health workers at a time. There are also other challenges such as shortages of material, poor record-keeping practices, the ever-increasing triple burden of disease, and injuries, and limited access to health care for the general population(Nowacki et al. 2013),(Rominski et al. 2014). Considering these complex differences, there is a surprising dearth of evidence on emergency triage implementation specifically in low-resource settings. In turn, ECs in developing countries have been given less attention by researchers and policymakers due to a general lack of locally relevant evidence and Ethiopia is one of these countries(Obermeyer et al. 2015).

Additionally, most regional and country-level hospitals did not have nurses or physicians trained in emergency medicine, and more than 70% of hospitals did not have written policies for standardized emergency care which is used in the department. Even though Many researchers have conducted different research in the area, in our country no one conducted the prediction of the triage outcome to solve these problems. Unsurprisingly, in this field, looking at these issues in the health field and applications of AI, we are motivated to look forward to predicting ED triage outcomes by applying uses of AI in the context of our country, to reduce the health burden and community discontent.

1.3. Statement of the problem

The triage system in the emergency department is to prioritize the patients who require the most urgent care and maximize efficiency when resources are not sufficient to treat all patients at once (Abdelwahab, Yang, and Teka 2017b). Nonetheless, emergency departments are considered one of the most stressful healthcare areas due to budget constraints(Zlotnik et al. 2016). EDs worldwide suffer from overcrowding, mainly risking lives and affecting the quality of care(Derlet and Richards 2000). The massive increase in demand can adversely impact healthcare services access and lead to many negative consequences such as Long waiting times, patient revisits, and leave without being seen (LWBS)(Pines et al. 2006)(Carter, Pouch, and Larson 2014) and patient dissatisfaction(Pines et al. 2008). Overcrowding caused more admissions to inpatient departments, and poorer care provision(Ro et al. 2015). And delays in diagnosis and treatment (van der Linden, Meester, and van der Linden 2016), affecting those in more need(van der Linden, Meester, and van der Linden 2016) in a way that increases the rate of morbidity and mortality(Sprivulis et al. 2006). And also, Adama Hospital's emergency department is encountering a problem that hinders giving the service effectively and efficiently. Of these problems, the triage process in this department is going manually and performed by triage nurses, and many of them are not trained in the field rather than working from experience, which resulted in less quality service, and also we have observed that patient cards are even lost and it is impossible to attend to the patient's condition.

Various scholars have conducted to give a response to the above kind of aforementioned problem. One of these scholars, Bruno P.(Roquette et al. 2020) proposed a model for predicting admission in pediatric emergency departments. In this study, they developed a model by focusing on textual data. However, the usage of numerical data from patient history is very important, as a result, prediction accuracy got less. In addition, the study considered only pediatrics admissions, which means the study is domain-specific and this leads to a poor generative. Another study was conducted by (Van Niekerk and Goris 1990) to determine triage performance among pre-hospitalized personnel in the case of Trauma, which is also domain-specific and they used a very small amount of data that resulted in minimum predictive accuracy. (Goto et al. 2019) have also tried to address the problem by examining the performance of ML algorithms to predict clinical outcomes and disposition in children in the emergency department. This work also with some limitations such as a class of society for whom the triage system is mandatory such as accidents are not included.

In general, however, for the emergency department triage-related problem a few works have been done so far, but no one has developed a model in the context of our country. This is an issue that initiated us to engage in this work and to become a role play in finding a solution for the above-mentioned problem. In this study, we come up with a domain-free deep-learning predictive model.

1.4. Research Questions

This study addresses the problem by answering the following questions:

RQ1. Which Attributes most contribute to the prediction of the Emergency Department Triage outcome?

RQ 2. What is the most important hyperparameter to deal with during the experiment that has a great impact on the model's performance?

RQ3. Which deep learning algorithm is most suitable for developing a model that can accurately predict triage for an emergency department?

1.5. Objective

General Objective

The general objective of this study is to develop a deep learning-based model which predicts triage outcomes for emergency departments.

Specific Objective

To achieve the general objective of the study, the following specific objectives are to be addressed.

- To build a dataset for model development.
- To examine and identify the most important features for the model development.
- To identify the best-performed deep learning algorithm for model development.
- To build a prediction model for triage outcomes.
- To find the optimum performance for the model.
- To evaluate the performance of the developed model.

1.6. Significance of the Study

Once this research is implemented, it will provide numerous benefits to the health system. First and foremost, it benefits Adama Comprehensive Specialized Medical College. By modernizing the system, it is possible to eradicate all human-made errors and build society's satisfaction with health services.

DL prediction modeling uses parametric algorithms that can incorporate a larger variety of complex predictors and maintain predictive performance. One of the most important advantages of ML(DL)-based methods is that they can identify patterns in the input data using algorithms, and then exploit the uncovered patterns to predict future data. It has been shown that AI can enhance the triage capacity(Leaver, Seydell-Greenwald, and Rauschecker 2016)(Traube et al. 2016) (Swain et al. 2020) to rapidly screen patients and ensure their timely treatment in function of their condition. This study has the potential to minimize the workload of the nurses, eradicate human errors, save patients time and costs, and improve the quality of care services. And also, the prepared dataset can be a wonderful resource for anyone interested in the problem.

1.7. Scope and Limitation

1.7.1. Scope of the Study

The study focuses on using deep learning techniques to provide a model that predicts emergency department triage outcomes. This is for patients who emergently come to the department, triaged as the priority of their case to be treated in the correct triage zone. Since there is no dataset to train models, a new dataset is constructed by collecting the data from Adama Comprehensive Specialized Medical College. Next to that, we performed different data preparation techniques. Then, different neural network models are built to predict the triage outcome. Lastly, the stratified k-fold cross-validation was applied to assess the model's performance using multiple metrics to choose the best-performed prediction model while training the dataset using recurrent and convolutional neural networks.

1.7.2. Limitations of the Study

This research comes across different limitations. The most difficult thing in Ethiopia especially in the medical area is finding the organized data in electronic format and it is time-consuming to convert the data from manually recorded to electronic data. And another difficulty is the lack of interest of the hospitals to give data because different hospitals are not interested to give the data for privacy purposes. Due to these cases, there was challenging to get data as needed for this work. The use of secondary data results in missing important information which can be collected in prospective observational studies and it may decrease the quality of the study. The other limitation of this study is it uses only data from the registered patients but in the emergency department, very emergent patients are not registered, but they are directly sent to severe emergency classes. However, building the model with a dataset collected from various medical centers and external validation could have better performance and more reliable features to predict the outcomes, but it is difficult to get. Besides, there are no conducted studies in Ethiopia with the same dataset for comparison of the relevant features and performance of models for the prediction.

1.8. Organization of the Study

The summary of the chapters in our thesis report is briefly covered in this section. It captures a total of seven chapters.

The introduction chapter, to which this section belongs, deals with the background of the study, the motivation behind the study, the statement of problems with research questions to be raised and addressed, the scope and limitation of the study, the objectives to be attained, and the significance of the study.

The second chapter contains a review of the literature and related works on the ED triage system, as well as an overview of the health domain, triage system, and Adama Comprehensive and Specialized Medical College. Methodologies used in triage outcome prediction: feature extraction and deep(machine) learning classifier by previous researchers, the related work with their comparison are discussed and seated.

The third chapter addresses research methodologies. In this, we discussed or included several techniques, tools, methods, and deep learning classification algorithms that we used in our study.

In chapter four, the architecture of the proposed solution is discussed. Emergency department triage outcome Prediction model architecture, and, data preprocessing architecture were described.

Chapter five contains the experimentation and implementation of the suggested method supported with some sample code and their explanation. Chapter six presents the results from experimenting with several deep learning models on our dataset and recommendation.

The final part, chapter seven, includes the conclusion of the thesis report and the future work direction of the study.

CHAPTER TWO

2. LITERATURE REVIEW AND RELATED WORKS

2.1. Introduction

In this chapter, the relevant literature related to the main objective of the study and the existing method is reviewed to clearly state and furtherly explore the research problem by providing techniques to give emergency department treatment. The review provides an introduction to the health domain and triage system, the application of AI in the health field, deep learning techniques and algorithms that have been used in triage system prediction, and related work previously studied in this area.

2.2. Worldwide Healthcare system

A healthcare system is the way all health services are provided. From how they are financed to the workforce, facilities, and existing supplies. A strong health system will ensure that everyone can access high-quality healthcare without being over-financed(Lux et al. 2011). Everyone has the right to access quality and efficient healthcare. There is a variety of health systems around the world. Implicitly, nations must design and develop health systems following society's needs and available materials, although common elements in virtually all health systems are primary healthcare and public health measures(White 2015).

Triage represents an area of high risk in the emergency department with that nursing practice as treatment and allocation decisions must be made based on limited data under time-pressured conditions. An efficient triage process is crucial to the safety of patients accessing the ED (Burgess, Kynoch, and Hines 2019). Early triage systems primarily originated for battlefield treatment. Today, triage is still deeply integrated into healthcare. Three phases of triage have emerged in modern healthcare systems. First, prehospital triage to dispatch ambulance and prehospital care resources. Second, triage at the scene by the first clinician attending to the patient. Third, triage on arrival at an emergency department or receiving hospital(Panesar 2019).

2.2.1. Application of Technologies in the Health Domain

Machine learning which is a branch of artificial intelligence performs on top of robust data that provides the ability to process big datasets which are beyond the scope of human capabilities. It can then be used to reliably transform the analysis datasets into clinical insights to plan and provide care. Machine Learning models can be developed to spot patterns, identify abnormalities, and highlight areas, thereby increasing the accuracy of all these processes. Ultimately, this leads to better outcomes and improves patient and healthcare professional stakeholder satisfaction(Davenport and Kalakota 2019).

Recently AI techniques have sent vast waves across healthcare, even fuelling an active discussion of whether human physicians replaced by AI doctors in the future. The increasing availability of healthcare data and the rapid development of big data analytic methods have made possible the recent successful applications of AI in healthcare(Jiang et al. 2017). The primary aim of health-related AI applications is to analyze relationships between clinical techniques and patient outcomes (Coiera 1997). AI programs are applied to practices such as diagnostic, treatment protocol development, drug development, personalized medicine, and patient monitoring and care to improve efficiency, reliability, and accuracy(Panesar 2019). Here are common ways AI is modifying healthcare today and will in the future. Managing Medical Records and other data(Dash et al. 2019), doing repetitive jobs(Luopajärvi et al. 1979), treatment design (Venkataramani, Bor, and Jena 2016), virtual nurses (Bickmore, Pfeifer, and Jack 2009) and, drug creation (Trenfield et al. 2019).

There are various triage systems implemented around the world that are supported by AI with a universal goal of supplying effective and prioritized care to patients while optimizing resource usage and timing(Robertson-Steel 2006), (Crumplin 2002), (Iserson and Moskop 2007b). The first one is Field and Disaster Triage:- they used different algorithms, such as START triage, which stands for "simple triage and rapid transport, and SALT triage or sort, assess, life-saving interventions, and treatment/transport”. As emergency responders arrive at the scene, victims are asked to walk to a designated area marked off for care for minor injuries. The benefit of the

SALT method vs. the START method is that there is a grey area that is provided for the population affected and allows providers to be more flexible with their decision-making.

The second is The Australian Triage Scale: originally named the international triage scale (ITS) which is a 5-level categorical scale. The ITS was adopted as the national triage scale (NTS) in 1993 by the Australian College of Emergency Medicine and become the ATS in 2000(Ebrahimi et al. 2015). Third, the Canadian Triage System Also known as CTAS, is based on the NTS of Australia. CTAS is a 5-level triage system based on the severity of the illness or the time needed before medical intervention combined with a standardized presenting patient complaint list. Similar to other 5-level triage systems, It starts with level one as the most severe patients to level five (non-urgent). Unique to CTAS is the first and second-order modifiers that are used after an initial acuity level is given to a patient that changes that patient's acuity level (Hodge et al. 2013).

2.3. Healthcare Systems in Ethiopia

Ethiopia adopted modern medicine in the 16th century, and every emperor after that embraced medicine and health(Ayano et al. 2017). According to a 1967 article in the Milbank Memorial Fund Quarterly by E. Torrey, the first hospital in Ethiopia was built in 1909 by the Russian Red Cross in Addis Ababa. Ethiopia has made various attempts to achieve universal primary health care, starting with the Alma-Ata Declaration of Primary Health Care in 1978. This act intended to provide health for all by the year 2000 and had support from the Ethiopian Government(Ajisegiri et al. 2021).

Ethiopia’s health service is structured into a three-tier system: primary, secondary, and tertiary levels of care(Manyazewal 2017)(Organization 2017). WHO presents that, Ethiopia’s healthcare sector is financed by multiple sources including loans and donations from all over the world (Organization 2017).

The triage system in some Ethiopian Hospitals for example, at the emergency triage in the new St. Paul’s Hospital patients will be registered and triaged at the same time. At the current St. Paul’s Hospital, patients need to check in at registration then speak with a triage nurse, and go back to complete their registration. At the new hospital patients will only have to make one step because registration clerks and the triage nurses will be working in pairs at the triage desk to quickly check in patients to help patients get to a bed sooner. Currently, the ED is organized by acuity, meaning the seriousness of a patient's illness or injury. Upon arrival patients are moved to a specific zone in the ED that treats their level of acuity so, low-acuity patients are treated in one area, and higher-acuity patients are in another. Patients may need to be moved around if more acute patients arrive or if different equipment is required.

As cited by (Abdelwahab, Yang, and Teka 2017a), the quality improvement study conducted to evaluate the implementation of nurse-led emergency triage in a large Ethiopian teaching hospital using the Donabedian model, the triage nurse gives a value from 0 to 3 on seven parameters (respiratory rate, pulse rate, systolic blood pressure temperature, mobility, level of consciousness, and exposure to trauma), and the total is the patient's TEWS. They use 4 coloring codes Red, Orange, Yellow, and Blue to show the severity level of the cases. As patient charts were collected and retrospectively reviewed by this study, from 251 randomly selected patients, 107 (42.6%) charts were retrieved. From these, only 45/107 (42.1%) contained the triage form filled within the chart. None of the triage forms were filled out completely, the under-triage rate was 30.7% and the over-triage rate was 21.9%.

And another one is triage knowledge and skills among nurses in emergency units of specialized hospitals in Hawassa, Ethiopia: a cross-sectional study(Duko et al. 2019). This study aimed to assess knowledge and skills of triage and associated factors among nurses in the emergency department of Hawassa University Comprehensive Specialized Hospital, South Ethiopia. Among the study participants, 51.5% had low triage knowledge scores, with the mean score being 9.54 (SD = 2.317), and 76.2% perceived their overall triage skill to be at a good level, with a mean score of 95.75 (SD = 9.562).

But still today, in Ethiopia the triage system is performed manually and Ethiopia also lacks human resources with the knowledge and essential skills to support a coordinated emergency medical care system. Moreover, this situation together with the lack of facilities, equipment, and some basic infrastructure for delivering emergency care makes it difficult to provide dedicated emergency care with appropriate triage protocols, rapid diagnosis, and timely treatment.

2.4. The Adama Comprehensive Specialized Medical College(ACSMC)

The ACSMC is one of the earliest medical colleges in Ethiopia as well as in Oromia. As time goes the hospital developed its facilities, technology, service standards, and personnel to become a renowned institution. The Hospital is serving a catchment population of more than 6 million; from five regions (Oromia, Amhara, Afar, Somali, and Dire-Dawa). The hospital has 232 beds capacity and serves an average of 1000 patients/day at six medical case teams (OPDs) and different specialty clinics. The college has governing board and the institutional structure of the college encompasses academic, health services, and administration wings, all of which are accountable to the provost of the college. The Senate of the college, constituting the provost and vice provosts, department heads, and coordinators of different units in the college, generally, the college is administered in a hierarchical management structure as shown in the diagram below.

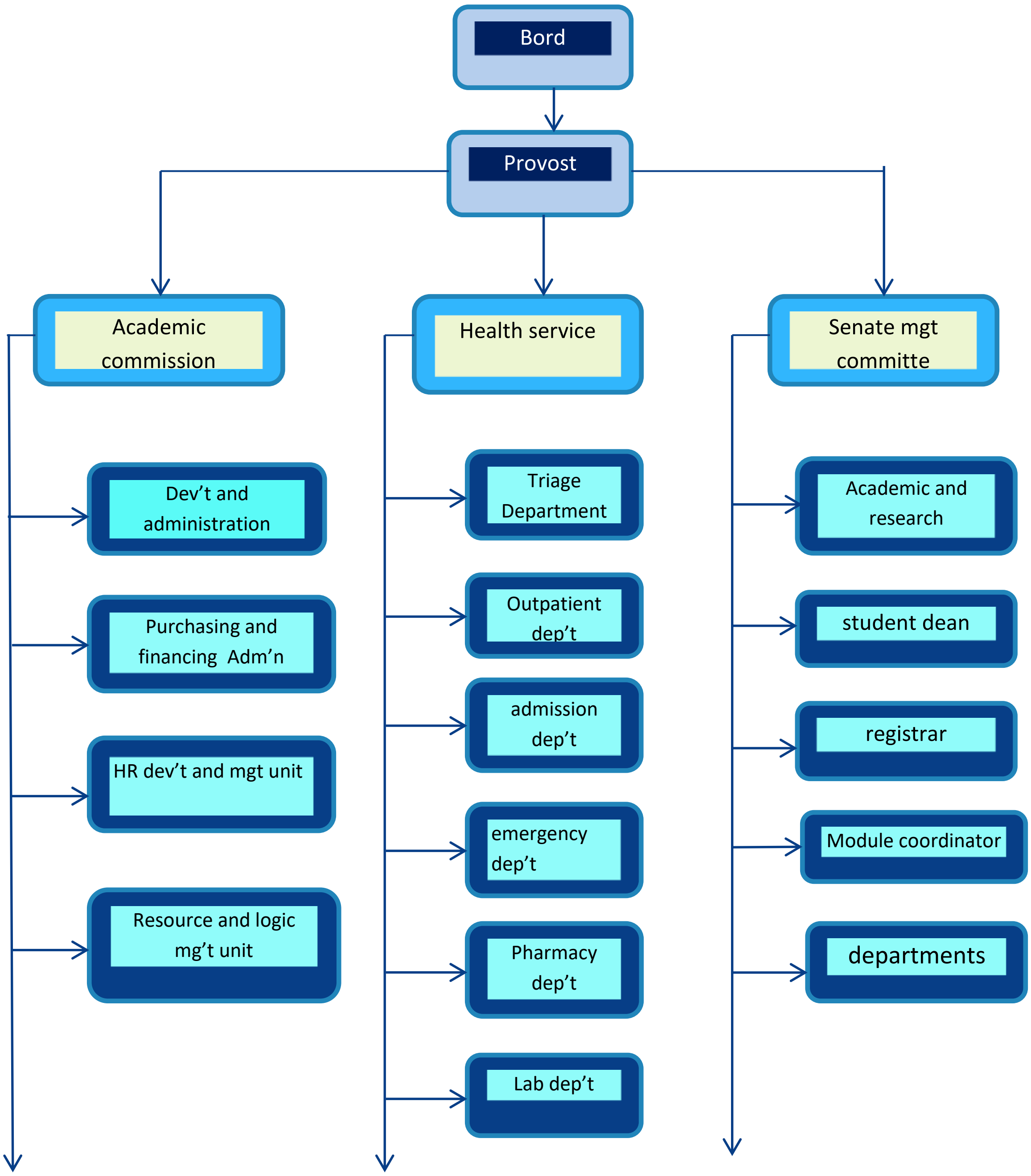

Figure 2. 1 Structure of the Hospital

Many facilities in the hospital would enable for provision of health services, execute the teaching-learning process, research undertakings, and community services. The triage system in the emergency department is one of the different departments that give healthcare in the hospital. This department is the first place where patient contact when come for emergency treatment. Here, triage nurses contact the patient and collect the patient history through interviewing and also using some measuring instruments. After the triage nurse identified the level of illness, then they assign the patients to the corresponding zone. These zones are categorized into four color codes which are: Red, Orange, Yellow, and Green. And, they send them to these different zones. This classification is based on the total TEWS score of the patient history and vital signs.

