Sukkur IBA Journal of Computing and Mathematical Sciences

Recognized by HEC Pakistan in "Y" Category

E-ISSN: 2522-3003

P-ISSN: 2520-0755

Volume: 7 | No: 2 | Jul - Dec | 2023

Sukkur IBA Journal of Computing and Mathematical Sciences (SJCMS) is the bi-annual research journal published by **Sukkur IBA University**, Sukkur Pakistan. **SJCMS** is dedicated to serve as a key resource to provide practical information for the researchers associated with computing and mathematical sciences at global scale.

Disclaimer: The opinions expressed in **Sukkur IBA Journal of Computing and Mathematical Sciences (SJCMS)** are those of the authors and contributors, and do not necessarily reflect those of the journal management, advisory board, the editorial board, Sukkur IBA University press or the organization to which the authors are affiliated. Papers published in **SJCMS** are processed through double blind peer-review by subject specialists and language experts. Neither the **Sukkur IBA University** nor the editors of **SJCMS** can be held responsible for errors or any consequences arising from the use of information contained in this journal, instead errors should be reported directly to corresponding authors of articles.

Publisher: **Sukkur IBA Journal of Computing and Mathematical Sciences (SJCMS)**
Office of Research Innovation & Commercialization – ORIC
Sukkur IBA University – Airport Road Sukkur-65200, Sindh Pakistan
Tel: (092 71) 5644429 Fax: (092 71) 5804425 Email: sjcms@iba-suk.edu.pk URL: sjcms.iba-suk.edu.pk

Mission Statement

The mission of **Sukkur IBA Journal of Computing and Mathematical Sciences (SJCMS)** is to provide a premier interdisciplinary platform to researchers, scientists and practitioners from the field of computing and mathematical sciences for dissemination of their findings and to contribute in the knowledge domain.

Aims & Objectives

Sukkur IBA Journal of Computing and Mathematical Sciences aims to publish cutting edge research in the field of computing and mathematical sciences.

The objectives of **SJCMS** are:

1. to provide a platform for researchers for dissemination of new knowledge.
2. to connect researchers at global scale.
3. to fill the gap between academician and industrial research community.

Research Themes

The research focused on but not limited to following core thematic areas:

Computing:

- Software Engineering
- Formal Methods
- Human Computer Interaction
- Information Privacy and Security
- Computer Networks
- High Speed Networks
- Data Communication
- Mobile Computing
- Wireless Multimedia Systems
- Social Networks
- Data Science
- Big data Analysis
- Contextual Social Network Analysis and Mining
- Crowdsource Management
- Ubiquitous Computing
- Distributed Computing
- Cloud Computing
- Intelligent devices
- Security, Privacy and Trust in Computing and Communication
- Wearable Computing Technologies
- Soft Computing
- Genetic Algorithms
- Robotics
- Evolutionary Computing
- Machine Learning

Mathematics:

- Applied Mathematical Analysis
- Mathematical Finance
- Applied Algebra
- Stochastic Processes

Publisher: **Sukkur IBA Journal of Computing and Mathematical Sciences (SJCMS)**
Office of Research Innovation & Commercialization – ORIC
Sukkur IBA University – Airport Road Sukkur-65200, Sindh Pakistan
Tel: (092 71) 5644429 Fax: (092 71) 5804425 Email: sjcms@iba-suk.edu.pk URL: sjcms.iba-suk.edu.pk

Patron's Message

Sukkur IBA University has been imparting education with its core values merit, quality, and excellence since its inception. Sukkur IBA University has achieved numerous milestones in a very short span of time that hardly any other institution has achieved in the history of Pakistan. The university is continuously being ranked as one of the best university in Pakistan by Higher Education Commission (HEC). The distinct service of Sukkur IBA University is to serve the rural areas of Sindh and also underprivileged areas of other provinces of Pakistan. Sukkur IBA University is committed to serve targeted youth of Pakistan who is suffering from poverty and deprived of equal opportunity to seek quality education. Sukkur IBA University is successfully undertaking its mission and objectives that lead Pakistan towards socio-economic prosperity.

In continuation of endeavors to touch new horizons in the field of computing and mathematical sciences, Sukkur IBA University publishes an international referred journal. Sukkur IBA University believes that research is an integral part of modern learnings and development. Sukkur IBA Journal of Computing and Mathematical Sciences (SJCMS) is the modest effort to contribute and promote the research environment within the university and Pakistan as a whole. SJCMS is a peer-reviewed and multidisciplinary research journal to publish findings and results of the latest and innovative research in the fields, but not limited to Computing and Mathematical Sciences. Following the tradition of Sukkur IBA University, SJCMS is also aimed at achieving international recognition and high impact research publication in the near future.

Prof. Dr Asif Ahmed Shaikh
Vice Chancellor, Sukkur IBA University
Patron SJCMS

Publisher: **Sukkur IBA Journal of Computing and Mathematical Sciences (SJCMS)**
Office of Research Innovation & Commercialization – ORIC
Sukkur IBA University – Airport Road Sukkur-65200, Sindh Pakistan
Tel: (092 71) 5644429 Fax: (092 71) 5804425 Email: sjcms@iba-suk.edu.pk URL: sjcms.iba-suk.edu.pk

Editorial

Dear Readers,

It is a pleasure to present to you the latest of Sukkur IBA Journal of Computing and Mathematical Sciences (SJCMS).

The stunning advances in various fields of science and technology have a profound impact on our lives in almost every sphere of our activity, such as health, agriculture, communication, transportation, and defense. These advances have been driven by an ever-growing volume of exciting discoveries, largely emanating from research community. In order to highlight the future technology challenges, the SJCMS aims to publish cutting-edge research in the field of computing and mathematical sciences for dissemination to the largest stakeholders. SJCMS has achieved milestones in very short span of time and is indexed in renowned databases such as DOAJ, Google Scholar, DRJI, BASE, ROAD, CrossRef and many others. SJCMS is now HEC recognized in Z-Category.

This issue contains the double-blind peer-reviewed articles that address the key research problems in the specified domain The SJCMS adopts all standards that are a prerequisite for publishing high-quality research work. The Editorial Board and the Reviewers Board of the Journal is comprised of renowned researchers from technologically advanced countries. The Journal has adopted the Open Access Policy without charging any publication fees that will certainly increase the readership by providing free access to a wider audience.

On behalf of the SJCMS, I welcome the submissions for upcoming issue (Volume-8, Issue-1, January-June 2024) and looking forward to receiving your valuable feedback.

Sincerely,

Prof. Dr. Javed Ahmed Shahani
Chief Editor

Publisher: **Sukkur IBA Journal of Computing and Mathematical Sciences (SJCMS)**
Office of Research Innovation & Commercialization – ORIC
Sukkur IBA University – Airport Road Sukkur-65200, Sindh Pakistan
Tel: (092 71) 5644429 Fax: (092 71) 5804425 Email: sjcms@iba-suk.edu.pk URL: sjcms.iba-suk.edu.pk

Publisher: **Sukkur IBA Journal of Computing and Mathematical Sciences (SJCMS)**
Office of Research Innovation & Commercialization – ORIC
Sukkur IBA University – Airport Road Sukkur-65200, Sindh Pakistan
Tel: (092 71) 5644429 Fax: (092 71) 5804425 Email: sjcms@iba-suk.edu.pk URL: sjcms.iba-suk.edu.pk

Vol. 7, No. 2 | July – December 2023

Review of Applications of Artificial Intelligence in Health Care

Muhammad Hibatullah Channa[1], Bushra Memon[1]

Abstract:

Twenty-first century is famously termed the age of the fourth industrial revolution, which is because of the massive amount of data being generated and stored which could be interpreted and analyzed by intelligible programs. Just as the discovery of the microscope in the sixteenth century led humans to discover things about human biology that the naked eye could not see, likewise artificial intelligence could be used to look for patterns in the data which humans otherwise would have less likely perceived. This paper will capitalize on this. How much potential could aid in the health care field A review and guide are compiled for any researcher or student who might want to practically implement the ideas discussed. The implementation of artificial intelligence for the analysis of medical images and beyond is to be discussed in this paper. Tools and software developed from these ideas could help medical practitioners make more accurate decisions.

Keywords: *Machine Learning; Neural Network; Convolutional Neural Network; Supervised Learning; Unsupervised Learning; Reinforcement Learning, Labelled dataset, medicine, health.*

1. Introduction

History is often described in eras. From a prehistoric perspective, each era brought on new innovations and a means to improve lives in general. Prehistoric times encompass the Paleolithic, Neolithic, Bronze, and Iron Ages. In each era, something new was discovered. Sometime during the Paleolithic age, the use of fire was discovered, the beginning of agriculture marked the starting of the Neolithic age, creation of metal tools marked the start of the Bronze Age, followed by the Iron Age. In the modern history medieval era emerges into modern times marked by the industrial revolution. Today we are living in the age of big data, as billions of terabytes of data are being generated every year and the size is just beginning to increase. The industrial revolution was marked by the invention of machines which replaced physical human labor work. However, the intellectual and cognitive skills of humans could not be mimicked by machines previously, until recently. Scientists have now understood that it is a matter-of-fact biochemical reaction in the human brain that equip humans with intelligence. Since these biochemical reactions can now be understood by science, they could be applied as algorithms in computers. At this time, there have been numerous instances where artificially intelligent systems have outperformed humans in analyzing skills.

[1], [2] Furthermore, day by day machines are improving in their cognitive skills such as learning and analyzing. The artificially intelligent systems simply work by accessing data in the form of inputs, processing it, and outputting the result.

[1] Department of Computer Science, SZABIST, Hyderabad Campus, Pakistan
Corresponding Author: bushra.memon@hyd.szabist.edu.pk

Artificial Intelligence processes data by various methods, Machine Learning is one of the subfields of artificial intelligence. Machine learning works by processing a dataset by using some algorithm and learning iteratively in the process. The use of machine learning could draw novel conclusions from studies that which human eye or mind could not encompass. The Artificial Intelligent system functions in the environment to meet its defined goals. As per Turning's test, AI should have its memory, be contagious, be able to conclude, and thereby, adapt to the new circumstances [3]. On the other hand, machine learning is just an aspect of artificial intelligence, machine learning algorithms are techniques that perform calculations and make predictions [4].

2. Literature Review

Several studies regarding review do guide applications of the domain of artificial intelligence in healthcare. However, studies are mostly limited to specific problems and are limited in scope. An article was published regarding the review of applications but its scope was limited to drug discovery, clinical trials, and patient care [39].

Mannee et al [40] published a review that explored areas such as Dermatology, Radiology, and Electronic Health Records the scope of this paper was mainly focused on image-processing tasks and the implementation of neural networks.

Jiang et al [41] published a review and comparison of different data types used for diagnosis techniques and proposed a roadmap for natural language processing from clinical data, commonly used machine learning algorithms in medical literature throughout the years. The paper greatly focuses on natural language processing tasks with limited reviews of diagnostic mechanisms.

Davenport et al [42] in their review focused on applications such as patient engagement, administrative activities, treatment recommendations, and diagnosis, ethical issues were also highlighted in the study.

Tran et al [44] focused on the applications of artificial intelligence in the context of infectious diseases, from laboratory diagnostics to clinical prognosis and clinical diagnosis, the infectious diseases applications would aid in a wide range of diseases such as COVID-19, Lyme disease, malaria, and tuberculosis.

The literature review was conducted to access the information provided within different reviews of applications in artificial intelligence in healthcare, the reviews were descriptive but were limited in their context and limited in their applications. Due to limitations in premises available limited deductive reasoning can be made and the review would aid little to the practitioner.

3. Methodology

In everyday life, there are many things where we encounter that apply artificial intelligence, such as the recommendation systems on Netflix. Machine learning can be characterized into three subfields: Supervised, unsupervised, and reinforcement. Which model to choose amongst three of these, depends upon the type of input. Each model has its algorithms.

To implement supervised learning algorithms, the data must be labeled. In supervised learning, features are extracted from the input and linked with the output labels [5] This is the process by which the algorithm performs predictions on the non-labeled data. Classification works by defining the elements into the discrete group based on their features which are extracted from training data calculations [6]. Supervised learning is broadly divided into classification and regression algorithms.

A probabilistic model is one of the classification algorithms, where the application of probability distribution is used to view unnoticed quantities and their relation to the data [7]. Logistic regression is also one

of the classification algorithms, it works by comparing a decision tree or logit function and predicting probability [8]. Naïve Bayes classifier is another classification algorithm. It functions by an assumption that the feature available in a class is not related to other elements' presence [6]. Support vector machine is one of the most famous classification algorithms, this algorithm works by deducing the hyperplane which is at mean distance from two or more classes [9].

Amongst regression, simple linear regression is an algorithm that makes value prediction, it works by deducing the relationship of a single independent variable to a single dependent variable using a straight line [10]. Multiple linear regression is another regression algorithm that functions by deducing the relationship of two or more independent to a single dependent variable by making use of hyperplane [11]. Polynomial regression, decision trees, and random forests are also algorithms applied in supervised learning.

Unsupervised learning does not have labeled data. In these models, the algorithms try to deduce correlations and patterns within data which is raw and is not classified, labeled, or categorized [12]. The techniques used in this subfield are clustering, association rules, and dimensionality reduction [13], [14]. K-means is a clustering algorithm used in unsupervised learning; it forms clusters by deducing points of clusters in proximity from the groups of data points [13]. DBSCAN is another algorithm that forms clusters of high-density regions, compared to low-density regions [3]. The dimensionality reduction follows two main techniques namely; feature selection and feature extraction [15]. The redundant features are refined and removed to fine-tune the compressed data. The principal component analysis is used for large-scale dimensionality reduction [16].

Reinforcement learning follows the path which increases the probability of the system being rewarded. It functions by trial-and-error method and iteratively improves upon receiving the feedback [17]. The system work by reacting to the environment, the actions of the system change the environment, a self-supervisory mechanism evaluates the environment, upon proper action it rewards the system, for example, a score of 1, and for error it penalizes the system, for example, 0 as a score [18].

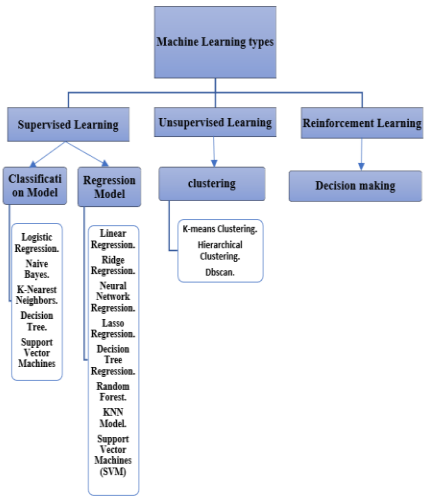


Fig. 1: Machine learning can be categorized into three subfields.

Deep Learning is also a subfield of Machine Learning. It employs the model of neural networks, modeled after the human brain. Artificial neural networks (ANN) have many layers of functions connected by weights just as neurons are connected in the human brain. ANNs that have more than one layer are termed as 'deep'. The networks of these layers self-evaluate the relationship in the raw data.

Deep learning can fall into the supervised, unsupervised, semi-supervised, or reinforcement learning categories. [21] The advantage of deep learning over other models is that the model does not demand any human intervention and automatically extracts features. It consists of an input layer followed by some hidden layers and an output layer. The layers are connected to the next level of

layers by a series of weights, upon training these weights iteratively change to make the best approximation and reach the desired output.

Amongst the categories of neural networks, feedforward is the simplest one where no feedback occurs between layers, it is often used with some backpropagation algorithm. Convolution neural network (CNN) is another type of deep neural network. Within these layers are the convolution layer, activation layer, pooling layer, and the fully-connected classification layer. These algorithms are good for classifiers, for example, the region is normal or tumor. CNN's ability to analyze and identify images makes it helpful in fields like radiology [18]. A full convolution network (FCN) is an improvement of CNN, it consists of convolutional layers instead of fully connected layers. This allows pixel-wise prediction. Therefore, this is a good model for semantic segmentation in medical imaging. Region-based CNN (R-CNN) classifies any sort of object within an image by mixing proposals of rectangle objects with CNN. Recurrent Neural Network (RNN) functions by forming cycles that allow the recycling of computational resources. RNN was facing a vanishing gradient problem during backpropagation [19]. So, its variant Long-short term memory network (LTSM) was created which replaced the recurrent hidden layer with a memory cell.

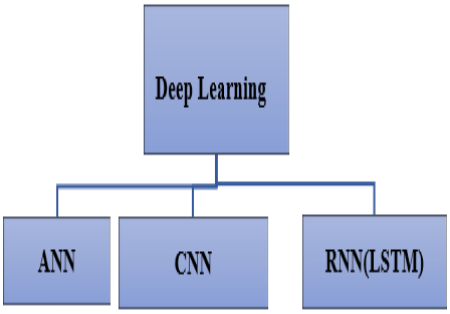


Fig. 2: categories of neural networks

The machine learning process can be described in five distinct stages; Model build-up, training, optimizing, evaluation, and prediction. In the first stage, the data is first, preprocessed, and the noise, redundant and missing values are dealt with. A high-quality database is necessary for the algorithm to perform better [23]. After this, the data is spitted into training, validation, and testing sets. Then the appropriate algorithm is selected. Then the model is trained on the dataset. In the third stage, hyper parameters are adjusted, the validation dataset is presented and the validation error is checked. In the fourth stage, the algorithm is tested on the test dataset and its performance is evaluated. Lastly, the model is used to conduct predictions on unlabeled raw data.

4. Proposed Models

4.1 Cardiology

Adler et al. depict that we can use the decision tree to evaluate the mortality risk in patients with a heart attack. Therefore, make timely decisions [24]. Li et al. findings could be utilized in the early detection of abdominal aortic aneurysms by applying the agonistic learning mechanism of machine learning [25].

4.2 Radiology

Image reconstruction and image analysis could be made possible by machine learning. Deep neural networks could be employed to attain high-resolution and quality images [26]. Furthermore, Convolutional neural networks could be applied for accurate and fast analysis [27].

TABLE 1. Summary of Proposed Models

Branch of medicine	AI Algorithm	Application	References
Cardiology	Decision tree, agnostic machine learning	Risk of heart attack, early detection of abdominal aortic aneurysms	[24-25]
Radiology	CNN, DNN	High quality images, fast and accurate analysis	[26-27
Nephrology	DNN, CNN	Kidney injury detection, global glomerulosclerosis detection	[28-29]
Psychiatry	SVM	Schizophrenia by MRI	[30]
Neurology	ANN	Parkinson's by handwriting	[31-32]
Dentistry	KNN, SVM, Decision tree, Naïve bayes, logistic regression	Growth of bone for orthodontal	[33]
Opthalmology	NN	Optical coherence tomography (OCT) images for diabetic retinopathy	[27]
Nutrition and Diabetology	Boosted decision tree	Personalized nutrition response to post meal glucose	[34]
Infectious Diseases	NN, SVM, Random forest	Epidemic trend of covid 19	[35-36]

4.3 Nephrology

Machine learning techniques could be applied to the prediction of organ damage. Deep neural networks could be applied to detect kidney injury 48 hours in advance, which can ensure timely treatment [28]. Altini et al. conducted a study where they analyzed the histological slides of the kidney and determines the global glomerulosclerosis, which is a necessary step in the pre-transplantation process. Using convolution neural networks, the whole analysis process could be quickened [29].

4.4 Psychiatry

Lei et al. conducted a study where schizophrenia could be analyzed at the level of the individual [30]. This is made possible by a support vector machine which analyzes the MRI images and makes a fast diagnosis of schizophrenia

4.5 Neurology

Cascara no et al. suggests that Parkinson's disease could be predicted by analyzing handwriting [31]. Artificial Neural Networks could be employed to analyze the handwriting of subjects and deduce Parkinson's disease. Interestingly, the model could detect even the progression such as mild to moderate course of the disease. Mellema et al. implemented machine learning techniques for MRI analysis and diagnosis of autism spectrum disorder [32]. A naïve Bayes, support vector machine, adaptive boosting, decision tree base, neural network, logistic regression, random forest,

all these techniques go into image analysis and diagnosis.

4.6 Dentistry

Kok et al. advised artificial intelligence techniques could be used for the prediction of the growth of bone [33]. These techniques could be applied to make personalized decisions as to when to seek orthodontal treatment. K-nearest neighbors, support vector machine, decision tree, a naïve Bayes, neural network, logistic regression, and random forest could be applied in this section.

4.7 Opthalmology

Kermany et al. conducted a study where they used neural networks to evaluate the optical coherence tomography (OCT) images of the retina of the human eye, Diabetic retinopathy or muscular degeneration was predicted with high accuracy [27].

4.8 Nutrition and Diabetology

Zeevi et al. advises personalized nutrition in response to post-meal glucose response [34]. A boosted decision tree could be utilized here to predict the glucose response of the patient and therefore make relevant decisions in the nutrition and diet setting of patients especially those with diabetes.

4.8.1 Infectious Diseases

At this time the world is struck by the novel Coronavirus. Artificial intelligence should aid the healthcare sector in many perspectives. Liu et al conducted a study aimed at evaluating the trend of the epidemic trend of Covid-19 [35]. Such an application could be useful in mobilizing public health services. Neural networks could be used to predict possible future cases of coronavirus so hospitals could be mobilized likewise. Furthermore, early diagnosis of covid-19 could be made by applying CNN, SVM, and random forests [36].

5. Conclusion

The discovery of the microscope in the sixth century transformed traditional medicine. Since the human eye could have a limitation and could not look at structures as small as cells in living things. The microscope enabled mankind to look beyond what the naked eye could see. Likewise, the advent of artificial intelligence helps researchers find patterns and correlations between the data that otherwise human intellect might not have been able to perceive. It is just the beginning of the big data era, every year more and more data is being produced and faster computers are being developed to process that data. Aforetime mentioned techniques could be used by students or researchers to make practical medical diagnostic tools growth of bone [33]. These techniques could be applied to make the personalized decision as to when to seek orthodontal treatment. K-nearest neighbors, support vector machine, decision tree, a naïve Bayes, neural network, logistic regression, and random forest could be applied in this section.

6. Future work

Yuval Noah Hariri, discusses in his book 21 Lessons for the 21st Century, the concept of AI doctors. As communication between every single doctor on earth is not possible but AI doctors could be connected with other doctors who could communicate in real-time. An AI doctor in Mongolia could communicate with other AI doctors in Kansas, which is not possible in the case of human doctors. The AI doctor could be connected to all other doctors in the world. The idea might be too futuristic but AI is rapidly advancing and the once thought-impossible cognitive skills are being developed within machines. The human senses such as hearing, sight, touch, and even smell is developed using an electric nose [37]. However, there are still some challenges such as if a human doctor performs malpractice he could be brought to a court of law. If an AI doctor makes a misdiagnosis, who is to blame? And will the whole connected system of AI doctors mimic that error? Even if each AI system is localized there are still moral and legal issues that would need to be solved.

REFERENCES

[1] C. Brooks, "Linear and non-linear (non-) forecastability of high-frequency exchange rates," *Journal of forecasting*, vol. 16, pp. 125-145, 1997.

[2] N. Ernest and D. Carroll, "Genetic Fuzzy based Artificial Intelligence for Unmanned Combat Aerial Vehicle Control in Simulated Air Combat Missions," *J. Def. Manag.*, vol. 06, no. 01, 2016, doi: 10.4172/2167-0374.1000144.

[3] F. Stulp and O. Sigaud, "Many regression algorithms, one unified model: A review," *Neural Networks*, vol. 69, pp. 60–79, Sep. 2015, doi: 10.1016/J.NEUNET.2015.05.005.

[4] M. W. Libbrecht and W. S. Noble, "Machine learning applications in genetics and genomics," *Nat. Rev. Genet. 2015 166*, vol. 16, no. 6, pp. 321–332, May 2015, doi: 10.1038/nrg3920.

[5] Z. Ghahramani, "Probabilistic machine learning and artificial intelligence," *Nat. 2015 5217553*, vol. 521, no. 7553, pp. 452–459, May 2015, doi: 10.1038/nature14541.

[6] J. S. Cramer, "The Origins of Logistic Regression," *SSRN Electron. J.*, 2005, doi: 10.2139/ssrn.360300.

[7] S. Neelamegam and E. Ramaraj, "Classification algorithm in Data mining : An Overview," *Int. J. P2P Netw. Trends Technol.*, vol. 4, no. 8, pp. 369–374, 2013.

[8] K. H. Zou, K. Tuncali, and S. G. Silverman, "Correlation and Simple Linear Regression1," *https://doi.org/10.1148/radiol.2273011499*, vol. 227, no. 3, pp. 617–622, Jun. 2003, doi: 10.1148/RADIOL.2273011499.

