ISSN: 3102-0496 (Print)

Annual Journal of

Technology Innovation & Management

Padmashree COLLEGE
Tinkune, Kathmandu, Nepal
Tel: 01-4112252 |01-4112057
Email: padmashreecollege@gmail.com
URL: www.padmashreecollege.edu.np

Annual Journal of

Technology Innovation & Management

Volume 1	Issue 1	October 2025	ISSN: 3102-0496 (Print)

Editorial Board

Reviewers Committee

Tel: 01-4112252 |01-4112057 **Email:**padmashreecollege@gmail.com

Message from the Chairperson

It gives me great pleasure to present this significant initiative by Padmashree College. This reflects our commitment to fostering excellence in education, research, and innovation.

In today's dynamic world, it is essential to create platforms that encourage critical thinking and practical application of knowledge. This publication provides our academic community with an opportunity to share innovative ideas and contribute meaningfully to their respective fields. I sincerely thank all the contributors, reviewers, and the editorial team for their dedication in making this initiative a success. I am confident this will inspire our students and faculty to pursue excellence in their academic endeavors.

Warm regards,

Er.Arna Raj Silwal
Chairperson
Padmashree College, Tinkune, Kathmandu, Nepal

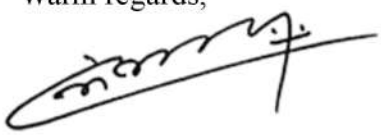
Message from the Editorial Chief

It is a pleasure to present the Annual Journal of Technology, Innovation and Management (AJTIM), Volume I, published by Padmashree College. This multidisciplinary journal offers a platform for sharing innovative ideas, research, and insights in technology, innovation, and management.

In today's rapidly evolving world, interdisciplinary research and innovation are essential for sustainable growth. This journal aims to foster academic excellence, collaboration, and knowledge exchange.

My sincere thanks to all contributors, reviewers, and the Padmashree College management for their support. I hope AJTIM inspires continued learning and innovation.

Warm regards,



Er. Ramesh Paudyal
Editorial Chief
Padmashree College, Tinkune, Kathmandu, Nepal

Network Anomaly Detection System Using Random Forest Algorithm

Prabhuram Karki, Prajwal Rai, Ramesh Paudyal
Nilai University, Malaysia
Padmashree College, Tinkune, Kathmandu
prabhu.bit_2022@padmashreecollege.edu.np

***Abstract*—In the modern digital world, network traffic is increasing rapidly not only in volume but also in diversity. Picking out unusual behavior in network traffic is now important for the security and normal operation of digital systems. The Network Anomaly Detection System Using Random Forest Algorithm (NADS) looks for behavior that seems out of the norm, as it sometimes appears in DDoS attacks, malware or unauthorized access matters. Network threats are no longer simple, so conventional tools cannot deal well with big and complicated datasets. This system uses Random Forest because it can handle such data well by combining different decision trees which helps minimize generalization errors and provides better accuracy. Data understanding, preprocesssing, feature transformation, model training, evaluation and deployment are the six steps of the CRISP-DM process used by NADS. Training and testing are done using the UNSW-NB15 dataset. With preprocessing, features are categorized, unnecessary observations are removed, class distribution is made equal with SMOTE and features are scaled. GridSearchCV is used to find the best settings for hyperparameters. The accuracy on the training data is 99.22% and the accuracy on the testing data is 90.40% and these results are further backed by high scores for precision, recall and F1. They prove that the system can tell apart regular behavior from unusual activity. Everything is saved using joblib so the model and pipeline fit together for use in real applications. Because cybersecurity infrastructure is so important in resource-poor regions, Nepal depends greatly on NADS. Because it monitors in real-time, produces reliable results and can grow with needs, the system helps set up a responsive and smart network security structure to handle new threats as they appear.**

***Index Terms*—Accuracy, CRISP-DM, Cybersecurity, F1 Score, GridSearchCV, Joblib, NADS, Network Anomaly Detection, Precision, Preprocessing, Random Forest, Recall, SMOTE, UNSW-NB15**

I. Introduction

Since digital networks are now used more and more for communication, business, and running governments, effective cybersecurity solutions are needed more than ever. Worries about cybercrimes such as data breaches, ransomware attacks, or DDoS incidents are rising because of unusual network traffic [1]. The digital infrastructure in Nepal is quickly improving as well. Recent events, including hacking of government sites and banks, have shown weaknesses in Nepal's IT security [2]. Currently, there is not enough infrastructure to keep an eye on and secure network traffic as it happens. Many companies are still protected by old, reactive security measures that allow threats to go unnoticed for some time, which makes them susceptible to malicious actions, stolen data, financial losses, and disrupted essential services. Moreover, having more connected things and using IoT has complicated networks, so older ways of watching traffic do not work as well [3].