In this hospital similar to other Ethiopian hospitals, the healthcare system specifically the triage system still now performing manually. But this mechanism has different problems. The first one is, in case of a large number of patients at a time and scarcity of hospital resources patients cannot get effective treatment, many of the triage nurses in the hospital are not trained in the triage system but they work with experience and this leads to inappropriate patient allocation and treatment. And another problem is hospital property misusing this means, when there is over-triaging and under-triaging occurs, patients are misallocated and it leads to the wastage of resources.

Since the use of artificial intelligence in healthcare has the potential to assist healthcare providers in many aspects of patient care and administrative processes, we can use these technologies to overcome challenges faster (Davenport and Kalakota 2019).

2.5. Deep learning

Deep learning is a subtype of machine learning with a neural network consists three or more layers based on a specific task. These neural networks try to make the activity of the human brain by enabling it to 'learn' from huge amounts of Information. DL networks are frequently enhanced as the quantity of data utilized to train them increases(Miotto et al. 2018). DL is different from traditional machine learning in how representations are learned from the raw data because it allows computational models that are composed of multiple processing layers based on neural networks to learn representations of data with multiple levels of abstraction (LeCun, Bengio, and Hinton 2015). There are several kinds of DL algorithms. CNN, RNN, LSTM, etc.

The integration of DL and other AI techniques in the health system will speed up the healthcare system's entire process. DL models will be used to minimize time, effort, and overall cost in tasks such as health data search, the ability to handle complex and multi-modality data, and the highest success rate and low resource consumption indicating that the field of medical health is gradually entering the era of intelligence and digitization. Here deep learning addresses several challenges relating to the characteristics of healthcare data (i.e. sparse, noisy, heterogeneous, and time-dependent). the following are the deep learning that we used in our study.

2.5.1. Recurrent Neural Network (RNN)

RNN is made to function with time series or sequence data common feed-forward neural networks just to handle disparate chunks of data. If the data in a series contain data points where one data point depends on the data point before it, the neural network must be changed to account for these dependencies. RNNs can "remember" the states or data of earlier inputs to create the next output in the sequence (Meng et al. 2021).

This makes them useful for time-based applications (audio, time-series data) as well as natural language processing RNNs perform well in situations where sequential information is critical since, without it, the meaning or grammar may be misinterpreted. Applications include image captioning, language modeling, and machine translation (Nguyen, Anh, and Yang 2019).

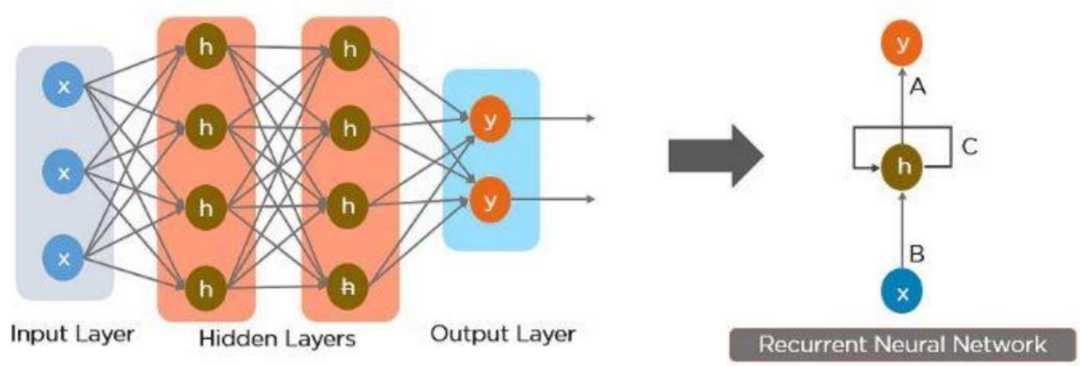

Figure 2. 2 Recurrent Neural Network (RNN) Architecture

At each given time, the output is collected from the network to enhance it. The RNN was developed in response to many concerns with the feed-forward neural network: cannot handle sequential data, just evaluates the current input and does not remember the earlier inputs. A CNN is trained to recognize patterns over space, whereas an RNN is trained to detect patterns across time.

2.5.2. Long Short-Term Memory Networks(LSTM)

When working with longer data sequences, standard recurrent neural Networks (RNNs) suffer from short-term memory due to a vanishing gradient problem. They have more powerful RNNs can maintain and carry over critical information from earlier sections of the sequence. Long short-term memory (LSTM) and gated recurrent units (GRU) are the two most well-known versions (Dobilas, 2022). Recurrent neural networks with order dependency, such as LSTM networks, can figure out sequence prediction issues. We used the LSTM because its recurrent unit is substantially more complicated than the RNN recurrent unit, which improves learning. LSTM has 3 main gates: forget gate, the input gate, and the output gate (Shraddha Shekhar, 2021). In LSTM, we may use multiple inputs to identify which class it belongs to. The LSTM model will be able to discern the true meaning of the input text and deliver the most correct output class if the right layers of embedding and encoding are employed.

2.5.3. Bidirectional gated recurrent unit (Bi-GRU)

BiGRUs are unique in that they can be trained to retain knowledge from the past without having to wash it away over time or delete information that is irrelevant to the prediction(McGranahan et al. 2017). It is a sequence processing model made up of two GRUs. One takes information in a forward direction, whereas the other takes it backward. Only the input and forget gates are present in them. There is no specified memory unit in GRU. The network and the memory unit are merged. It takes into account both past and future information in the patient data sequence to make an accurate prediction about the expected number of patients in the ED at different times of the day or week.

2.5.4. Attention mechanism

The attention mechanism is one of the algorithms that we select because they attempt to mimic the comparable behavior of selectively focusing on a few important items while disregarding others in a neural network (Hore and Chatterjee 2019). And, a neural network in which it exists is a computer program that attempts to emulate the operations of the human brain in a simplified fashion. Since the decoder would only have limited access to the data provided by the input when utilizing a fixed-length encoding vector, the attention technique was developed by(Chorowski et al. 2015) to address this bottleneck issue. Instead of using the entire sequence to predict an output, the model just uses portions of the input where the most important information is concentrated. In other words, it only pays attention to a small number of inputs. Utilizing this structure, the model may focus only on the most important portions of the input sequence and therefore understand the relationships between them. As a result, the model can handle longer input phrases more easily(Acharya et al. 2014).

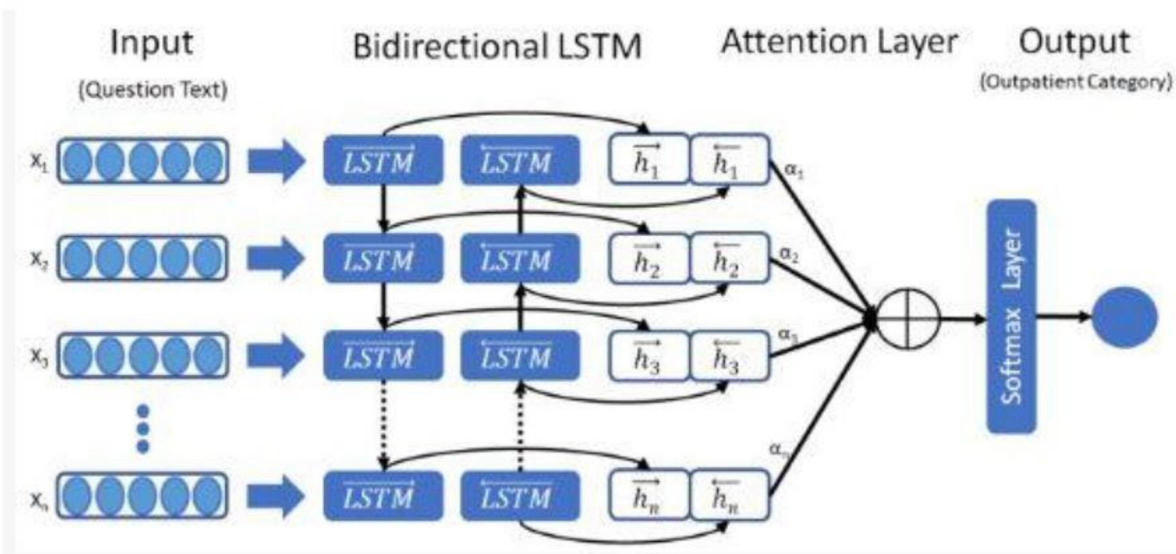

Figure 2. 3 Architecture of attention-based Bi LSTM model (Shen et al. 2020).

2.5.5. Convolutional Neural Network (CNN)

CNN is a feed-forward artificial neural network (ANN) that employs a form of multilayer perceptron designed to need minimal preprocessing and to make use of local connection and weight sharing, which dramatically decreases the number of parameters required to learn in the network. CNN is just a sequence of convolutional layers with non-linear activation functions added to the output such as ReLU, Tanh, or SoftMax. CNN in number classification is used by applying a series of convolutional, pooling, and fully connected layers involving one-dimensional convolution with a small size kernel to extract features, and max pooling to condense or summarize the features extracted from the convolution. Finally, the fully connected layer takes the features through activations and fits into the training data, and makes predictions (Q. Liang et al. 2019). Assume that the input text sequence has n inputs, each of which is represented by a dimension word vector. the majority of the CNN calculation is as follows.

- Convolution operations on the inputs are carried out using multiple one-dimensional convolution kernels that are defined. The correlation of various numbers of neighboring data may be captured using convolution kernels of various widths.
- After all output channels have been subjected to max over-time pooling, combine the pooling output values of these channels into a vector.
- Through the fully connected layer, the concatenated vector is changed into the output for each category. To address overfitting, a dropout layer might be applied.

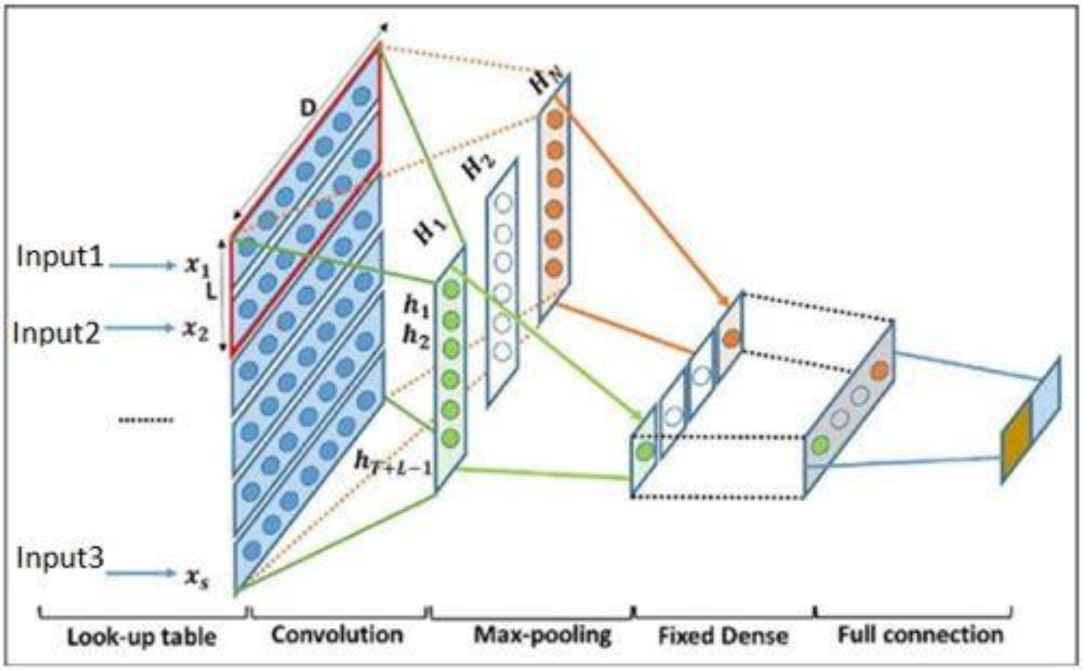

Figure 2. 4 1D CNN model architecture

2.6. Feature Extraction

Feature extraction, which extracts information to represent a message is the foundation of a large number of data processing algorithms (S. K. Singh, Tang, and Tachiev 2013). It is a method that involves selecting a group of features from a variety of effective approaches to lower the dimension of feature space. After the original data has been cleaned, it must be converted to characteristics that may be utilized in modeling. Before modeling we employ a specific way of giving weights to certain phrases in our content. Individual Inputs are represented numerically since our dataset features numbers that are easier for the computer to process(Xiao et al. 2021).

2.6.1. CNN(Convolutional Neural Network)

CNN can learn features directly from the raw input data by applying a sliding window over the input and learning local patterns and structures. To use the CNN for feature extraction on the number data we have to preprocess the data to convert it into a suitable format. Then train the CNN on the suitable format of the dataset and during this time, CNN would learn to identify patterns in the data that are relevant to the task. After training the CNN can be used to extract features from new numerical data. This involves passing the data through trained CNN and using the output of one of the Intermediate layers as a set of features that can be input to the Deep learning model for classification purposes.

Feature Selection: this is a technique used to select a subset of relevant features. It reduces overfitting and may lead to performance improvement. Furthermore, a model becomes easier to comprehend when it has fewer variables (Figueroa Barraza, J., López Droguett, E., & Martins, M. R. (2021). There are numerous techniques for feature selection. This is useful for finding accurate data models. Since an exhaustive search for an optimal feature subset is infeasible in most cases (Jović, Brkić, and Bogunović 2015).

2.7. Related Works

The possibility of applying machine learning and other new technologies to the healthcare profession is a very promising and lucrative prospect and a lot of research is being done in this area using different techniques. Because there is a lot of significant evidence that these technologies could replace or support human activities. Recently, advancements in artificial intelligence (AI), particularly machine learning (ML), and deep learning (DL), have provided us with the tools to automatically analyze health information and predictive models of medical outcomes to create on given properties. Given the large number of health medical documents that healthcare professionals must sift through to retrieve the necessary information on a date or the repetitive nature of reviewing a patient's treatment, the combination of ML and other technologies can play an important role in this area. Several research studies have focused on building a model to predict the outcome of the triage system in emergency departments. In this section, different related works have been conducted locally and globally that are done by different researchers in the health domain using machine-learning techniques that are closely related to emergency department triage prediction have been reviewed.

(Abdelwahab, Yang, and Teka 2017b) proposed a quality improvement study of the emergency center triage. The study was conducted to evaluate the implementation of nurse-led emergency triage using the Donabedian model. They introduced an adapted version of the South African Triage Scale. They collect Patient charts and reviewed them retrospectively. Presence and proper completion of the triage form were appraised. Triage level was abstracted and compared with patient outcome (admitted to the hospital or died and discharged alive from the emergency center) to quantify over- and under-triage. From 251 randomly selected patients, only 45/107 (42.1%) contained the triage form filled within the chart. None of the triage forms were filled out completely. From 13 (28.9%) admitted or deceased patients, the under-triage rate was 30.7% and from 32 (71.1%) patients discharged alive from the EC the over-triage rate was 21.9%. Aside from potentially disastrous impacts on the quality of patient care, they haven't drawn any generalizable conclusions from the study results. And also, they didn't include all the important features.

(Rominski et al. 2014) proposed the implementation of the South African Triage Score (SATS) in an urban teaching hospital in Ghana. The work was done for evaluating the accuracy of the nurses by SATS at KATH. They conducted the cross-sectional study for years 12 and above with complete triage form information. As a result, the study shows that after that Expert assessment of all 903 adult patients, 52 (5.7%) patients were assessed as mistreated by the triage nurse; 49 were under-sorted and 3 were over-sorted. They used a small amount of data which is 909 and didn't include all relevant variables. However, there is a need for the predictive validity of the algorithm, they have not built it.

Berhanu desu(Ababa 2019) proposed an evaluation of triage implementation and factors influencing it in patients visiting the specialist emergency department of Tikur Anbessa Hospital. This study identifies whether the triage tool is appropriately populated, measured, and documented, whether the TEWS score is calculated and treated in respected areas, and the factors influencing its implementation. In this study, they have produced that from the total eligible sample, 43.7% were incorrectly triaged. Among them, 25.22% (n=29) were over triaged while 74.78% (n=86) were under triage. Nevertheless, the study is not to predict the triage system but rather than focusing on the factors that influence the system.

(Roquette et al. 2020) presented a Prediction of admission in the pediatric emergency department with deep neural networks and triage textual data. The study was to propose and compare a model for admission prediction by using both structured and unstructured data that exist in triage. From a total dataset, the admission rate is 5.76%. They used a model that consists of a 2-stage architecture with a deep neural network (DNN) to extract information from unstructured data with a gradient boosting classifier. And, by using a combination of these models, they got a result of 0.89 for the Area Under the Curve (AUC) in the test data. However the study was only for admission prediction which is domain-specific, not including the level of emergency prediction, and they didn't include common triage variables.