[9] "The Concise Encyclopedia of Statistics - Yadolah Dodge–Google 368.&ots=9n21631kn_&si Books."https://books.google.com.pk/books?hl=fnd&lr=&id=k2zklGOBRDwC&oi=fnd&pg=PP6&dq=Multiple+Linear+Regression.+In+The+Concise+Encyclopedia+of+Statistics%3B+Springer:+New+York,+NY,+USA,2008%3B+pp.+364–368.&ots=9n21631kn_&sig=T-1Cw1nSjl14TzvU17XBDvYu0c&redir_esc=y#v=onepage&q&f=false(accessed May 26, 2023).

[10] S. Lloyd, M. Mohseni, and P. Rebentrost, "Quantum algorithms for supervised and unsupervised machine learning," Jul. 2013, Accessed: May 26, 2023. [Online]. Available: https://arxiv.org/abs/1307.0411v2

[11] M. Wang, F. Sha, and M. I. Jordan, "Unsupervised Kernel Dimension Reduction," *Adv. Neural Inf. Process. Syst.*, vol. 23, 2010.

[12] K. J. Cios, R. W. Swiniarski, W. Pedrycz, and L. A. Kurgan, "Unsupervised Learning: Association Rules," *Data Min.*, pp. 289–306, Oct. 2007, doi: 10.1007/978-0-387-36795-8_10.

[13] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "DBSCAN Revisited, Revisited," *ACM Trans. Database Syst.*, vol. 42, no. 3, Jul. 2017, doi: 10.1145/3068335.

[14] C. Lazar *et al.*, "A survey on filter techniques for feature selection in gene expression microarray analysis," *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, vol. 9, no. 4, pp. 1106–1119, 2012, doi: 10.1109/TCBB.2012.33.

[15] C. Peng, Y. Chen, Z. Kang, C. Chen, and Q. Cheng, "Robust principal component analysis: A factorization-based approach with linear complexity," *Inf. Sci. (Ny).*, vol. 513, pp. 581–599, Mar. 2020, doi: 10.1016/J.INS.2019.09.074.

[16] G. Choy *et al.*, "Current Applications and Future Impact of Machine Learning in Radiology," *https://doi.org/10.1148/radiol.2018171820*, vol. 288, no. 2, pp. 318–328, Jun. 2018, doi: 10.1148/RADIOL.2018171820.

[17] D. Silver *et al.*, "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play," *Science (80-.).*, vol. 362, no. 6419, pp. 1140–1144, Dec. 2018, doi: 10.1126/SCIENCE.AAR6404/SUPPL_FILE/AAR6404_DATAS1.ZIP.

[18] Learning Approaches for Health Informatics," *Deep Learn. Parallel Comput. Environ. Bioeng. Syst.*, pp. 123–137, Jan. 2019, doi: 10.1016/B978-0-12-816718-2.00014-2.

[19] T. M. Navamani, "Efficient Deep Learning Approaches for Health Informatics," *Deep Learn. Parallel Comput. Environ. Bioeng. Syst.*, pp. 123–137, Jan. 2019, doi: 10.1016/B978-0-12-816718-2.00014-2.

[20] S. Zhang, C. Zhang, and Q. Yang, "Data preparation for data mining," *http://dx.doi.org/10.1080/713827180*, vol. 17, no. 5–6, pp. 375–381, May 2010, doi: 10.1080/713827180.

[21] E. D. Adler *et al.*, "Improving risk prediction in heart failure using machine learning," *Eur. J. Heart Fail.*, vol. 22, no. 1, pp. 139–147, Jan. 2020, doi: 10.1002/EJHF.1628.

[22] J. Li *et al.*, "Decoding the Genomics of Abdominal Aortic Aneurysm," *Cell*, vol. 174, no. 6, pp. 1361-1372.e10, Sep. 2018, doi: 10.1016/J.CELL.2018.07.021.

[23] B. Zhu, J. Z. Liu, S. F. Cauley, B. R. Rosen, and M. S. Rosen, "Image reconstruction by domain-transform manifold learning," *Nat. 2018 5557697*, vol. 555, no. 7697, pp. 487–492, Mar. 2018, doi: 10.1038/nature25988.

[24] D. S. Kermany *et al.*, "Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning," *Cell*, vol. 172, no. 5, pp. 1122-1131.e9, Feb. 2018, doi: 10.1016/J.CELL.2018.02.010.

[25] N. Tomašev *et al.*, "A clinically applicable approach to continuous prediction of future acute kidney injury," *Nat. 2019 5727767*, vol. 572, no. 7767, pp. 116–119, Jul. 2019, doi: 10.1038/s41586-019-1390-1.

[26] V. Bevilacqua *et al.*, "A comparison between two semantic deep learning frameworks for the

autosomal dominant polycystic kidney disease segmentation based on magnetic resonance images," *BMC Med. Inform. Decis. Mak.*, vol. 19, no. 9, pp. 1–12, Dec. 2019, doi: 10.1186/S12911-019-0988-4/TABLES/6.

[27] D. Lei *et al.*, "Integrating machining learning and multimodal neuroimaging to detect schizophrenia at the level of the individual," *Hum. Brain Mapp.*, vol. 41, no. 5, pp. 1119–1135, Apr. 2020, doi: 10.1002/HBM.24863.

[28] G. D. Cascarano *et al.*, "Biometric handwriting analysis to support Parkinson's Disease assessment and grading," *BMC Med. Inform. Decis. Mak.*, vol. 19, no. 9, pp. 1–11, Dec. 2019, doi: 10.1186/S12911-019-0989-3/TABLES/18.

[29] C. Mellema, A. Treacher, K. Nguyen, and A. Montillo, "Multiple deep learning architectures achieve superior performance diagnosing autism spectrum disorder using features previously extracted from structural and functional mri," *Proc. - Int. Symp. Biomed. Imaging*, vol. 2019-April, pp. 1891–1895, Apr. 2019, doi: 10.1109/ISBI.2019.8759193.

[30] H. Kök, A. M. Acilar, and M. S. İzgi, "Usage and comparison of artificial intelligence algorithms for determination of growth and development by cervical vertebrae stages in orthodontics," *Prog. Orthod.*, vol. 20, no. 1, pp. 1–10, Dec. 2019, doi: 10.1186/S40510-019-0295-8/TABLES/5.

[31] G. D. Cascarano *et al.*, "Biometric handwriting analysis to support Parkinson's Disease assessment and grading," *BMC Med. Inform. Decis. Mak.*, vol. 19, no. 9, pp. 1–11, Dec. 2019, doi: 10.1186/S12911-019-0989-3/TABLES/18.

[32] C. Mellema, A. Treacher, K. Nguyen, and A. Montillo, "Multiple deep learning architectures achieve superior performance diagnosing autism spectrum disorder using features previously extracted from structural and functional mri," *Proc. - Int. Symp. Biomed. Imaging*, vol. 2019-April, pp. 1891–1895, Apr. 2019, doi: 10.1109/ISBI.2019.8759193.

[33] H. Kök, A. M. Acilar, and M. S. İzgi, "Usage and comparison of artificial intelligence algorithms for determination of growth and development by cervical vertebrae stages in orthodontics," *Prog. Orthod.*, vol. 20, no. 1, pp. 1–10, Dec. 2019, doi: 10.1186/S40510-019-0295-8/TABLES/5.

[34] D. Zeevi *et al.*, "Personalized Nutrition by Prediction of Glycemic Responses," *Cell*, vol. 163, no. 5, pp. 1079–1094, Nov. 2015, doi: 10.1016/J.CELL.2015.11.001.

[35] D. Lei *et al.*, "Integrating machining learning and multimodal neuroimaging to detect schizophrenia at the level of the individual," *Hum. Brain Mapp.*, vol. 41, no. 5, pp. 1119–1135, Apr. 2020, doi: 10.1002/HBM.24863.

[36] B. P. Little *et al.*, "Artificial intelligence-enabled rapid diagnosis of patients with COVID-19," *Nat. Med.*, doi: 10.1038/s41591-020-0931-3.

[37] D. Karakaya, O. Ulucan, and M. Turkan, "Electronic Nose and Its Applications: A Survey", doi: 10.1007/s11633-019-1212-9.

[38] R. V. Yampolskiy, "Turing test as a defining feature of AI-completeness," *Stud. Comput. Intell.*, vol. 427, pp. 3–17, 2013, doi: 10.1007/978-3-642-29694-9_1.

[39] M. Y. Shaheen, "Applications of Artificial Intelligence (AI) in healthcare: A review," *Sci. Prepr.*, Sep. 2021, doi: 10.14293/S2199-1006.1.SOR-.PPVRY8K.V1.

[40] R. Manne, S. K.-C. J. of A. S. and, and undefined 2021, "Application of artificial intelligence in healthcare: chances and challenges," *papers.ssrn.com*, vol. 40, no. 6, pp. 78–89, 2021, doi: 10.9734/CJAST/2021/v40i631320.

[41] F. Jiang *et al.*, "Artificial intelligence in healthcare: past, present and future," *svn.bmj.com*, doi: 10.1136/svn-2017-000101.

[42] T. Davenport, R. K.-F. healthcare journal, and undefined 2019, "The potential for artificial intelligence in healthcare," *ncbi.nlm.nih.gov*, Accessed: May 30, 2023. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6616181/

[43] N. Tran, S. Albahra, L. May, … S. W.-C., and undefined 2022, "Evolving applications of artificial intelligence and machine learning in infectious diseases testing," *academic.oup.com*, Accessed: May 30, 2023. [Online]. Available: https://academic.oup.com/clinchem/article-abstract/68/1/125/6490223

[44] Y. Harari, "21 lessons for the twenty-first century," 2018, Accessed: May 30, 2023. [Online]. Available: http://www.waverleyseniorcitizensclub.org.au/files/21-Lessons-For-the-Twenty-First-Century.pdf

Vol. 7, No. 2 | July – December 2023

Towards a Trustworthy and Efficient ETL Pipeline for ATM Transaction Data

Muhammad Ahmad Ashfaq [1], Nimra Haq [2], Usman Arshad[3], Muhammad Farooq[4], and Shuja ur Rehman Baig

Abstract:

ATMs generate vast amounts of data daily, which needs to be analyzed and stored. Dealing with this data also termed big data, is a complex task, and here comes the role of ETL pipelines. ETL pipelines need extensive resources for operations, and their performance optimization is necessary as data must be dealt with in near or even real-time. If the pipeline deals with financial data such as ATM transactions, steps should be taken to ensure the data's security, privacy, confidentiality, and integrity. This can be achieved using Blockchain technology. It is a distributed ledger technology having an immutable nature. It has significant advantages in terms of providing security, but it has disadvantages as well, such as low throughput and transactional latency. If blockchain is used in an ETL pipeline, it will affect the overall performance. So, to prevent the decline in performance, steps should be taken to optimize it. In this paper, we are using parallelization and partitioning as techniques to optimize performance. The primary goal here is to achieve maximum security while maintaining performance.

Keywords: *ETL Pipeline, Big Data, Blockchain, Performance optimization, Kafka, Spark*.

1. Introduction

Today, technology has digitized the finance sector and changed how customers interact with financial institutions and get their services. This change can be seen in the wide use of Automated Teller Machines(ATMs). ATMs play a vital role in providing certain banking services at any time with convenience and ease. As users perform transactions using ATMs, a considerable amount of valuable financial data is generated. Businesses want to use this data for their benefit as it can provide insights into customer behaviors and preferences when analyzed.

This data can also be used for better-aimed marketing, analyzing overall market patterns, and getting more business insights. The banking institutions can use this data to take finer steps to improve customer experience and optimize their operations.

Big data can be explained in terms of volume, velocity, and variety. It refers to massive datasets that can be wide-ranging (structured, unstructured, or semi-structured) and have complex structures. Ten percent of data collected and generated by businesses is structured, ten percent is semi-structured, and the rest is unstructured. These datasets can pose significant difficulty in storing, analyzing, and visualizing them. Big data analytics is the research process into such datasets to identify patterns and hidden correlations. The data generated by ATM daily transactions is so massive that it can be termed Big Data.

[1]FCIT, Faculty of Computing and Information Technology, Shahrahe Quaid-e-Azam Allama Iqbal Campus (Old Campus), Lahore, Pakistan
Corresponding Athor: shuja@pucit.edu.pk

Managing, processing, and analyzing this vast data offers great complexity and difficulty. So, robust and scalable data pipelines are required for this purpose. Data pipelines are classified into many categories, one of them is ETL. ETL stands for Extract, Transform, and Load. The ETL pipelines function like circulatory systems, moving data from its source to the intended destination and enabling near real-time analysis and decision-making. The source and the destination can be separated physically, and transformations may take place in between. Data is retrieved from various heterogeneous sources such as databases, APIs, or other structured or unstructured sources during the extraction phase. The extracted data can be of different formats, e.g., text files, images, videos, emails, XML, JSON, CSV, etc. The transformation phase is quite diverse. For instance, basic transformation may include replacing NULL values with a zero or removing duplicate values. Transformation may have joins, which can be complex sometimes, aggregation of rows, splitting of columns, etc. In this phase, the data is transformed to make it usable at the destination. In this phase, the transformed data is loaded into the destination. The destination can be a traditional or non-traditional database, visualization tools, machine learning models, or deep learning models.

2. Literature Review

In the "Age of Data," industries and public bodies are producing vast amounts of new data at an unprecedented rate. Organizations invest heavily in utilizing this data to create value through big data analytics. The premise is that by analyzing large volumes of unstructured data from various sources, actionable insights can transform businesses and provide a competitive edge. These data-driven insights are crucial, especially for organizations in fast-paced environments where informed decisions are vital[15]. Collecting data from multiple resources, processing it for analytical purposes, and transporting it to the destination is challenging, and data pipelines are used to manage it efficiently. Data pipelines have become a necessity for all data-driven companies[1].

Raj et al. [2] created a pipeline for analyzing datasets containing trip records of Uber, yellow, and green taxis using big data technologies such as MapReduce, Hive, and Spark. The analysis enabled us to suggest whether yellow, green, or Uber is the right choice for a rider. This system could suggest the regions to focus on for drivers depending on competitor presence and historical pickups. Mehmood and Anees[3] focused on designing distributed real-time ETL architecture for unstructured big data. They proposed an architecture using Apache Kafka, MongoDB, and Apache Spark. The method they presented and employed for experimentation can be easily applied when distributed data needs to be combined with a fast incoming unstructured stream of data in real-time. Farki and Noughabi [4] suggested a real-time blood pressure prediction method. Apache Kafka and Apache Spark were utilized to handle the large influx of incoming signals from diverse sources, encompassing wearable technology and IoT sensors. Machine learning algorithms such as K-means and Random Forest Regression are implemented using Spark MLlib to improve the precision of this model.

Leveraging big data technologies like Apache Kafka and Apache Spark simplifies the management of data pipelines. Apache Kafka streamlines the processing of vast volumes of real-time data from diverse sources, offering fault tolerance, scalability, and efficient data handling. On the other hand, Apache Spark provides a scalable and practical approach to both machine learning model development and real-time data processing tasks.[4].

With the growing volume of data, the ETL jobs of many enterprises may take hours or days to complete. This latency may cause incorrect decision-making. So, there is a need to optimize the ETL pipeline data flow as the demand for shorter time processing time for ETL processes is increasing. Various case studies have provided evidence of the efficacy

of these approaches in practical settings. For instance, a research investigation conducted by Ranjan J (2009)[10] discovered that implementing data warehousing and business intelligence tools, alongside optimized data extraction processes, yielded notable enhancements in the performance of ETL tasks within a prominent financial services organization. According to [11], parallel processing is a powerful technique that enhances the performance of ETL processes by executing multiple tasks simultaneously. It increases overall throughput, allows scalability with additional resources, reduces latency, and improves fault tolerance. Task parallelism divides large ETL jobs into smaller tasks executed through thread-based, process-based, or cluster-based parallelism. Pipeline parallelism divides jobs into stages, executed via multi-threaded, multi-process, or multi-node pipelines. Cloud-based parallel processing utilizes cloud services for distributed execution. Data partitioning divides large datasets for parallel processing. Conversely, caching stores frequently accessed data in temporary storage, reducing retrieval time source system load and improving scalability and data consistency. In-memory, disk-based, and distributed caching strategies can be used. The choice of parallel processing and caching strategies depends on data, processing time, and resource requirements. These techniques collectively optimize ETL performance and efficiency.

Blockchain is the technology behind the birth of Bitcoin and cryptocurrency. According to Bhutta et al.[5], Blockchain's key characteristics include decentralization, transparency, autonomy, security, immutability, traceability, democratization, and fault tolerance. Blockchain is a transformational technology that can provide a basis to develop distributed and secure applications for industries like finance, health care, government, manufacturing, distribution, etc. One use case described by Teogenes & Gomes[6] is using blockchain in e-voting systems. The current voting methods, electronic or not, cause an unsatisfactory level of voter confidence. Blockchain would leverage security, transparency, and immutability to increase voter confidence and strengthen democracy.

Ali Syed et al.[7], talks about the use of blockchain in the vehicle industry. BMW has implemented blockchain technology to handle its asset and logistics operations; since 2016, Toyota has invested in blockchain-based supply chain management. Furthermore, BMW, Ford, Renault, and General Motors are part of the Mobility Open Blockchain Initiative (MOBI), including IBM, Bosch, and Blockchain at Berkeley, among 30 other companies. MOBI's primary objective is to encourage the adoption of blockchain technology and establish industry-wide collaboration.

According to Monrat et al.[8], Blockchain can be used in health care to trace medicines and patient data. One of the major concerns for the healthcare industry is managing patient data integrity. Blockchain can solve data integrity problems because of its immutable and secure nature. Haderet al.[9], presented a framework that integrates blockchain and big data to enhance supply chain traceability and facilitate information sharing within the textile industry.

In conclusion, the literature review highlights the significance of data pipelines in managing and processing data efficiently in today's data-driven landscape. It explores the implementation of big data technologies, such as Apache Kafka and Apache Spark, for optimized data management. Moreover, it emphasizes the potential of blockchain technology in various industries, including e-voting, the automotive sector, healthcare, and supply chain management. The reviewed studies demonstrate the real-world efficacy of these approaches and lay the foundation for further research and innovation in data pipeline management and blockchain integration.

3. Technological Frameworks

3.1 Apache Kafka

In distributed stream processing applications, ensuring strong correctness guarantees in the face of unexpected failures and out-of-order data is crucial. This provides reliable and authoritative results without depending on complementary batch results. While existing systems tackle issues like consistency and completeness, finding the optimal balance between correctness, performance, and cost remains a practical challenge for users. Apache Kafka addresses this challenge through its core design for stream processing, leveraging its persistent log architecture for storage and communication between processors. By doing so, it achieves the desired correctness guarantees. Kafka Streams, a scalable stream processing client library within Apache Kafka, utilizes read-process-write cycles to capture state updates and outputs as log appends, providing a robust and reliable solution [13].

In the current era of big data, the primary challenge lies in collecting the vast amounts of data generated [15]. Apache Kafka, a free and open-source distributed streaming platform/messaging system, stands out for its capability to manage large volumes of incoming data streams. It is widely utilized for data extraction from diverse and heterogeneous sources, owing to its ability to ingest expanding data volumes from unstructured or semi-structured data sources. Renowned organizations like Twitter, Walmart, and others extensively use Kafka. Apache Kafka's key features, such as high throughput, scalability, fault tolerance, and reliability, make it an excellent and preferred choice for handling ATM transaction data.

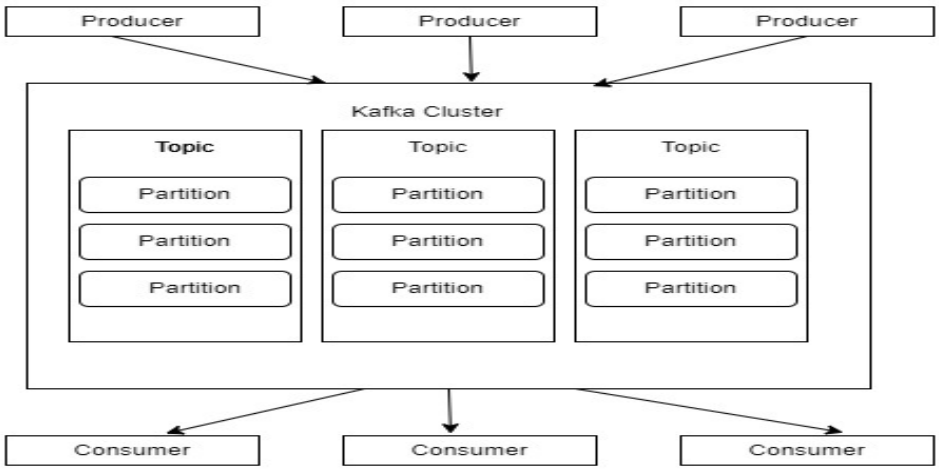


Fig. 1: A typical ANN model

Kafka consists of clusters of multiple brokers that store data assigned to different Kafka topics. A topic can have multiple partitions and be replicated across multiple brokers.

Data producers write data on different Kafka topics. The number of partitions and replication factors for a Kafka topic can be defined at the time of Kafka topic creation. A partition consists of messages in a sequence, and new messages are added at the end of the partition. Replication of topics across multiple brokers prevents data loss in case of a broker

failure. Consumers subscribe to a specific Kafka topic and fetch messages from it. A consumer may belong to a consumer group. A consumer can be a database such as HBase Cassandra or a real-time consumer such as Spark or Storm.

Numerous companies across various industries have adopted Apache Kafka as the fundamental infrastructure for data pipelines, streaming analytics, data integration, and critical applications. Data in Kafka is organized into topics, each of which can have multiple partitions. These partitions are maintained as immutable sequences of records, functioning like logs. Producers can continuously add data to partitions, while consumers can continuously read from them [16].

3.2 Apache Spark

Big data analytics, crucial in storing, processing, and analyzing massive datasets, has become indispensable [15]. With the emergence of distributed computing frameworks like Spark, efficient solutions to explore vast amounts of data are now available. Spark's popularity has surged due to its accessible application programming interface (API) and exceptional performance, surpassing the MapReduce framework. The default system parameters in Spark make it effortless for system administrators to deploy their applications and measure specific cluster performance using factory-set parameters [12].

Apache Spark, a widely adopted open-source framework, is renowned for its ability to handle extensive data processing tasks. It offers a programming interface that facilitates cluster programming with implicit data parallelism and ensures fault tolerance.

The process of training machine learning models faces challenges that cause slowdowns, such as the dataset size and the optimization parameters needed to create the best-fitting model. To address these issues, researchers have sought a more suitable approach. One potential solution is employing the Apache Spark tool, a high-speed cluster computing framework and open-source distributed programming tool for clusters. Additionally, Spark performs operations in memory, further enhancing its efficiency [17].

Spark provides Java, Scala, Python, and R APIs and an optimized engine that executes general execution graphs. Spark excels in iterative computations, making it an ideal choice for creating large-scale machine-learning applications.

In the Apache Spark architecture, when the Driver Program executes, it calls the actual application program and establishes a SparkContext containing all the fundamental functions. Alongside the SparkContext, the Spark Driver comprises other essential components such as the DAG Scheduler, Task Scheduler, Backend Scheduler, and Block Manager. These components combine to convert user-written code into jobs executed on the cluster.

The Cluster Manager is responsible for managing the execution of various jobs within the cluster. The Spark Driver works hand in hand with the Cluster Manager to oversee the execution of different jobs. The Cluster Manager allocates resources for the job, divides them into smaller tasks, and distributes them to worker nodes. The Spark Driver takes charge of controlling this execution process.

Multiple worker nodes can be employed to process an RDD created in SparkContext, and the results can also be cached for optimization. The Spark Context receives task information from the Cluster Manager and enqueues it on worker nodes. The executor manages the execution of these tasks. The lifespan of executors aligns with that of the Spark Application, and if desired, increasing the number of workers can enhance the system's performance, allowing for the division of jobs

into more manageable parts.

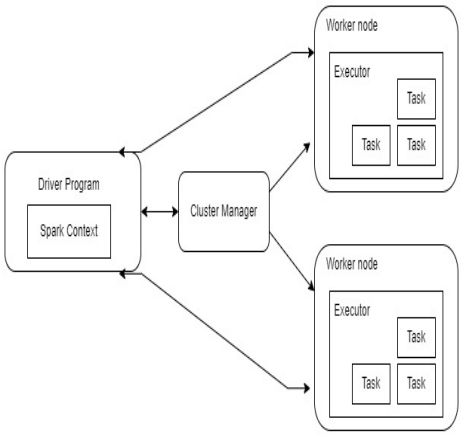


Fig. 2: Apache Spark Internal Architecture

There are several platforms and engines available for transforming the data, for example, Hadoop MapReduce, Apache Flink, and Apache Storm, but the spark is preferred for our ETL pipeline considering the following reasons:

Scalability: Spark is designed to handle extensive scale data processing, making it one of the best choices for data transformation. Distributing computation across a cluster of machines enables parallel processing and effective utilization of available resources.