Around the world, people are paying close attention to network anomaly detection as a key means to fight today's cybersecurity threats. Nowadays, the use of AI and machine learning (ML) has greatly improved how anomalies are both uncovered and addressed [4]. This means traffic monitoring systems can spot when something unusual occurs in real-time, allowing fast action against threats. Healthcare, banking, and governance, which require steady internet connections, are especially open to risks from poor infrastructure security. If there is a data breach in a healthcare system, important patient data might be put at risk, whereas a DDoS attack on financial institutions could stop important transactions and result in many people losing money [5].

Digital technologies are increasingly used in Nepal for communication, business, governance, and public services. Correspondingly, the associated challenges in cybersecurity have risen significantly. Advanced cybersecurity systems with constant monitoring and proactive measures are the ways through which these issues are being addressed in most developed countries. For example, the developed countries like the USA and Germany have imposed a robust security structure in the digital space, while Nepal is still struggling with old security tools and manual monitoring. According to reports, Nepal faces an increased rate of cyberattacks, instances are defacement of governmental websites and data breaches that have hurt businesses and sensible citizens. The cyberattacks continue to disturb systems, including phishing, ransomware, DDoS attacks, and unauthorized access. With the rapid growth of the usage of IoT in Nepal, vulnerabilities also increase since more entry points for attackers come with increased numbers of connected devices, further complicating the cyber landscape.

These cyberattacks in critical sectors of Nepal, such as healthcare, energy, and education, have grave consequences. For example, breaching a healthcare system might threaten sensitive data about patients, while attacks on energy infrastructure disrupt essential services. All the while, this ever-heightening threat of cybercrime has been chasing away foreign investment-one thing that directly hurts Nepal's digital

economy. As digital financial services and online commerce expand, there is also growing erosion of trust driven by cyber-attacks. For SMEs in Nepal, well-endowed with resources to deploy advanced cybersecurity solutions, cybercriminals find easy victims. In addition, the shortages of skilled cybersecurity professionals increase the challenge of organizations' vulnerabilities to such evolving threats.

The Network Traffic Anomaly Detection System using a Random Forest algorithm will, be able to make organizations of all sizes move with the evolving threats and provide scalable solutions. Unlike traditional security tools, this will use machine learning from historic data for continuous improvement, hence the identification of new attack vectors will be quicker. It will strengthen the cybersecurity awareness and build resilience in Nepal's digital ecosystem by automating threat detection, offering safer and more reliable online services, enhancing overall trust in the growing digital economy of Nepal.

II. Related works

Many researchers have explored machine learning approaches to detect unusual patterns in network traffic. A particularly popular objective has been to create systems capable of real-time threat detection, handling high volumes of data, and minimizing false positives. Ensemble techniques such as Random Forest have received extensive research on anomaly detection in cybersecurity. To illustrate, Kaur and Singh [6] demonstrated the effectiveness of Random Forest compared to such classifiers as SVM and Naive Bayes in accuracy and detection rate when using the NSL-KDD dataset. In the same manner, Thomas et al. [7] evaluated various machine learning models on the UNSW-NB15 dataset and reported that Random Forest provided good performance uniformly across the network attack types.