(Goto et al. 2019) proposed machine learning–based prediction of clinical outcomes for children during ED Triage. The research was conducted to examine the performance of machine learning approaches to predict clinical outcomes and disposition in children in the ED and to compare their performance with conventional triage approaches. The outcomes were critical care and hospitalization. As predictors, they derived 4 machine learning–based models: lasso regression, random forest, gradient-boosted decision tree, and deep neural network. As a result, Of eligible ED visits by children,47.1 % had a critical care outcome and 45.2 % had a hospitalization outcome. Nevertheless, the patient characteristics used are not common or known in the emergency department and it leads to poor generativity. As well as the data used for this study was not currently collected. Additionally, there is no clear justification for the result found in the study.

Summary of Related Work

In this section, we will cover various related works done in different places using different techniques. In general, many researchers have tried to create a suitable environment for medical care in different countries especially related to the triage system. Many of them have focused only on the binary classification (inpatient or discharged), or (ICU or Hospitalization). But have not specified the classification of severity levels of emerging cases. But without the proper classification of the emergency levels, it is difficult to treat the patient correctly.

Table 2. 1 Summary of related work

S/N	Title	Author	Objective	Gap	Algorithm
1	Construct an Optimal Triage Prediction Model: A Case Study of the ED of a Teaching Hospital in Taiwan	Wang Shen-tsu (2013)	To effectively predict anomaly detection and triage	There is an exclusion of common features All important The study was focused on anomaly detection rather than the severity levels of the triage outcomes prediction.	By Integrating (PCA) and (SVM), BPNN and (RBFN)
2	Early triage of critically ill COVID-19 patients using deep learning	Wenhua Liang, Jianhua Yao, Ailan Chen, (2020)	develop a deep learning-based survival model To early predict the risk of COVID -19	They included only three levels of triage classes. They used small size dataset They excluded very common triage features.	Cox proportional hazards model (CPH),CNN, LASSO)
3	Machine Learning–Based Prediction of Clinical Outcomes for Children During E D Triage	Tadahiro Goto, Jr,Dr PH:Mammad Kamal Faradi (2019)	To examine ML approaches and to compare them with conventional triage approaches	It is a binary classification, not specifying different classes of triage. They didn't build a predictive Validity evaluation for the algorithm they used.	lasso regression, RF, gradient-boosted decision tree, and DNN

4	Prediction of admission in the pediatric emergency department with DNN	Bruno P.Roquette,Hitoshi Nagano, (2020)	To propose and evaluate ML algorithm to predict admission in ED.	They did not include most contributing variables It is only for admitted patients in the hospital.	SVM, ElasticNet, DNN, Caboose
5	Machine learning for prediction of septic shock at initial triage in the emergency department	Scatt Levin Ph.D., Matthew Toerper BS, February (2020)	Assessing the performance of ML-based triage tools in screening patients with septic shock in ED.	The study is domain-specific; only for septic shock screen They used a small number of features. And also the features are not described.	(qSOFA) score and (MEWS), score

Generally, as we tried to describe several studies were conducted for the prediction of triage in emergency departments using different approaches in different countries. However, the previously conducted researches were with some gaps. Firstly, their study was not precisely predicting the triage severity levels of triage outcomes such as Red, Green, and Orange rather than determining a binary classification as either ICU or Hospitalization. Secondly, the dataset used was small in size, noisy, and not recently collected data. Thirdly, they are domain-specific and excluded very contributing classes of data. And these results in accuracy reduction and poor generative for the models. Besides, no study conducted in the context of our country. Therefore, this thesis is to build a Deep-learning based model that can predict the severity level of triage classes as a solution.

CHAPTER THREE

3. METHODOLOGY

3.1. Introduction

As mentioned in chapter one, the objective of this study is to build a deep-learning-based model to predict triage outcomes in the emergency department. In this chapter, we have discussed procedures that are used in accomplishing this objective. This includes methods for collecting, developing, and preprocessing data utilized by different deep learning models, splitting data into train and test, building deep learning models and feature extraction methods, and the model performance evaluation approach used. Additionally, it covers systematic methods and approaches to conduct the study starting from identifying research areas up to deploying a model.

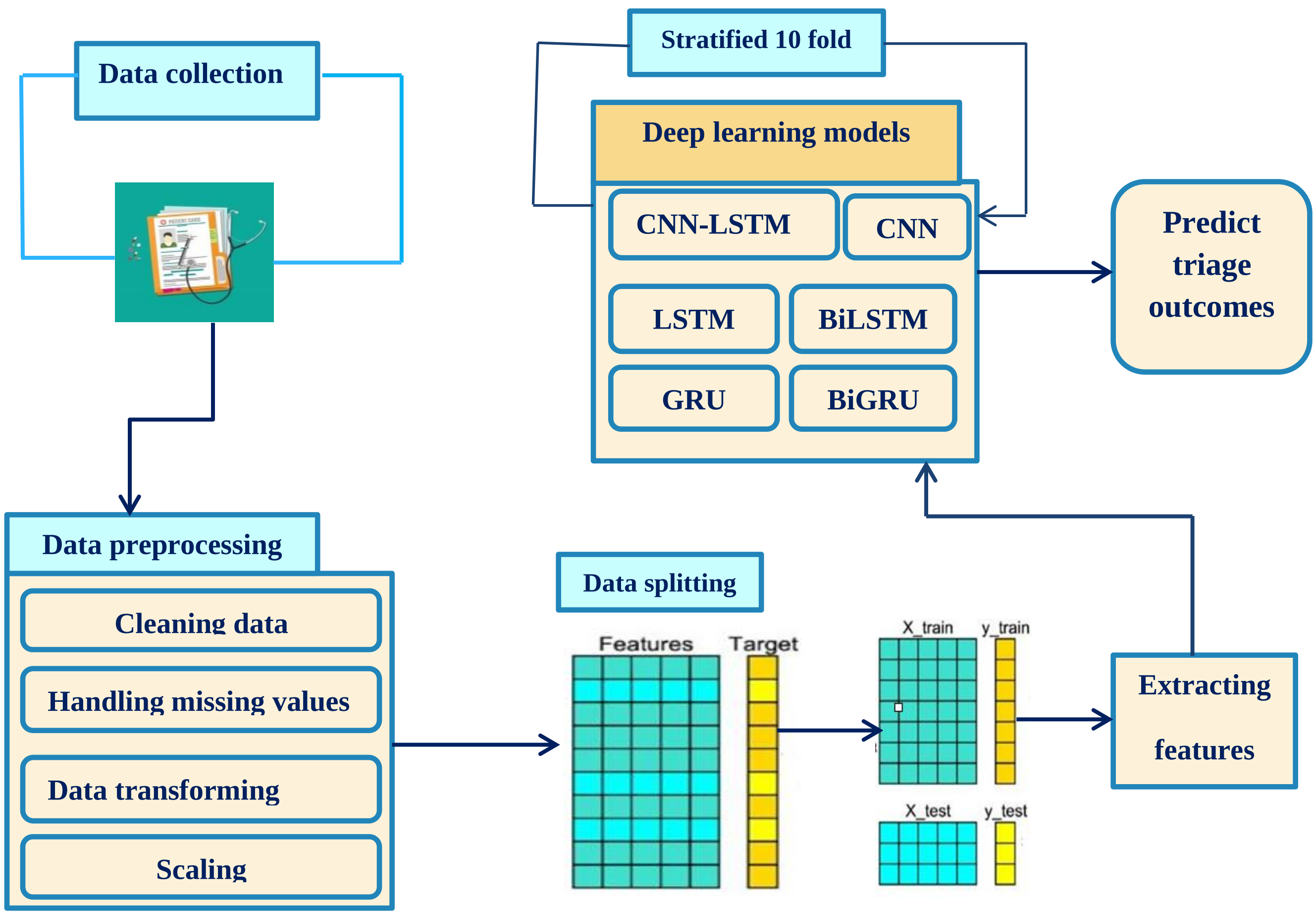

Figure 3.1 Research Methodology

3.2. Research Design

Research design is a plan designed to answer raised research questions. So, to answer these research questions, we have designed our research plan as described in Figure 3.2. There are several kinds of research design approaches such as qualitative, quantitative, and participatory. Our study followed quantitative or experimental research for the design approach which includes identifying research objectives and building deep learning models to validate the concept. Quantitative research design approaches used in this study find the relationship between two variables i.e., independent and dependent variables in data samples (Kamiri and Mariga 2021).

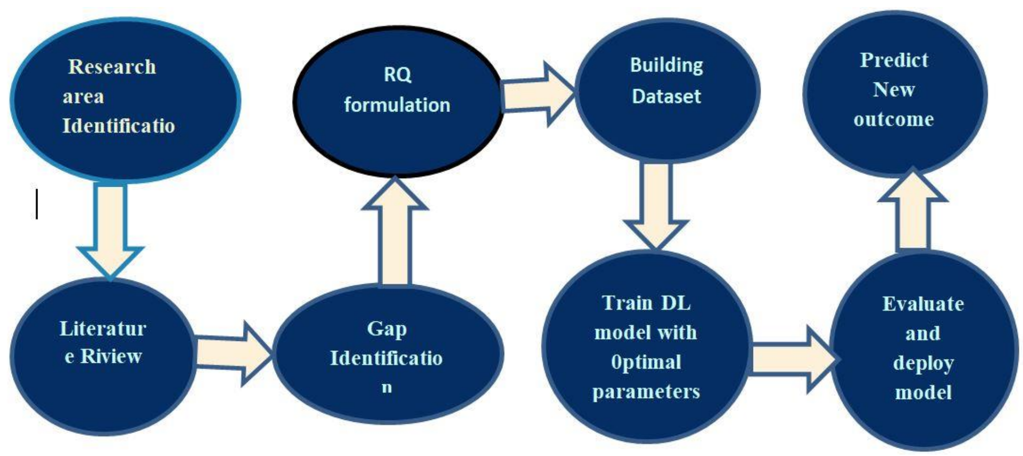

Figure 3.2 Research design procedure of EDTP

3.3. Building Datasets

The health domain is too complex and too broad to interpret by health professionals. Due to this, there is no excessive dataset and pre-trained models in our country's hospitals. As the case regarding dataset scarcity in the study predicting the emergency department triage outcome for the emergency department is also with the same above-mentioned problems, it's needed to build a new dataset because there is no locally published or prepared dataset for this purpose. The data collected for this study is in the form of hardcopy documents that cannot be processed directly. And, since there is no OCR or other system to convert the handwritten texts we prepared the data by typing from the patient card.

3.3.1. Data Source

The data to conduct this research was collected from Adama Comprehensive Specialized Medical College. The hospital is envisaged to conduct problem-solving research and provide community services. The hospital is led by the board and includes three basic wings (Academic and research, Health service, and development and administration). The hospital has 232 beds capacity and serves an average of 1000 patients/day at six medical case teams (OPDs) and other specialty clinics. And in this hospital, the emergency department triage filtering is one of the fields. In the emergency department, the triage filter is given by triage nurses. The triage nurses filter and send the patients to the next class i.e., one of the four classes: Red, Orange, Yellow, or Green depending on the level of the case which is determined by the total TEWS score of the patient.

3.3.2. Data Collection

The data for this study was secondary and collected from the patient history data without direct communication with the patient. The data collected and used in this study included patients' records which are collected by the Adama Comprehensive Specialized Medical College emergency department. When we collect this data first, we collected the identification number of the patients registered in the emergency department and by using this number we find the patient card. The patient card contains information like patient name, age, sex, kebele, identification number, etc. The collected data from the hospital includes trauma cases(car accidents, wounds in battle, blooding) and other cases including surgical, medical, and the like. In this study the collected cards of patient history data from the emergency department are in hard copy and by collecting them we prepared a dataset. The total sample size is 6000 cards and all of them are emergency department cases for adult patients with age 15 and above. Since we collected the data selectively for its usability, from a total sample most of them are usable. But some of the data are non-valid which include patient cards that have no triage form at all, others have the form but the form is not filled and the left is unclear or understandable.

The data collected for this study contains a total of twelve features which include eleven input features and one target variable. These input features include mobility which indicates the status of the patient whether he or she comes by walking, with help, or by stretcher. Hypertension rate, Respiratory rate, systolic blood pressure(SBP), and Central nervous system(CNS) represent to which action the patient can give a response and it is either alert, verbal, pain, or unresponsive. Another input feature is trauma which shows whether the patient has a trauma case or not represented by yes or no. And also, age, sex, Oxygen saturation which is the amount of oxygen level in the red blood cell, body temperature, and pain score which indicate the level of the patient's illness. We used these features because they are a common feature used for triage filtering in our country hospitals. The last feature is a target which is an outcome that includes four classes. The first class is green which shows that the patient's illness is a non-urgent condition for TEWS score of 0-2. The second class is yellow and it is for less urgent illnesses that have a TEWS score of 3-4. The third class is orange which indicates the patient is with urgent illness for a patient with a 5-6 TEWS score. And, the fourth class is red which shows the patient is in a life-threatening condition for patients with a TEWS score of 7 and above.

Table 3. 1 initially collected data from ACSMC Emergency Department

#	Emergency department Registered patient cards	Number of cards	Description
1	Trauma cards	2000	Accepted
2	Other cards(surgical, medical, etc.)	3000	Accepted
3	Invalid cards	1000	Not Accepted
4	Total	6000	

3.3.3. Data Preparation

The data preparation is the important step after data is collected for building the dataset. In the real world, raw data are often inconsistent, incomplete, and lacking specific characteristics. They are also more likely to lead to mistakes. Therefore, the collected data should pass through some data processing to alleviate the data quality issue, which may have a great effect on building models. The data collected by Adama Comprehensive Specialized Medical College (ACSMC) contains unstructured data. Therefore, once collected, data needs to be cleared and prepared. It starts with selecting valid patient cards. The next step is typing and storing these pieces of information in Excel form. During this, some cards have missed fields (not full information), unreadable forms, and duplicated data will be eliminated. So, after performing data preparation, 1000 records are eliminated and we have got 5000 records. The amount of prepared data is 5000.

3.3.4. Data Preprocessing Techniques

Whenever collected raw data is difficult to analyze, we have to transform these raw data into a data format that is analyzed and understood easily by a computer program by applying data preprocessing techniques. It is used to convert data into a clean dataset (Yang et al. 2020). This method is used to make ready the dataset for the models to use it. This involves selecting the most relevant features for analysis and removing irrelevant or redundant features. In this study, the following methods are used in pre-processing procedures to make them more suitable for deep learning.

Cleaning: removing a special character, or punctuation, removing or correcting missing or incorrect data, removing duplicates and extra space from our dataset.

Feature scaling: normalizing the range of features to improve the optimization process by helping algorithms to reach the minimum of the cost function more quickly.

Data transformation: to convert data into a more usable format, such as converting text data to numerical data using encoding techniques.

Handling Categorical Data: In this step, real data is transformed into the required format. The collected data consists of real data, decimal values, and nominal data. Therefore, in this step, nominal data is converted into numerical data. For instance, ‘Gender’ has a nominal value that can be labeled as 0 for females and 1 for males. After preprocessing the data then the resultant CSV file comprises all the integer and float values for different EDT-related features.

3.4. Feature Extraction Method

Feature extraction is a crucial step in building an effective predictive model for the prediction of emergency department triage outcomes. It involves selecting and transforming the relevant information from a numerical dataset into a set of features that can effectively represent the data and capture its underlying patterns. It helps to select relevant information, reduce dimensionality, capture patterns, handle non-linear relationships, and improve interpretability, all of which contribute to better prediction performance and actionable insights. Because The field of computer science, machine learning, and DL focuses on teaching computers to process huge amounts of human language input.

3.5. Deep learning models

Deep learning is a type of machine learning and artificial intelligence (AI) that mimics the way humans acquire certain types of knowledge. And, is an important part of data science, which includes statistics and predictive modeling, and it is extremely beneficial for tasks involving the collection, analysis, and interpretation of large amounts of data which is used to make the process faster and easier. The deep learning algorithm can learn without human supervision and can be used on structured and unstructured data types. While traditional machine learning algorithms are linear, based on this theory, we propose the following different deep learning algorithms:

3.5.1. RNN-Based Model

A recurrent model-based approach in deep learning is the most engaging. The ability of RNN-based models to interpret the Input as a collection makes them ideal for identifying input dependencies and categorization. The most common RNN variant, long short-term memory (LSTM), is designed to better capture long-term situations than other RNN variations.

Gated recurrent units (GRU) are the alternative effective option. In this study, bidirectional LSTM and GRU models are used in addition to LSTMs and GRUs.

3.5.2. Long Short-Term Memory (LSTM) Model

The LSTM models can be used to capture temporal dependencies and make predictions based on a sequence of data points effectively. This makes them a suitable choice for predicting emergency department triage outcomes based on sequential data. Its architecture contains memory cells and gates which enable the model to learn and remember important information over time. The model processes the input sequences step-by-step, updating its internal state at each time step and using it to make predictions.

Similar to another neural network, LSTM comprises a few layers that help in pattern recognition and learning for improved performance. The fundamental purpose of LSTM can be thought of as capturing the necessary information while discarding the information that is not necessary for the prediction.

3.5.3. Hybrid CNN-LSTM Model

It is a hybrid classification model that consists of long short-term memory (LSTM) and convolutional neural network (CNN) models. The proposed CNN-LSTM model included an LSTM model for classification and a CNN model for feature extraction.

3.5.4. CNN-Based Model

Convolution Neural Networks (ConvNets) are neural networks that reduce the original phrase matrix into its low-dimension matrices by convolving (rolling over it). They perform it by using a series of filters of various sizes and shapes. A CNN is made from Two primary layers: the first is, a convolution layer that obtains features from the data, and a pooling layer that reduce the size of the feature map.

3.6. Model Evaluation

3.6.1. Prediction Model Evaluation

To ensure that the model is compatible with the prepared dataset and works well on new unseen input data, the prediction model must be evaluated through model evaluation techniques. The model performance evaluation is used to estimate the overall accuracy of a model on unseen sample data(Raschka 2018). There are two types of performance evaluation techniques; holdout and cross-validation(D. Singh et al. 2019).

In this study, the cross-validation performance evaluation method is used. It performs by dividing data into two partitions for evaluating and comparing models: one part is used to train a model and the rest part is to test or validate the model. The cross-validation technique is the most widely used evaluation method to avoid over-fitting (Stone 1974). There are different types of cross-validation for evaluation methods.