Speed: Spark is renowned for its exceptional processing speed. It achieves this by conducting computations in memory, minimizing disk input/output (I/O), and considerably expediting data transformation operations. Moreover, Spark's capability to store intermediate data in memory through caching further amplifies its performance, resulting in accelerated data transformation tasks.

Fault Tolerance: Spark incorporates inherent fault tolerance mechanisms, providing a robust system for handling failures. It can automatically recover from errors, guaranteeing uninterrupted progress in the data transformation process.

Flexibility: Spark offers a wide range of programming interfaces, including Java, Scala, Python, and R. Moreover, It also boasts a comprehensive ecosystem comprising numerous libraries and extensions.

3.3 Random Walk Model

Blockchain technologies have become prominent in recent years, with many experts citing the technology's potential applications regarding different aspects of any industry, market, agency, or governmental organization. In the brief history of blockchain, many achievements have been made regarding how blockchain can be utilized and the impacts it might have on several industries.[18].

Blockchain is recognized for its decentralized, autonomous, and immutable characteristics, providing various features such as verification, fault tolerance, anonymity, auditability, and transparency [14]. Blockchain is a distributed and decentralized ledger that records transactions across a network of computers, ensuring immutability. It provides essential features such as authentication, integrity, traceability, privacy, confidentiality, and fault tolerance.

There are three main types of blockchains:

Permissionless or Public blockchains: These allow anyone to join the network and participate in managing the blockchain.

Permissioned or Private blockchains: Only invited individuals from a single organization can join the network and take part in managing the blockchain.

Consortium blockchains: Invited members from various organizations can join and participate in the consortium blockchain's management.

Here are some key aspects that contribute to blockchain's security and immutability:

1. **Decentralization:** To collectively maintain and validate the database, blockchain operates on a decentralized network of nodes. These nodes working in peer-to-peer architecture form a blockchain network. This decentralized system removes the need for a central authority, making it resistant to single points of failure and reducing the risk of unauthorized tampering or data manipulation.

2. **Distributed Ledger:** The blockchain functions as a distributed ledger, chronologically recording all transactions or data entries. Within the network, each node retains a copy of the complete blockchain, and the addition of transactions to the ledger follows a consensus mechanism like proof-of-work or proof-of-stake. This decentralized approach guarantees the existence of multiple copies of the ledger, rendering it challenging for attackers to alter the data without consensus from the majority of nodes.

3. **Cryptographic Hash Functions:** Blockchain employs cryptographic hash functions to safeguard the data's integrity. Every block within the blockchain contains a unique hash value generated based on its data content. If any alteration is made to the data within a block, it will lead to a distinct hash value, making any tampering easily detectable. This crucial characteristic guarantees that once a block is added to the blockchain, it becomes practically impossible to modify or erase the data without being noticed.

4. **Immutable Records**: Once data is added to the blockchain, it becomes virtually immutable. The decentralized and distributed nature of the blockchain, coupled with the cryptographic hash functions, ensures that historical transactions or data entries resist modification. This immutability provides high trust and transparency, as it becomes difficult to dispute or alter past records.

5. **Consensus Mechanism:** Blockchain networks rely on consensus mechanisms to agree on the validity of transactions or data entries. Consensus algorithms ensure that all nodes in the network reach an agreement on the order and validity of transactions, preventing fraudulent or conflicting entries. This consensus process strengthens the security of the blockchain by requiring a majority of nodes to validate and agree on the data being added.

6. **Encryption:** Blockchain can integrate encryption techniques to safeguard sensitive data. Encryption guarantees that the data stored on the blockchain remains confidential and can only be accessed by authorized parties possessing the correct decryption keys. Through data encryption, blockchain adds an extra layer of security, particularly for sensitive information like personal or financial data.

By combining these elements, blockchain technology provides a secure and immutable database resistant to tampering, fraud, and unauthorized access. Its decentralized nature, cryptographic principles, and consensus mechanisms create a trustless environment where participants can confidently interact and rely on the integrity and security of the stored data.

4. Proposed Solution

4.1 Data Collection and Ingestion

Apache Spark is used for data collection and ingestion. The whole process is described in this section.

4.1.1 Overview of ATM transaction data

Financial transactions conducted at ATMs provide valuable information about customer

behavior, banking patterns, and cash flow. Here is an overview of the typical information captured during ATM transactions:

Date and Time: The timestamp indicates when the transaction took place. It includes the date, time, and time zone.

Transaction Type: Specifies the nature of the transaction, such as cash withdrawal, cash deposit, funds transfer, or other services.

ATM Location: Records the physical location of the ATM, identified by an address or geographic coordinates (latitude and longitude).

Card Information: Encrypted details related to the card used for the transaction, including the card number and expiration date. Note that sensitive cardholder data like the cardholder's name, PIN, or CVV (Card Verification Value) is typically not stored in the transaction data.

Transaction Amount: Indicates the monetary value involved in the transaction.

Account Information: Identifies the bank account associated with the transaction, usually by an account number or an encrypted identifier.

Transaction Result: Specifies the transaction's outcome, whether it was successful, declined, canceled, or encountered an error.

ATM Terminal ID: A unique identifier assigned to each physical ATM terminal, distinguishing it from other ATMs within a network.

Currency: Denotes the currency in which the transaction was conducted, such as USD (United States Dollar), EUR (Euro), GBP (British Pound), etc.

Additional Messages: Records any additional message utilized during the transaction, like language selection, receipt printing, or screen customization.

The dataset used in this paper can be accessed at https://www.kaggle.com/datasets/sparnord/danish-atm-transactions.

4.1.2 Integration of Kafka for ATM Transaction Data Ingestion

We are using Kafka to collect ATM transaction data as it can handle high-throughput data streams, provide fault tolerance, and enable real-time processing. Collecting ATM transaction data from different ATMs involves setting up a Kafka infrastructure to receive, store, and process the data. Here's an explanation of the process:

ATMs are the source here. An ATM does not have its own dedicated Kafka producer. Instead, a middleware layer containing several Kafka producers is usually responsible for collecting the transaction data from multiple ATMs. This system acts as a producer and forwards the data to appropriate Kafka topics.

A Kafka topic is explicitly created for storing ATM transaction data. The middleware produces the collected ATM data for this Kafka topic. A Kafka topic is a channel where ATM transaction data is organized and published. Think of it as a virtual container or a labeled stream of data.

After the data is produced into the Kafka topic, the Kafka cluster provides the infrastructure to handle the data flow. A Kafka cluster consists of multiple Kafka brokers that form a distributed system. Brokers are individual server instances that include the distributed messaging system. Each broker is responsible for handling a portion of the data, including storing and replicating the data across the cluster. It facilitates the streaming of data in real-time. As new data arrives, it is immediately made available to consumers subscribed to the corresponding Kafka topic, enabling real-time processing, analytics, and integration with downstream systems. A

Kafka consumer refers to an application or service subscribing to a Kafka topic and consuming the transaction message published to that topic for further processing. The following figure explains the process:

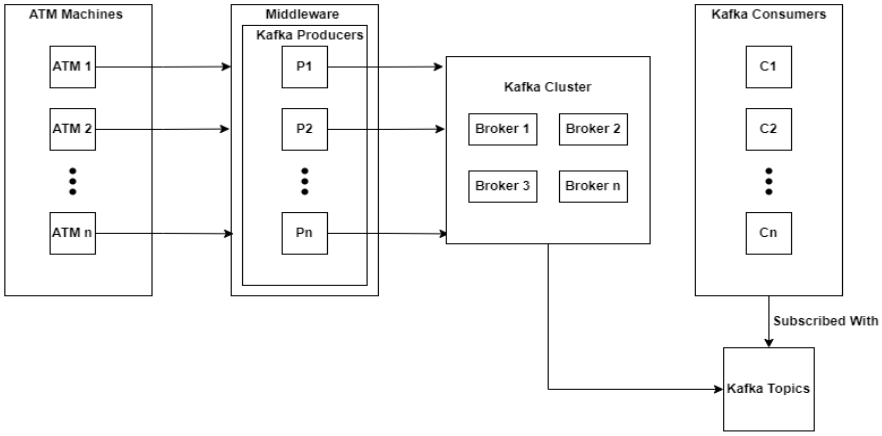


Fig. 3: Integration of Kafka for ATM transaction data ingestion

4.2 Data Transformation and Analysis

Various transformations can be applied depending on the dataset and specific use case. These transformations include Filtering, Mapping, Aggregation, Data Cleansing, and Machine Learning Transformations.

In our particular use case, the initial step involves mapping the dataset onto a specific schema, eliminating non-essential attributes. Subsequently, Data Cleansing is executed to address missing values and duplicates. Lastly, the Data Frame is aggregated based on the transaction month. The desired output comprises grouped rows of data that have been cleansed, mapped, and organized according to the month of the transactions.

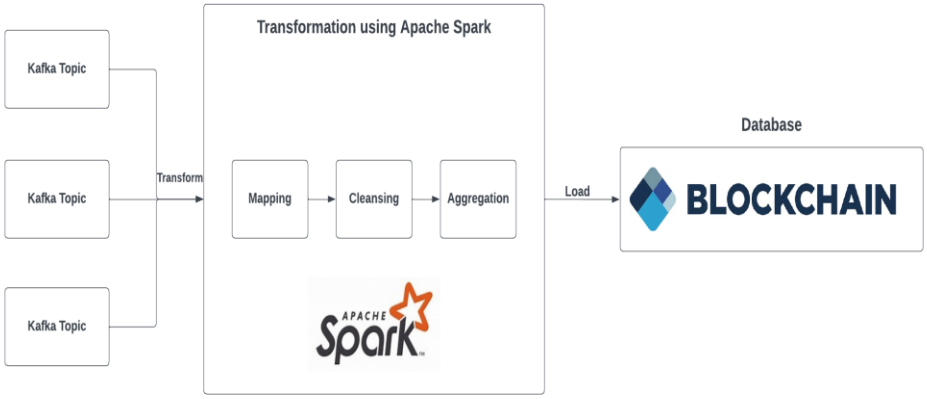


Fig. 4: Data Transforming using Apache Spark

The depicted diagram illustrates the workflow of the ETL (Extract, Transform, Load) process. Initially, the ATM Transaction Data is extracted and forwarded to Kafka topics. Following extraction, the data is transformed using Apache Spark. This transformation phase encompasses Mapping, Cleansing, and Aggregating the data frame based explicitly on the month attribute. After the transformation step, the next stage is loading, during which the transformed data is loaded into a database, depending on the ETL configuration in use.

4.3 Data Storage and Security

Implementing blockchain for ATM transaction data storage can provide security, transparency, and immutability to the transaction records. Multiple blockchain platforms can be used depending on the requirement. The most popular platforms include Ethereum, Hyperledger Fabric, and Corda. The choice of platform depends on the scalability, consensus mechanism, smart contract, and community support.

Using a platform like Ethereum, you can create smart contracts to define the rules and logic for processing transactions. Smart contracts are self-executing contracts with predefined conditions, enabling automated and trustworthy processing. Define the necessary functions and events to handle the ATM transaction data. Define the data structure through which information should be stored on the blockchain.

```
struct Transaction {
    uint256 dateAndTime;
    string transactionType;
    string atmLocation;
    CardInformation cardInfo;
    uint256 transactionAmount;
    AccountInformation accountInfo;
    string transactionResult;
    string atmTerminalID;
    string currency;
    string additionalMessages;
```

Write a smart contract that defines the functions and events to handle ATM transactions. Smart contracts are crucial in handling ATM transaction data in a blockchain-based system. Smart contracts are the system's backbone, ensuring ATM transaction data's accuracy, transparency, security, and trustworthiness in a blockchain-based environment. They provide a decentralized and automated approach to processing and storing transaction data, eliminating the need for intermediaries and enhancing the efficiency and reliability of the overall ATM transaction process.

Input: Kafka topic (string)

Processing:

```
contract KafkaDataLoader {
mapping(string => bool) private processedRecords;
event RecordLoaded(string recordId);
function loadFromKafka(string memory kafkaTopic) public {
KafkaConsumer consumer = createConsumer();
consumer.subscribe(kafkaTopic);
while (true) {
Message message = consumer.consume();
string memory recordId = extractRecordId(message);
string memory recordData = extractRecordData(message);
if (!processedRecords[recordId]) {
bool isValid = validateRecordData(recordData);
if (isValid) {
```

```
        storeOnBlockchain(recordId,
          recordData);

        processedRecords[recordId] =
              true;

        emit RecordLoaded(recordId);

    }   }   }   }   }
```

Output: Records loaded on the Ethereum blockchain, with the event RecordLoaded containing the recordId (string).

This represents a smart contract called "KafkaDataLoader" that facilitates loading data from a Kafka topic into the blockchain. The contract defines a mapping to keep track of processed records and emits an event when a record is successfully loaded. The "loadFromKafka" function creates a Kafka consumer, subscribes to the specified topic, and continuously consumes messages. It extracts the record ID and data from each message, validates the data, and stores it on the blockchain if it is deemed valid. The contract ensures that each record is processed only once by checking the mapping. Overall, this smart contract enables the integration of Kafka data with the blockchain, providing transparency, immutability, and audibility to the loaded data.

In the next step, we will submit transactions to the blockchain. Ethereum SDK, such as Web3.js, connects with the deployed contract. The "storeOnBlockchain" function is responsible for submitting the transactions to the Ethereum blockchain. Monitor the Ethereum network to ensure the transactions are successfully added to the blockchain. Listen for the NewTransaction event emitted by the smart contract to track new transactions.

4.4 Overall Architecture of Pipeline

Using Kafka to collect ATM transaction data enables high-throughput data streams, fault tolerance, and real-time processing. ATM data is collected by a middleware layer with Kafka producers, forwarded to dedicated Kafka topics, and consumed by applications for processing and analytics. ETL process applies filtering, mapping, aggregation, and data cleansing. Use cases involve mapping to a schema, data cleansing for missing values and duplicates, and aggregation.

Based on transaction months. Blockchain implementation for ATM transaction data offers security, transparency, and immutability. Smart contracts play a vital role, defining functions and events to process and store transaction data, ensuring accuracy, trustworthiness, and decentralization.

The following diagram describes the architecture of the pipeline proposed in this paper:

Fig. 5: Overall Architecture of Pipeline

5. Results and Discussion

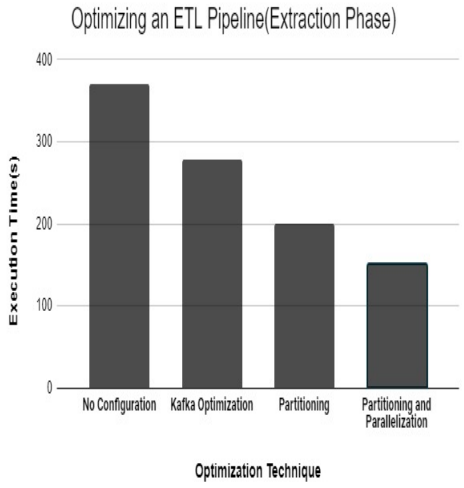
5.1 Analysis of ETL Pipeline Performance

The following section presents the performance evaluation of the ETL Pipeline. The experiment aimed to optimize the performance of the ETL Pipeline while considering the security factor.

Firstly, the goal was to improve the performance of the extraction phase. This phase includes sending the 24M transaction records data to Kafka topics. With the default configuration of Kafka, topics with one partition, and No Optimization method, the execution time is almost 6 minutes. Now, the goal is to optimize the methods involved so that the execution time can be reduced.

The first optimization method undertaken was to change the default configuration for Kafka. Here are the changes made to the Kafka configuration:

- Acks was set to 1, meaning the producer will only wait for the acknowledgment from the leader broker.
- Batch size was set to 32kb so that more data to a broker can be sent at a time.



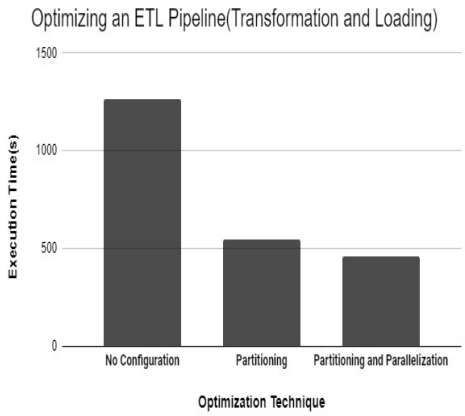
- Linger. ms, the delay before sending a new data packet was set to 5ms so that producers have enough time to store data in the batch before sending.
- Topics are created with three partitions rather than one by default.

After all these configuration changes, execution time was recorded to be almost 4.6 minutes. Then, the most critical optimization step suggested in [2] is the process of Partitioning and Parallelization. To achieve the following, the data source was partitioned into multiple sources, and multiple producers sent the partitioned data to Kafka topics. But only Partitioning wasn't enough for a good result as the execution time was decreased to almost 3.5 minutes. The last method to optimize this phase was Parallelization, which is achieved using multi- processing. Processes that send data to Kafka-topics were run in parallel, and the whole data was sent to Kafka-topics in around 2.5 minutes, which appears suitable for sending 24 million rows of data.

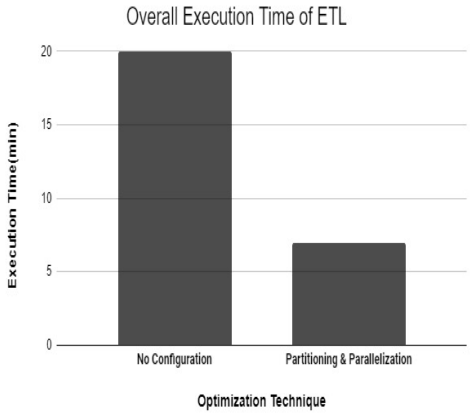
The next phase is the transformation and loading phase. This phase includes reading from Kafka topics, mapping, cleaning, aggregating, and loading the transformed data into a database. The non-optimized transformation phase took around 20 minutes, which is unacceptable.

For better performance, the main data frame containing the whole data was partitioned into multiple sub-data frames. Now, transformation methods like filtering based on month cleaning were applied on sub-data frames instead of the main data frame, resulting in a sufficient decrease in execution time. Each sub-data frame was executed in parallel to optimize the complete processes.

This results in a final execution time of almost 7 minutes.



Now, to compare how much optimization we have achieved, the overall execution time of the whole ETL Process of non-optimized and optimized ETL Pipeline is shown in the figure:



5.2 Data Security Evaluation Using Blockchain

Data security is critical when utilizing blockchain technology to store ATM transaction data. Here's an evaluation of the security measures mentioned in the provided information:

1. **Access Control:** Implementing roles and permissions helps ensure that only authorized individuals can perform specific actions. By defining roles for ATM operators, bank administrators, and auditors, access to sensitive operations can be restricted. Using modifiers and require statements in the smart contract enforces these permissions effectively.

2. **Encryption:** Encrypting sensitive data fields before storing them on the blockchain adds an extra layer of protection. Utilizing encryption algorithms like AES or RSA helps ensure the confidentiality of card information, account details, and transaction amounts. Encrypting the data before storing it on the blockchain makes it more difficult for unauthorized parties to access sensitive information.

3. **Event Logging:** Emitting events for critical operations and storing event logs off-chain in a secure and centralized logging system aids in auditing and analysis. By keeping track of successful transactions, account balance updates, and access control changes, it becomes easier to monitor and review activities for any potential security breaches.

4. **Secure Data Handling:** Avoiding the storage of sensitive information such as card PINs or CVV numbers on the blockchain is crucial. These details should be handled securely outside the blockchain, with the blockchain storing only the necessary transaction metadata. Adhering to secure coding practices helps mitigate common vulnerabilities and ensures the integrity of the stored data.

5. **Code Auditing and Testing:** Conducting thorough code reviews and security audits is essential to identify and address potential vulnerabilities. By testing the smart contract extensively,

including simulated attack scenarios, weaknesses can be discovered and rectified. Regular audits and testing help maintain a robust and secure smart contract for handling ATM transaction data.

6. **Contract Upgradability:** Implementing a mechanism for contract upgradability is essential for addressing security patches or adding enhancements. However, caution must be exercised to ensure that contract upgrades do not compromise the integrity of stored transaction data or introduce new security risks. Best practices should be followed to maintain the security of the data during the upgrade process.

7. **External Dependency Security:** Using verified and audited libraries for cryptographic operations and other external dependencies is crucial. Relying on trusted code reduces the risk of introducing vulnerabilities or compromising data security. Care should be taken to thoroughly evaluate and vet any external dependencies in the blockchain solution.

8. **Regular Updates and Patching:** Staying informed about security updates and patches for the blockchain platform and smart contract frameworks is necessary. Promptly applying these updates ensures that known vulnerabilities are addressed, and the latest security measures are in place. Regular updates and patching are vital for maintaining a secure environment for ATM transaction data.

9. **Third-Party Audits:** Engaging independent security auditors helps ensure a comprehensive evaluation of the smart contract handling ATM transaction data. Third-party auditors can identify potential security weaknesses, validate the effectiveness of implemented security measures, and provide recommendations for improvement. These audits offer an objective perspective on the security of the blockchain solution.

By implementing these security measures and regularly evaluating the data security aspects, ATM transaction data stored on the blockchain can benefit from increased integrity and confidentiality.

5.3 Data Security Evaluation Using Blockchain

The following table highlights the differences between blockchain and traditional databases:

	Block Chain	**Traditional Databases**
Immutable and Tamper-Resistant	Blockchain databases store data in linked blocks with cryptographic hashes, ensuring highly secure and tamper-resistant storage.	Traditional databases may rely on centralized servers or administrators, which can be vulnerable to unauthorized modifications.
Decentralization	Blockchain databases' decentralized nature, distributing copies across a node network, mitigates single points of failure and centralized attack targets.	On the other hand, traditional databases often have a central server, which can be a potential weakness.

Consensus Mechanisms	Consensus mechanisms like proof-of-work or proof-of-stake in blockchain databases validate and require agreement among most network participants before adding transactions to the chain.	In traditional databases, transactions are often validated and controlled by a central authority, which may be more susceptible to corruption or hacking attempts.
Encryption and Security Measures	Due to their distributed ledger nature, blockchain databases employ robust encryption for securing transactions and identities.	Traditional databases can implement encryption techniques to protect data.
Privacy and Confidentiality	While blockchain offers transparency and immutability, it introduces privacy concerns as certain data might be visible to all participants based on the design.	Traditional databases, on the other hand, can implement access controls and encryption to restrict access to sensitive information.
Performance and Scalability	Blockchain databases may need help with scalability and performance due to slower transactions and limited scalability tied to consensus mechanisms and distributed structure.	Traditional databases, especially those designed for high-performance environments, can often handle larger volumes of data and provide faster response times.

6. Conclusion

ETL processes are known to be complex and resource-consuming. With the advancement of financial technology, the security and integrity of customer data have emerged as a big concern. So, there is a definite need to focus on the security aspects of a pipeline. However, the enhancement of pipeline security comes with a compromise on pipeline performance. To achieve a better level of security, we need to increase pipeline performance to maintain an optimum balance between security and performance.

Blockchain is known for its secure nature. In this paper, we propose that the ATM transaction data should be stored in a blockchain in the load phase of the ETL pipeline. This ensures the security and integrity of crucial financial data as Blockchain provides immutability, transparency, traceability, decentralization, reliability, privacy, confidentiality, and fault tolerance. Blockchain offers encryption/decryption, cryptography, digital signature and timestamp, and hash trees to ensure security. Compromising on security is synonymous with compromising on customer trust. However, blockchain has the disadvantage of low throughput, making the pipeline slow. So, we tried to optimize the pipeline performance using parallelization and partitioning in the extraction phase. Before optimization, for 24 million transactions, it took 6 min to send data to Kafka topics. After optimization, it took 2.5 minutes. In this way, we enhanced the pipeline security without affecting performance.

REFERENCES

[1] A. Raj, J. Bosch, H. H. Olsson, and T. J. Wang, “Modelling Data Pipelines,” 2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Portoroz, Slovenia, 2020, pp. 13-20, doi: 10.1109/SEAA51224.2020.00014.

[2] H. Sun, S. Hu, S. McIntosh, and Y. Cao, "Big data trip classification on the New York City taxi and Uber sensor network," Journal of internet Technology, vol. 19, no. 2, pp. 591–598, 2018.

[3] Mehmood, E., Anees, T. Distributed real-time ETL architecture for unstructured big data. Knowl Inf Syst 64, 3419–3445 (2022). https://doi.org/10.1007/s10115-022-01757-7

[4] A. Farki and E. A. Noughabi, "Real-Time Blood Pressure Prediction Using Apache Spark and Kafka Machine Learning," 2023 9th International Conference on Web Research (ICWR), Tehran, Iran, Islamic Republic of, 2023, pp. 161-166, doi: 10.1109/ICWR57742.2023.10138962.

[5] M. N. M. Bhutta et al., "A Survey on Blockchain Technology: Evolution, Architecture and Security," in IEEE Access, vol. 9, pp. 61048-61073, 2021, doi: 10.1109/ACCESS.2021.3072849.