Other methods have used Random Forest together with feature selection techniques to enhance detection. Patel et al. [8] preprocessed their data by a correlation-based feature selection followed by Random Forest which yielded a higher efficiency and less time on training. This concept is equal to the application of preprocessing and transformation to enhance performance. Network anomaly detection has also been done using deep learning techniques like LSTM and CNNs. As an example, Yin et al. [9] applied LSTM networks to KDD data to perform sequence-based intrusion detection. Although deep learning gives decent accuracy, it is usually more computationally demanding and more difficult to deploy in real-time or resource-limited environments such as in Nepal. Hybrid models that mix various classifiers also exist. As another instance, Zhang et al. [10] used Random Forest together with K-means to cluster the data and subsequently classify it. Although these techniques can be used to improve the accuracy of detection, they make it more complex. Others have worked on real-time deployment and systems. Mohammadi et al. [11] implemented a real-time intrusion detection system based on Random Forest and Apache Kafka, demonstrating that live traffic monitoring can be integrated with machine learning. Majority of the past works have been tried in well resourced settings. Nonetheless, little research has revolved around implementing such models in low-resource settings or areas such as Nepal, where real-time processing, ease of implementation, and scalability are of primary concern.The system enhances previous work by including a full CRISP-DM pipeline, optimizing with GridSearchCV, balancing class representation with SMOTE, and saving the model with joblib to be used in the real world. It is meant to be precise and feasible in the developing world.

III. Methodology

A. Problem Understanding

Nepal is experiencing a boom in adoption of digital technologies in areas such as communication, online services, governmental activities and finance. However, as this has taken shape, cyber attacks such as website defacement, data leaks, ransomware, phishing and DDoS attacks have also increased. The rest of small business owners use manual observation and outdated security programs, which puts them at risks of advanced threats. Nepal lacks such tools, and there are no trained professionals in the field related to cybersecurity, although, in contrast to Nepal, such states as the USA and Germany use modern and automated security systems. Hence, small and medium sized enterprises (SMEs) are experiencing an issue of network security. The critical infrastructure including healthcare, education, and energy are especially vulnerable sectors. To give an example, a cyberattack on a hospital can violate the privacy of patients, and attacks on an energy plant can cause the suspension of essential local services. These risks reduce the trust of people in the online platform and in foreign investment.

The solution to this is to have a system which can identify abnormal or suspicious activities on the network in real time. Machine learning is a decent solution since it is not based on set in stone rules. It improves with time and learns on past data, this makes it better than the traditional methods of identifying various forms of attacks. Machine learning algorithm Random Forest is chosen due to high accuracy rates, speed, and suitability to work with large, structured data. The system has the ability to react swiftly to dangers by using real-time information along with offline logs. The aim is to develop a safe, expandable, and comfortable system of the Nepalese population. It must assist any kind of organization, either large or small to enhance their network security and have awareness of the current cybersecurity practice. The strategy will assist in minimizing cyber risks and providing additional protection to the Nepali digital environment that is being developed.

B. Data Understanding

Data Understanding is all about carefully analyzing the dataset to prepare it for use in building a cybersecurity model. In this project, the UNSW-NB15 dataset is chosen. It contains normal and malicious network traffic, and it is split into training set (used to train the model) and testing set (used to test the model on new data). The dataset is first examined to get an idea about its volume, shape, and characteristics. It contains such

information as packet size, protocol type, service type, and connection time. They can be either numeric or categorical and the knowledge of feature types allows to determine how it will be further processed.Statistical summaries of features are computed (minimum, maximum, average, and median values) to get an overview of general trends in data. Patterns and outliers (such as very large packets or extended connection times which may be associated with a cyberattack) are identified using visualization tools such as histograms and box plots. An important observation is that the data is imbalanced, containing a large number of normal records compared to malicious ones. Such imbalance may mislead the model so that it becomes accurate on normal traffic but misses most of the attack instances. The identification of this issue at this stage enables to plan how to address it at later stages

Missing or wrong values are also vital to look at as they may damage model training. Intending to clean or repair those makes the data reliable. Correlation analysis is also utilized to research the relationship between features. features Some features might be too similar or not useful and can be dropped to get better model performance. All in all, this step is useful to gain insights into the data, uncover difficulties, and outline the process that should be taken during the features selection and cleaning. It establishes a good foundation to develop an intelligent and precise system to identify cyber threats.