K-fold cross-validation: in the k-fold cross-validation method, the first samples are randomly divided into k equal sub-samples. Of the k sub-samples, one subsample remains as the validation data set for testing the model, and the remaining k - 1 sub-sample serves as the training dataset (J. Brownlee 2018). The average of k-scored accuracy called CV accuracy is served as a performance metric of the model.

Stratified K-Fold Cross Validation: this is a technique for an enhancement of the cross-validation approach used for classification issues. Cross-validation is used to conserve the imbalanced class distribution in each fold and It forces class distribution in each split or division of the data to match the distribution in a complete training set(Sharif et al. 2020). In this study, since the data collected contains imbalanced class distribution, a stratified k-fold cross-validation technique was used.

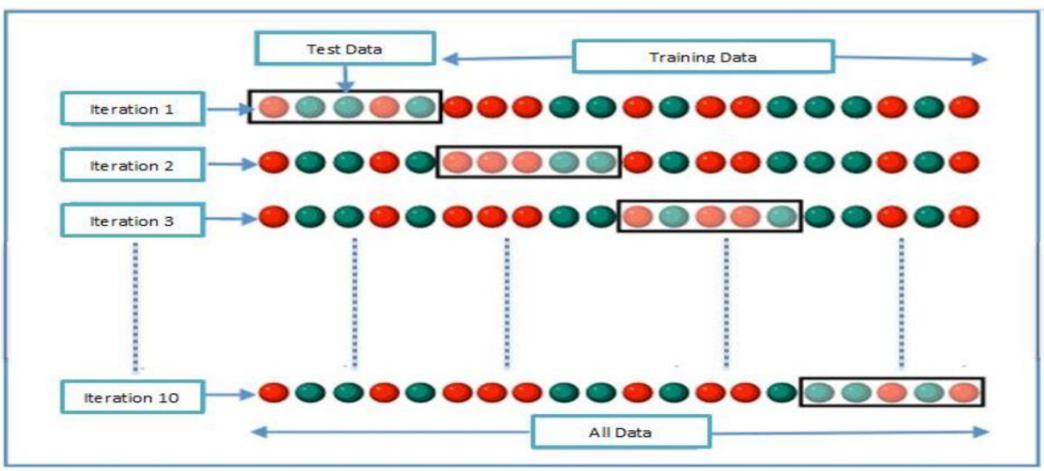

Figure 3.3 10-fold cross-validation

3.6.2. Classification Metrics

Since classification algorithms do not provide the same results, predictive classification model evaluation metrics are important to measure the performance of the model. There are different kinds of classification model evaluation (M & M.N, 2015). From these classification metrics, f1-score, precision, recall, and confusion matrix are used in this study with cross-validation. The followings are the common terms in performance measures:

In multiclass classification, the concepts of TP, TN, FP, and FN are not directly applicable as they are specific to binary classification.

True Positives (TP): TP represents the instances that are correctly predicted as belonging to a specific class.

True Negatives (TN): TN represents the instances that are correctly predicted as not belonging to any of the classes.

False Positives (FP): FP represents the instances that are incorrectly predicted as belonging to a specific class.

False Negatives (FN): FN represents the instances that are incorrectly predicted as not belonging to a specific class.

A. Confusion Matrix

The confusion matrix in Table [3.1] is used to explain how these parameters are calculated. Here, the values for false negative (FN) and false positive (FP) give, for a given patient, the number of classes that are incorrectly predicted. Similarly, the values for true negative (TN) and true positive (TP) gives the number of classes correctly predicted for four classes.

Table 3. 2 Confusion matrix for multiclass classification

		Prediction			
		Class 1	Class 2	Class 3	Class 4
	Class 1	Class 1 True	Class 2 False	Class 3 False	Class 4 False

actual	Class 2	Class 1 False	Class 2 True	Class 3 False	Class 4 False
	Class 3	Class 1 False	Class 2 False	Class 3 True	Class 4 False
	Class 4	Class 1 False	Class 2 False	Class3 False	Class 4 True

B. Accuracy

It is the most common and popular performance evaluation metric for classification tasks and is defined as the percentage of correct predictions for the test data.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \qquad \textbf{\textit{Equation 3- 1}}$$

C. Precision

Precision is a measure of the likelihood of getting the correct positive class classification, and it is how close a measured value is to each other.

$$Precision = \frac{TP}{TP+FP} \qquad \textbf{\textit{Equation 3- 2}}$$

D. Recall

A recall is the total number of true positives divided by the number of all relevant samples, or it is the ratio of a true positive to a true positive plus a false negative.

$$Recall = \frac{TP}{TP+FN} \qquad \textbf{\textit{Equation 3- 3}}$$

E. F1-score

The harmonic means of the precision and recall values for a classification task is known as the F1-Score. When false positives and false negatives are more important than True Positives and True negatives, the F1 score is a better choice for measuring Accuracy. The following is the F1-Score formula:

$$F1\text{-}score = 2 * \left(\frac{precision * recall}{precision + recall}\right) \qquad \textbf{\textit{Equation 3- 4}}$$

3.7. Research Tools

The following tools were utilized to create the proposed model.

3.7.1. Data Preparation Tools

MS-office OneNote: Along with applications such as Word, Excel, and PowerPoint, OneNote is a component of Microsoft Office. OneNote is a limitless digital notebook.

MS-excel: Microsoft Excel is a piece of software produced by Microsoft, that allows users to do simple math, work with graphing tools, make tables, etc. It was used to save the selected and cleaned data as a single file in a tabular format.

3.7.2. Design Tools

Edraw Max is UML builder software that is used to create illustrations using ready-made symbols and templates. It enables us to import drawing into different file formats including Word, HTML, PDF, PPT, etc. And also, provides a user-friendly interface similar to MS Word and allows to share designs anytime and anywhere. Particularly, we used it to draw organizational structure, proposed system architecture, and research methodology workflow process.

3.7.3. Hardware Tools

We used a personal computer with an Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz 2.30 GHz, 8 Gigabytes of physical memory, 1 Terabyte of hard disk storage capabilities, and 64 bits of Windows 10 Pro operating system to see the results of the experiment.

3.7.4. Implementation Tools

The following Data analytics and visualization tools are used to implement the proposed models.

Pandas: Python-based toolkit pandas is used to offer a wide range of utilities, from parsing various file formats to transforming an entire data table into a NumPy matrix array.

Matplotlib: we used it to build static, animated, and interactive visualizations simply.

NumPy: it's a Python module that offers support for multidimensional, big arrays and matrices, as well as the ability to perform on them.

Scikit-learn: It is the most useful and robust library in Python that is used to provide several powerful tools for statistical modeling and machine learning via a Python consistency interface including classification, clustering, regression, and dimensionality. We applied it to train our model with stratified k-fold cross-validation and the generation of classification reports based on the results of each fold's metrics.

Jupyter Notebook: Using the free, open-source, interactive Jupyter application, it is possible to combine software code, computational results, explanatory text, and multimedia resources in a single document(Perkel 2018). During developing the proposed model, we used the Jupyter Notebook to write, run, and process data using a Python program.

Keras: It is a lightweight Python package in deep learning which runs on top of TensorFlow. The application was designed to facilitate the research and creation of deep learning models as rapidly and straightforwardly as possible(W. Brownlee et al. 2020). As it provides a high degree of abstraction in Python front and the option of numerous back-ends for computation, Keras is very straightforward to understand and

Summary

Our objective is to predict the emergency department triage outcome using a deep learning approach. A variety of research methods, tools, and techniques have been used to achieve this objective. We used a quantitative research method for our study because the study answers the research question through the experiment. It is necessary or mandatory to build a dataset for experiments since there is no dataset to train the model. As a result, the emergency department triage dataset has been built using the patient documents that are collected from ACSMC. Various techniques have been used to clean up and make suitable data for the deep-learning models: Convolutional Neural Network (CNN), Long Short-Term Memory Network (LSTM), and Hybrid CNN-LSTM Model are used in this study. The evaluation method like cross-validation, confusion matrix, accuracy, precision, recall, and f1 score was used to evaluate the model. To design model architecture, build the dataset, and write implementation code different tools, packages, and libraries were used.

CHAPTER FOUR

4. PROPOSED EMERGENCY DEPARTMENT TRIAGE OUTCOMES PREDICTION MODEL

4.1. Introduction

In this chapter, the design and overview of the proposed solution have been discussed. The proposed model architecture shows how the emergency department triage prediction model would be designed and developed to address the research questions raised in chapter one and cover the gaps identified in the literature study part. The proposed solution demonstrates the methods and approaches we used for the thesis. It includes dataset construction procedures, feature extraction methods, and deep learning algorithms that we used. Also, the methods we used to evaluate the models are discussed.

4.2. Model Architecture

The goal of this study is to predict the emergency department triage outcomes. Model architecture is used to show how the emergency department triage prediction model or proposed model is developed. In this, we developed a model that was used to predict the outcomes of the emergency department triage system. In most Ethiopian hospitals, the triage system is classified by color code to indicate the emergency level of the patient case starting from very emergent cases to non-emergent ones. Therefore, the proposed model predicts these different triage outcomes depending on the total calculation of the TEWS score which contains the patient history.

The data is gathered from ACSMC. To prepare the dataset for data preprocessing, as stated in Chapter 3, section 3.1, it first goes through a preparation stage that includes data selection, cleaning, and storing. The model was built on top of the prepared dataset. First, the data preprocessing technique will be applied to the stored data. Preprocessing is about removing unwanted characters, handling null values, and transforming. So that the data can easily be utilized in the next step which is feature extraction. Feature extraction is the process of extracting relevant numerical features while keeping the original data set's content unchanged. CNN is the feature extraction method we proposed in this study with various models. It learns hierarchical representations of the numerical data and captures patterns in the dataset. Next to that, deep learning algorithms are built. To recognize patterns and create predictions, extracted vectors were submitted to LSTM, BiLSTM, Bi GRU, and GRU models. To determine the best-performed model, these models were tested through the stratified k-fold cross-validation approach and other classification metrics.

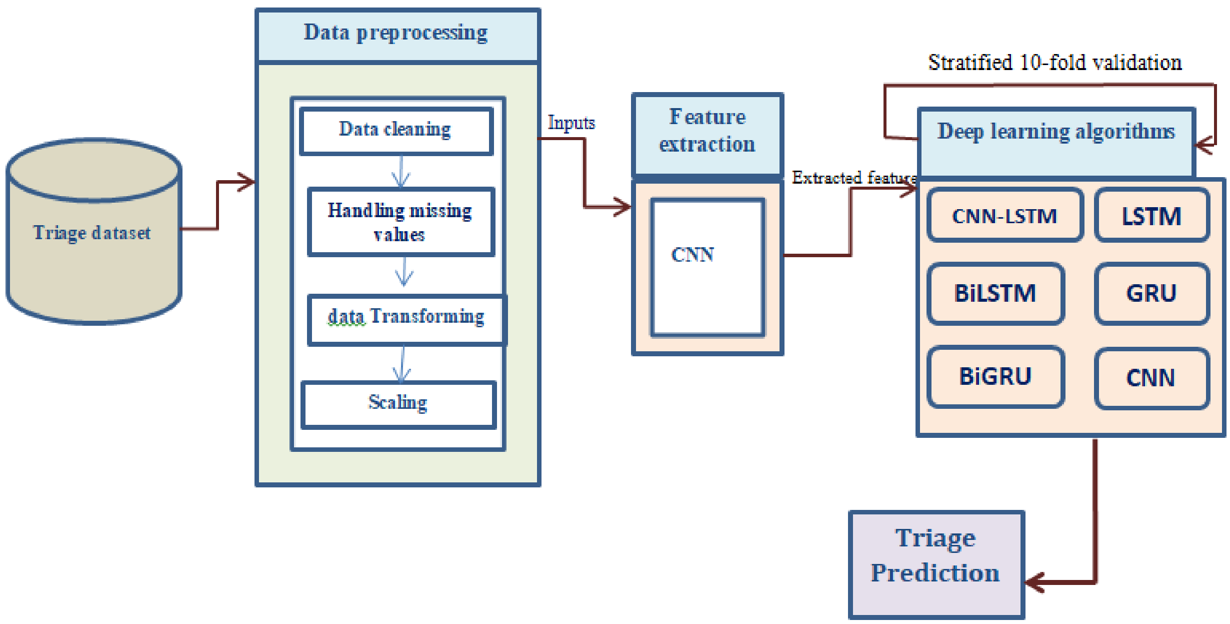

Figure 4. 1 Emergency Department Triage prediction model architecture

The above diagram is to describe an architectural representation of the proposed solution. It shows the overall flow steps followed in this study. First of all, we applied preprocessing techniques such as data cleaning to clean some characters in the DataFrame, handling missing values, transforming categorical data into a suitable format, and then converting the datatypes of the DataFrame. Then after, we give the preprocessed data to the CNN that we used as a feature extraction to extract relevant features. Next, we used these features to train and test different deep learning algorithms on it to predict different classes of the triage. Lastly, we applied stratified

10-fold cross-validation to test the performance of models to select the most performed model with high accuracy and other metrics.

4.3. Data Preprocessing

As the quality of the data and the information that can be gained from it have a direct impact on our model's capacity to learn, data preprocessing is a crucial stage. The preprocessing techniques were applied to the already prepared dataset file and converted into a common separated value (CSV) format with .csv.

4.3.1. Cleaning the Data

In principle, our preprocessing should be identical to the preprocessing carried out before training. The first thing we did was remove punctuation and other special characters from our dataset.

4.3.2. Handling Missing Values

While collecting data some results have missing values. The dataset is created from the data gathered from patient history data, which contains missing values. The missing values in this study were handled.

4.3.3. Handling Categorical Data

The created dataset has some categorical values that need to be transformed to the form 1 and 0. Thus, this transformation task transforms categorical values. label encoding methods were used to handle categorical data in this study.

4.4. Feature Extraction Methods

Both dependent and independent variables are accepted by deep learning algorithms in numerical form. Therefore, we need to use certain feature extraction techniques to make vectors and prepare the data for our DL model. CNN feature extraction techniques were used in this study.

4.5. Deep Learning Algorithm

The goal of our study is to develop a deep learning-based model that predicts the Triage outcomes in the emergency department. In the emergency department triage system, the patients are treated with the priority of their cases. Different emergency levels in the triage system mean it is multiclass with one Target Label(outcome). These classes are Red(most severe) which indicates the severe or most emergent class for a patient with a TEWS score of >=7, Orange(mild-emergent) shows the next emergent case with a TEWS score of 5-6, Yellow(less urgent) with TEWS score of 3-4 and Green(non-emergent) which have TEWS score of <=2. This classification is given depending on the patient histories stored during treatment. These patient histories include vital signs(Temperature, Hypertension Rate, Respiratory Rate, Oxygen saturation, systolic blood pressure), demographic and mobility status, and also, CNS response. To accomplish our goal, we built LSTM, BILSTM, GRU, CNN-LSTM, and BiGRU-based models, which are the most used algorithms in the Health domain.

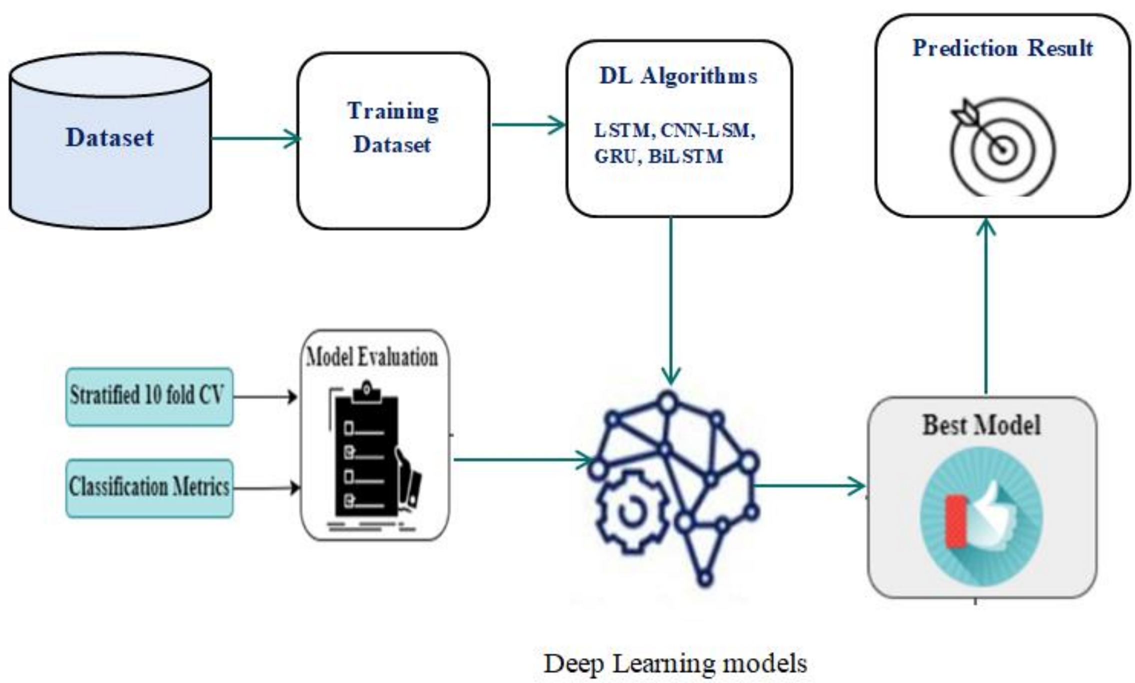

Figure 4. 2 Model training and testing diagram

4.5.1. Hybrid CNN-LSTM Model

in the CNN-LSTM architecture, Convolutional Neural Network (CNN) layers are used to extract features from input data, and LSTMs are used for a sequence prediction. It is possible to use CNN to classify the input and find patterns of several closely related Inputs By adjusting the kernel sizes and concatenating their outputs. Although the Convolutional Neural Networks (CNN)-based data analysis approach may extract important features by pooling, it is difficult to get contextual information but the LSTM algorithm can do that. Therefore, combining CNN with LSTM could provide some better results. We used the hybrid convolutional neural network with a long short-term memory model for prediction. In this, extracting features from the dataset make it faster and more accurate. We used pooling, convolutional layers, and rectified linear activation units (ReLu), as well as dropout layers. Since neurons can get overfit easily, we used the dropout layer and ReLu activation function to handle it. Then we defined the LSTM model that takes the CNN model's output as input parameters. We used a single LSTM layer. Here also we used the activation functions and dropout layer. Unlike CNN, we used the softmax activation function for the LSTM prediction model. The last one was the dense layer.