[6] Moura, Teogenes & Gomes, Alexandre. (2017). Blockchain Voting and its Effects on Election Transparency and Voter Confidence. 574-575. 10.1145/3085228.3085263.

[7] T. Ali Syed, A. Alzahrani, S. Jan, M. S. Siddiqui, A. Nadeem and T. Alghamdi, "A Comparative Analysis of Blockchain Architecture and its Applications: Problems and Recommendations," in IEEE Access, vol. 7, pp. 176838-176869, 2019, doi: 10.1109/ACCESS.2019.2957660.

[8] A. A. Monrat, O. Schelén, and K. Andersson, "A Survey of Blockchain From the Perspectives of Applications, Challenges, and Opportunities," in IEEE Access, vol. 7, pp. 117134-117151, 2019, doi: 10.1109/ACCESS.2019.2936094.

[9] Hader, Manal & Tchoffa, David & El Mhamedi, Abderrahman & Ghodous, P. & Dolgui, Alexandre & Abouabdellah, Abdellah. (2022). Applying integrated Blockchain and big data technologies to improve supply chain traceability and information sharing in the textile sector. Journal of Industrial Information Integration. 28. 100345. 10.1016/j.jii.2022.100345.

[10] Jayanthi Ranjan, "Business Intelligence: Concepts, Components, Techniques and Benefits," Journal of Theoretical and Applied Information Technology, vol. 9, no. 1, pp. 60-70, 2009.

[11] Dhamotharan Seenivasan (2023). We are improving the Performance of the ETL Jobs. 71(3), pp.27–33. Doi https://doi.org/10.14445/22312803/ijctt-v71i3p105.

[12] Ahmed, N., Barczak, A.L.C., Susnjak, T. et al. A comprehensive performance analysis of Apache Hadoop and Apache Spark for large scale data sets using HiBench. J Big Data 7, 110 (2020). https://doi.org/10.1186/s40537-020-00388-5

[13] Wang, G., Chen, L., Dikshit, A., Gustafson, J., Chen, B., Sax, M.J., Roesler, J., Blee-Goldman, S., Cadonna, B., Mehta, A., Madan, V. and Rao, J. (2021). Consistency and Completeness. Proceedings of the 2021 International Conference on Management of Data. doi:https://doi.org/10.1145/3448016.3457556.

[14] Guo, H. and Yu, X. (2022). A Survey on Blockchain Technology and its security. Blockchain: Research and Applications, 3(2), p.100067. doi https://doi.org/10.1016/j.bcra.2022.100067.

[15] Mikalef, P., Boura, M., Lekakos, G. and Krogstie, J. (2019). Big data analytics and firm performance: Findings from a mixed-method approach. Journal of Business Research, [online] 98(2), pp.261–276. Doi https://doi.org/10.1016/j.jbusres.2019.01.044.

[16] Wang, G., Chen, L., Dikshit, A., Gustafson, J., Chen, B., Sax, M.J., Roesler, J., Blee-Goldman, S., Cadonna, B., Mehta, A., Madan, V. and Rao, J. (2021). Consistency and Completeness. Proceedings of the 2021 International Conference on Management of Data. doi:https://doi.org/10.1145/3448016.3457556.

[17] Haggag, M., Tantawy, M.M. and El-Soudani, M.M.S. (2020). Implementing a Deep Learning Model for Intrusion Detection on Apache Spark Platform. IEEE Access, 8, pp.163660–163672. doi:https://doi.org/10.1109/access.2020.3019931.

[18] Berdik, D., Otoum, S., Schmidt, N., Porter, D. and Jararweh, Y. (2021). A Survey on Blockchain for Information Systems Management and Security. Information Processing & Management, 58(1), p.102397. doi:https://doi.org/10.1016/j.ipm.2020.102397.

Vol. 7, No. 2 | July – December 2023

Dynamic Time Quantum Computation for Improved Round Robin Scheduling Algorithm Using Quartiles and Randomization (IRRQR)

Asif Yar[1], Bushra Jamil[1*], Humaira Ijaz[1]

Abstract:

Scheduling is a decision-making process through which large numbers of tasks compete for various system resources. The availability of limited resources makes scheduling a challenge. Among resources, the processoris the most important resource for in-time completion of tasks therefore; developing an efficient processor scheduler is still a topic of interest. We have applied a statistical approach that combines concept of median quartiles, upper quartiles and randomness to adjust the time quantum for each process with the objective of optimizing the allocation of CPU time to a set of processes. We have considered average waiting time, average turnaround time, and the number of context switches as performance metrics and compared our algorithm (IRRQR) with nine other dynamic time quantum adjustment algorithms. The comparisons with these algorithms reveal the effectiveness of IRRQR in terms of minimizing number of contexts switches up to 25%, average waiting time up to 13.7%, and average turnaround time by 8.4%.

Keywords: *Scheduling Algorithm; Round Robin; Quartiles; Time Quantum; Randomization; Average Waiting Time; Average Turnaround Time*

1. Introduction

Modern operating systems were designed with the idea of multi-programming to deal with the problem of underutilization of system resources that enabled the interleaved execution of multiple jobs on a single processor machine [1]. These multi-programmed operating systems need a plan for the specific ordering of these jobs in the ready queue according to some defined criteria to select amongst them for CPU allocation to maximize resource utilization. This process is called CPU scheduling in which a scheduler deals with ordering and selection of jobs from the ready queue which mainly concerns throughput, latency, and response time. After that, the dispatcher transfers the control of the CPU to the selected job. As the nature of multi-programming-based CPU scheduling like First Come First Serve (FCFS), and Shortest Job First (SJF), priority scheduling is non-preemptive and therefore is not well suited for I/O bound jobs and online users.

The idea of the time-sharing concept was introduced in multi-programming-based operating systems to overcome issues of non-preemption. In time-sharing systems, the CPU time is shared in short time intervals/slots among multiple jobs simultaneously by frequent context switching so that all jobs run seamlessly without any problem. This time interval is called a time quantum (TQ), time slice, or time slot. If the job completes its CPU burst before its time slice expires the job preempts out of CPU like the FCFS algorithm. However, if the job completely consumes its

[1] Department of CS & IT, University of Sargodha, Sargodha, Pakistan
Corresponding Authors: bushra.jamil@uos.edu.pk

time quantum, it preempts and moves to the tail of the ready queue. As the time quantum is usually small, switching among jobs occurs so frequently that the users are unaware of the fact that jobs are sharing system resources and can also interact with running jobs. Some examples of preemptive-based job scheduling algorithms are Shortest Remaining Time First (SRTF), Priority Scheduling, and Round Robin (RR).

Round Robin (RR) is the most widely used preemptive scheduling algorithm for a time-sharing environment [2]. In RR a fixed small static time quantum (TQ) is defined for which the dispatcher transfers CPU control to each job in the ready queue in FCFS order [3]. The selection of TQ greatly affects the performance of the RR scheduling algorithm. The selection of small-time quantum results in an increased number of Context Switches that increase the overhead as the CPU does nothing except save and restore the context of jobs. Whereas, a large quantum leads to increased waiting time causing RR to behave as FCFS (First Come First Serve) [4]. Increasing the context switching increases the overhead as the CPU does nothing except save and restore the context of jobs.

Dynamic time quantum selection dynamically adjusts the time quantum for each process based on its behavior. It not only gives a fair share of CPU time to every process by prioritizing them but also improves system response time along with optimal utilization of resources. In the field of operating systems dynamic time quantum selection is an active area of research to improve the performance of system. Therefore, there is need to develop a dynamic time quantum selection techniques that optimizes CPU time allocation to different processes minimizing turnaround time along fare share to each process. In the field of operating systems dynamic time quantum selection is an active area of research to improve the performance of system. Some researchers have already proposed a modified version of Round Robin, in which dynamic time quantum is continuously computed for currently running jobs based on burst times of currently running jobs [5]. However, existing algorithms are still not efficient enough in reducing the number of context switches (CS), average waiting time (AWT), and average turnaround time (ATT). To achieve these objectives, we propose a dynamic time quantum computing algorithm, for an improved Round Robin Scheduling algorithm that calculates TQ dynamically based on Quartile and Randomization (IRRQR) of the remaining burst time of jobs in the ready queue. The median quartile is used to group processes into short CPU bursts and long CPU bursts for adjusting time quantum to shorter for processes with short CPU burst lengths and longer time quantum for long CPU bursts. Whereas the upper quartile is used as a threshold for long-running processes in which the longer time quantum is assigned to processes with CPU bursts above the upper quartile and shorter time quantum with CPU bursts below upper quartile. Afterwards, randomization is used to select any random value from the ready queue. The quartiles are used to group processes with similar CPU burst lengths and assign them similar time quantum values, while randomization can be useful in preventing long-running processes from monopolizing the CPU.

Therefore, we have used a combination of median quartile, upper quartile, and randomness in dynamic time quantum selection that provides a fair share of CPU time to every process to ensure efficient allocation of the CPU time to different processes, preventing long-running processes from monopolizing the CPU. The objectives of this algorithm are to minimize the total number of context switches, average turnaround time, and average waiting time.

The rest of this paper is organized in the following order. Related work is described in Section 2. The design and implementation of the proposed algorithm is presented in Section

3. Experimental results are discussed in Section 4. We conclude and describe the future work in section 5.

2. Related Work

In this section, we review some existing modified Round Robin Scheduling algorithms that have been proposed till now for enhancing the performance of the classical RR algorithm by computing dynamic time quantum. In [5], Gowda proposes a statistical approach for dynamic time quantum. They divide jobs into four categories based on burst time and time quantum. For each category min-max spread size is used. They compute dynamic time quantum by taking the square root of the multiple of a total number of jobs, mean, and standard deviation. In [6], Mishra et al. proposed dynamic time quantum computation by sorting jobs in ascending order of their burst times. Then, the time quantum is selected equal to the burst time of the first job. An amended Dynamic Round Robin (ADRR) [7] is proposed by Shafi et al. in which they arrange all jobs in increased order of their burst times and the time quantum is set to the lowest burst time. In each cycle, if the time quantum is less than 20, set the time quantum to 20. If the remaining burst time of a current job is less than half of the time quantum, let this job complete its execution. Therefore, a job with minimum burst waits for minimum time. After each cycle, the quantum is readjusted.

The technique suggested in [8] by Berhanu et al. is to initially sort the jobs in increasing order of their burst times and after that set the quantum equal to the burst of the first job and if the remaining burst time of a currently running job is less than time quantum, assign CPU to same job again till it completes its execution.

In [9], LaxmiJeevani et al. suggest the calculation of dynamic time quantum, firstly by fixing quantum to k unit of time statically and then assigning CPU for that time quantum to the first coming job from the queue. After that, the time quantum is set to the burst time of the job with the lowest burst time present in the ready queue. Tajwar et. al. in [10], propose to arrange jobs according to their burst time in ascending order and set a time quantum equal to the average of these jobs, then assign CPU to each job equal to that time quantum.

In [11], Kumar et al. propose an algorithm that sorts the jobs in ascending order and computes dynamic time quantum by calculating the harmonic mean of jobs that have arrived in the ready queue. According to [12], Ranjan et al. dynamic time quantum is calculated by summing up the burst time of all jobs in the ready queue and dividing that sum by the total number of jobs, which is called arithmetic mean. Emami in [13] proposes a harmonic-arithmetic mean (HARM) algorithm for dynamic time quantum. The quantum is set to the Harmonic mean of the burst time of all the jobs if some jobs have greatly larger burst times than other jobs and jobs are heterogeneous then set time quantum to the arithmetic mean of the burst time of jobs.

In [14], Mohanty et al. propose to compute dynamic time quantum by sorting jobs in ascending order, calculating the median of the burst time of a job that has arrived in the ready queue. Another technique is presented in [15] by Matarneh, in which jobs are sorted in ascending order to calculate the median, and the time quantum is set to the median. If the quantum is less than the threshold value of 25, it is set equal to the threshold value. In [16], Nayak et al. computes the dynamic time quantum by sorting jobs in ascending order and finding median and highest burst time. They fixed the time quantum to an average of the median and the highest burst time.

Varma et al. in [17], arrange jobs in increased order to their burst times and calculate time quantum by taking the square root of the sum of squares of burst time divided by the total number of jobs, also called root

mean square. In [18], Khokhar et al. propose a dynamic time quantum technique using an average of mean and median. Allocate CPU to each job and if the remaining burst time of a current job is less or equal to the burst time, allocate CPU again to it till it completes its execution. In [19], Zhang et al., suggest dynamic time quantum computation based on median theory. In another experimental study, Iqbal and fellows compared four variations of RR with conventional RR [20]. The study reveals that variations of RR perform better than conventional RR.

Najafi and Samira propose a method to calculate dynamic time quantum using machine learning [21]. They use several processes and their average, maximum, and minimum burst times are used as features for training the data set. After training ML classifier predicts the time quantum. Vayadande et al. propose a method to calculate dynamic time quantum for RR. In this method, the dynamic TQ for each cycle is calculated using a prescribed formula [22].

Sakshi and fellows present an algorithm to calculate dynamic time quantum using the median and average burst time of every process. The proposed algorithm is compared with four other state-of-the-art algorithms to reveal the superiority of the proposed algorithm [23]. In another study, the priority of the process and RR scheduling algorithm are combined to calculate the dynamic time quantum to take advantage of both algorithms [24].

Table 1 presents the summary of the existing dynamic time quantum computation techniques used in different studies.

TABLE I**.** *Comparison of*

Sr. #	**Tech.**	**Short Comings**	**Ref.**
1	Statistical Method		[5]
2	Sorting	Unfair CPU allocation due to prioritization.	[6,7,8 ,9,10]
3	Harmonic mean	It gives priority to smaller values causing more context switches	[11]
4	Arithmetic mean	It assumes that the distribution of time quantum is normal so not suitable for real-world cases.	[12]
5	Median	Not suitable for systems with highly variable workloads	[14,15,16, 19, 23]
6	Root mean Square	Calculating RMS is time-consuming	[17]
7	Mean and median	Needs more computations to result in long delays.	[18]
8	ML	May not be accurate for future workloads	[21]

The value of time quantum can greatly affect the performance of the RR algorithm, and existing algorithms are still not efficient enough in reducing the number of context switches, average waiting time, and average turnaround time. To achieve these purposes, we implement a variant of Round Robin Scheduling Algorithm using Quartiles and Randomization (IRRQR).

3. Design and Implementation

Static time quantum is a limitation of the RR algorithm that degrades its performance. Therefore, in this work, we propose a methodology for dynamic time quantum computation that would be computed in each cycle of the RR algorithm. Our dynamic time quantum results in:

1. Minimizing the number of context switches

2. Minimizing average waiting time
3. Minimizing average turnaround time

For dynamic time quantum computation, we are using the concepts of median quartile (MQ), upper quartile (UQ), and randomization. For MQ we compute the median of the total number of jobs, while for UQ median of the second half of jobs is computed. We compute these medians using equation 1.

$$Median = \begin{cases} E(\frac{N}{2}) & \text{if N is odd} \\ \frac{E(\frac{N}{2})+E(\frac{N}{2}-1)}{2} & \text{otherwise} \end{cases} \quad (1)$$

Where E(N) indicates the element at index x and N is the total number of jobs. After MQ and UQ are computed, the time quantum is selected randomly from both computed values in each cycle of the RR algorithm. Therefore, at least 50 percent of the jobs will complete their execution in the first round of the algorithm.

We present our proposed algorithm IRRQR as follows.

Algorithm 1 IRRQR Algorithm for Task Scheduling

INPUT: RQ ($P1, P2, \ldots PN$)
OUTPUT: AWT, ATT

```
do
    Sort RQ in ascending according to BT of jobs
    Compute M_Q
    Compute U_Q
    TQ = Rand(M_Q , U_Q)
    for ( i ← 1 to N ) do
        Assign TQ to Job P_i
        if BT_i ≤ TQ then
            remove P_i from RQ
        else
            BT_i = BT_i - TQ
            RQ ⟸ P_i
        end if
        Compute WT_i for all jobs in RQ
    end for
while RQ!=∅
calculate AWT,ATT
```

The symbols used in the IRRQR algorithm are shown in Table 2.

TABLE II. Algorithm symbols

Symbol	Meaning
RQ	Ready Queue of N Jobs
AWT	Average waiting time
ATT	Average turnaround time
MQ	Median of burst time of all jobs RQ
UQ	Quartile of jobs in the second half of RQ
TQ	Time quantum
WT	Waiting time

In IRRQR, initially, jobs in the ready queue are ordered in ascending order according to their burst times. Then, we compute MQ and UQ. TQ is calculated as a random number between MQ and UQ. This TQ is assigned to every job in the current iteration of the RR algorithm. If the job completes its execution and terminates, it is removed from the ready queue otherwise, it is put back at the tail of the ready queue. This whole job continues until the ready queue becomes empty.

3.1 Illustration

We illustrate the performance of our proposed algorithm using a simple scenario. Let us consider there are five jobs P0, P1, P2, P3, and P4 in a ready queue. The arrival time for each job is zero while the burst time is 48, 22, 70, 74, 10. According to our proposed methodology, jobs in the ready queue will be arranged in ascending based on their burst times and the new sequence will be P4, P1, P0, P2, P3. Random time quantum between the median quartile and upper quartile is calculated i.e. TQ = 70. CPU will be allocated to each job and in the first iteration of the algorithm time, the quantum value will be 70, so jobs P4, P1, P0, and P2 will complete their execution as their remaining burst time is zero and will be taken out of the ready queue. The remaining burst time of job P3 will be 4. In the second iteration of the algorithm TQ = 4 as there is a single job in the ready queue. So, CPU will be allocated to job P3 when job P3

terminates ready queue will be empty. The average turnaround is 99.2 units and the average waiting time is 54.4 units.

3.2 Complexity Analysis

The time complexity of the presented algorithm depends on the length of the ready queue that is N. As the scheduler sorts this queue, therefore, the time complexity of sorting is O(nlogn). Computation of UQ, MQ, and TQ takes O(1) time. TQ has to be assigned to every job so it will take O(n) time. As a result, the time complexity of the IRRQR algorithm is O(nlogn) time while the space complexity is O(n).

4. Performance Evaluation

In this section, we present the results of our proposed IRRQR algorithm for different scenarios against existing state-of-the-art algorithms. These algorithms are: Round Robin (RR) [4], Improved Round Robin Variant Quantum (IRRVQ) [6], Amended Dynamic Round Robin (ADRR) [7], Dynamic Time Quantum Round Robin (DTQRR) [8], Round Robin Based Effective Time Slice (RRBETC) [10], SubContrary Mean Dynamic Round Robin (SMDRR) [11], Shortest Remaining Burst Round Robin (SRBRR) [14], Self Adjustment Time Quantum Round Robin (SATQRR) [15], Improved Round Robin (IRR) [16], Improved Shortest Remaining Burst Round Robin(ISRBRR) [17] and Median Based Round Robin (MBRR) [18].

For comparison, we select three metrics which are the number of average waiting time, average turnaround time and no of context switches.

Average Waiting Time(AWT): is the average waiting time of all the jobs that can be calculated using equation 2.

$$\text{AWT} = \sum_{1=1}^{m} WT_i \qquad (2)$$

Where m is the number of jobs in the ready queue. The waiting time for each job can be computed using Equation 3.

$$WT_i = CT_i + BT_i + AT_i \quad \forall i \in T \quad (3)$$

Average Turnaround Time (ATT): the average turn time of all the jobs in the ready queue can be computed usingequation 4.

$$\text{ATT} = \sum_{1=1}^{n} TT_i \qquad (4)$$

Turnaround time can be computed using equation 5.

$$TT_i = CT_i - AT_i \quad \forall i \in T \quad (5)$$

Where CT_i is the completion time and AT_i is the arrival time of i^{th} job.

1) Assumptions

We assume that all the jobs in the ready queue have equal priority. All the jobs are independent of each other. The arrival time and burst time of all jobs are known in advance. All jobs submitted for execution are CPU-bound. Forall scenarios, we have taken five jobs. We have not considered the overhead of context switching also sorting timefor jobs is considered zero.

2) Simulation Environment

To evaluate the performance of the proposed IRRQR scheduling algorithm, we have performed simulations. These simulations are carried out in Scala 2.11.8 on the Ubuntu 16.04 operating system, on a 4-core system with 4GB RAM and a 2.67GHz processor. In the next section, numerical and simulation results are discussed.

3) Results

In this section, we present six cases and compare our proposed algorithm results with the traditional round robin algorithm as well as eight other existing algorithms proposed by different researchers. For each case, we take seven jobs with different burst times. We divide six cases into two categories.

1. Zero arrival time

2. Heterogeneous arrival times

In each category, there are three different cases given below.

1. **First case**: All jobs are heterogeneous means they have burst times of different lengths.
2. **Second case**: One of the jobs has less burst time.
3. **Third case**: One of the jobs has more burst time.

4.3.1Zero Arrival Time

Case 1: Burst Times in ascending order In this case, a ready queue with seven identical jobs is considered where jobs are in increasing order of their burst time. Table 3 shows the burst time of each job.

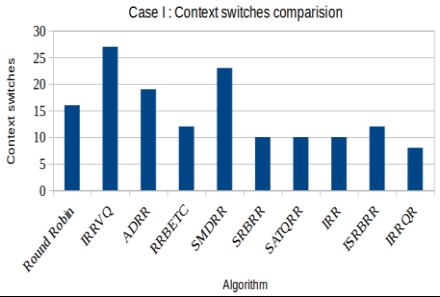
TABLE III. Burst Times in ascending order

Job	P1	P2	P3	P4	P5	P6	P7
Burst Time	5	23	34	41	66	78	80

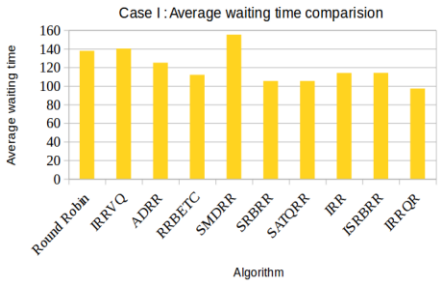
Table 4 presents the comparison results of algorithms.

TABLE IV. Comparison Results

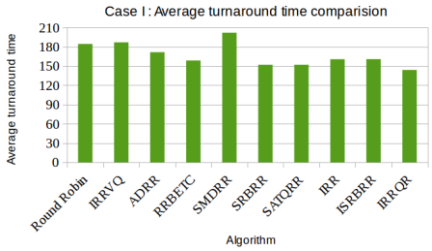
Algorithm	**CS**	**AWT**	**ATT**
Round Robin	16	137.7	184.4
IRRVQ	27	140.1	186.9
ADRR	19	124.9	171.6
RRBETC	12	111.9	158.6
SMDRR	23	155.1	201.9
SRBRR	10	105.3	152
SATQRR	10	105.3	152
IRR	10	113.9	160.6
ISRBRR	12	114	160.7
IRRQR	8	97.1	143.9



(a) Context Switches



(b) Average Waiting Time



(c) Average Turnaround Time

Fig I: Comparison Results of Burst Times in Ascending Order

Figure 1 shows the comparison of selected algorithms with the proposed algorithm. Algorithms are shown along the x-axis while the y-axis presents no of CS, AWT and ATT in figures 1a, 1b, and 1c respectively. The figure presents that our IRRQR algorithm gives the least no of CS, AWT, and ATT as compared to all existing algorithms while SRBRR and SATQRR give competitive results. The results show that the CS, AWT, and ATT reduce by 25%, 8.4%, and 5.6% respectively in comparison with other best approaches.

Case 2: Burst Times in descending order In this case, a ready queue with seven jobs has been considered where jobs have burst times in descending order as shown in Table 5.

TABLE V. Burst Times in descending order

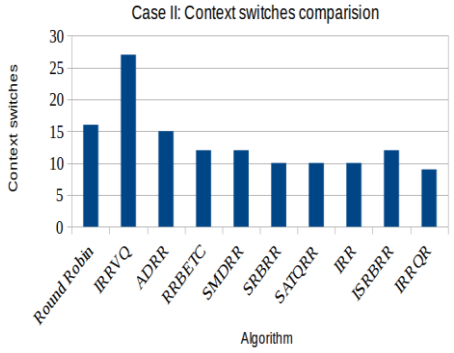
Job	P1	P2	P3	P4	P5	P6	P7
Burst Time	94	89	79	60	57	52	50

Table 6 presents the comparison results of algorithms.

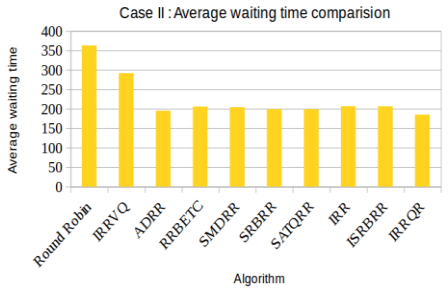
TABLE VI. Comparison Results

Algorithm	CS	AWT	ATT
Round Robin	16	362.9	431.6
IRRVQ	27	291.8	360.6
ADRR	15	195	263.7
RRBETC	12	205.7	274.4
SMDRR	12	204.2	273
SRBRR	10	199.2	268
SATQRR	10	199.2	268
IRR	10	206.6	275.3
ISRBRR	12	206.4	275.1
IRRQR	9	185	253.8

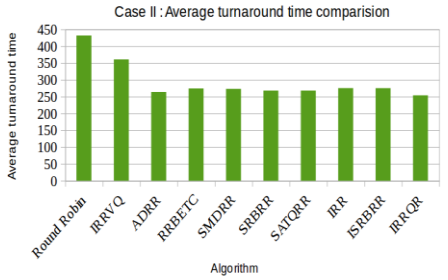
Figure 2 shows the comparison of selected algorithms with the IRRQR algorithm. Algorithms are shown along the x-axis, while the y-axis presents number of CS, AWT, and ATT in figures 2a, 2b, and 2c respectively. The figure presents that the presented IRRQR algorithm gives the least no of CS, AWT, and ATT as compared to all existing algorithms while ADRR, SRBRR, and SATQRR give competitive results. Our algorithm decreases CS, AWT, and ATT by 11.1%, 7.7%, and 5.6% respectively in comparison with SRBRR and SATQRR algorithms.



(a) Context Switches



(b) Average Waiting Time



(c) Average Turnaround Time

Fig II: Comparison Results of Burst Times in Descending Order

Case 3: Random Burst Times

In this case, seven jobs with random burst times are considered as shown in Table 7.

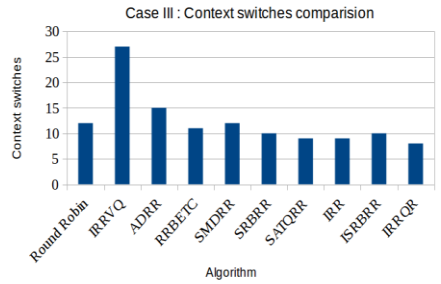
TABLE VII. Jobs in random order

Job	P1	P2	P3	P4	P5	P6	P7
Burst Time	39	20	43	26	31	42	28

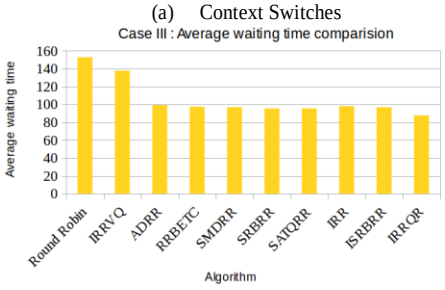
Table 8 shows the comparison results of the algorithms.

TABLE VIII. Comparison Results

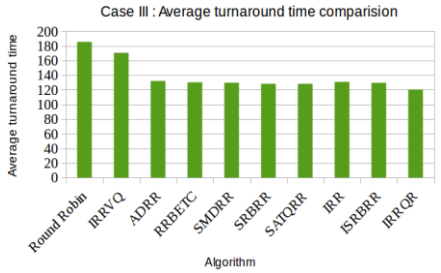
Algorithm	CS	AWT	ATT
Round Robin	12	152.7	185.4
IRRVQ	27	137.7	170.4
ADRR	15	99.2	132
RRBETC	11	97.4	130.1
SMDRR	12	96.9	129.6
SRBRR	10	95.4	128.1
SATQRR	9	95.4	128.1
IRR	9	98	130.7
ISRBRR	10	96.7	129.4
IRRQR	8	87.7	120.4



(a) Context Switches



(b) Average Waiting Time



(c) Average Turnaround Time

Fig III: Comparison Results of Random Burst Times

Figure 3 shows the comparison of selected algorithms with the proposed algorithm. Algorithms are shown alongthe x-axis while the y-axis presents no context switches, average waiting time, and average turnaround time in figures 3a, 3b, and 3c respectively. The figure presents that the IRRQR algorithm gives the least no of CS, AWT, and ATT as compared to all existing algorithms while SATQRR and SRBRR give competitive results. Our algorithm gives 12.5%, 8.7% and 6.3% reduced CS, AWT, and ATT as compared to SATQRR while 25%, 8.7%, and 6.3% least CS, AWT, and ATT in comparison with SRBRR.

4.3.2Non- Zero Arrival Time

Case 4: Burst Times in ascending order In this case, a ready queue with seven identical jobs is considered where jobs are in ascending orderof their burst time. Table 9 shows the arrival time and burst time of each job.

TABLE IX. Jobs in ascending order

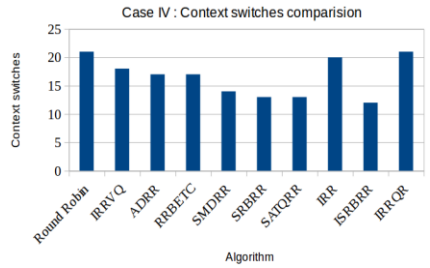
Job	P1	P2	P3	P4	P5	P6	P7
Arrival Time	4	0	7	11	19	13	23
Burst Time	7	11	38	53	61	72	74

Table 10 shows the comparison results of the IRRQR algorithm with other algorithms.

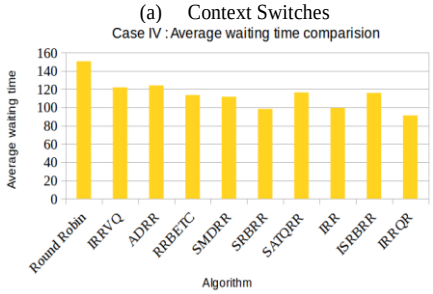
TABLE X. Comparison Results

Algorithm	**CS**	**AWT**	**ATT**
Round Robin	16	150.6	195.7
IRRVQ	21	122	167.1
ADRR	18	124.1	169.3
RRBETC	17	113.6	158.7
SMDRR	17	111.7	156.9
SRBRR	14	98.4	143.6
SATQRR	13	116.4	161.6
IRR	13	99.4	144.6
ISRBRR	20	115.9	161
IRRQR	12	91.2	136.4

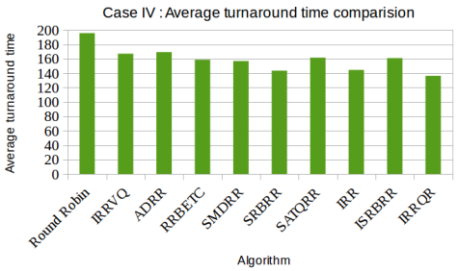
Figure 4 shows the comparison of IRRQR with other selected algorithms. Algorithms are shown along the x-axis and the y-axis presents number of CS, AWT, and ATT in figures 4a, 4b, and 4c respectively. The figure presents that the proposed IRRQR algorithm gives the least no of CS, AWT, and ATT as compared to all existing algorithms while SRBRR and IRR give competitive results. Our algorithm gives 8.3%, 8.9% and 6.01% reduced CS, AWT, and ATT as compared to the best-performing IRR algorithm.



(a) Context Switches



(b) Average Waiting Time



(c) Average Turnaround Time

Fig IV: Comparison Results with non-zero arrival time

Case 5: Burst Times in descending order

In this case, a ready queue with seven identical jobs P1, P2, P3, P4, P5, P6, and P7 has been considered where jobs are in descending order of their burst time. Table 10 shows the arrival time and burst timeof each job.

TABLE XI. Jobs in ascending order

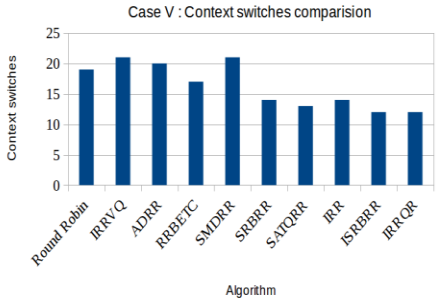
Job	P1	P2	P3	P4	P5	P6	P7
Arrival Time	11	2	7	4	16	0	9
Burst Time	103	96	85	72	46	19	7

Table 12 shows the comparison results of the algorithms

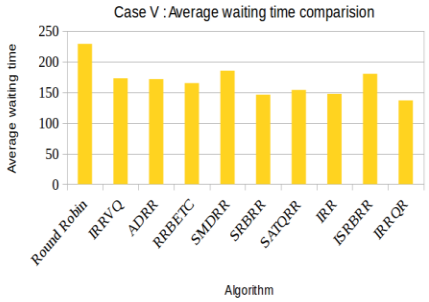
TABLE XII. Comparison Results

Algorithm	**CS**	**AWT**	**ATT**
Round Robin	19	228.9	290
IRRVQ	21	172.7	233.9
ADRR	20	171.4	232.6
RRBETC	17	164.9	226
SMDRR	21	185.1	246.3
SRBRR	14	146	207.1
SATQRR	13	153.7	214.6
IRR	14	147.3	208.4
ISRBRR	12	180.1	241.3
IRRQR	12	136.6	197.7

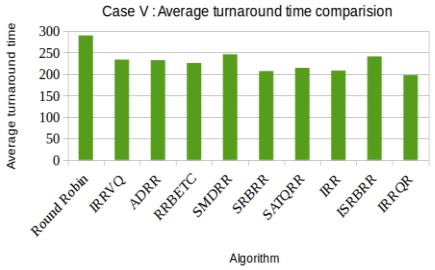
Figure 5 shows the comparison of selected algorithms with our IRRQR algorithm. Algorithms are shown alongthe x-axis while the y-axis presents no of context switches, average waiting time, and average turnaround time in figures 5a, 5b, and 5c respectively.



(a) Context Switches



(b) Average Waiting Time



(c) Average Turnaround Time

Fig V: Comparison Results with descending burst times for non-zero arrival time

The figure presents that the presented IRRQR algorithm gives the least no of CS, AWT, and ATT as compared to all existing algorithms while SRBRR and IRR give competitive results. IRRQR decreased CS, AWT, and ATT by 16.7%, 7.8%, and 5.4% respectively as compared to IRR.

Case 5: Random Burst Times In this case, a Ready queue with seven identical jobs P1, P2, P3, P4, P5, P6, and P7 has been considered where jobs are in random order of their burst time. Table 13 shows the arrival time and burst time of each job.

TABLE XIII. Jobs in ascending order

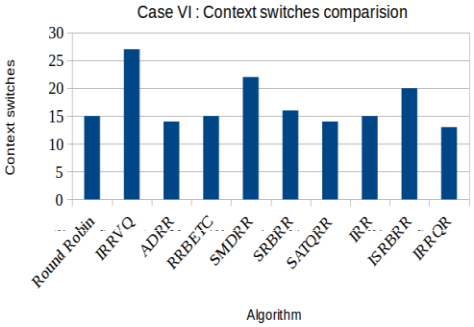
Job	P1	P2	P3	P4	P5	P6	P7
Arrival Time	2	3	0	5	17	9	12
Burst Time	31	26	3	11	85	40	76

Table 14 shows the comparison results of the algorithms

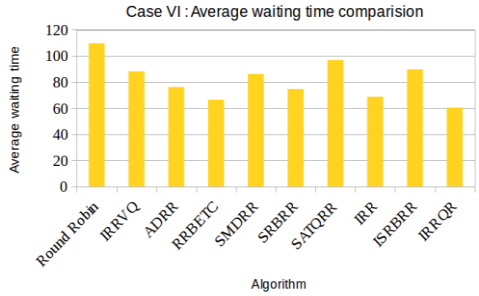
TABLE XIV. Comparison Results

Algorithm	**CS**	**AWT**	**ATT**
Round Robin	15	109.6	148.4
IRRVQ	27	88.1	127
ADRR	14	76.1	115
RRBETC	15	66.4	105.3
SMDRR	22	86.1	125
SRBRR	16	74.6	113.4
SATQRR	14	96.8	135.7
IRR	15	68.7	107.6
ISRBRR	20	89.7	128.6
IRRQR	13	60.4	99.2

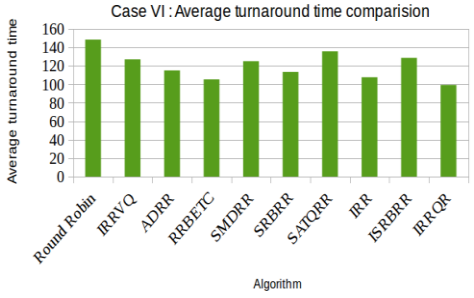
Figure 6 shows the comparison of selected algorithms with the proposed algorithm. Algorithms are shown along the x-axis while the y-axis presents no of context switches, average waiting time, and average turnaround time in figures 6a, 6b, and 6c respectively. The figure presents that our IRRQR algorithm gives the least no of CS, AWT, and ATT as compared to all existing algorithms while RRBETC and IRR give competitive results. Our IRRQR gives 15.3%, 9.9%, and 6.1% least CS, AWT, and ATT as compared to RRBETC.



(a) Context Switches



(b) Average Waiting Time



(c) Average Turnaround Time

Fig V: Comparison Results with random burst times for non-zero arrival time

5. Conclusion and Future Work

RR algorithms are widely used in real-time operating systems but their performance badly suffers from the wrong selection of time quantum. An optimal time quantum may decrease turnaround time, waiting time, and the numberof context switches. In this paper, we propose an improved quartile and randomization based dynamic round-robin scheduling algorithm (IRRQR) for optimal time quantum computation. IRRQR combines the concept of median quartile, upper quartile, and randomness in dynamic time quantum selection by providing a fair share of CPU time to all processes ensuring efficient allocation of the CPU time to different processes, preventing long-running processes from monopolizing the CPU. The quantum time is randomly selected based on the median of the burst time of all jobs and the upper quartile. Then, each job gets its time quantum. As many of the jobs complete their burst, therefore hence results in a decreasing number of context switches. The dynamic quantum is computed after one pass resulting in low complexity. Simulation results showed that our IRRQR algorithm results in reducing number of context switches up to 25%, average waiting time up to 13.7%, and average turnaround time by 8.4% as compared to existing algorithms.

In the future, we intend to use a meta-heuristic BAT algorithm with our proposed technique for large-scale scheduling problems in the cloud for optimal resource scheduling.

AUTHOR CONTRIBUTION

Asif Yar executed the research, whereas Bushra Jamil and Humaira Ijaz conceived the idea and supervised the work.

DATA AVAILABILITY STATEMENT

Not applicable.

CONFLICT OF INTEREST

The Authors declare that there is no conflict of interest.

FUNDING

(No funding available)

REFERENCES

[1] K. E. Arzen, A. Cervin, J. Eker, L. Sha, "An introduction to control and scheduling co-design", In *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No. 00CH37187)*, Vol. 5, Sydney, NSW, Australia, 2000, pp. 4865-4870.

[2] A. Noon A. Kalakech, S. Kadry, "A new round robin based scheduling algorithm for operating systems: dynamic quantum using the mean average, *International Journal of Computer Science Issues*, Vol. 8, No. 1,2011, pp. 224-229.

[3] S. K. Panda, D. Dash, J. K. Rout, "A Grouped Based Time Quantum Round Robin Algorithm Using Min-Max Spread Measure", *International Journal of Computer Applications*, Vol. 64, No. 10, 2013, pp. 1-7.

[4] A. Silberschatz, P. B. Galvin, *Operating System Concepts*, 9th Edition, John Wiley & Sons, Inc., 2012.

[5] S.N. Gowda, "Statistical Approach to Determine Most Efficient Value for Time Quantum in Round Robin Scheduling." *AIRCC's International Journal of Computer Science and Information Technology* 8, 2016, pp. 33-39.

[6] M. K. Mishra, D. F. Rashid, "An Improved Round Robin CPU Scheduling Algorithm With Varying Time Quantum", *International Journal of Computer Science, Engineering and Applications*, Vol. 4, No. 4, 2014, pp. 1-8.

[7] U. Shafi, M. Shah, A. Wahid, K. Abbasi, Q. Javaid, M. Asghar, "A Novel Amended Dynamic Round Robin Scheduling Algorithm for Time shared Systems", *International Arab Journal of Information Technology*, Vol.17, No. 1, 2020, pp. 90-98.

[8] Y. Berhanu, A. Alemu, M. K. Mishra, "Dynamic Time Quantum based Round Robin CPU Scheduling Algorithm", *International Journal of Computer Applications*, Vol. 167, No. 13, 2017, pp. 48-55.

[9] M. LaxmiJeevani, T. S. P. Madhuri, Y. S. Devi, "Improvised Round Robin Scheduling Algorithm and Comparison with Existing Round Robin CPU Scheduling Algorithm", *IOSR Journal of Computer Engineering*, Vol. 20, Issue. 3, Version I, 2018, pp. 01-04.

[10] M. M. Tajwar, M. N. Pathan, L. Hussaini, "CPU

Scheduling with a Round Robin Algorithm Based on an Effective Time Slice", *Journal of Information Processing System*, Vol. 13, No. 4, 2017, pp. 941-950.

[11] S. K. Bhoi, S. K. Panda, D. Tarai, "Enhancing CPU performance using Subcontrary Mean Dynamic Round Robin (SMDRR) Scheduling Algorithm", *Journal of Global Research in Computer Science*, Vol. 2, No. 12,2011, pp. 17-21.

[12] A. R. Dash, S. KK. Sahu, S. K. Samantra, "An Optimized Round Robin CPU Scheduling Algorithm with Dynamic time Quantum", *International Journal of Computer Science, Engineering and Information Technology*, Vol. 5, No. 1, 2015, pp. 7-26.

[13] A. E. A. Agha, S. J. Jassbi, "A New Method to Improve Round Robin Scheduling Algorithm with QuantumTime Based on Harmonic-Arithmetic Mean (HARM)", *International Journal of Information Technology and Computer Science*, Vol. 5, No. 7, 2013, pp. 56-62.

[14] P. R. Mohanty, P. H. S. Behera, K. Patwari, M. R. Das, M. Dash, Sudhashree, "Design and Performance Evaluation of a New Proposed Shortest Remaining Burst Round Robin (SRBRR) Scheduling Algorithm", *In Proceedings of International Symposium on Computer Engineering & Technology (ISCET)*, Vol. 17, 2010, pp. 126-137.

[15] R.J. Matarneh, "Self-Adjustment Time Quantum in Round Robin algorithm Depending on Burst Time of Now Running Processes", *American Journal of Applied Sciences*, Vol. 6, No. 10, 2009, pp. 1831-1837.

[16] D. Nayak, S. K. Malla, D. Debadarshini, "Improved Round Robin Scheduling using Dynamic Time Quantum", *International Journal of Computer Applications*, Vol. 38, No. 5, 2012, pp. 34-38.

[17] P. S. Varma, "Improved Shortest Remaining Burst Round Robin (ISRBRR) Using RMS as its time quantum", *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, Vol. 1, No.8, 2012, pp. 60-64.

[18] D. Khokhar, M. A. Kaushik, "Median Based Round Robin CPU Scheduling Algorithm", *International Journal of Computer Science and Information Technology Research*, Vol. 5, No. 2, 2017, pp. 198-203.

[19] C. Zhang, P. Luo, Y. Zhao, J. Ren, "An Efficient Round Robin Task Scheduling Algorithm Based on a Dynamic Quantum Time", *International Journal of Circuits, Systems and Signal Processing*, Vol. 13, 2019, pp. 197-204.

[20] S. Z. Iqbal, H. Gull, S. Saeed, M. Saqib, M. Alqahtani, Y.A. Bamarouf, G. Krishna, and M.I. Aldossary, "Relative Time Quantum-based Enhancements in Round Robin Scheduling", *Computer Systems Science & Engineering*, Vol. 41 No. 2, 2022, pp. 461-477.

[21] S. Najafi, S. Noferesti, "Determining dynamic time quantum in round-robin scheduling algorithm using machine learning*", Soft Computing Journal*, Vol. 10 No. 2, 2022, pp.32-43.

[22] K. Vayadande, A. Bodhankar, A. Mahajan, D. Prasad, R. Dhakalkar, S. Mahajan. "An Improved Way to Implement Round Robin Scheduling Algorithm" *International Conference on Computer Vision, High-Performance Computing, Smart Devices, and Networks*, 2022, pp. 403-414.

[23] C. Sharma, S. Sharma, S. Kautish, S. A. Alsallami, E.M. Khalil, A.W. Mohamed. "A new median-average round Robin scheduling algorithm: An optimal approach for reducing turnaround and waiting time" *Alexandria Engineering Journal*, vol. 61, no. 12. pp. 10527-10538.

[24] N. M. Najm, "New hybrid priority scheduling algorithm based on a round Robin with dynamic time quantum." *In AIP Conference Proceedings*, vol. 2787, no. 1. AIP Publishing, 2023.

Vol. 7, No. 2 | July December 2023

PREDICTIVE ANALYSIS OF CLIMATE DISASTER DATA

Anum Aziz[1], Shaukat Wasi[2], Muhammad Khaliq-ur-Rahman Raazi Syed[3]

Abstract:

In this paper, the Total deaths and Cost per Index (CPI) of worldwide climate disaster dataset has been modelled. The time period of the dataset is from 1900 to 2021. The Autoregressive Integrated Moving Average (ARIMA) has been applied to forecast the Total Deaths and CPI of the study area. The total of 75% of the train data is used for construction of the model and the remaining 25% dataset is used for testing the model. The ARIMA model is general provides more accurate projection especially interval forecast and is more reliable than other common statistical techniques. The best-fitted model is identified as ARIMA*(2,0,1) and (2,1,2) for Cost per Index CPI and Total Deaths* respectively, generated on the basis of minimum values of Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) procedures. The accuracy parameter considered as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) both parameters shows the model is accurate respectively. There is a 7% difference between the auto and manual models for the CPI feature, similarly, there is a 4% difference for Total Deaths, indicating that CPI plays a significant impact in climatic disasters. In order to identify best fitted model, we applied the model manually and automatic processing. By means of Auto Correlation Function (ACF) and Partial Auto Correlation Function (PACF) plots, the most appropriate order of the ARIMA model are determine and evaluated. Accordingly the created model can help in determining future strategies related to climate disaster dataset of the world. From the forecast result it is found that the results seems to show an increasing trend in CPI values and the minimal decreasing in total death condition and economic activities of the world.

Keywords: *Climate Disaster,Climate Predictions, Climate evolution, Disaster Management*

1. Introduction

The utilization of large-scale climate data is becoming increasingly important in forecasting climate change and understanding its influence on a variety of issues, including environmental illnesses. Climate data can now be collected on a worldwide scale thanks to technological improvements and the availability of sophisticated sensors, reflecting seasonal oscillations and offering vital insights for weather forecasting, climate modeling, and seasonal disease analysis. The availability of comprehensive global climate data has also resulted in breakthroughs in environmental legislation and international accords. Regulators and policymakers rely on current and precise data to develop plans and make educated decisions about reducing greenhouse gas emissions, adapting to climate change, and allocating resources to combat it. Furthermore, combining large-scale climatic data with socioeconomic aspects allows for a better understanding of the social implications

[1] Department of Computer Science, Muhammad Ali Jinnah University, Karachi 75400, Pakistan.
Corresponding Author: anumaziz9900@gmail.com

of climate change. Stakeholders and legislators can design effective ways to alleviate the detrimental consequences of climate change on those most vulnerable by examining relationships between climate-related factors and health outcomes, food security, migratory patterns, and economic indicators.

The effects of climate change are becoming more visible, resulting in natural catastrophes and extreme weather conditions that significantly impact people in need, particularly in countries that are developing. Rising temperatures, shifting rainfall patterns, and extreme weather events all have serious repercussions, including higher fatality rates. Effective temperature forecasting and understanding the implications of climate change are critical for future planning and reducing the effects of natural catastrophes. Intelligent infrastructure is critical in disaster management for gathering, integrating, managing, and analyzing disparate dispersed data sources. Climate change-related difficulties are exacerbated by shifting rainfall patterns. Prolonged droughts in certain areas cause water scarcity, agricultural failure, and food instability. Others may experience more heavy rains and a higher danger of floods, resulting in relocation, infrastructure devastation, and the spread of waterborne illnesses. Understanding the effects of shifting rainfall patterns is critical for planning for the future, particularly managing water resources, agricultural practices, and susceptible urban growth. Disaster prevention and response activities rely heavily on intelligent infrastructure. Intelligent infrastructure may acquire, integrate, manage, and analyze various data sources by using contemporary tools such as remote sensing, satellite photography, and real-time data-gathering systems. This allows for precise and timely information on weather patterns, environmental conditions, and possible threats, allowing for early warning systems, evacuation strategies, and resource allocation during natural disasters. Intelligent infrastructure also aids in disaster recovery and resilience-building by promoting data-driven decision-making and effective collaboration among multiple stakeholders. Predictive analytics seeks to estimate future system behaviors based on past data. One of the most significant benefits of machine learning is its capacity to deliver insights via various sorts of analytics. Descriptive analytics is concerned with analyzing historical data in order to comprehend underlying processes, discover patterns, and solve critical concerns. It assists stakeholders in gaining a full knowledge of previous events, assessing their impact, and making well-informed choices on the basis of that information. On the other hand, predictive analysis seeks to forecast future system behaviors by analyzing historical data. Machine learning algorithms can anticipate and predict the future climate by recognizing patterns and connections in previous climate data. This allows stakeholders to predict future hazards, arrange for mitigation measures, and efficiently allocate resources. Another key part of machine learning is predictive analytics. It extends further descriptive and predictive analytics by making suggestions and recommending actions to improve results. Prescriptive analytics can provide practical insights on how to mitigate the effect of climate-related disasters by analyzing large-scale climate data and taking into account numerous influencing factors.