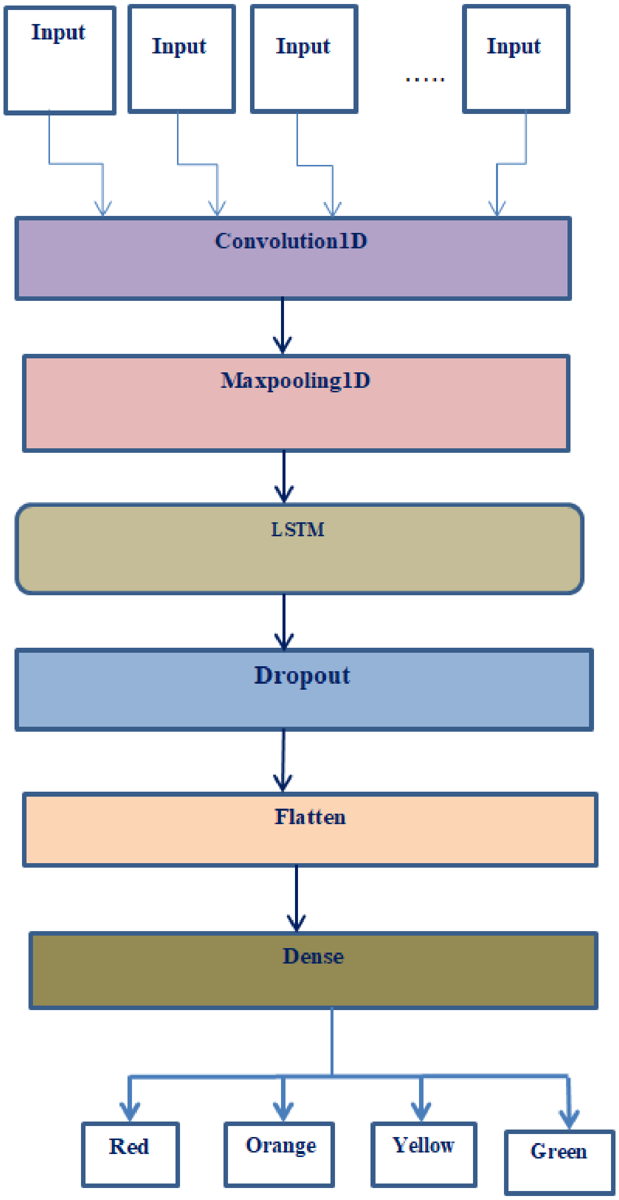

Figure 4. 3 Architecture of the proposed hybrid CNN-LSTM model

4.5.2. GRU Model

The GRU is a kind of RNN that has advantages over long short-term memory (LSTM) in certain cases. GRU is faster than LSTM and it uses less memory. However, LSTM is more accurate when using datasets with longer sequences. GRU has fewer gates and fewer parameters when compared with LSTM and that makes it simpler and faster. But they are less powerful and less adaptable. LSTM has a separate cell state and output that enable it to store and output different information while GRU has a single hidden state that serves both purposes which may result in a limited capacity. GRUs are an improved version of recurrent neural networks. The information that should be passed to the output is decided by the two vectors. The special thing about the GRUs is that they can be trained to keep information from long apart without washing it through time or removing irrelevant information for the prediction.

4.5.3. BiGRU Model

GRU is a kind of newer version of RNN that is widely used for learning sequential data of prediction models. It is capable of handling vanishing gradients which exist in the traditional RNN model. It contains only 2 gates (Reset and Update Gate), and GRU is faster than LSTM. As with every other thing they both have some layers which help them to recognize and learn the pattern in the case for better performance. GRU works in one direction, while BiGRU tries to capture information from both sides, left to right and right to left. This model has a BiGRU layer, a dropout layer, and a dense layer. As recurrent networks are prone to errors, a dropout layer with a 50% error rate was added after the BiGRU Layer.

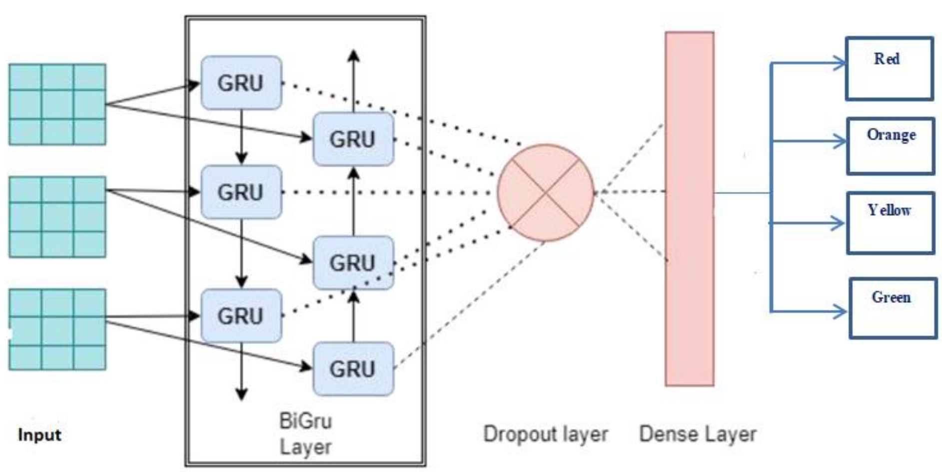

Figure 4. 4 BiGru model architecture for EDT prediction

4.5.4. BiLSTM Model

Bidirectional LSTM (BiLSTM) uses information from both sides and allows input to flow in both directions. One additional LSTM layer is added by BiLSTM which reverses the information flow's direction. Our proposed BiLSTM model's architecture is similar to that of the BiGRU model.

4.5.5. CNN Model

The only way that makes CNN different from other neural networks is its convolutional layer. The model that was used has three layers. A one-dimensional convolutional layer with 32 filters and a kernel size determined by the number of Inputs size to be read simultaneously. In this layer, predefined filters roll over the input matrix and reduce it to a low-dimensional matrix. The output from the convolutional layer is combined in the next layer which is a max pooling layer. The other layer is a flattened layer that converts the output from three dimensions to two dimensions for concatenation. We took advantage of layers that all models share the dropout layer and the dense layer. As depicted in the above architectures, they used to predict one of the four classes of the emergency department by their encoded form for instance, green is encoded as ‘0’, yellow as ‘1’, orange as ‘2’, and red as ‘3’.

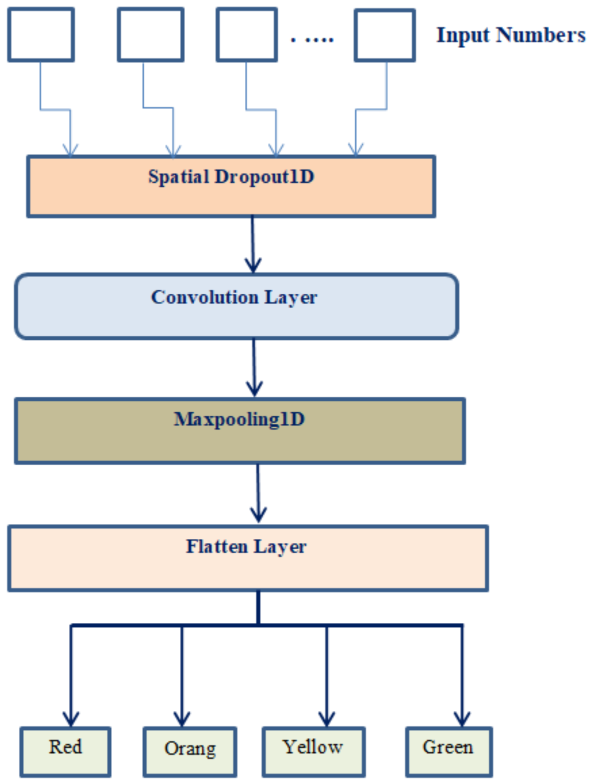

Figure 4. 5 CNN model architecture

4.6. Hyperparameter Tuning

It is a prerequisite to build the architecture for neural networks before they can be integrated into a model. The "hyperparameters" affect both the architecture of the network and how the network is trained. Hyperparameter tuning is Finding hyperparameter values that optimize the model's performance. We used a grid search to optimize the number of neurons, batch size, and epochs as hyperparameters.

4.7. Model Evaluation

A deep learning model's performance as well as its strengths and weaknesses are understood through the process of model evaluation which employs many evaluation metrics. It's important to assess the model's efficiency. We used stratified k-fold cross-validation with k=10 and performance will be assessed after training using the confusion matrix, accuracy, precision, recall, and F1 metrics.

CHAPTER FIVE

5. EXPERIMENT AND IMPLEMENTATION

5.1. Introduction

This chapter presents how the methodology covered in chapter three and the proposed solution implemented in chapter four can be used to predict ED triage outcomes. The Ethiopian health system categorizes triage levels based on the urgency of needed treatment for the patients. We created a model for the prediction of severity levels of the triage outcomes. To create our model, we combined deep learning techniques with appropriate feature extraction. We evaluate the efficiency of each model after putting the model into practice.

5.2. Working Environment

This study used several development tools and packages, as was mentioned earlier in Chapter 3 section 3.7 to put the suggested solution into action. To analyze and visualize data, we used the tools pandas, matplotlib, numpy, sci-kit-learn, and Jupiter notebook. Several more tools and libraries were employed in the implementation of our prediction model. In addition, Tensorflow and Keras as modeling tools are used.

Table 5. 1 Description of the Tools and Python Packages Used During Implementation

Tools and Packages	Description
Jupyter Notebook	It is a free web application that enables us to create and share documents. It is useful for data cleaning and transformation, modeling, and, data visualization.
Python	Python is a popular platform that is easy to learn, and a powerful programming language to develop ML and DL applications
Microsoft Excel	We used it for dataset preparation
Microsoft word	For document preparation
Scikit-learn	A Python module in ML that is used for handling basic ML

	tasks, clustering, regression, and classification
Pandas	An important package that provides a fast, flexible, and expressive data structure that can work with 'Relational' or 'Labeled' data that enables filtering and integrating of data, loading it from external sources, and also handling a dataset
Numpy	Used for array processing, used to convert text to numeric data for feature training and model testing
Matplotlib	For result visualization
Seaborn	For statistical data visualization

5.3. Dataset Description

The dataset used for this study is collected from ACSMC located in Adama, Ethiopia. Data were gathered from the emergency department of the hospital. We created a dataset of the ED triage prediction from patient documents registered at the department. The dataset comprises one target variable and eleven independent variables. And additional features exist in the form used in the department but the features are excluded because the features are not filled for most of the patients.

The dataset contains the patient history data columns in sex, age, mobility, HR, RR, SPO2, TEMP, CNS/AVPU, SBP, TRAUMA, PAIN_SCORE, and OUTCOME with the last column as the target variable.

Table 5. 2 Dataset Features Description

No	Symbols	Features full name	Type	Class
1	Sex	Sex of patient	Nominal	Predictor
2	Age	Age of the patient	Numeric	Predictor
3	mobility	Mobility	Nominal	Predictor

4	HR	Hypertension Rate	Numeric	Predictor
5	RR	Respiratory Rate	Numeric	Predictor
6	SPO2	Oxygen saturation	Numeric	Predictor
7	TEMP	Temperature	Numeric	Predictor
8	CNS/AVPU	central nerves system	Nominal	Predictor
9	SBP	Systolic blood pressure	Numeric	Predictor
10	TRAUMA	Trauma	Nominal	Predictor
11	PAIN_SCORE	pain_score	Numeric	Predictor
12	OUTCOME	Outcome	Nominal	Target

5.4. Preprocessing and Label Encoding

5.4.1. Importing Libraries

Python is a programming language with varieties of libraries. These Python Libraries are a collection of pre-built Python modules used for a variety of functions. Some libraries were employed to create a friendly environment and implement the proposed solution, as seen in the following figure.

```
import tensorflow as tf
from keras.models import Sequential
from keras.layers import Input, Dense, Dropout, LSTM
import pandas as pd
import numpy as np
from sklearn import preprocessing
from keras.models import Sequential
from sklearn.model_selection import train_test_split
from keras.initializers import Constant
from sklearn.model_selection import KFold
from keras.layers import SpatialDropout1D
from keras.layers.convolutional import Conv1D, MaxPooling1D
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten, BatchNormalization,LSTM, GRU, Dropout, Bidirectional, SpatialDropout1D
from tensorflow.keras.utils import to_categorical
```

Figure 5. 1 Sample code for importing libraries

Let's see the purpose of some libraries

Importing TensorFlow: is to provide a platform for building and training models because it offers a wide range of tools and functions for numerical computation, model building, and deployment.

Importing Keras: is a high-level neural network API in Python that is built on TensorFlow, which allows defining of models using sequential or functional API and offers pre-built layers, loss, and optimization.

Importing Pandas: to provide data structure like DataFrame to easily handle and manipulate structured data for preprocessing and feature engineering.

Importing Numpy: supports large, multidimensional and metrics, is extensively used for numerical computation and data manipulation.

Importing Sklearn: used for classification, regression, clustering, and dimensionality reduction.

Importing Scikit-learn: to implement varieties of Algorithms and evaluation metrics.

Importing tf. Keras.layer: it contains different types of layers such as Dense(fully connected layer), LSTM, GRU, Dropout(regularization).

Importing Tf.keras.preprocessing.sequence: for padding sequence, text vectorization, splitting data into test and train set.

5.5. Analyzing Data

5.5.1. Loading Dataset

When running a Python program for data analytics and experimentation, we use a dataset and it must be loaded. And, the data should be made available to the Python environment we're using it. To allow importing external data into a Python program in a variety of file formats, Python uses a variety of modules. We use pandas to import our dataset. The panda’s method read the CSV file and loads it as DataFrames, where instances of the data are available via column name.

```
df=pd.read_csv('/content/triage_datasetshm.csv', encoding='cp1252')
```

Figure 5. 2 Sample code of loading dataset

This line of code is used to read CSV(comma-separated values) and store its contents into pandas DataFrame object df.

5.5.2. Cleaning Dataset

Following dataset loading, we cleaned the features like ("SBP"), (PAIN_SCORE), and ("TEMP") columns. Because they include categorical values and characters like(“.,” ”/”).

```
df['SBP'] = df['SBP'].str.replace('/', '')
df['TEMP']= df['TEMP'].apply(str)
df['TEMP'] = df['TEMP'].str.replace('.', '')
```

Figure 5. 3 Sample code of cleaning dataset

The first line of code is to replace the forward slash (‘/’) character in the SPB column of the DataFrame with an empty string. The second line is, to convert the values in the ‘TEMP’ column to a string by using apply () method with the string function. It is used to apply string conversion to each element in the ‘TEMP’ column. The third line of code is used to replace the (‘.’) character in the ‘TEMP’ column of DataFrame ‘df’ with an empty string.

5.5.3. Feature Scaling

Then after we cleaned our data, we scaled our features to normalize the range of our independent variables or features of data. This preprocessing technique is to bring all the columns or features of the data to the same scale.

```
scaler = StandardScaler()
scaled_data = scaler.fit_transform(input_data)
```

Figure 5. 4 Implementation of the Feature Scaling

The first line is used to create an instance of the StandardScaler class and assigns it to the variable scaler. The StandardScaler is a preprocessing class in sci-kit-learn that standardizes features by subtracting the mean and scaling to unit variance. And, the second line of code applies the feature scaling transformation to the input_data using the fit_transform() method. The fit_transform() method calculates the mean and standard deviation of each feature in the input_data and then applies the standardization transformation to the data. The transformed data is assigned to the scaled_data variable.

```
grads = gradient_input(model, X)
importance_scores = np.abs(grads) / np.sum(np.abs(grads))
sorted_indices = np.argsort(importance_scores)[::-1]
sorted_features =df.columns[sorted_indices]
for feature, score in zip(sorted_features, importance_scores[sorted_indices]):
    print(f'Feature: {feature}, Importance Score: {score:.4f}')
```

Figure 5. 3 Implementation of feature contribution score

In the above figure, the line of code grads = gradient_input(model, X) calculates the gradients of the model's output for the input data X. The gradient_input() function takes the model and the input data X as inputs and returns the gradients. The second line of code is to calculate the importance scores for each feature by taking the absolute values of the gradients and dividing them by the sum of the absolute values. This normalization ensures that the importance scores sum up to 1. The third line of code sorts the indices of the importance scores in descending order. The resort () function from NumPy returns the indices that would sort the array, and [::-1] reverses the order to obtain descending sorting. The next line retrieves the column names (features) of the DataFrame df based on the sorted indices of the importance scores. The for loop iterates over the sorted features and their corresponding importance scores, printing them using the print() function.

5.5.4. Label Encoding

Our model's target variables are categorical. We must encode categorical features into a representation that is compatible with the models because deep learning algorithms only operate with numeric values. Some of our input features like (sex), (mobility), (and CNS) are categorical. So, We must encode the categorical variables into a numerical representation to make it usable by deep learning algorithms. We used the label encoder class for this purpose.

```
le = preprocessing.LabelEncoder()
df['OUTCOME'] = le.fit_transform(df['OUTCOME'])
```

Figure 5. 4 Implementation of the encoding target variable

The first line of code is used to create instances of LabelEncoder class and assign it to the variable 'le'. The LabeEncoder is used to encode columns with categorical variables into numerical labels. The second line of code is to apply the transformation to the 'OUTCOME' column of the DataFrame. The fit_transform() method of the LabelEncoder class is used to fit the encoder to the column data and then transform it into encoded numerical labels. The encoded labels are then assigned to the 'OUTCOME'column of the DataFrame.

5.6. Model Implementation

After we got pre-processed data which is used to train the deep learning model now it is possible to implement all the architectures considered in the study. Before training or fitting the model we categorized our dataset into independent and dependent variables as X and Y. X is a concatenation of all features which is encoded, while Y is the labeled target class.

```
X=df[['sex', 'age','Mobality','HR','RR','SPO2','TEMP','CNS/AVPU','SBP','TRAUMA','PAIN_SCORE']].values
Y=to_categorical(df['OUTCOME'].values)
```

Figure 5. 5 Splitting dataset into dependent and independent variables for classification

The first line of code extracts the specified columns from the DataFrame df and assigns them to the variable. It selects these columns by their column names. The values attribute converts the selected data into a NumPy array which is the expected format for many deep learning algorithms. And, the second line prepares the target variable (Y) by converting it into categorical form using the to_categorical() function. The df ['OUTCOME'].values retrieves the values from the 'OUTCOME' column of the DataFrame df and to_categorical() converts these values into encoded representation.