Finally, prescriptive analytics entails making the best future judgments based on the outcomes of analytical methods such as descriptive and predictive. Given these improvements and obstacles, the goal of this thesis is to perform a thorough examination of several data processing prototypes, such as spatial autocorrelation models, binary segmentation models, closest neighbor algorithms, and principal component analysis. Forecasting and future forecasts are critical in climate-related research for understanding the possible implications of climate change and

establishing effective strategies for mitigation and adaptation. This method makes use of historical climate data, statistical modeling, and advanced analytical tools.

2. Literature Review

Large-scale climate data have been used to anticipate climate changes and disease from the environment. It consists of two increase scalability and feasibility, various data processing prototypes have been developed, including spatial autocorrelation models, binary segmentation models, nearest neighbor algorithms, principal component analysis as an unsupervised model, and nearest neighbor algorithms. This work conducts a thorough analysis of the aforementioned approaches to develop a fresh paradigm to handle complex climate data. Data analytics is becoming increasingly important in fields such as healthcare, social networking, climate modeling, and so on. Climate data, which reflects seasonal fluctuations, might be acquired with the sophisticated sensor. Meteorological data is used to anticipate the weather, and weather data is also valuable for analyzing seasonal diseases and reflecting seasonal changes [1][12]. The ongoing difficulty in worldwide healthcare research is determining the risk presented by epidemics of infectious diseases as our understanding of them improves and the geographical range that exists naturally increases. As the size of spatial epidemiology data expands, utilizing usable intelligence in these data has become a priority. They share volume, velocity, diversity, value, and authenticity, which are all data analysis properties. Spatial epidemiology data is a critical component of big data and healthcare analytics in digital epidemiology. The purpose of this study is to examine the geographical climate data issues in infectious illness monitoring, with an emphasis on influenza epidemics [2]. Climate change is a crucial role in determining the magnitude of other factors. Handling catastrophes has played an important role in reducing and minimizing loss of life and property damage. Intelligent infrastructure for the gathering, integration, administration, and analysis of disparate distributed data sources is required to facilitate efficient disaster management [3]. Infrastructure disaster including those caused by hydrological, climatic, and climatological consequences, have become more severe and frequent, putting cities around the world to the test. The effects of climate change have been related to higher snow loss, faster sea level rise, more frequent heat waves and droughts, stronger hurricanes, and, most importantly, a continuous and rapid rise in global temperatures. [4][11]. Descriptive analytics is focused with analyzing historical data in order to comprehend the processes under consideration, answering critical questions about these processes, and making meaningful conclusions. Predictive analytics seeks to forecast the future behavior of systems and entities based on such findings. Finally, prescriptive analytics focuses on determining the optimum future decision(s) based on descriptive and predictive analytics results [5]. The current study uses data analytics and machine learning approaches to provide a performance prediction for CPI infrastructure networks [13]. Data analytics, which is separated into descriptive, predictive, and prescriptive analytics, tries to find hidden information that cannot be investigated using traditional statistical and mathematical methods [6]. An interesting application topic for social sensing is emergency management. Despite this, little effort has been made to acquire fast estimates of the effects of disasters on the people and infrastructure. The use of crowd sourced social data, such as eyewitness testimonies, in estimating damage had long been claimed. However, the present methods dependent on citizen reporting may take days to get final conclusions [7][8].

Many studies have been published that investigate shifting patterns in temperature, precipitation, and discharge, as well as their interactions across the world. Trend analysis of historical climatic data is a critical step in

determining a region's climate status. It offers an overall assessment of the fluctuations in climatic variables during a certain time period [9].

3. Methdology

This research work comprises of many essential phases to achieve the goals of analyzing complicated climate data, building a new paradigm for dealing with such data, and utilizing data analytics for climate modeling death monitoring, and CPI. The essential components of the technique are outlined below.

3.1 Climate Dataset

Data gathering for climate catastrophe research may come from a variety of sources of information, and Kaggle is one site where you can get datasets connected to climate and catastrophic occurrences. It is a great resource for field researchers. The dataset chosen for study has a total number 16,127 records spanning the years 1900 to 2021. This broad time span enables the study of climate-related disasters over a long period of time. The dataset has 45 Characteristics, which provide a comprehensive collection of variables capturing various elements of the events. The dataset contains metadata about past disasters, such as the type of disaster (e.g., floods, hurricanes, wildfires), the date and time period when the disasters occurred, the geographical area or location where the events occurred, and potentially additional details such as severity or magnitude. These qualities provide useful information for understanding the characteristics, trends, and effects of climate-related disasters throughout history.

However, it has been shown that certain records include a high amount of null or trash data, rendering them unsuitable for analysis. As a result, these faulty entries must be removed from the dataset. Similarly, records containing trash values must be detected and eliminated, as they may reveal data input mistakes or discrepancies. These records can skew the analysis and lead to incorrect conclusions. By removing them, we verify that the dataset contains valid and useful data. After preprocessing, just 18 features of the initial 45 features with 15,071 records regarded compatible and adequate for the specified study purpose.

3.2 Dataset Analysis

This EDA process gives a basic knowledge of the dataset and aids in hypothesis formulation, modelling tool selection, and assumptions. We often investigate many features, such as summary statistics, distributions, correlations, and visualizations.

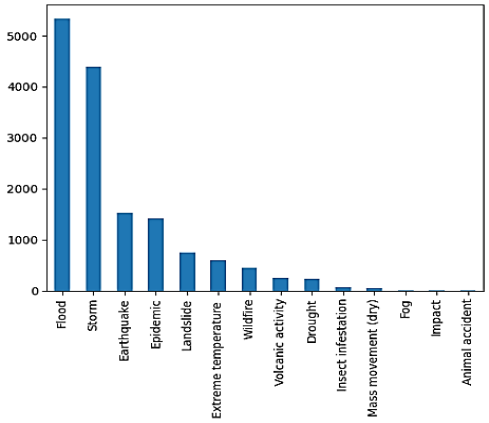


Fig. 1: Trend of Disaster Type

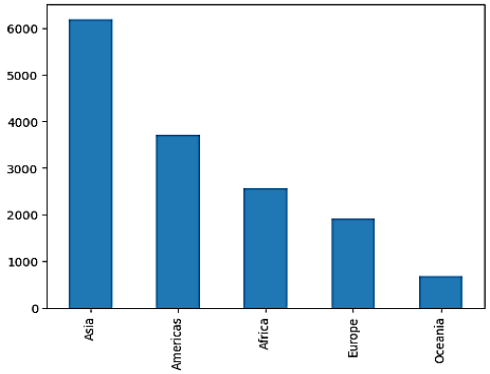


Fig. 2: Number of Disasters Occur in Regions

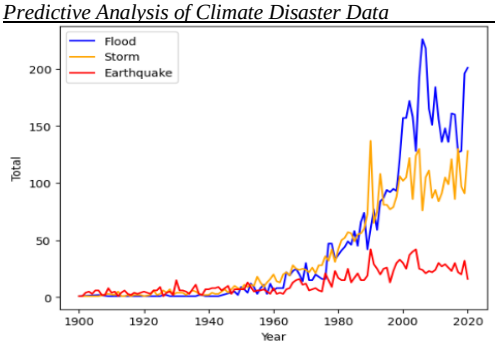


Fig. 3: Trend of Flood, Storm, Earth Quake over the time period

3.2.1 Check Data is Stationary or not?

The Augmented Dickey-Fuller (ADF) test is one of the most frequent and commonly used methodologies for determining stationarity. The ADF test is a statistical test used to detect whether or not a time series is stationary. Many statistical software packages and computer languages support it, notably Python's statsmodels module**.** The null hypothesis in the ADF test asserts that the time series is non-stationary. If the p-value produced from the test is less than or equal to the desired significance threshold (often set to 0.05), the null hypothesis can be rejected, indicating that the data is stationary.
After calculating,
p-value: 0.00010390767452394873 is less than 0.5 that means data is stationary

3.3 Predictive Analysis Modelling

The ARIMA model is used for prediction and forecasting in the dataset trend analysis. Two strategies are used: ARIMA modeling, both automatic and manual. The first step in choosing an appropriate model is ensuring that the time series data is stationary. Stationarity is a time series property in which statistical parameters like mean, variance, and autocorrelation stay constant across time [14].

Basic Components of ARIMA:

1. Autoregressive (AR) model: To predict Y_t by one or multiple lagged value. This is represented by equation mentioned below.

$$Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \ldots + \phi_p Y_{t-p} + \epsilon_t$$

2. Moving Average (MA) model: To predict Y_t by one or multiple lagged value of the error. This is represented by equation mentioned below.

$$Y_t = c + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + \ldots + \phi_q \epsilon_{t-q}$$

3. Differencing (Integration): In ARIMA Model, Time Series data must be stationary to obtaining useful information.

$$Y'_t = Y_t - Y_{t-1}$$

A stationary time series is essential for ARIMA modeling since it implies data stationarity. The next step is to choose the AR (Autoregressive) and MA (Moving Average) parameters for the ARIMA model [3][10[15]. The ACF plot, also known as the autocorrelation plot, aids in establishing the order of the MA component. After a given lag, the autocorrelation values in an MA process drop to zero. This suggests that the MA component is capturing the time series' random shocks or mistakes. The autocorrelation plot for an AR process, on the other hand, diminishes gradually or geometrically. This suggests that prior observations in the time series have persisted. We can establish the optimal order of the AR component in the ARIMA model by watching the decline of autocorrelation in the ACF plot. Following are the four main stages the model takes inputs and provides desired outputs:

3.3.1 Stage 1: Model Inputs

Several factors are included in the current study on disaster prediction to forecast the (CPI) and Total Deaths. Among the variables chosen are: *Start_Month_Year, End_Month_Year, Disaster Type, Disaster Subgroup, Region, Dis Mag value, Continent.* We can investigate the correlations, trends, and possible predictive value of each variable alone or in combination by including them in the study. The combining of several characteristics allows for a more complete study and can improve the accuracy and resilience of CPI and Total Deaths prediction models in the context of disaster situations.

3.3.2 Stage 2: Model Selection and Prediction

The ARIMA (Autoregressive Integrated Moving Average) model was chosen as the optimal way for analyzing and predicting the dataset throughout the model selection and prediction procedure. The ARIMA model is frequently used for time series forecasting and is particularly excellent at capturing the data's temporal patterns and statistical behavior. To use the ARIMA model, you must first establish the AR (Autoregressive) and MA (Moving Average) components. The ARIMA model is chosen and applied by analyzing the statistical behavior of the data, choosing the suitable AR and MA values, training the model, and generating future predictions [3][10][15].

3.3.2.1 The Auto ARIMA Model

The Auto ARIMA is a valuable tool since it automates the laborious task of manually picking the best parameters for an ARIMA model, which may be difficult and time-consuming. It is especially useful for those who do not have a comprehensive grasp of time series modeling but wish to make use of its forecasting capabilities. This approach is intended to make it easier to choose the right order of lags and variance variables for an ARIMA model. It uses a stepwise strategy to search through various parameter combinations, assessing the performance of each model using a chosen criterion (such as AIC, BIC, or AICs), and identifying the hypothesis with the best performance. We calculate the Auto ARIMA model for CPI Table1 and Total Deaths Table2 below.

Table 1: Auto AR and MA values for CPI

ARIMA(p,d,q)	*AIC*	*TIME*
ARIMA(0,0,0)	132463.091	0.15 sec
ARIMA(0,0,1)	infinity	1.28 sec
ARIMA(0,0,2)	103230.402	4.77 sec
ARIMA(0,0,3)	infinity	23.71 sec
ARIMA(1,0,0)	infinity	0.69 sec
ARIMA(1,0,1)	33134.731	6.40 sec
ARIMA(1,0,2)	33041.430	1.36 sec
ARIMA(1,0,3)	33036.118	8.94 sec
ARIMA(2,0,0)	infinity	7.25 sec
ARIMA(2,0,1)	33041.399	11.05 sec
ARIMA(2,0,2)	33138.72	8.68 sec
ARIMA(2,0,3)	33038.861	11.42 sec
ARIMA(3,0,0)	infinity	8.67 sec
ARIMA(3,0,1)	33083.273	8.46 sec
ARIMA(3,0,2)	33018.448	12.34 sec

Table 2: Auto AR and MA values for Total Deaths

ARIMA(p,d,q)	*AIC*	*TIME*
ARIMA(0,0,0)	145385.231	0.18 sec
ARIMA(0,0,1)	144316.049	0.85 sec
ARIMA(0,0,2)	143833.524	2.38 sec
ARIMA(0,0,3)	143628.520	4.20 sec
ARIMA(1,0,0)	143832.983	0.97 sec
ARIMA(1,0,1)	141902.743	3.41 sec
ARIMA(1,0,2)	141787.031	5.94 sec
ARIMA(1,0,3)	141765.330	10.00 sec
ARIMA(2,0,0)	143315.159	0.94 sec
ARIMA(2,0,1)	141776.757	6.66 sec
ARIMA(2,0,2)	infinity	9.17 sec
ARIMA(2,0,3)	infinity	19.75 sec
ARIMA(3,0,0)	143071.613	1.41 sec
ARIMA(3,0,1)	141764.384	8.01 sec
ARIMA(3,0,2)	infinity	16.45 sec

The model that has a minimum AIC Value is the best-fitted model. *ARIMA (3,0,2)* has minimum AIC for CPI and *ARIMA(3,0,1)* for Total Deaths . These (p,d,q) values will be used for auto-modeling in a dataset.

3.3.2.2 The Manual ARIMA Model

The manual ARIMA model incorporates the autoregressive (AR), differencing (I), and moving average (MA) components, as does the Auto ARIMA model. The manual technique, on the other hand, gives researchers greater power and flexibility in determining the optimal parameters for the ARIMA model. It is critical to validate the dataset's stationarity before using the manual ARIMA model. The next stage in the manual ARIMA model is to find the appropriate AR and MA parameters after confirming stationarity.

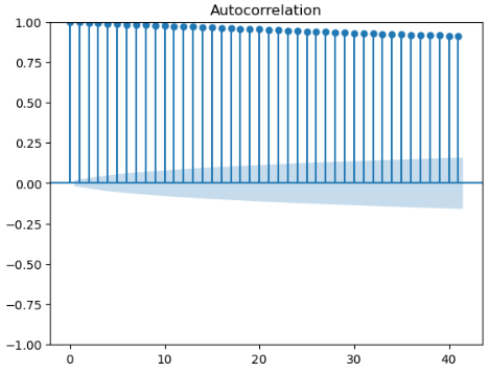


Fig. 4: ACF for CPI

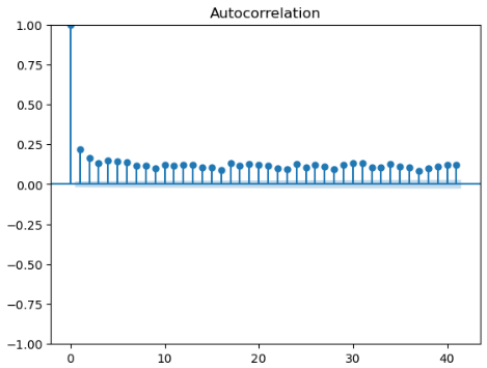


Fig. 5: ACF for Total Deaths

The Autocorrelation Function (ACF) is a useful tool for finding the lag value (p) for strong correlation in a collection of independent features. Figures 4 and 5 are most likely ACF plots demonstrating correlation coefficients at various lag levels. We can uncover significant association patterns and appropriate values for p and q by analyzing the ACF and PACF plots.

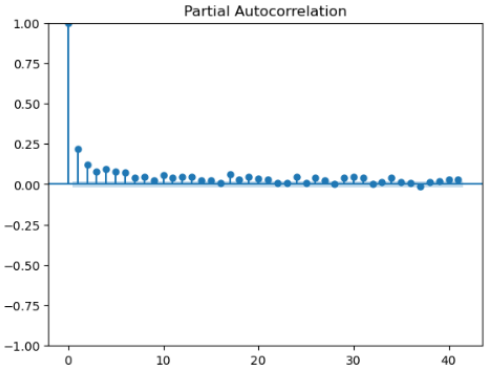


Fig. 6: PACF for CPI

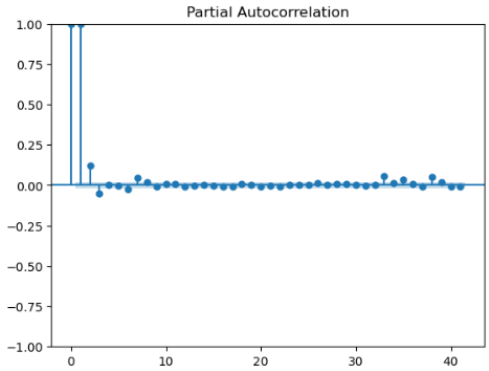


Fig. 7: PACF for Total Death

These parameters are critical in building the manual ARIMA model. According to the text, ARIMA (2,0,1) is the best-fitted model for CPI, with an autoregressive order of 2 (AR = 2), no differencing (d = 0), and a moving average order of 1 (MA = 1). Similarly, the best-fitting model for Total Deaths is ARIMA (2,1,2), which implies an autoregressive order of 2 (AR = 2), differencing order 1 (d = 1), and a moving average order of 2 (MA = 2). These parameter values are calculated by analyzing the ACF and PACF plots, which correspond to the patterns and statistical behavior seen in the corresponding datasets.

We generate predictions and projections for CPI and Total Deaths using the manually specified ARIMA (2,0,1) and ARIMA(2,1,2)

models, which incorporate the indicated autoregressive, differencing, and moving average components. These models provide a more tailored approach, allowing researchers to fine-tune the ARIMA model to their individual dataset and analysis needs.

3.3.3 Stage 3: Model Validation

It entails assessing the model's performance, correctness, and generalizability in order to assure its dependability and use in real-world applications. Accurate forecasts and credible models are critical for successful disaster management, risk assessment, and decision-making in the context of climate disaster frameworks. In this research, we will go deeper into the model validation process, its significance, and the numerous strategies used. Model validation evaluates how well the model works on previously unknown or future data, since it is critical to guarantee that the model can generalize beyond the data provided for training.

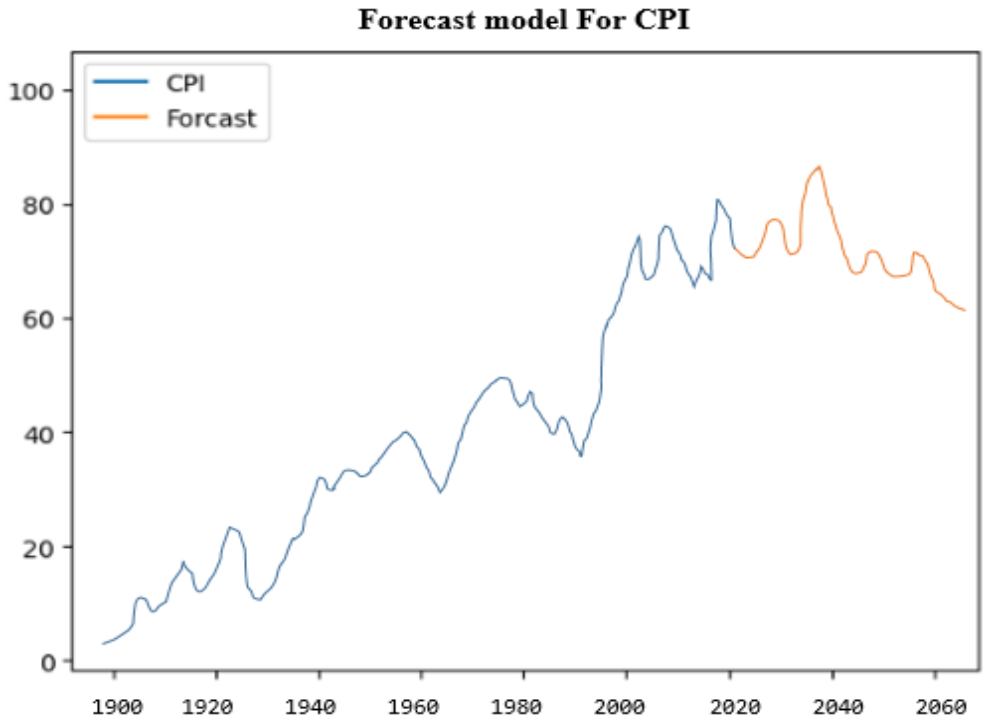


Fig. 8: Actual and Forecast for CPI

3.3.4 Stage 4: Actual and Forecast Prediction

To evaluate the accuracy of the model's forecasts, the forecast and actual lines are frequently shown on the same graph for comparison. This combined graph shows the model's predictions match the actual data. Figure 08 and 09 shows the actual and forecast values of CPI and Total Deaths with respect to the time period of 1900 to 2021. The Forecast were generated from the best fitted ARIMA (2,0,1) and ARIMA(2,1,2) models respectively. This shows the model seems to be accurately. The models' accuracy will be measured by MAE and RMSE tests.

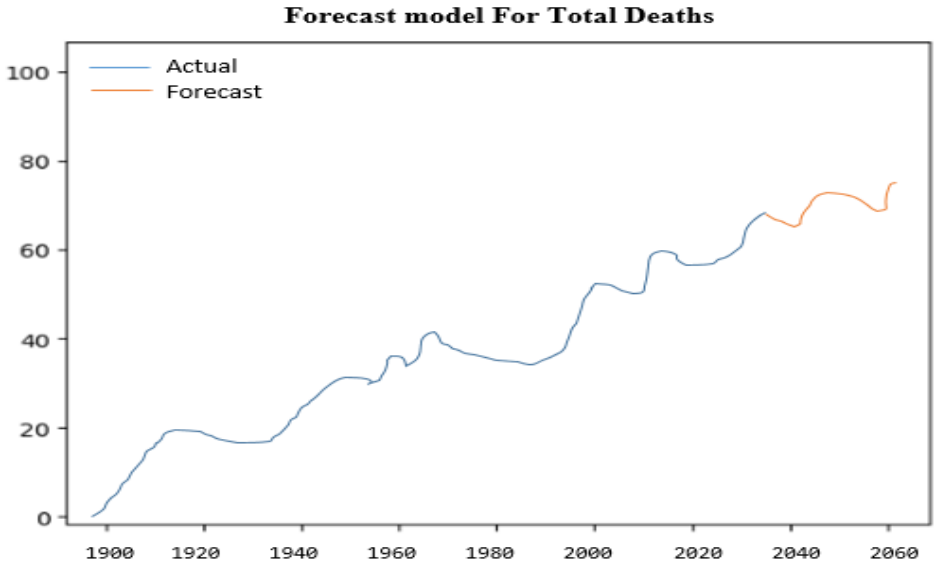


Fig. 9: Actual and Forecast for Total Deaths

4. Results

It is critical to validate the ARIMA model in order to evaluate its performance and correctness. Both the auto and manual ARIMA models are examined in this scenario using two assessment metrics: Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). These metrics give insight into the models' forecast performance and quantify the Gap between expected and actual values. Tables 3 and 4 are most likely the anticipated performance indicators for CPI and Total Deaths, respectively. These tables offer an overview of the model's predictive accuracy for the various variables.

Table 03: Forecast Performance measures for CPI feature

1900-2021	MAE	RMSE
Auto	0.32193142421649	0.3403954082930152
Manual	0.23143605915805	0.2612724085844486

Table 04: Forecast Performance measures for Total Death feature

1900-2021	MAE	RMSE
Auto	0.5240172128499798	0.5990340634941405
Manual	0.4602817487618751	0.5580509383657063

In summary, the MAE and RMSE evaluation metrics give quantitative assessments of the forecast performance of the auto and manual ARIMA models. By comparing these measurements, researchers may identify which model predicts CPI and Total Deaths more accurately, allowing them to make educated judgments and interpretations based on the anticipated numbers.

5. Conclusion

Time series analysis is critical in forecasting and predicting the CPI and Total Deaths during global climatic disasters. The goal of this work is to analyze the Climate Disaster dataset from 1900 to 2021 and construct models to make forecasts. To find the best-fitted model for the dataset, both automated

and human testing methods are used. This assumption is critical to the model's dependability. The MAE and RMSE measures are used to assess how well the auto and manual models predict CPI and Total Deaths. These measures assess the difference between anticipated and actual values. The analytical findings show that both traits are important. There is a 7% difference between the auto and manual models for the CPI feature, indicating that CPI plays a significant impact in climatic disasters. Similarly, there is a 4% difference between the two models for Total Deaths, underscoring the importance of this variable in assessing the effect of climatic catastrophes. This gives important insights into the probable effects and implications of climate change on economic parameters like the CPI and human lives as indicated by Total Deaths.

The study's findings and insights might be useful to policymakers, researchers, and practitioners working in the fields of climate science and disaster management. These findings add to a better understanding of the effects of climate change and serve as a foundation for future studies employing more advanced modeling methodologies in risk reduction initiatives. Researchers can use the tools and approaches employed in this work to expand on and construct more complex and robust models. Future research may build on these limits to refine and improve climate disaster prediction models, eventually leading to more effective risk reduction measures and climate change adaption plans.

References

[1] V.Nandhini & Geetha Devasena, 2019, *Predictive Analytics for Climate Change Detection and Disease Diagnosi.* 5th International Conference on Advanced Computing & Communication Systems (ICACCS).

[2] M. Gunasekaran, Senthil Murugan, Harpreet Kaur, Kaja M. Abbas, 2014, *Spatial Big Data Analytics of Influenza Epidemic in Vellore, India*, IEEE International Conference on Big Data.

[3] Muhammad Amjad, 2022, Analysis of Temperature Variability, Trends, and Predictions in the Karachi Region of Pakistan Using ARIMA Model, Pakistan, Academic Editor

[4] Cheng Zhang, Chao Fan, Wenlin Yao, Xia Hu, Ali Mostafavi, 2019, *Social media for intelligent public information and warning in disasters: An interdisciplinary review*, International Journal of Information Management.

[5] Jedsada Phengsuwan, 2021**,** Use of Social Media Data in Disaster Management: A Survey, Future Internet

[6] May Haggag, 2021, Infrastructure performance prediction under Climate-Induced Disasters using data analytics, International Journal of Disaster Risk Reduction.

[7] Marco Avvenuti, 2017, *Nowcasting of Earthquake Consequences using Big Social Data*, IEEE Internet Computing

[8] Gary Feng, 2016, *Trend analysis and forecast of precipitation, reference evapotranspiration and rainfall deficit in the Blackland Prairie of Eastern Mississippi*, Journal of Applied Meteorology and Climatology.

[9] Rashid Mahmood, 2017, *Spatial and temporal hydro-climatic trends in the transboundary Jhelum River basin*, Journal of Water and Climate Change

[10] Andrea L. Schaffer, 2021, *Interrupted time series analysis using autoregressive integrated moving average (ARIMA) models: a guide for evaluating large-scale health interventions*, BMC Medical Research Methodology.

[11] Benoˆıt Sarr, 2012, *Present and future climate change in the semi-arid region of West Africa: a crucial input for practical adaptation in agriculture*, Royal Meteorological Society.

[12] Brian R. Pickard, 2015, *Translating Big Data into Big Climate Ideas*, Research Gate

[13] N. W. Arnell, 2019, *Global and regional impacts of climate change at different levels of global temperature increase*, Climate Change

[14] Rashid Mahmood, 2019**,** *Global and regional impacts of climate change at different levels of global temperature increase*, Scientific Report.

[15] YUCHUAN LAI, 2019, *Use of the Autoregressive Integrated Moving Average (ARIMA) Model to Forecast Near-Term Regional Temperature and Precipitation*, American Meteorological Society

Vol. 7, No. 2 | July – Dec 2023

Prediction of Next Word in Balochi Language Using N-gram Model

Sharan Bashir[1]*, Sohail A. Sattar[1], Muhammad Umer Farooq[1], Saad Ahmed[2], Mustafa Latif[3]

Abstract:

Balochi Language is among the oldest languages, spoken by approximately 10 million people worldwide. The Balochi language has been spoken for a very long period. In comparison to other languages like English, Urdu, French etc. it has a research gap in Natural language processing (NLP). The next word prediction system is one of the techniques of NLP for suggesting standardization and corpus collection. This research aims to provide a next word prediction system and a corpus with no ambiguity for the Balochi language. N-gram model for the next word prediction has been utilized, i.e. Unigram, Bigram, Trigram, Quad-gram and so on. A trained model has been embedded in an application after being evaluated extrinsically and intrinsically. It plays a crucial role in typing through a keyboard and helps users to type faster. Additionally, it helps native users to have fewer typing errors in less time. The results of the research show that Five-gram model has the highest performance of 93% while Quad-gram model has 80% and Trigram model has 76% respectively.

Keywords: *NLP; N-gram Model; Word Prediction; Intrinsic evaluation; extrinsic evaluation; Laplace smoothing; Lidstone smoothing*.

1. Introduction

Balochi is among the oldest living languages of west Iranian languages; approximately 10 million people speak it as their first or second language in Pakistan, Iran, Afghanistan, Turkmenistan, and Gulf countries [1]. The language has three dialects Eastern, Western, and Southern. Its script is Latin (Roman), written from left to right, and Perso-Arabic, from right to left. The language has short and long vowels, both with consonants. Arabic script is preferable in writing, as most books and magazines are published in Arabic script. Nevertheless, there is hardly any work done to digitize the Balochi language. In recent times, writing has shifted to digital devices. Many tools for language intelligence have been made possible by natural language processing, including language translators, semantic analysts, spell checkers, word predictor etc.

Next word prediction is the primary feature of language in NLP. It facilitates swift and

[1]Department of Computer Science and information Technology, NED University of Engineering and Technology, Karachi, Pakistan
[2]Department of Computer Science, Iqra University Karachi
[3]Department of Software Engineering, NED University of Engineering and Technology, Karachi, Pakistan
Sharan Bashir: [sharanbaloch2018@gmail.com]

error-free typing. It suggests the most probable next word for the currently typed word. Moreover, it can be used for multiple devices like smartphones, tablets or laptops. On next word prediction, a lot of research have been conducted in different languages like Urdu, Sindhi, Hindi, Bangla, Hebrew, Kurdish, and Assamese. Unlike other languages, no significant research has been doneon Balochi language by using next word prediction system. Therefore, there is a need for research in Balochi language to implement the next word prediction to help users to type more frequently and accurately in this Language.

N-gram Language Model is a statistical language model broadly used to predict the next word. In this paper, N-gram Model is utilized for next word prediction in the Balochi language with the addition of Laplace smoothing to avoid data sparseness. N-gram aims to predict the next word according to its previous N-1 words. It relies on the combination of the words like Bigram two words pair, Trigram three words together, Quad-grams four words together and so on. The model was trained on the Balochi corpus according to its number of N-grams. Moreover, the corpus is based on 300,000 words approximately of western dialect taken from a Balochi novel.

The paper consists of the following sections: Section 2 deals with the literature review, the Balochi Language is discussed in Section 3 and methodology of the proposed model is mentioned in Section 4. The result and discussion are illustrated in Section 5. Finally, the conclusion and future work is drawn in Section 6.

2. Literature Review

The Word prediction system has been an essential practice in augmentative communication for over 20 years [2]. Early word prediction systems, introduced in the 1980s, were used as assistance for those who had difficulties in learning [3]. For the last few years predictive techniques on the previous word prediction have become the need of any language. Social media has a significant role in daily life, so each language has to cope with its requirements. Unlike the Balochi language, most work has been done for languages like Urdu, Sindhi, Bangla, Hindi, Kurdish etc.

S. Shahzadi et al. [5], 2013, have worked on the Urdu language keyboard for android mobile phones to have a word prediction function, which works on less memory with less processing speed by using bigram. M. Hassan et al. [4] 2018 also implemented a system for the Urdu language to predict the next word with reference to a current word using the Hidden Markov model. J. Mahar and G. Memon [6] have used N-gram for Sindhi Language next word prediction in a sentence by using its previous history. The Add-One smoothing method is also utilized to assign non-zero probabilities to those N-grams with zero possibilities for increasing N-grams' performance.

In a paper for the Bangla language, M. Haque et al. [7] applied unigram, bigram, Trigram, deleted Interpolation, and back-off models to complete the sentence automatically by having a word prediction system. However, the back-off model's accuracy is higher than other models, which is 63.50, unigram at 21.24%, bigram at 45.84%, Trigram at 63.04%, and deleted Interpolation at 62.86%. Habib et al. [8] have worked on the Bangla language using unigram, bigram, Trigram, back-off and linear Interpolation. The accuracy level of Trigram, linear Interpolation and back-off is the same; however, in accuracy and failure together, linear Interpolation is the most accurate in the word prediction process, is 77%. Mittra et al. [9] have applied the N-gram term frequency matrix to the total count of each stored term in the Bangla language. They also measured the semantic similarity of sentences after word prediction with the help of Word2Vector, and for the most probable word suggestion, they used the Stupid Back off model, which worked well for large N-grams.

Shah and Kshetra [10] have introduced Sn-Grams, i.e. "Syntactic N-grams" that follow the grammar while making a prediction. Likewise, two deep learning techniques have been utilized to predict the Hindi Language's [11] next word: Long Short Term Memory (LSTM) and Bi-LSTM. At the same time, their

accuracy rate is measured as 59.46 % and 81.07 %, respectively. Hamarashid et al. [12] implemented the N-gram model with the addition of the Stupid Back-off (SBO) algorithm to create the word prediction system that is based on two dialects of Kurdish, Kurmanji and Sorani. The system could not find the proper result to predict the next word each time. It decreases N-gram size, for example, from Trigram to bigram.

It is claimed that the corpus size for training purposes can acquire the accuracy of word prediction [13]. However, accuracy can also depend on various other factors like the methodology of prediction, speed of prediction, dictionary structure, user interface and several words suggested to the user [14].

In 2008 research was conducted by M. Herold et al. [15] to examine whether using next word prediction can improve typing speed and spelling accuracy. For the experiment, 80 students from grades 4-6 were selected who had difficulty with spelling. The students entered 30 words via an on-screen keyboard with the next word prediction ability and, without it, software. Surprisingly, an increase in spelling accuracy had arisen with the use of software with next-word prediction.

In 2020, Nandini et al. [16] discussed a word prediction system for the Kannada language by using Naïve Bayes. However, for model optimization, they implemented stochastic gradient descent. Ali Pourmohammad et al. [17] proposed Hidden Markov Models (hmms) for next word prediction in the Azeri language based on Natural language processing techniques.

A. Yazdani et al. [18] evaluated three measurement terms to find the efficiency in the next word prediction system using trigram as keystrokes, time reduction and text generation rates. The results show typing time reduction of 33.36% and 73.53% in the number of keystrokes.

In a study, M. Parekh and Y. Patel [19] applied a linear probability model for word completion. The results were shown to efficiently reduce keystrokes by about 51% and 0.019s. Using Impala, the 2-grams of Google N-grams dataset was used on Hadoop Distributed File System (HDFS).

Bhuyan and Sarma [20] have described two predictive models: a traditional model for general word prediction and an enhanced model for ambiguous word prediction. They used 6 grams for the improved model and quad-grams for the general model. In addition, they implemented Katz's back-off smoothing model to avoid zero probability of words. The accuracy of the enhanced model is 66.88%, and the failure rate is 29.17%. Moreover, the accuracy of the traditional model is 60.68%, and the rate of failure is 32.35%.

Nagalavi and Hanumanthappa [21] proposed an exponential interpolation language model, which combines the POS language model and the N-gram language, model. The model is to predict an aligned sequence of words in blocks of articles in e-Newspaper, and the model's accuracy is 98.8%.

Gosh, et al. [22] have introduced a collaborative filtering algorithm with the Pearson correlation coefficient (PCC) to calculate the word similarity for predicting the frequencies of a bigram, which are missed. Furthermore, they elaborated that filtering or smoothing can increase the efficiency in next word prediction.

Word prediction and auto-complete are very useful in Search Engines and hand-held devices like smartphones for typing purposes, and it helps reduce incorrect spellings and efforts of typing, increasing the communication rate [23]. Further, it can diminish the gap between fast and slow typing, and it also helps people with disability in typing [24]. Likewise, W. Tesema and D. Tamirat [25] have implemented a system to help disabled people in typing as the next word prediction. The accuracy and precision of the model were found to be 90% and 73.34%, respectively.

Nowadays, word prediction is not unique to a language. A system is introduced that can predict emoticons (highly used in social media to express their thoughts) for a text [26]. Also, Yogesh et al. [27] included punctuation marks

and semantic rules of the language in the word prediction model. Furthermore, K. C. Arnold et al. [28] have introduced a system for phrase suggestion in mobile communication with the help of the 5-gram model. Unfortunately, no research has been conducted on the next word prediction system for the Balochi Language. Only one study conducted for the Balochi language in computer science is based on Optical Character Recognition (OCR) [29].

So far according to literature review N-gram model is not used for Balochi Language, therefore in this paper, a novel system is developed, based on a combination of N-gram model and Laplace smoothing. The proposed system helps users to quickly type in the Balochi language with a high rate of keystrokes and without spelling mistakes. The system is a simple model effective for next word prediction requirements that is smart enough to suggest words without any need for grammatical rules that help to save time.

3. Balochi Language

Balochi is among the oldest languages in the world. But, there is no official estimation of the number of Balochi speakers. However, grim statistics say that balochi is the first language of at least 10 million people worldwide. These people belong to Pakistan, Iran, Afghanistan, Oman, UAE, Turkmenistan, East Africa and India [1]. Unlike other languages, it has a recent history of scripting. Balochi had no written language before 19th century, hence most historical events and tales were transmitted verbally.

Nevertheless, it is said that in mid of 18th century Osman Kalmati wrote a book containing Balochi poems. That book is kept in British Library, London [30]. Due to their proximity to Iran and the fact that certain Balochs had studied the Holy Quran, they developed an Arabic and Persian writing system that was compatible with Balochi. Later, after the British had taken control in the 19th century, they gave the Balochi language a Roman/Latin script. British priests taught their people Balochi and translated the Bible into Balochi [31].

Table I. Samples of Balochi Language Letters

Balochi Academy	Phon-emes	Words	Pronunciation	Meaning
ا	\a\	اپس	Aps	Horse
ب	\b\	باہوٹ	Bahot	To be in Custody
پ	\p\	پاد	Paad	Foot
ت	\t\	تو	Tou	You
ٹ	\ ṭ\	ٹپ	Ta,p	Mark
ج	\j\	جار	Jaar	Announcement
چ	\c\	چار	Chaar	See
د	\d\	دپ	Dap	Mouth
ڈ	\ḍ\	ڈڈ	Dadd	Hard
ر	\r\	رد	Rad	Wrong
ڑ	\ŕ\	مڑاہ	Marrah	To be shy
ز	\z\	زبگ	Zahg	Child
ژ	\ż\	ژند	Zhand	Tired
س	\s\	سر	Sar	Head
ش	\sh\	شام	Shaam	Evening/ dinner
ک	\k\	کار	Kaar	Work
گ	\g\	گٹ	Gatt	Busy
ل	\l\	لال	Laal	Red
م	\m\	مات	Maat	Mother
ن	\n\	نان	Naan	Bread
و	\w\	وہد	Wahd	Time
ہ (ھ)	\h\	ھشک	Hushk	Dry
ی	\i\	یک	Yak	One
ے	\e\	نودربرے	Nodarbare	A Student

After the creation of Pakistan, the Baloch scholars adopted the Perso-Arabic script for their language. In the early 1950s, Gul Khan Nasir, who is called the father of modern Balochi poetry, published his first poetry book, "Gulbang". Also, Sayad Zahoor Shah Hashmi, "The Father of Balochi", created complete guidance on Balochi language writing to give it a standardized orthography. He also wrote the first dictionary of Balochi, named as "Sayad Ganj".

There has also been a Cyrillic script used for Balochi in addition to the Perso-Arabic

alphabet. In the 1980s, the Baloch of Turkmenistan invented this writing. However, the Cyrillic script was unable to gain international acceptance once the USSR was divided [32]. In May 2000, in a workshop at Uppsala University, the Latin script for the Balochi language was adopted again. Tab. I, presents Balochi alphabets of Arabic and Latin script.

3.1. Balochi Orthography

Sayad Hashmi [33] considers these excluded letters already have their alternative sounds like ص ،ث like س \s\, ذ ،ض ،ظ have alternative sounds as ز \z\, ع as ا \a\, غ as گ \g\ , ق as ک \k\ and ف as پ \p\. R Regarding standardization of language, academies like Balochi academy, Balochistan academy, Uppsala University and other Balochi language publishers use different alphabets of Arabic script. So, adding those letters to a Balochi keyboard also becomes necessary.

Thus far, in the world of technology, the Balochi language does not have proper research, and one of the reasons is the lack of adequate corpus. There is a need of non-ambiguous corpus for efficient word prediction system in the Balochi language to bring the language to the standard of other languages of the world.

4. Methodology

The proposed model is divided into three steps: in step one, data is preprocessed; in step two, the data is trained to generate N-grams as Unigram, Bigram, Trigram, Four-gram and so on, and in step three, the accuracy of the model is calculated through the test data by using perplexity and accuracy formula. These steps are elaborated further below and Fig. 1 represents the complete workflow of proposed model.

4.1. Data Pre-Processing

Data pre-processing is one of the primary and critical steps in processing the text to obtain pertinent data from the raw document. The text pre-processing steps are followed, which are required to gain an appropriate form of data for the model processing.

4.1.1. Corpus Collection

The whole data is based on a Balochi novel named “Bahisht o Doza” “بهشت ءُ دوزه” [34]. It is in a text file format that is easy to access. The corpus consists of 317590 words; it is divided into 80% for training purposes and 20% for testing. The total counting of words and sentences of corpus are presented Tab. 2.

Table II. Total Number of Words and Sentences in Corpus

	No of Words	**No of Sentences**	**Divided Percentage of corpus**
Training	253962	26511	80%
Testing	63629	6707	20%

As earlier explained, the Balochi language does not have a standard form. It has an ambiguous structure. One word has two different spellings, which makes the whole corpus confusing. This problem decreases the probability of the number of occurrences of the word. Therefore, it can directly affect the accuracy of the model is calculated through the test data by using perplexity and accuracy formula. These steps are elaborated further below and Fig. 1 represents the complete workflow of proposed model.

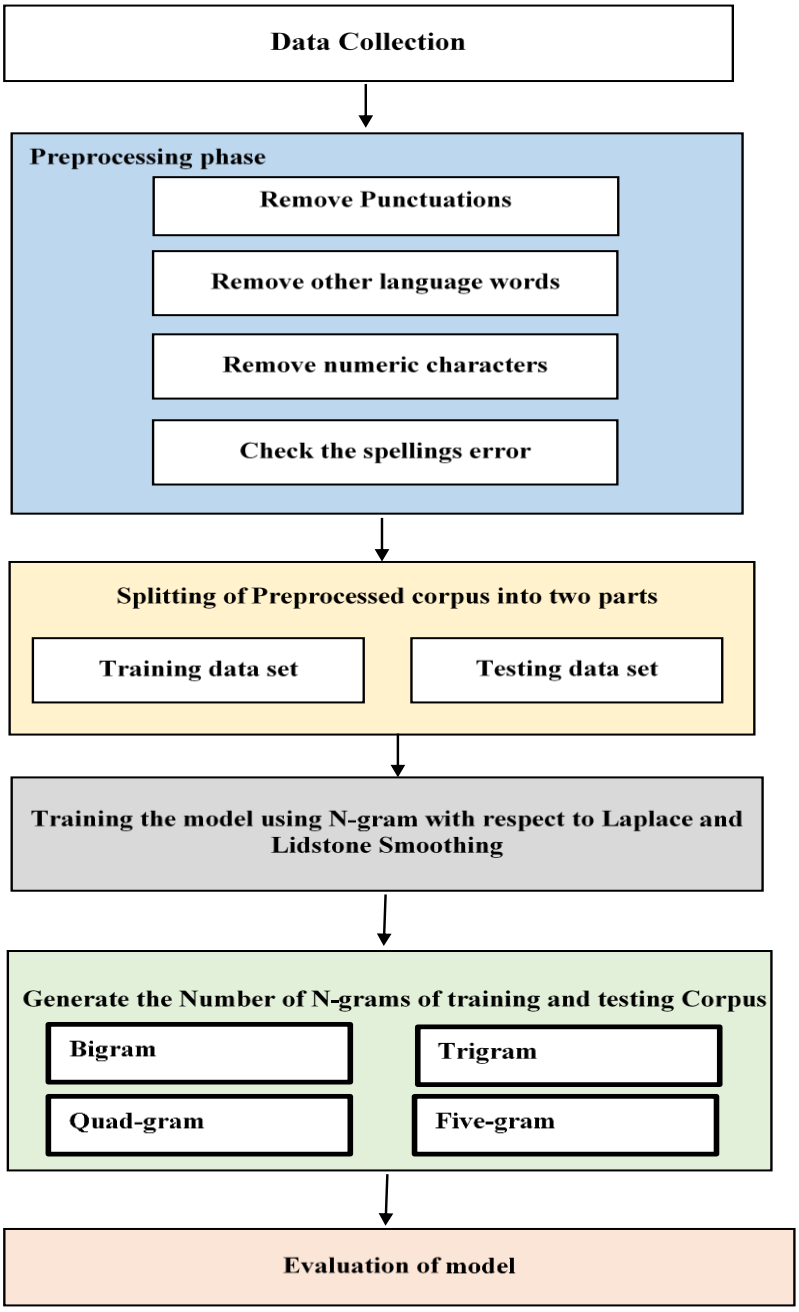


Fig.1. Methodology of the Proposed Model

Table III.: Example of Ambiguous words

Meaning	**Balochi Word**	**Balochi Word**	**Pronunciation**
Tell, say	گوش	گش	Gosh / Gwash
Sister	گوهار	گہار	Gohar
Think, thought	حیال	هیال	Hayal
History	راجدپتر	راج دپتر	Raj Daptar
You	شما	شُما	Shuma
Hani (Name)	هانی	حانی	Hani

In Tab. 3, it can be seen the word گوش that is pronounced as "Gwash" and the other one is گش that is pronounced as "Gosh", both the words are same but spoken in different dialects. The other word حیال and هیال as "Hayal", their pronunciation is same but spellings are different. As lack of language standardization the discussion whether to include ح in Balochi language or not. So those words which are started with H are sometimes written with ح and ھ. Due to these ambiguity the system considers these words different from one another.

4.1.1.Text Pre-Processing

There are several steps in text pre-processing, like noise removing, stemming and lemmatization, stop words removing, spell-checking etc.; each corpus is pre-processed according to the requirement of the model. Following are the steps that are required for the model before training.

- The first step is to clean the noise like words from other languages and numbers.
- Spellings of the words are checked to avoid ambiguity among words.
- The machine considers punctuation marks as words, which are not required in this research. After converting the docx file into txt format, the whole corpus is segmented into sentences with the use of (. ؟ ، !) As a separator. Then all other punctuation marks (: ؛ ' ") are removed from the corpus.
- The complete corpus is tokenized through NLTK library, those tokenized words are also considered Unigram.
- While testing the model there may occur words that are not seen in the training corpus those words are known as out of vocabulary words. These are tagged as Unknown <unk>.

Once the model is cleaned from the noice the next step is to train the model with respect to N-gram model that is proposed for the word prediction system.

4.2. N-Gram Language Model

Word predicting is a probabilistic task to check the appearance of the next word's probability with the current word. The main task of Language Models is to assign probabilities to the word sequences. One of the basic approaches is to implement word prediction on the N-gram Language model. It is a statistical model used to predict the possible word that follows the sequence of the given word [35]. The N-gram language model predicts the probability of a word occurrence with its previous word, also called the Markov assumption [7]. The computing task is done in Eq. (1).

$$WP=P(w|h) \qquad (1)$$

Where WP is word prediction, P is the probability of the current word w, and h is the history of the current word. Thus, the N-gram model predicts (n-1) words. Hence, the Markov assumption provides the technique to look at the current word's probability to its last term. The N-1 Markov assumption model measures the N-gram model. It assumes that the next word can only be predicted with the awareness of the previous N-1 word. The formula is obtained by using Markov Model with the N-gram model as shown in Eq. (2).

$$P(W_1^N) \approx \prod_{K=1}^{N} P(W_k|\, W_{k-1}) \qquad (2)$$

When N = 1, it is known as Unigram, that predicts how often a word has occurred in a

sentence sequence. When N = 2, it is known as Bigram, also called Markov first order. This is because it checks one prior word's probability with the current word. The general formula for the sequence of bigram to predict the next word conditional probability is mentioned in Eq. (3), and the generalized form is shown in Eq. (4).

$$P(W_n|W_{1,n-1}) \approx P(W_n \mid W_{n-1}) \quad (3)$$

$$P(W_N \mid W_1^{N-1}) \approx P(W_N \mid W_{N-N+1}^{N-1}) \quad (4)$$

However, the probability estimation is calculated using MLE, that is, Maximum Likelihood Estimation; it is done by taking the counts from the corpus and then normalizing them. After the normalization, the general formula is derived in Eq. (5).

$$P\ (W_n|W_{n-1}) = \frac{P(W_n|W_{n-1})}{P(W_{n-1})} \quad (5)$$

Thus when N = 3, it is known as a trigram that checks the probability of the next word with two previous words; it is called Markov second order and the general formula is presented in Eq. (6).

$$P(W_{1,n}) = P(W_n|W_{n-2},W_{n-1}) \quad (6)$$

After normalization, the formula obtained is shown in Eq. (7).

$$P(W_n|W_{n-2},W_{n-1}) = \frac{P(W_n \mid W_{n-2}, W_{n-1})}{P(W_{n-2},W_{n-1})} \quad (7)$$

Likewise, when N = 4, 5, 6 and so on, it checks the occurrence of the next word to three to four to five previous words as N-1.