5.6.1. Create the Model

The Keras model is a representation of the real neural network model. Simple and user-friendly Sequential API and sophisticated Functional API are the two methods offered by Keras for creating models. The Functional API is used when a model has to share layers or have many inputs or outputs. For modeling more basic neural networks where each layer only accepts one input and transmits one output there is the Sequential API a little less detailed API. For our study of emergency department triage predictions, we created a sequential model by using the Sequential () API.

```
model = Sequential()
```

Figure 5. 6 Implementation of creating a model

In the above line of code model = Sequential() is used to initialize a sequential model in Keras. While each layer is added sequentially a sequential model is a linear stack of layers in Keras. Then after, we can build a neural network by adding different types of layers to the sequential model which is done by calling the add () method.

Implementation of LSM: LSTMs are specifically designed to capture and model long-term dependencies in sequential data. This ability is crucial when dealing with sequences of numerical data. LSTMs can remember relevant information from earlier time steps and utilize it to make predictions later in the sequence. LSTMs have memory cells that can store and update information over time. These memory cells enable LSTMs to handle and process sequential data by selectively remembering or forgetting information. The gating mechanisms (such as the forget gate, input gate, and output gate) help to regulate the flow of information and allow the LSTM to retain important information while discarding irrelevant or noisy signals. LSTMs typically use activation functions to introduce nonlinearity that allows the LSTM to model complex patterns and relationships in the data.

```
model.add(LSTM(128, activation='relu', return_sequences=True))
model.add(Dropout(0.2))

model.add(LSTM(32, activation='relu'))
model.add(Dropout(0.2))

model.add(Dense(32, activation='relu'))
model.add(Dropout(0.2))

model.add(Dense(4, activation='softmax'))
```

Figure 5. 9 Implementation of the LSTM model

In the first line, the first LSTM layer is added with 128 units and return_sequences= True and ReLU activation functions. The return_sequences parameter is set to return the output sequence from each time step which is useful when stacking multiple LSTM layers. Then the first dropout layer of rate 0.2 is added to prevent overfitting. Next, the second LSTM layer with 32 units is added with the ReLU activation function which only returns the output from the last time step. The dense layer with 32 units and ReLU activation is added to serve as a fully connected layer for further feature transformation. Finally, the last dense layer with 4 units and softmax activation is added which produces the classification probabilities for the 4 classes.

implementation of CNN-LSTM: Combining these two models aims to produce a model that benefits from the advantages of CNN and LSTM capturing the CNN-extracted features and using them as an LSTM input. As a result, we create a model that achieves this goal and uses the vectors as the input for convolutional neural networks. After each filter, a layer of max pooling is applied to update and reduce the size of the data. Then, the results of all max-pooling layers are concatenated to build the LSTM input which applies an LSTM layer to filter the information. Finally, to assign classes to levels and produce the appropriate output, we use the softmax function as an activation function for the prediction.

```
cnn_model = tf.keras.models.Sequential([
    tf.keras.layers.Conv1D(filters=64, kernel_size=3, activation='relu', input_shape=(11, 1)),
    tf.keras.layers.MaxPooling1D(pool_size=2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(4, activation='softmax'),
])
```

```
lstm_model = tf.keras.models.Sequential([
    tf.keras.layers.LSTM(units=64, return_sequences=True, input_shape=(num_features, 1)),
    tf.keras.layers.LSTM(units=32),
    tf.keras.layers.Dense(4, activation='softmax')
])
```

Figure 5. 10 Implementation of the CNN-LSTM model

In the above code the line cnn_model = tf.keras.models.Sequential() is for sequential model initialization. In the next line, the Conv1D layer performs 1-dimensional convolution with 64 filters and a kernel size of 3. It uses the ReLU activation function and expects input with shape (num_features, 1). The MaxPooling1D layer performs max pooling with a pool size of 2 for spatial dimension reduction. The Flatten layer converts the output from the previous layers into a 1-dimensional vector. The next serves as a fully connected layer for further feature transformation. The final Dense layer with 4 units and softmax activation produces the classification probabilities for the classes.

And for LSTM, the first line returns the output sequence from each time step. The second LSTM layer with 32 units does not have return_sequences=True, indicating that it only returns the output from the last time step. The final Dense layer with 4 units and softmax activation produces the classification probabilities for the classes.

Implementation of GRU model: GRU is a kind of newer version of RNN widely used for learning sequential data prediction model problems. It is capable of handling vanishing gradients that exist in the traditional RNN model. GRU contains Reset and Update Gate and It is faster than LSTM. As with every other thing they both have some layers which help them to recognize and learn the pattern in the case for better performance.

```
model = Sequential()
model.add(GRU(units=64, input_shape=(11, 1)))
model.add(Dense(4, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

Figure 5. 11 Implementation of the GRU model

Implementation of CNN*:* The method used to produce features is convolution. The CNN is then fed these features. The most crucial features are found via the convolution process. We used CNN in one dimension for classification. The weighted total is filled in the output after calling each row in the dimension and multiplying each block element by the weight value. This is true across all channels. Each channel's weights are learned by it. In this method, the values are combined into smaller values. When the max-pooling layer is eventually invoked, the network is forced to keep only the highest value of each feature vector, which is typically the most helpful local feature. Finally, a flattened layer was called. The pooled feature map is flattened into a single column and then sent to the fully linked layer.

```
model = Sequential()
model.add(BatchNormalization())
model.add(SpatialDropout1D(0.2))
model.add(Conv1D(32, kernel_size=3, padding='same', activation='relu'))
model.add(MaxPooling1D(pool_size=2))
model.add(Conv1D(32, kernel_size=3, padding='same', activation='relu'))
model.add(MaxPooling1D(pool_size=2))
model.add(Flatten())
model.add(Dense(4, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

Figure 5. 12 Implementation of the CNN model

Implementation of BiGru and BiLSTM: Both BiGru and BiLSTM are a type of recurrent neural network. Since they use two hidden layers they process data in two ways. We used Convolution, BiGRU/BiLSTM, and dense layer.

```
model = Sequential()
model.add(Bidirectional(LSTM(units=64, input_shape=(X_train.shape[1], 1))))
model.add(Dropout(0.2))
model.add(Flatten())
model.add(Dense(units=4, activation='softmax'))
```

Figure 5. 13 Implementation of BiLSTM

5.6.2. Compile the Model

To build the model the Keras model provides a method called *compile ()*. We included three arguments in the compile method: loss, optimizer, and metrics. The loss function is used to detect errors or deviations in the learning process. Binary_crossentropy is used for a two-class, or binary classification problem. Categorical_crossentropy is used for a multi-class classification problem. In our case there is multiple class classification, we used categorical_crossentropy. Optimization is an important technique that compares the prediction and the loss function to optimize the input weights. Keras has several optimizers as modules one of which was Adam. The Adam method is easier to build, has a shorter run time, consumes less memory, and requires less tuning than any other optimization technique. Model performance is assessed using metrics.

```
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

Figure 5. 14 Implementation of model compiling

5.6.3. Model Training

Fit is used in the training of models. This fit function's objective is to assess the model's performance during training. The model evaluation is the next step after training the model. When building a model evaluation is a procedure that determines whether the model is the best fit for the given problem and associated data.

```
history = model.fit(X[train], Y[train], epochs=100, batch_size=10, verbose=1)
scores = model.evaluate(X[test], Y[test], verbose=0)
```

Figure 5. 15 Implementation of model training

In this code, the model is trained using the fit function. The training data X[train] and Y[train] are passed as input. The epochs and batch_size parameters specify the number of times the model will iterate over the entire training dataset and the number of samples per gradient update respectively. The verbose parameter is set to 0 which means no output will be displayed during training.

5.7. Model Evaluation

We evaluated our models by using stratified 10-fold cross-validation. From the 10 folds, 9 folds are used for training and 1-fold for testing iteratively. An improvement on the cross-validation method used for classification issues is the stratified k-fold cross-validation. It aims to address the issue of imbalanced target classes.

```
kfold = KFold(n_splits=10, shuffle=True, random_state=7)
cvscores = []
```

Figure 5. 16 Implementation of Stratified Cross-validation

The first line is implemented to create instances of stratified Kfold class which for cross-validation during n_splits number of folds from the sklearn library. setting n_splits to 10 indicating that the data will be divided into 10 folds. And, the second line initializes an empty list named cv scores to store the cross-validation scores in it. Additionally, we evaluated our model's performance using evaluation metrics such as the confusion matrix, precision, recall, and F1-score.

CHAPTER SIX

6 EVALUATION, RESULTS, AND DISCUSSIONS

6.1. Introduction

The experimental findings of the proposed methodology for applying deep learning to predict emergency department triage are described in this chapter. The used dataset summary and class distribution are covered first. The confusion matrix and classification report are then given depending on the validation result. A comparison of the six models is offered in the third section. Finally, a thorough comparison of this study with the prior research is offered to ensure that it addresses the research topic and covers the gaps found in the literature review.

6.2. Dataset Class Distributions Result

After preprocessing the dataset utilized for this study has 5000 rows and 12 columns comprising one target class. All of the data 5,000 rows was used for the classification. Consequently, there is a class imbalance in the dataset since instances are not distributed equally among the target classes. When it comes to the machine learning technique training the model with dataset distributions that have a class imbalance causes the model to be biased toward the majority class which results in prediction issues for the minority class. We used stratified cross-validation to handle class imbalance. All classification metrics were used to assess the model's performance.

6.3. Model Evaluation Result

In this experiment, we evaluate the performance of models in terms of emergency department Triage Prediction. Cross-validation has been used to prevent overfitting and assess and evaluate the performance of models on unseen data. Our decision to use stratified cross-validation rather than other cross-validation types was based on the fact that models deal with imbalanced classification. All of our evaluations are carried out on top of 10-fold stratified cross-validation. The emergency department triage outcome Prediction model is multiclass that classifies the patient's triage level in the following form: Red labeled as 3, Orange labeled as 2, Yellow labeled as 1, and Green labeled as 0. We used twelve features including the target class of the built dataset which has 5000 instances. To evaluate the performance of this model variety of classification metrics are used in the following section.

Confusion Matrix for LSTM model

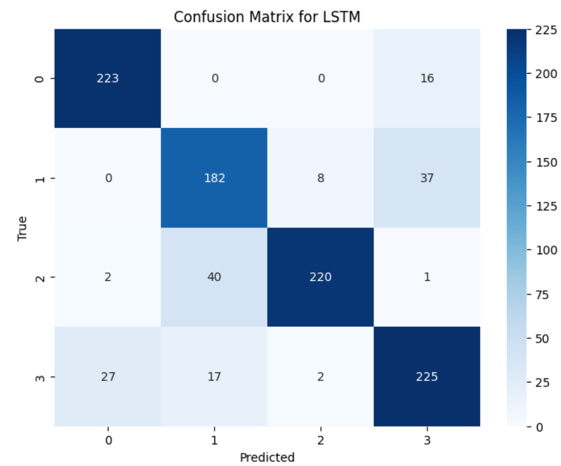

Figure 6. 1 Confusion Matrix for LSTM model

We employed LSTM model Confusion Matrix as indicated in the above figure. It yields an accuracy of 83.7% and 83.5 % of F1. The LSTM is our best-performed model.

Confusion Matrix for GRU

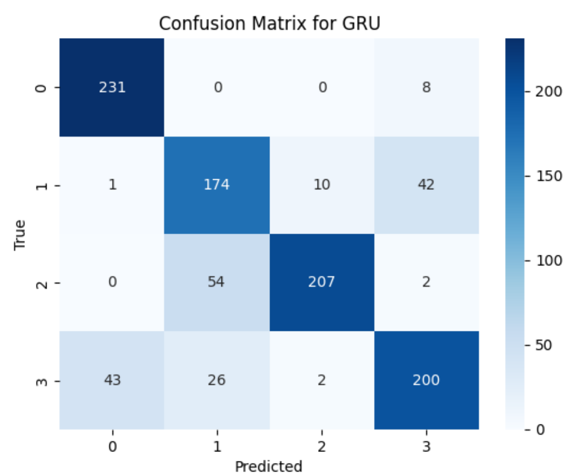

Figure 6. 2 *Confusion matrix for GRU model*

The above is the GRU model Confusion Matrix With an accuracy of 77.3 % and F1_score of 77.5 %.

Confusion matrix for BiLSTM

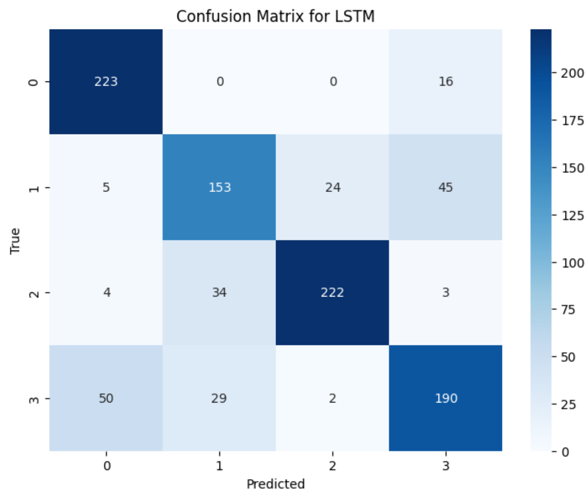

Figure 6. 3 Confusion matrix for the BiLSTM model

Confusion matrix for BiGRU

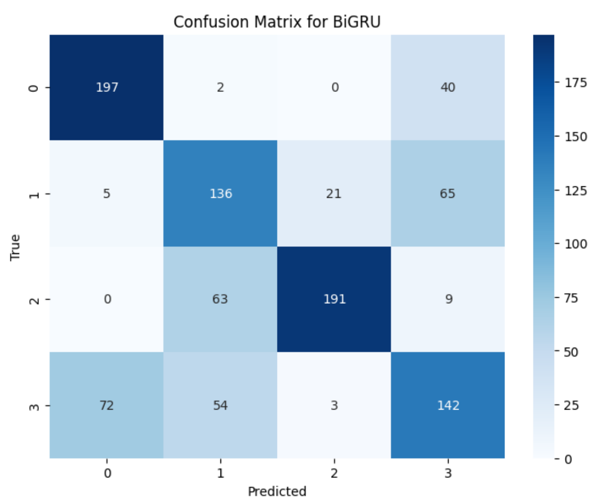

Figure 6. 4 *Confusion matrix for the BiGRU model*

Table 6. 1 Summary of classification performance of the EDTP model

Models	**Accuracy(%)**	**Precision(%)**	**F1_score(%)**	**Recall(%)**
LSTM	**83.7**	83.5	**83.5**	83.4
CNN-LSTM	76.5	76.3	76.5	77.4
GRU	77.2	77.1	77.3	77.1
BiLSTM	77.3	77.2	77.5	77.3
BiGRU	66.7	66.6	66.8	66.7

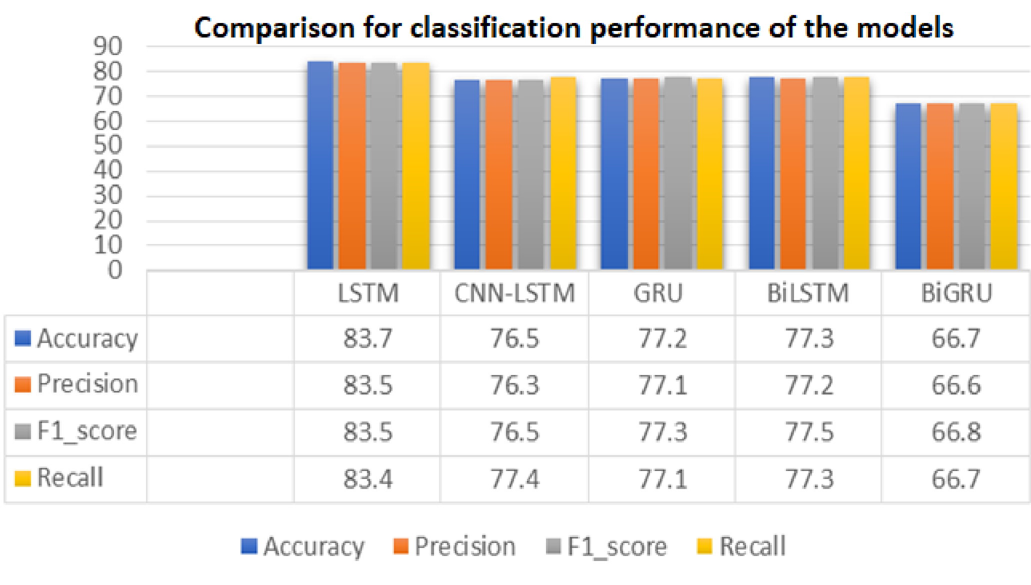

Figure 6. 5 Classification performance of model comparison

6.4. Tuning Hyperparameters

The reason behind hyperparameter optimization is that neural networks have a lot of parameters and are a little bit difficult to configure in such a way that they can perform well on a given problem. We checked for hyperparameters by using different values of the parameters to optimize them. We tuned parameters for our models starting from several neurons. The number of neurons in a layer is an important parameter to tune. Generally, the number of neurons in a layer controls the representational capacity of the network.

```
Best: 0.776357 using {'model__neurons': 25}
0.750152 (0.039915) with: {'model__neurons': 10}
0.763957 (0.021122) with: {'model__neurons': 15}
0.755360 (0.024488) with: {'model__neurons': 20}
0.776357 (0.027023) with: {'model__neurons': 25}
0.771350 (0.027375) with: {'model__neurons': 30}
0.763355 (0.020137) with: {'model__neurons': 32}
0.757346 (0.025211) with: {'model__neurons': 50}
0.741554 (0.039052) with: {'model__neurons': 64}
0.767956 (0.033189) with: {'model__neurons': 100}
```

Figure 6. 6 Hyperparameter tuning number of neurons for the GRU model

We also tuned batch size and epochs. They define how many patterns to read at a time and the number of times that the training dataset is shown to the network. To optimize dropout and learning rate, the same approach was used.