The probability estimation of a sentence in Unigram model is as "منی نام هانی اِنت" "My name is Hani". Tab. 4 represents that how the Balochi language is pronounced, and the meaning of each word is presented in English.

Table IV. Balochi Words Pronunciation

Balochi Words	**Pronunciation**	**Meaning in English**
منی	Mani	My
نام	Naam	Name
هانی	Hani	Hani
انت	Int	Is

P(My name is Hani) = P (My) × P(name)× P(is) × P(Hani)

P (منی نام هانی اِنت) = P (منی) × P(نام) × P(هانی) x P(اِنت)

In the generalized form, the Unigram formula is given in Eq. (8).

$$P(Wi) = \frac{C(Wi)}{C(W)} \quad (8)$$

P is the probability of the current word wi, c is the count and w is the entire sentence,

$$P(\text{هانی}) = \frac{C(\text{هانی})}{C(\text{منی نام هانی اِنت})}$$

For the bigram model, the probability sequence of a sentence is,

P(My name is Hani) = P(My|<sos>) × P(name | My)× P(is |name) × P(Hani | is) × (</eof> | Hani)

P (منی نام هانی اِنت) = P(منی |<sos>) × P(| نام) × P(منی | هانی) x P(هانی |اِنت) x P(</eof> | اِنت)

<sos> represents the start of sentence, it is used in training to determine the start of the sentence. While </eof> represents the end of sentence, these are the padding sequences that are used separate one sentence probability from another. From the generalized formula, the Eq. (5) becomes,

$$P\ (\text{منی} \mid \text{نام}) = \frac{P(\text{منی} \mid \text{نام})}{P(\text{منی})}$$

For trigram model, the probability estimation is given as:

P(My name is Hani) = P (My|<sos><sos>) × P(name | My <sos>)× P(is | My name) × P(Hani | name is) × P(</eof> | is Hani) × P(</eof> | </eof> Hani)

P (منی نام هانی اِنت) = P(منی |<sos> <sos>) × P(نام | منی<sos>) × P(هانی| منی نام) x P(| اِنت) × P(نام هانی) × P(هانی اِنت | </eof>) × P(</eof> | </eof> اِنت)

Hence Eq, (7) becomes,

$$P\ (\text{هانی}|\text{نام منی}) = \frac{P(\text{منی نام}|\text{هانی})}{P(\text{منی نام})}$$

Thus, the N-gram model is a probabilistic sequential model that suggests the word according to its n number of previous words. It represents the nth order of Markov assumption that it depends on the preceding structure of the N-gram. However, in the above example, if the probability of a word "تئی" (your) is checked instead of the word "منی" (my) which never occurred in training due to multiplication and division, the probability becomes 0. That brings data sparsity, due to which the MLE model is unsuitable. To counter this problem, the smoothing technique is utilized.

4.2.1. Laplace Smoothing (Add-one Smoothing)

Maximum Likelihood estimation, the generalized form of N-gram, has the data sparseness problem. Due to that, the problem of zero probability occurs. However, the smoothing technique plays a vital role to avoid the sparsity of data which means looking ahead. Among various smoothing techniques, one of the simplest is the Laplace smoothing. The Laplace smoothing adds one to any number of N-gram before normalization.

By modifying Eq. (5) with Laplace smoothing, 1 is added to the count and Vocabulary V to its denominator, as represented in Eq. (9).

$$P_{add-1}(W_n|W_{n-1})=\frac{P(W_n, W_{n-1})+1}{P(W_{n-1})+V} \quad (9)$$

In spite of this, Laplace gives too much probability to unseen data instead of seen data. This problem can be avoided by using Lidstone smoothing.

4.2.2. Lidstone Smoothing

Lidstone smoothing is also named as Expected Likelihood Estimator. It adds a value (λ), smaller than and equal to 1 to the unseen data. It supposes that each n-gram has been seen λ times that is $0 < \lambda \leq 1$, Eq. (10) represents the formula of Lidstone,

$$P_{Lid}(W_1 \ldots W_n) = \frac{C(W1 \ldots wn)+\lambda}{N+B\lambda} \quad (10)$$

Where P is the probability of N-gram, C is the training count of N-gram, N is the total number of n-grams in training, and B represents the possible number of N-grams, and λ is the small positive number for unseen data to reduce sparsity.

4.3. Model Evaluation

The primary purpose of model evaluation is to determine a model's simplified accuracy on unseen future data. In this phase, the decision is taken on how well the model works. For example, the perplexity evaluation matrix and accuracy to evaluate the N-gram model are used.

4.3.1. Perplexity

Perplexity is the intrinsic evaluation metric used to determine how well the model predicts the next word. The model assigns the highest probability to the test data by that the model does not get confused it gets a good understanding of the model from the test set. It is the inverse of the probability which is assigned to the test set, as shown in Eq. (11).

$$PP(W) = \frac{1}{P(W_1, W_2, \ldots, W_N)^{\frac{1}{N}}} = \sqrt[N]{\frac{1}{P(W_1, W_2, \ldots, W_N)}} \quad (11)$$

The higher the probability, makes the perplexity lesser. The model performs well if it has lower perplexity. It is also known as cross-entropy, 2H (W), which means the average number of bits needed to encode one word. However, the perplexity of the words is encoded in those bits [36], like if the branching factor is 200. In that case, prediction is based on among these 200 predictions, which are the best to be chosen, which is shown in Eq. (12).

$$PP(W)=2H(W)=2^{-\frac{1}{N}P(w1,w2,\ldots,WN)} \quad (12)$$

In Eq. 12, H (W) = the average bits required for encoding one word, and 2H (W) = the average words which could be encoded by the use of H (W) bits.

4.3.2. Accuracy

The model's accuracy means the total number of predictions divided by the sum of correct and incorrect predictions. It has been calculated by embedding the model into an application then correct and inaccurate predictions are manually checked. However,

the task is time-consuming but worthy enough to estimate the result. After all, the user determines how accurate the model is. Eq. (13) represents the formula to calculate the model's accuracy.

$$Accuracy=\frac{numberofprediction}{correctprediction+incorrectprediction} \quad (13)$$

5. Result and Discussion

In the proposed system, approximately 300,000 words of Balochi book are divided into two parts 80% for training and 20% for testing. The most frequent Unigrams, Bigrams, Trigrams, Quad-grams and Five-grams generated from training corpus are illustrated in Figs. 2-6 respectively.

These N-gram counts are produced using training corpora, which implies smoothing to prevent data sparsity. Sparse data is a computational issue that is the phenomenon of having insufficient data in a dataset or data with value of zero. This issue is solved using smoothing techniques, where the zero value data are given a value of 1 for Laplace smoothing and a value of 0.5 (according to the requirement) for Lidstone smoothing.

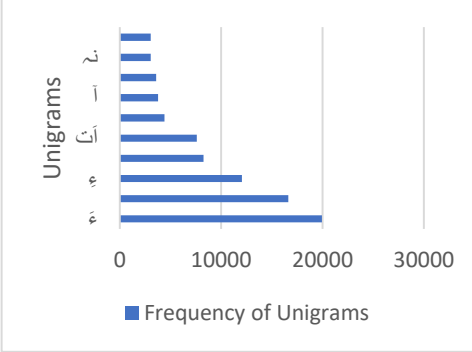


Fig. 2. Most Frequent Words in Unigrams from training corpus

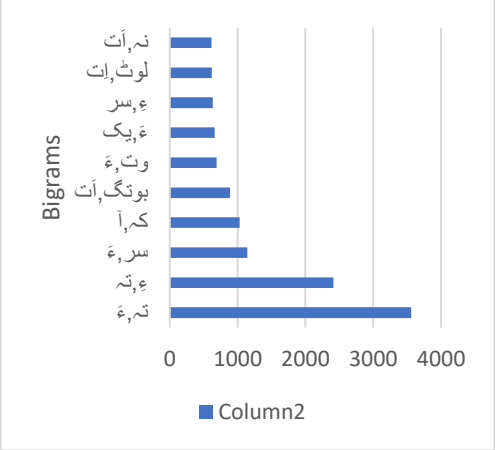


Fig. 3. Most Frequent Words in Bigrams from training corpus

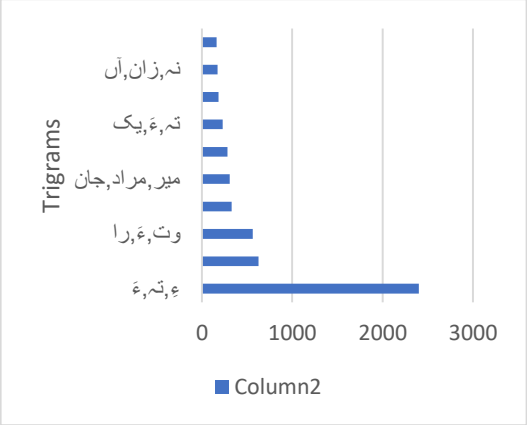


Fig. 4. Most Frequent Words in Trigrams from training corpus

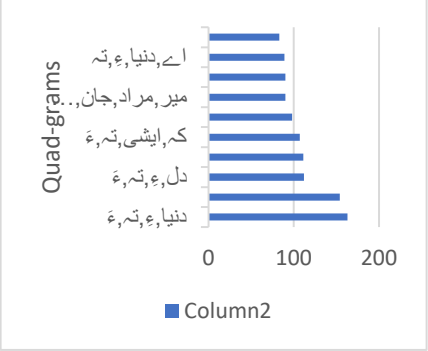


Fig. 5. Most Frequent Words in Quad-grams from training corpus

Fig. 6. Most Frequent Words in Five-grams from training corpus

5.1. Intrinsic Evaluation Result

For various N-gram models, the perplexity (i.e., intrinsic evaluation) result is obtained using Laplace and Lidstone smoothing, as shown in Tab. 5 and Fig. 7. Lidstone smoothing employs a 0.5 value for unseen words in the testing corpus since it outperforms Laplace by a quite margin. Laplace, however, employs one value for the testing corpus's unseen data. After applying Laplace and Lidstone smoothing, the difference in Bigram perplexity is 42, trigram is 56, quad-gram difference is 42, and five-gram difference is 35. The average difference is 43.75, which indicates that the N-gram performs 43.75 times better than Laplace smoothing when Lidstone smoothing is being used. It demonstrates how much more effectively Lidstone smoothing decreases data sparsity. This benefits in lowering word prediction errors.

Table V. Perplexity results after Laplace and Lidstone Smoothing

Model	**Perplexity (Laplace)**	**Perplexity (Lidstone)**	**Difference in Perplexity**
Bigram	419	377	42
Trigram	359	303	56
Quad-gram	312	270	42
Five-gram	275	240	35
Average			43.75

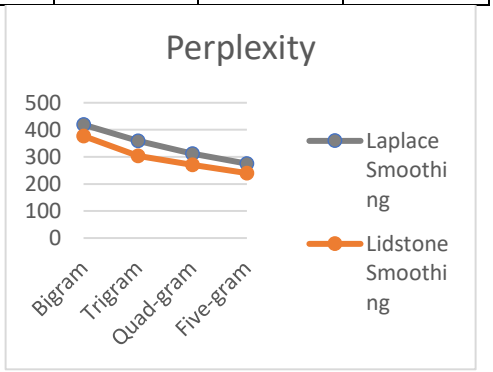


Fig. 7. Perplexity measurements through Laplace and Lidstone smoothing

5.2. Extrinsic Evaluation Result

Further, the trained model is embedded in an application to predict the next word (i.e. extrinsic evaluation). The following conditions are applied in the application:

- When a user enters a word, it uses the bigram model and predicts one word to the next.
- When the user enters two words, it uses the trigram model and predicts two words to the next word.
- When a user enters three words, it uses the Quad-gram model and predicts three words to the next word.

5.2.1. User Interface

The user interface helps to interact with the model in the real world, which is considered a final product. A user checks the model through the user interface to see whether the result is correct. The user enters one word; after pressing space, it predicts the next most common word. The following figures represent the user interface; Figs. 8-10 show the model embedded in an application that how it is suggesting the next possible word.

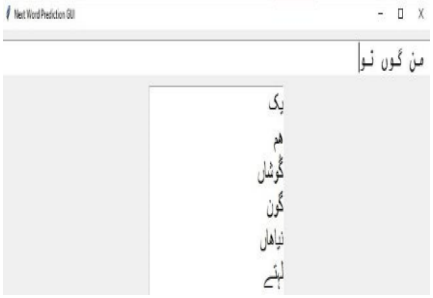


Fig. 8. Predicts the next word based on trigram

Fig. 9. Predicts the next word based on quad-gram

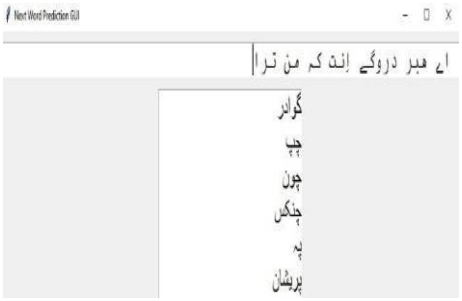


Fig. 10. Predicts the next word based on five-gram

From the above application, 100 sentences are generated for five-gram, quad-gram, Trigram, and bigram to determine the accuracy. The accuracy of Bigram is 68%, Trigram is 76%, Quad-gram is 80 %, and Five-gram is 93%. Tab. 6 and Fig. 11 represent the accuracy of these models.

Table VI. Accuracy generated from training corpus

Model	**Accuracy**
Bigram	68%
Trigram	76%
Quad-gram	80%
Five-gram	93%

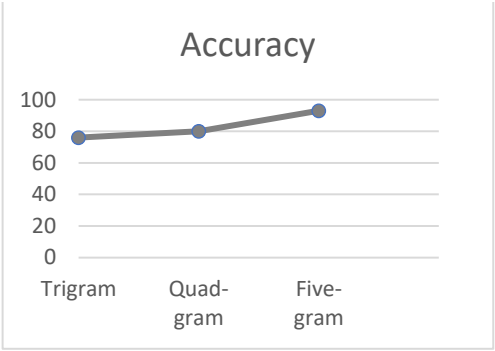


Fig. 11. Measurement of Accuracy

5.2. Result Analysis

A large corpus of approximately 3 lac words was utilized to build the Next Word Prediction System for the Balochi Language. The corpus was cleaned from noises like punctuation marks, numbers, and other languages letters that are not the required characters to be predicted. As Balochi language is unstandardized the spell-checking of each word was done manually to avoid the ambiguity of the data and this is represented in Tab. 4. However, N-gram Model was used to analyze the frequency of the words. N-gram Model is beneficial as it depends on the frequency of terms. The results from the perplexity (intrinsic evaluation metric) and accuracy (extrinsic evaluation metric) showed that the accuracy of Five-gram model is much better than other models. It has a perplexity result of 240, and its accuracy is 93%.

This research determined that Five-gram is a suitable model for the system. Moreover, it can also generate new sentences that are reliable for real-world application**.**

6. Conclusion and Future Work

The word prediction implemented in the Balochi language with the help of the N-gram

model is an innovative study. In the area of NLP, the Balochi language hardly has any research. Further, the Balochi language does not have a large corpus to do a wide range of research. A corpus of 317590 words is created with no ambiguity. The N-gram model is used to build a word prediction system with that Balochi corpus. The N-gram model has shown promising results in predicting the next word in the Balochi language. To avoid the sparsity of data, Lidstone smoothing has been used, which improved perplexity has compared to Laplace smoothing. The trained model has been embedded in an application to achieve the model's accuracy. However, the result of the five-gram model is 93%, and Quad-gram has 80% of the prediction system.

The accuracy of the model mainly depends upon the training corpus. Therefore, we must have a non-ambiguous corpus to achieve a precise result. Nevertheless, the feature has brought ease in typing for native users; it will benefit Balochi language users.

Nevertheless, many other smoothing techniques, such as the Back-off model, Good turning model, KneserNey smoothing etc., are included in future work to check whether it gives better results in perplexity. Furthermore, a word prediction based on POS tagging can enhance the efficiency of the word prediction system. It can also be great to include the corpus of other Balochi dialects to broaden the service area. This research will help develop a plan to complete the sentence according to the grammar and correct the spelling.

REFERENCES

[1] C. Jahani, “A Grammar of Modern Standard Balochi Language,” in Acta Universitatis Upsaliensis, 1st ed., Upsala, Sweden, 2019.

[2] G. W. Lesher, B. J. Moulton and D. J. Higginbotham, "Effects of N-gram order and training text size on word prediction," In Proceedings of the RESNA'99 Annual Conference, pp. 52-54. 1999.

[3] M. Ghayoomi and E. Daroodi, "A POS-Based Word Prediction System for the Persian Language," in Advances in Natural Language Processing., Springer, Berlin, Heidelberg, 2008.

[4] S. Shahzadi, B. Fatima, K. Malik and S. M. Sarwar, "Urdu Word Prediction System for Mobile Phones," World Applied Sciences Journal, vol. 21, no.1, pp. 1260-1265, 2013.

[5] M. Saeed, A. Nawaz, K. Ahsan, S. Jabeen, K. Islam, M. Hassan and F. A. Siddiqui, "Effective Word Prediction in Urdu Language Using Stochastic Model," SJCMS, vol. 2, no.2, pp. 38-46, 2018.

[6] J. Mahar and G. Memon, "Probabilistic Analysis of Sindhi Word Prediction using N-Grams," Australian Journal of Basic and Applied Sciences, vol. 5, no.5, pp. 1137-1143, 2011.

[7] M. Haque, M. Habib and M. Rahman, "Automated Word Prediction in Bangla Language Using Stochastic Language Models," International Journal in Foundations of Computer Science & Technology (IJFCST), vol. 5, no.6, pp. 67-75, 2015.

[8] T. M. Habib, A. Al-Mamun, S. M. Rahman, S. M. T. Siddiquee and F. Ahmed, "An Exploratory Approach to Find a Novel Metric Based Optimum Language Model for Automatic Bangla Word Prediction," International Journal of Intelligent Systems and Applications, vol. 2, no.2, pp. 47-54, 2018.

[9] T. Mittra, L. Islam and D. C. Roy, "Prediction of Semantically Correct Bangla Words Using Stupid Backoff and Word-Embedding Model," in 2nd International Conference on Applied Information Technology and Innovation (ICAITI), Denpasar, Indonesia, pp. 66-70, 2019.

[10] N. Shah and N. Khetra, "Syntactic Word Prediction for Hindi," IJSART, vol. 3, no.3, pp. 1191-1195, 2017.

[11] R. Sharma, N. Goel, N. Aggarwal, P. Kaur and C. Prakash, "Next Word Prediction in Hindi Using Deep Learning Techniques," in International Conference on Data Science and Engineering (ICDSE), Patna, India, pp. 55-60, 2019.

[12] H. Hamarashid, S. Saeed and T. Rashid, "Next word prediction based on the N-gram model for Kurdish Sorani and Kurmanji," Neural Computing and Applications, vol. 33, no.5, pp. 4547-4566, 2020.

[13] Y. Hacohen-Kerner and I. Greenfield, "Basic Word Completion and Prediction for Hebrew," in String Processing and Information Retrieval, Springer, Berlin, Heidelberg, pp. 237-244, 2012.

[14] G. S. Mahi and A. Verma, "PURAN: Word Prediction System for Punjabi Language News," in Data Management, Analytics and Innovation, Proceedings of ICDMAI, Springer, Singapore, 2019, pp. 383-400.

[15] M. Herold, E. Alant and J. Bornman, "Typing speed, spelling accuracy, and the use of word-

prediction," South African Journal of Education, vol. 28, no.1, pp. 117-134, 2008.

[16] Nandini, P. Hamsaveni and P. Charunayana, "Hybrid Machine Learning based Kannada Next Word Prediction," International Research Journal of Engineering and Technology (IRJET), vol. 7, no.7, pp. 5605-5608, 2020.

[17] A. Pourmohammad, M. Gulami, J. Mahmudov, Y. Aliyev and R. Akberov, "The First Azeri (Azerbaijani) Language Next Word Predictor," Information Systems and Signal Processing Journal, vol. 5, no.1, pp. 1-4, 2020.

[18] A. Yazdani, R. Safdari, A. Golkar and S. R. N. Kalhori, "Words prediction based on N-gram model for free-text entry in electronic health records," Health Information Science and Systems, vol. 7, pp. 1-6, 2019.

[19] M. Parekh and Y. Patel, "Word Completion and Word Prediction using Probabilistic Model," 2019.

[20] M. P. Bhuyan and S. K. Sarma, "A Higher-Order N-gram Model to enhance automatic Word Prediction for Assamese sentences containing ambiguous Words," International Journal of Engineering and Advanced Technology (IJEAT), vol. 8, no.6, pp. 2921-2926, 2019.

[21] D. Nagalavi and M. Hanumanthappa, "A Model to Predict Words in the Sentence to Identify an Aligned Sequence of Article Blocks in e-Newspaper," International Journal of Advanced Research in Computer and Communication Engineering, vol. 5, no.2, pp. 160-164, 2016.

[22] S. Ghosh, H. S. Rana and R. Tomar, "Word Prediction using Collaborative Filtering Algorithm," IJCTA, vol. 9, no.22, pp. 115-122, 2016.

[23] R. Makkar, M. Kaur and D. V. Sharma, "Word Prediction Systems: A Survey," Advances in Computer Science and Information Technology (ACSIT), vol. 2, no.2, pp. 177-180, 2015.

[24] M. Bhuyan and S. Sarma, "An N-gram based model for predicting of word formation in Assamese language," Journal of Information and Optimization Sciences, vol. 14, no.2, pp. 427-440, 2019.

[25] W. Tesema and D. Tamirat, "Enhancing the Text Production and Assisting Disable Users in Developing Word Prediction and Completion in Afan Oromo," Journal of Information Technology & Software Engineering, vol. 7, no.2, pp. 1-4, 2017.

[26] R. Mahte, R. Nair, V. Nair, A. Pillai and P. M. Kulkarni, "Emoticon Suggestion with Word Prediction using Natural Language," International Research Journal of Engineering and Technology (IRJET), vol. 07, no.5, pp. 3104-3108, 2020.

[27] Y. Sharma, J. S. Bindra, K. Aggarwal and N. Dahiya, "Word Prediction and Sentence Completion," IJSRD - International Journal for Scientific Research & Development, vol. 7, no.3, pp. 744-747, 2019.

[28] K. C. Arnold, K. Z. Gajos and A. T. Kalai, "On Suggesting Phrases vs. Predicting for Mobile Text Composition," in Symposium on User Interface Software and Technology, Tokyo, Japan, pp. 603-608, 2016.

[29] G. J. Naseer, A. Basit, I. Ali and A. Iqbal, "Balochi Non Cursive Isolated Character Recognition using Deep Neural Network," International Journal of Advanced Computer Science and Applications, vol. 11, no.4, pp. 717-722, 2020.

[30] "Baask," [Online]. Available: http://baask.com/diwwan/index.php?Topic=4381.0.

[31] Z. Baloch, "بلوچی راست نبیسَی," Raeesi Chaap o Shing Jah, 2015.

[32] P. Kokaisl and P. Kokaislová, "The Ethnic Identity of Turkmenistan's Baloch," Asian Ethnology, vol. 78, no.1, pp. 181-196, 2019.

[33] S. Hashmi, "بلوچی سیاهگ ءِ راست نبیسی," in Sayad Hashmi Academy, Karachi, Pakistan, 1964.

[34] M. A. Badini, "بهشت ءُ دوزه," in New College Publication, Quetta, Pakistan, 2013.

[35] D. Jurafsky and J. H. Martin, "Speech and Language Processing: An introduction to natural language processing," in Prentice Hall, 1st ed., New Jersey, NJ, USA, 2000.

[36] C. Campagnola. "Perplexity in Language Models", 2020, [Online]. Available: https://towardsdatascience.com/perplexity-in-language-models-87a196019a94.