```
Best: 0.786360 using {'batch_size': 10, 'epochs': 50}
0.758348 (0.021514) with: {'batch_size': 10, 'epochs': 10}
0.786360 (0.008182) with: {'batch_size': 10, 'epochs': 50}
0.753954 (0.015154) with: {'batch_size': 10, 'epochs': 100}
0.762555 (0.009064) with: {'batch_size': 20, 'epochs': 10}
0.781760 (0.012259) with: {'batch_size': 20, 'epochs': 50}
0.777159 (0.011251) with: {'batch_size': 20, 'epochs': 100}
0.759152 (0.009089) with: {'batch_size': 30, 'epochs': 10}
0.783158 (0.026559) with: {'batch_size': 30, 'epochs': 50}
0.767956 (0.013297) with: {'batch_size': 30, 'epochs': 100}
0.712153 (0.040244) with: {'batch_size': 60, 'epochs': 10}
0.783558 (0.011205) with: {'batch_size': 60, 'epochs': 50}
0.778156 (0.014057) with: {'batch_size': 60, 'epochs': 100}
```

Figure 6. 7 Grid search tuning the number of epochs and batches for the GRU model

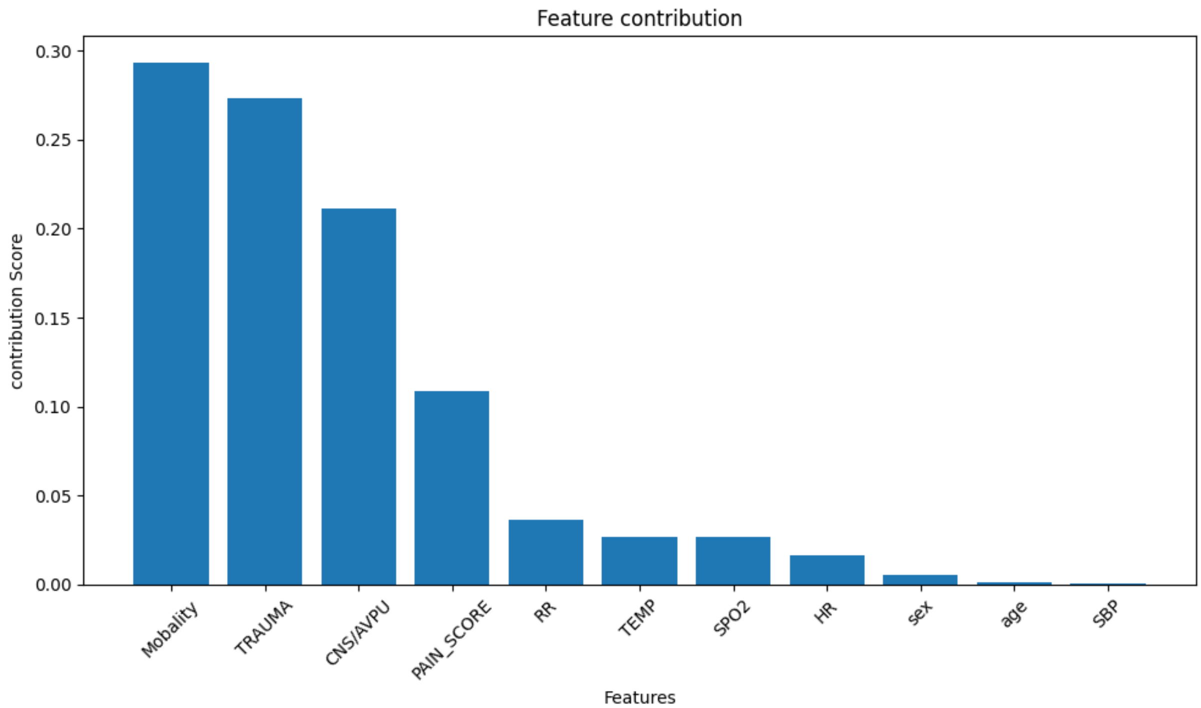

Figure 6. 8 Feature contribution score

6.5. Discussions

Objectively this study is to build a deep learning model that predicts the emergency department triage outcome for Adama Comprehensive Specialized Medical College. To our knowledge, this study is the first in our country. To accomplish the objective first we constructed a new dataset. And then we examined a variety of Deep learning models. Lastly, we have conducted several experiments on it and we found promising results.

As we summarized in Figures 6.1 and Tables 6.1 the LSTM model performed better in terms of f1_score, accuracy, and precision to predict the triage outcomes. Generally found results confirm obtaining good results by training the LSTM model with tuned parameters. And the CNN is a model with less performance in this study with an accuracy of 52.8 %. This is Because CNNs are primarily designed to capture spatial information from image data by leveraging their hierarchical structure. But with numerical datasets, the CNN is not able to effectively extract meaningful features from the data because it does not possess the inherent spatial structure that images have rather CNN is used to operate over the textual data(Elbattah, Jules, and Dequen 2020).

And also, in this section, we discuss how our study answers the research questions raised in the first chapter Section 1.4. To answer these questions we used a variety of techniques. Let's see at used techniques and how the proposed model tried to respond to each research question.

RQ1: Which Attribute most contributes to the prediction of the emergency department Triage outcome? To answer this question different techniques were employed. The first technique is using gradient input and LASSO techniques we identified contribution scores for each attribute which means all columns of a dataset. Then after we examined the model performance with the most contributed features and less contributed features depending on a feature contribution score arranged in descending order. As Figure 6.8 shows, mobility is the most contributing attribute which is followed by Trauma as the second contributing feature while SBP is the least contributing feature. And results that model performance with a high contribution score is great and with a low contribution score we got reduced model performance in terms of Accuracy, F1_ score, precision, and recall.

RQ2: Which deep learning algorithm is most suitable for emergency department triage prediction? To answer this question first we set out an assumption to select a deep learning model. After we selected the model we have been trained and tested models using the dataset prepared for this purpose. To identify the best-performed models model performance evaluation (stratified 10-fold CV) and classification metrics (accuracy, precision, recall, and f1-score), would be used. Our empirical analysis indicates that the LSTM model is best by the performance evaluation.

RQ3. What is the most important hyperparameter to deal with during the experiment that has a great impact on the model's performance? To answer the question we conducted the hyperparameter tuning. To do this first we have defined a set of hyperparameters. Then we designed the experiment to evaluate by using defined parameters. Next, we performed training and evaluation of the model using a combination of hyperparameters by using a 10-fold cv. then by analyzing the performance of the model, we identified the hyperparameter as consistently better.

Different related studies have been conducted using machine learning and different techniques as discussed in chapter two in section 2.7 for the prediction of triage outcome. Let us see some of the studies which are closely related to our study.

The first (Wang 2013) this study predicts anomaly detection: overestimation and underestimation which is identified after the completion of the patient's treatment by evaluating the determination of the nursing staff. It is to determine the correctness of the criticality judgment of nursing staff by evaluating that of the criticality judgment of physicians. In this study, they developed SVM with PCA and ANN models to detect the anomaly. They didn't show the triage level to where the patients were assigned but it was to evaluate the work of the nurses.

Second (W. Liang et al. 2020) this study is to predict the critically ill patients of laboratory-confirmed Covid-19 positive patients. They used LASSO to identify the feature significance and they found age is the most recognized risk factor prognosis for covid -19. However, they used a small amount of data(only 1,590 instances) which results in less accuracy score. And also, they were only to identify critically ill patients there is no priority prediction for other-level patients. Another work is (Roquette et al. 2020) this study tried to predict only the admitted patient to the hospital rather than including the severity level of the triage there are no contributing features explained in this work and they didn't put a general idea about the result of their study.

Additionally, studies like (Raita et al. 2019), (Goto et al. 2019) are conducted to predict the ward for the patient whether goes to an Intensive care unit or Hospitalization. So both of study are binary classification that does not answer the levels of triage outcomes and also these studies are not general for emergency department activities as they focus on specific classes. For the latter one for ICU prediction the accuracy is o.86 using DNN and 0.74 using Logistic Regression while for Hospitalization, o.82 for DNN and 0.69 using Logistic Regression. However, the study is with mensioned gaps. And, the study (Kim et al. 2020) was to assess the performance of machine ML-based triage tools in screening septic shock. But this work was not included other kinds of disease only for septic screening which made it domain-specific, second, not included different levels of triage cases, and also there is no justification for the features and a result of the study.

Generally, the studies conducted before in this area are with some gaps. These gaps include the exclusion of a very common class of the patient for whom the triage system is mandatory and the exclusion of a very common attribute in the triage system, which results in poor generative for the study. Another is they are domain specific i.e, they work only with specific types of disease rather than applying for total emergency department patients which leads to less generality also, they used small size datasets and as a result, they are with less accuracy

The current study looked at various deep-learning neural networks and processing techniques to predict the triage outcomes in emergency departments. The study attempted to deal with patients who are registered in the emergency department of the hospital to give treatment by giving priority to the patient with the most severe conditions. For this work, we prepared or arranged our dataset based on the hospital's guidelines. The guideline shows that vital signs and other contributing features are used for the emergency department triage prediction. After the dataset is prepared, we implemented five different types of deep learning algorithms for the prediction. These algorithms include LSTM, GRU, CNN-LSTM, BiLSTM, and BiGRU. From the overall experiment results, as shown in Tables 6.1 and Figure 6.1, we can conclude that LSTM is the best prediction model achieving an accuracy of 83.7% and F1_score of 83.5 %. It is followed by the BILSTM model with an accuracy of 77.3 % and 77.5 % of F1_score. A grid search was carried out to identify the most suitable parameters for these models. Along with classification metrics including accuracy, precision, recall, f1-score, and confusion matrix, stratified 10-fold cross-validation was employed to evaluate their results. This study is the first one we are aware of that took into account in our country to predict the emergency department triage outcome.

6.6. Contribution

We used different deep learning techniques and approaches to address the problem mentioned in chapter one in section 1.3 and to answer the raised research questions. In the study, to obtain results, we did experiments. And also, provided some contributions or additional work. The basic contributions are:

- We have created a new triage dataset: to use deep learning a dataset is very important and mandatory. Previously there is no triage dataset prepared in the context of our country's triage system. So We have constructed by collecting the hard copy form of the triage form from Adama Comprehensive Specialized Medical College.
- Predicting the emergency department triage outcomes: by using the proposed deep learning models, we predicted different classes of emergency department triage outcomes from the prepared dataset.
- Improving the performance of prediction: to do this, First, we identified what factors affect the emergency department triage prediction, and second, we used different performance metrics and hyperparameter tanning techniques.

CHAPTER SEVEN

7. CONCLUSION AND FUTURE WORK

7.1. Conclusion

The emergency department triage prediction system is used to support triage nurses, Doctors, and nonprofessionals of the health system to know the outcome of patients. To help or support them, we have proposed an emergency department triage prediction model. The general objective of this study is to predict the ED triage outcome using deep learning. We predict the outcome of the triage patients by using patient history collected from the hospital. To accomplish this we have used different techniques. The first technique is building a dataset. Since there was no prepared dataset available for the work, we prepared a new dataset. For this purpose, we collected patient data from the Adama Comprehensive Specialized Medical College Emergency Department. The collected patient document is in hard copy form. So we collected the dataset from that hard-copy patient cards during this time since no software to scan the handwritten texts we prepared by typing. Then after we implemented different data preprocessing techniques to make a more suitable dataset for the DL algorithm. And this study provides a prediction model implemented on top of a newly collected dataset from patient history data. The dataset comprises 5000 instances with twelve variables one is a target variable. We utilized CNN feature extraction to make the data we gathered intelligible to the deep learning algorithm. For the classification, we compared six deep-learning models including CNN, and tested each of them using a 10-fold stratified cross-validation technique.

To maximize the performance of the LSTM, GRU, CNN-LSTM, BiLSTM, and BiGRU models the optimal hyperparameters were chosen using the grid search CV hyperparameter tuning technique. In this study, the best-performing model for emergency department triage prediction was determined to be LSTM. Under stratified 10-fold cross-validation, it received scores of 83.7 % accuracy and 83.5 % f_core. The BILSTM is 77.3 % accuracy and 77.5 % F1 score.

7.2. Future Work

In this study, to build the prediction of Emergency Department Triage outcomes, we have built a new dataset, and we applied a deep learning algorithm with a feature extraction method. In addition to this, we have implemented important data preprocessing techniques. There are numerous other works for future research in this area. The first is improving the accuracy of the work itself by gathering more data that is efficient for the deep learning model. In case of limitation of time, the amount of data used for this work is not sufficient as Deep Learning Algorithm needs a very large amount of data. the second is that there are additional conditions that can affect the classification purpose. There is additional information exists on the guideline used for Triage screening but since that information is not filled in for more of the patients, we have not included it in this study it is better to add it as an additional feature. The proposed model supports the experts to give a fast service, it is better to make it a mobile-based system that enables the experts to follow the status of the patients and help the patients to use the system to know their status. It is better to work with real-time datasets to achieve higher accuracy.

* * *

Special Acknowledgement: This research was funded by Adama Science and Technology University with a grant number **ASTU/SM-R/872/23**

REFERENCES

Ababa, Addis. 2019. "OF HEALTH SCIENCE, DEPARTMENT OF EMERGENCY MEDICINE FOR June 2019." (June).

Abdelwahab, Rehab, Hannah Yang, and Hareya Gebremedhin Teka. 2017a. "A Quality Improvement Study of the Emergency Centre Triage in a Tertiary Teaching Hospital in Northern Ethiopia." *African Journal of Emergency Medicine* 7(4): 160–66. http://dx.doi.org/10.1016/j.afjem.2017.05.009.

2017b. "A Quality Improvement Study of the Emergency Centre Triage in a Tertiary Teaching Hospital in Northern Ethiopia." *African Journal of Emergency Medicine* 7(4): 160–66.

Acharya, Bishwas, et al. 2014. "Prevalence and Socio-Demographic Factors Associated with Overweight and Obesity among Adolescents in Kaski District, Nepal." *Indian Journal of community health* 26(Supp 2): 118–22.

Ajisegiri, Whenayon Simeon et al. 2021. "Aligning Policymaking in Decentralized Health Systems: Evaluation of Strategies to Prevent and Control Non-Communicable Diseases in Nigeria." *PLOS Global Public Health* 1(11): e0000050.

Ayano, Getinet, et al. 2017. "Mental Health Training for Primary Health Care Workers and Implication for Success of Integration of Mental Health into Primary Care: Evaluation of Effect on Knowledge, Attitude, and Practices (KAP)." *International Journal of mental health systems* 11: 1–8.

Bickmore, Timothy W, Laura M Pfeifer, and Brian W Jack. 2009. "Taking the Time to Care: Empowering Low Health Literacy Hospital Patients with Virtual Nurse Agents." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1265–74.

Brownlee, Jason. 2018. *Deep Learning for Time Series Forecasting: Predict the Future with MLPs, CNNs, and LSTMs in Python*. Machine Learning Mastery.

Brownlee, Wallace, et al. 2020. "Treating Multiple Sclerosis and Neuromyelitis Optica Spectrum Disorder during the COVID-19 Pandemic." *Neurology*.

Burgess, Luke, Kathryn Kynoch, and Sonia Hines. 2019. “Implementing Best Practice into the Emergency Department Triage Process.” *International Journal of Evidence-Based Healthcare* 17(1): 27–35.

Carter, Eileen J, Stephanie M Pouch, and Elaine L Larson. 2014. “The Relationship between Emergency Department Crowding and Patient Outcomes: A Systematic Review.” *Journal of Nursing Scholarship* 46(2): 106–15.

Chorowski, Jan K et al. 2015. “Attention-Based Models for Speech Recognition.” *Advances in neural information processing systems* 28.

Coiera, Enrico. 1997. *Guide to Medical Informatics, the Internet and Telemedicine*. Chapman & Hall, Ltd.

Crumplin, M K. 2002. “The Myles Gibson Military Lecture: Surgery in the Napoleonic Wars.” *Journal of the Royal College of Surgeons of Edinburgh* 47(3): 566–78.

Dash, Sabyasachi, Sushil Kumar Shakyawar, Mohit Sharma, and Sandeep Kaushik. 2019. “Big Data in Healthcare: Management, Analysis and Future Prospects.” *Journal of Big Data* 6(1): 1–25.

Davenport, Thomas, and Ravi Kalakota. 2019. “The Potential for Artificial Intelligence in Healthcare.” *Future healthcare journal* 6(2): 94.

Derlet, Robert W, and John R Richards. 2000. “Overcrowding in the Nation’s Emergency Departments: Complex Causes and Disturbing Effects.” *Annals of emergency medicine* 35(1): 63–68.

Duko, Bereket et al. 2019. “Triage Knowledge and Skills among Nurses in Emergency Units of Specialized Hospital in Hawassa, Ethiopia : Cross-Sectional Study.” *BMC Research Notes*: 19–22. https://doi.org/10.1186/s13104-019-4062-1.

Ebrahimi, Mohsen, Abbas Heydari, Reza Mazlom, and Amir Mirhaghi. 2015. “The Reliability of the Australasian Triage Scale: A Meta-Analysis.” *World Journal of emergency medicine*

6(2): 94.

Elbattah, Mahmoud, Université De Picardie Jules, and Gilles Dequen. 2020. “Deep Learning to Predict Hospitalization at Triage : Integration of Structured Data and Unstructured Text.” : 4836–41.

Ericksen-Pereira, Wendy G., Nicolette V. Roman, and Rina Swart. 2018. “An Overview of the History and Development of Naturopathy in South Africa.” *Health SA Gesondheid* 23(November).

FitzGerald, Gerard, George A. Jelinek, Deborah Scott, and Marie Frances Gerdtz. 2010. “Emergency Department Triage Revisited.” *Emergency Medicine Journal* 27(2): 86–92.

Gilboy, Nicki, Paula Tanabe, Debbie Travers, and Alexander M Rosenau. 2012. “Emergency Severity Index (ESI): A Triage Tool for Emergency Department Care, Version 4.” *Implementation handbook* 2012: 12–14.

Goto, Tadahiro et al. 2019. “Machine Learning-Based Prediction of Clinical Outcomes for Children during Emergency Department Triage.” *JAMA Network Open* 2(1): 1–14.

Health, Federal Democratic Republic of Ethiopia Ministry of. 2015. “Health Sector Transformation Plan: 2015/16-2019/20.”

Hodge, Alister, Andrew Hugman, Wayne Varndell, and Kylie Howes. 2013. “A Review of the Quality Assurance Processes for the Australasian Triage Scale (ATS) and Implications for Future Practice.” *Australasian Emergency Nursing Journal* 16(1): 21–29.

Hore, Prodip, and Sayan Chatterjee. 2019. “A Comprehensive Guide to Attention Mechanism in Deep Learning for Everyone.” *American Express*.

Iserson, Kenneth V., and John C. Moskop. 2007a. “Triage in Medicine, Part I: Concept, History, and Types.” *Annals of Emergency Medicine* 49(3): 275–81.

2007b. “Triage in Medicine, Part I: Concept, History, and Types.” *Annals of Emergency Medicine* 49(3): 275–81.

Jiang, Fei et al. 2017. “Artificial Intelligence in Healthcare: Past, Present, and Future.” *Stroke and vascular neurology* 2(4).

Jović, Alan, Karla Brkić, and Nikola Bogunović. 2015. “A Review of Feature Selection Methods with Applications.” In *2015 38th International Convention on Information and Communication Technology, Electronics, and Microelectronics (MIPRO)*, Ieee, 1200–1205.

Kamiri, Jackson, and Geoffrey Mariga. 2021. “Research Methods in Machine Learning: A Content Analysis.” *International Journal of Computer and Information Technology (2279-0764)* 10(2).

Kim, Joonghee et al. 2020. “Machine Learning for Prediction of Septic Shock at Initial Triage in Emergency Department.” *Journal of critical care* 55: 163–70.

Leaver, Amber M, Anna Seydell-Greenwald, and Josef P Rauschecker. 2016. “Auditory–Limbic Interactions in Chronic Tinnitus: Challenges for Neuroimaging Research.” *Hearing research* 334: 49–57.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. 2015. “Deep Learning.” *nature* 521(7553): 436–44.

Liang, Qijie et al. 2019. “High-performance, Room Temperature, Ultra-broadband Photodetectors Based on Air-stable PdSe2.” *Advanced Materials* 31(24): 1807609.

Liang, Wenhua et al. 2020. “Early Triage of Critically Ill COVID-19 Patients Using Deep Learning.” *Nature Communications* 11(1): 1–7. http://dx.doi.org/10.1038/s41467-020-17280-8.

van der Linden, M Christien, Barbara E A M Meester, and Naomi van der Linden. 2016. “Emergency Department Crowding Affects Triage Processes.” *International emergency nursing* 29: 27–31.

Luopajärvi, Tuulikki, Ilkka Kuorinka, Markku Virolainen, and Mia Holmberg. 1979. “Prevalence of Tenosynovitis and Other Injuries of the Upper Extremities in Repetitive Work.”

Scandinavian Journal of Work, Environment & Health: 48–55.

Lux, M P et al. 2011. “Health Services Research and Health Economy–Quality Care Training in Gynaecology, with Focus on Gynaecological Oncology.” *Geburtshilfe und Frauenheilkunde* 71(12): 1046–55.

Manyazewal, Tsegahun. 2017. “Using the World Health Organization Health System Building Blocks through Survey of Healthcare Professionals to Determine the Performance of Public Healthcare Facilities.” *Archives of Public Health* 75(1): 1–8.

McGranahan, Nicholas, et al. 2017. “Allele-Specific HLA Loss and Immune Escape in Lung Cancer Evolution.” *Cell* 171(6): 1259–71.

Meng, Bo et al. 2021. “Recurrent Emergence of SARS-CoV-2 Spike Deletion H69/V70 and Its Role in the Alpha Variant B. 1.1. 7.” *Cell reports* 35(13).

Miotto, Riccardo et al. 2018. “Deep Learning for Healthcare: Review, Opportunities and Challenges.” *Briefings in Bioinformatics* 19(6): 1236–46. https://doi.org/10.1093/bib/bbx044.

Nguyen, Van, Nguyễn Anh, and Hyung-Jeong Yang. 2019. “Real-Time Event Detection Using Recurrent Neural Network in Social Sensors.” *International Journal of Distributed Sensor Networks* 15: 155014771985649.

Van Niekerk, J., and R. J.A. Goris. 1990. 1 Clinical Intensive Care *Management of the Trauma Patient*.

Nowacki, Anna K, Megan Landes, Aklilu Azazh, and Lisa M Puchalski Ritchie. 2013. “A Review of Published Literature on Emergency Medicine Training Programs in Low- and Middle-Income Countries.” : 1–10.

Obermeyer, Ziad et al. 2015. “Emergency Care in 59 Low- and Middle-Income Countries : A Systematic Review.” (April): 577–86.

Organization, World Health. 2017. *Primary Health Care Systems (Primasys): Case Study from*

Nigeria. World Health Organization.

Panesar, Arjun. 2019. *Machine Learning and AI for Healthcare*. Springer.

Perkel, Jeffrey M. 2018. “Why Jupyter Is Data Scientists’ Computational Notebook of Choice.” *Nature* 563(7732): 145–47.

Pines, Jesse M et al. 2008. “The Effect of Emergency Department Crowding on Patient Satisfaction for Admitted Patients.” *Academic Emergency Medicine* 15(9): 825–31.

Pines, Jesse M, Judd E Hollander, A Russell Localio, and Joshua P Metlay. 2006. “The Association between Emergency Department Crowding and Hospital Performance on Antibiotic Timing for Pneumonia and Percutaneous Intervention for Myocardial Infarction.” *Academic Emergency Medicine* 13(8): 873–78.

Raita, Yoshihiko et al. 2019. “Emergency Department Triage Prediction of Clinical Outcomes Using Machine Learning Models.” *Critical Care* 23(1).

Raschka, Sebastian. 2018. “Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning.” *arXiv preprint arXiv:1811.12808*.

Ro, Young Sun et al. 2015. “Triage-based Resource Allocation and Clinical Treatment Protocol on Outcome and Length of Stay in the Emergency Department.” *Emergency Medicine Australasia* 27(4): 328–35.

Robertson-Steel, Iain. 2006. “Evolution of Triage Systems.” *Emergency medicine journal* 23(2): 154–55.

Rominski, Sarah, et al. 2014. “The Implementation of the South African Triage Score (SATS) in an Urban Teaching Hospital, Ghana.” *African Journal of Emergency Medicine* 4(2): 71–75. http://dx.doi.org/10.1016/j.afjem.2013.11.001.

Roquette, Bruno P., Hitoshi Nagano, Ernesto C. Marujo, and Alexandre C. Maiorano. 2020. “Prediction of Admission in Pediatric Emergency Department with Deep Neural Networks and Triage Textual Data.” *Neural Networks* 126: 170–77.

https://doi.org/10.1016/j.neunet.2020.03.012.

Sharif, Salmaan et al. 2020. “Detection of SARs-CoV-2 in Wastewater, Using the Existing Environmental Surveillance Network: An Epidemiological Gateway to an Early Warning for COVID-19 in Communities.” *MedRxiv*: 2006–20.

Shen, Chao et al. 2020. “RNA Demethylase ALKBH5 Selectively Promotes Tumorigenesis and Cancer Stem Cell Self-Renewal in Acute Myeloid Leukemia.” *Cell stem cell* 27(1): 64–80.

Singh, Dave, et al. 2019. “Global Strategy for the Diagnosis, Management, and Prevention of Chronic Obstructive Lung Disease: The GOLD Science Committee Report 2019.” *European Respiratory Journal* 53(5).

Singh, Shrawan K, Walter Z Tang, and Georgio Tachiev. 2013. “Fenton Treatment of Landfill Leachate under Different COD Loading Factors.” *Waste Management* 33(10): 2116–22.

Sprivulis, Peter C et al. 2006. “The Association between Hospital Overcrowding and Mortality among Patients Admitted via Western Australian Emergency Departments.” *Medical Journal of Australia* 184(5): 208–12.

Stone, Mervyn. 1974. “Cross-validatory Choice and Assessment of Statistical Predictions.” *Journal of the royal statistical society: Series B (Methodological)* 36(2): 111–33.

Swain, Sandra M et al. 2020. “Pertuzumab, Trastuzumab, and Docetaxel for HER2-Positive Metastatic Breast Cancer (CLEOPATRA): End-of-Study Results from a Double-Blind, Randomised, Placebo-Controlled, Phase 3 Study.” *The Lancet Oncology* 21(4): 519–30.

Traube, Chani et al. 2016. “Cost Associated with Pediatric Delirium in the Intensive Care Unit.” *Critical care medicine* 44(12): e1175.

Trenfield, Sarah J et al. 2019. “Shaping the Future: Recent Advances of 3D Printing in Drug Delivery and Healthcare.” *Expert opinion on drug delivery* 16(10): 1081–94.

Venkataramani, Atheendar S, Jacob Bor, and Anupam B Jena. 2016. “Regression Discontinuity Designs in Healthcare Research.” *BMJ* 352.

Wang, Shen Tsu. 2013. “Construct an Optimal Triage Prediction Model: A Case Study of the Emergency Department of a Teaching Hospital in Taiwan.” *Journal of Medical Systems* 37(5).

White, Franklin. 2015. “Primary Health Care and Public Health: Foundations of Universal Health Systems.” *Medical Principles and Practice* 24(2): 103–16.

White, Franklin, Lorann Stallones, and John M Last. 2013. *Global Public Health: Ecological Foundations*. Oxford University Press.

Xiao, Fei et al. 2021. “Recent Advances in Electrocatalysts for Proton Exchange Membrane Fuel Cells and Alkaline Membrane Fuel Cells.” *Advanced Materials* 33(50): 2006292.

Yang, Guang-Zhong et al. 2020. “Combating COVID-19—The Role of Robotics in Managing Public Health and Infectious Diseases.” *Science Robotics* 5(40): eabb5589.

Zlotnik, Alexander et al. 2016. “Building a Decision Support System for Inpatient Admission Prediction with the Manchester Triage System and Administrative Check-in Variables.” *CIN: Computers, Informatics, Nursing* 34(5): 224–30.

APPENDICES

Appendix A: Feature descriptions

Table 7. 1

No	Symbols	Features full name	Type	Class
1	Sex	Sex of patient	Nominal	Predictor
2	Age	Age of the patient	Numeric	Predictor
3	mobility	Mobility	Nominal	Predictor
4	HR	Hypertension Rate	Numeric	Predictor
5	RR	Respiratory Rate	Numeric	Predictor
6	SPO2	Oxygen saturation	Numeric	Predictor
7	TEMP	Temperature	Numeric	Predictor
8	CNS/AVPU	central nerves system	Nominal	Predictor
9	SBP	Systolic blood pressure	Numeric	Predictor
10	TRAUMA	Trauma	Nominal	Predictor
11	PAIN_SCORE	pain_score	Numeric	Predictor
12	OUTCOME	Outcome	Nominal	Target

Appendix B: Sample Code of Model Implementation

10-Fold Models Result

A. LSTM Implementation

```
from sklearn.model_selection import KFold
import numpy as np
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dropout, Dense

kfold = KFold(n_splits=10, shuffle=True, random_state=7)
cvscores = []
for train, test in kfold.split(X, Y):
    max_length = 50
    model = Sequential()

    model.add(LSTM(128, activation='relu', return_sequences=True))
    model.add(Dropout(0.2))

    model.add(LSTM(32, activation='relu'))
    model.add(Dropout(0.2))

    model.add(Dense(32, activation='relu'))
    model.add(Dropout(0.2))
    model.add(Dense(4, activation='softmax'))

    model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

    X_train_reshaped = np.reshape(X[train], (X[train].shape[0], X[train].shape[1], 1))
    Y_train_reshaped = Y[train]  # Assuming Y_train is already one-hot encoded

    # Fit the model
    history = model.fit(X_train_reshaped, Y_train_reshaped, epochs=100, batch_size=10, verbose=0)

    X_val_reshaped = np.reshape(X_val, (X_val.shape[0], X_val.shape[1], 1))
    scores = model.evaluate(X_val_reshaped, Y_val, verbose=0)

    print("%s: %.2f%%" % (model.metrics_names[1], scores[1]*100))
    cvscores.append(scores[1] * 100)

print("%.2f%% (+/- %.2f%%)" % (np.mean(cvscores), np.std(cvscores)))
```

```
accuracy: 83.00%
accuracy: 85.60%
accuracy: 82.80%
accuracy: 83.40%
accuracy: 84.90%
accuracy: 86.40%
accuracy: 79.80%
accuracy: 82.80%
accuracy: 84.30%
accuracy: 84.20%
83.72% (+/- 1.74%)
```

B. GRU model Implementation

```
import numpy
from sklearn.model_selection import StratifiedKFold
kfold = StratifiedKFold(n_splits=10, shuffle=True, random_state=7)
cvscores = []
print('Result of SCV for GRU')
for train, test in kfold.split(input_data, target_variable):
    max_length=50
    model=Sequential()
    model = Sequential()
    model.add(GRU(units=64, return_sequences=True, input_shape=(11, 1)))
    model.add(GRU(units=32, return_sequences=False))
    |
    model.add(Dense(4, activation='softmax'))  # Assuming 4 triage levels
    model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
    history=model.fit(X_train, y_train, epochs=100, batch_size=10, verbose=0)
    # evaluate the model
    scores = model.evaluate(X_test, y_test, verbose=0)
    print("%s: %.2f%%" % (model.metrics_names[1], scores[1]*100))
    cvscores.append(scores[1] * 100)
print("%.2f%% (+/- %.2f%%)" % (numpy.mean(cvscores), numpy.std(cvscores)))
```

```
Result of SCV for GRU
accuracy: 77.50%
accuracy: 78.10%
accuracy: 77.90%
accuracy: 76.30%
accuracy: 77.70%
accuracy: 77.20%
accuracy: 75.80%
accuracy: 77.10%
accuracy: 77.90%
accuracy: 77.20%
77.27% (+/- 0.70%)
```

CNN model Implementation

```
from sklearn.model_selection import StratifiedKFold
import numpy

kfold = KFold(n_splits=10, shuffle=True, random_state=7)
cvscores = []
for train, test in kfold.split(X, Y):

    model = Sequential()
    model.add(BatchNormalization())
    model.add(SpatialDropout1D(0.2))

    model.add(Conv1D(32, kernel_size=3, padding='same', activation='relu'))
    model.add(MaxPooling1D(pool_size=2))
    model.add(Flatten())
    model.add(Dense(4, activation='softmax'))
   model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
    history=model.fit(X_train, y_train, epochs=100, batch_size=10, verbose=0)
    # evaluate the model
    scores = model.evaluate(X_test, y_test, verbose=0)
    print("%s: %.2f%%" % (model.metrics_names[1], scores[1]*100))
    cvscores.append(scores[1] * 100)
print("%.2f%% (+/- %.2f%%)" % (numpy.mean(cvscores), numpy.std(cvscores)))
```

```
accuracy: 52.80%
accuracy: 54.00%
accuracy: 50.00%
accuracy: 48.20%
accuracy: 48.20%
accuracy: 53.60%
accuracy: 51.40%
accuracy: 59.00%
accuracy: 55.20%
accuracy: 55.91%
52.83% (+/- 3.29%)
```

Appendix C: Results for different models

10-Fold Models Result

A. BiLSTM model

```
Result of SCV for BiLSTM
accuracy: 78.60%
accuracy: 77.20%
accuracy: 77.10%
accuracy: 77.70%
accuracy: 76.60%
accuracy: 77.40%
accuracy: 77.30%
accuracy: 75.70%
accuracy: 77.90%
accuracy: 78.40%
77.39% (+/- 0.80%)
```

B. BiGRU model

```
Result of SCV for BIGRU
accuracy: 64.60%
accuracy: 67.50%
accuracy: 65.30%
accuracy: 66.30%
accuracy: 67.60%
accuracy: 67.20%
accuracy: 66.60%
accuracy: 65.90%
accuracy: 67.80%
accuracy: 68.40%
66.72% (+/- 1.14%)
```

B. CNN-BiGRU model

```
Result of SCV for CNN-BIGRU
accuracy: 67.20%
accuracy: 66.90%
accuracy: 67.10%
accuracy: 65.70%
accuracy: 65.10%
accuracy: 63.40%
accuracy: 67.10%
accuracy: 66.10%
accuracy: 66.00%
accuracy: 66.90%
66.15% (+/- 1.14%)
```

Sample code for Grid search Implementation

```
# Use scikit-learn to grid search the batch size and epochs
import numpy as np
import tensorflow as tf
from sklearn.model_selection import GridSearchCV
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from scikeras.wrappers import KerasClassifier
EMBEDDING_DIM=50

def create_model():
    # create model
    model=Sequential()
    model = Sequential()
    model.add(GRU(units=64, return_sequences=True, input_shape=(11, 1)))
    model.add(GRU(units=32, return_sequences=False))
    model.add(Dense(4, activation='softmax'))  # Assuming 4 triage levels
    model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])

    return model
seed = 7
tf.random.set_seed(seed)

model = KerasClassifier(model=create_model, verbose=0)
# define the grid search parameters
batch_size = [10,20,30,60]
epochs = [10, 50, 100]
param_grid = dict(batch_size=batch_size, epochs=epochs)
grid = GridSearchCV(estimator=model, param_grid=param_grid, n_jobs=-1, cv=3)
grid_result = grid.fit(input_data, target_variable)
# summarize results
print("Best: %f using %s" % (grid_result.best_score_, grid_result.best_params_))
means = grid_result.cv_results_['mean_test_score']
stds = grid_result.cv_results_['std_test_score']
params = grid_result.cv_results_['params']
for mean, stdev, param in zip(means, stds, params):
     print("%f (%f) with: %r" % (mean, stdev, param))
```

```
Best: 0.786360 using {'batch_size': 10, 'epochs': 50}
0.758348 (0.021514) with: {'batch_size': 10, 'epochs': 10}
0.786360 (0.008182) with: {'batch_size': 10, 'epochs': 50}
0.753954 (0.015154) with: {'batch_size': 10, 'epochs': 100}
0.762555 (0.009064) with: {'batch_size': 20, 'epochs': 10}
0.781760 (0.012259) with: {'batch_size': 20, 'epochs': 50}
0.777159 (0.011251) with: {'batch_size': 20, 'epochs': 100}
0.759152 (0.009089) with: {'batch_size': 30, 'epochs': 10}
0.783158 (0.026559) with: {'batch_size': 30, 'epochs': 50}
0.767956 (0.013297) with: {'batch_size': 30, 'epochs': 100}
0.712153 (0.040244) with: {'batch_size': 60, 'epochs': 10}
0.783558 (0.011205) with: {'batch_size': 60, 'epochs': 50}
0.778156 (0.014057) with: {'batch_size': 60, 'epochs': 100}
```