C. Data Preparation

Data Preparation is a crucial step to make the dataset ready for training a machine learning model. Missing values, irrelevant columns, and mixed formats are some of the issues that might be present in the raw data and lower the model performance once they are not correctly addressed. First is the data cleaning exercise where mistakes, missing values, and columns with ID numbers are cleaned or removed as they do not aid in network anomaly detection and instead add noise. Categorical features like protocol type or service name are then one-hot encoded after cleaning. This technique converts each category into individual columns of binary values (0 or 1) and simplifies the data interpretation by the model. The significant problem with the data is the imbalance between classes there are many more normal traffic records than attack records. It may lead to the situation when a model disregards a minority class, in which case cyber threats are found.

As a solution, those synthetic samples of the minority class are created with the SMOTE technique to balance the dataset and allow the model to learn the attack patterns more effectively. After balancing the features are scaled with StandardScaler which transforms all the numeric features to the same range with mean zero and standard deviation one. This is to make sure that large values features do not dominate smaller ones during training of the model. Lastly, the completely ready data, cleaned, encoded, balanced, and scaled is divided into training and testing data. The model is constructed by training on the training set and the performance of the model on unseen data is tested on the testing set. In short, the Data Preparation stage consists of all processes required to transform raw network traffic data into clean, balanced, and well-structured format applicable to effective and precise anomaly detection in cybersecurity systems.

D. Modeling

Modeling focuses on using the prepared data to build and train a machine learning model that can detect unusual activities in network traffic. Random Forest Classifier is a powerful, accurate, and reliable tool which is selected to perform this task. It operates by growing numerous decision trees and averaging their outcomes, which is useful in minimizing errors and enhancing stability. To arrange the steps in the right order and prevent the usual errors such as data leakage, a pipeline is involved. SMOTE is used in conjunction with pipeline internal data balancing (to generate synthetic attacks) to ensure that the model does not get biased towards normal traffic. The Random Forest model is fitted on the balanced data. The hyperparameters of the model are optimized with the help of GridSearchCV in order to achieve better results. This procedure automatically experiments with different settings combinations based on cross-validation that divides the training data into portions and examines the performance of the model on each portion. This prevents overfitting, in which the model fits the training data but underfits new data.

After the optimal settings have been identified, the model is applied on a different set of data to determine its ability to identify actual network threats. This is done by several performance measures, including accuracy (was it overall correct), precision (of the attacks that were identified, how many were actually real), recall (of the real attacks, how many were identified), F1-score (balance between precision and recall), and ROC-AUC score (was it able to distinguish between normal and attack traffic). Also a confusion matrix is plotted to demonstrate how many predictions were correct or incorrect. Moreover, feature importance chart is employed to determine features that played the major role in model decisions. All in all, the Modeling stage uses an ensemble of Random Forest, SMOTE, parameter optimization, and evaluation to create a robust and precise system to identify cyber threats. **Equations:**

Gini Impurity

$$Gini(t) = 1 - \sum_{i=1}^{C} p_i^2$$

Majority Voting

$$\hat{y} = \text{mode}\{T_1(x), T_2(x), \ldots, T_N(x)\}$$

Feature Importance

$$Importance(f) = \frac{1}{N} \sum_{i=1}^{N} \sum_{j \in S(f,T_i)} \Delta Impurity_j$$

E. Evaluation

Evaluation is essential to check how well the machine learning model performs after training. This is the step that evaluates the model on a different dataset that the model has never been exposed to, and aids in testing whether the model is able

to generalize and perform well on real network traffic. The model is not overfitting (that is, learning useful patterns rather than remembering training data) by testing on new data. In order to assess the performance of the model, a number of evaluation measures are applied. Although accuracy gives the overall percentage of correct predictions, it is not sufficient on its own since the dataset is unbalanced, containing a large amount of normal traffic compared to attacks. A model might categorise all as normal and achieve high accuracy but miss threats. This is the reason why such measures as precision, recall, and F1-score also matter. Precision tells the number of predicted attacks that were actually correct, whereas recall tells the number of actual attacks that the model managed to detect. F1-score provides a balanced perspective as it averages precision and recall.

It is also presented in the form of a confusion matrix indicating the true positives, true negatives, false positives, and false negatives. This will aid in realizing in what senses the model is going wrong. As an illustration, a false positive can result in unnecessary notifications whereas a false negative implies that real attacks are being overlooked. The other important metric is the ROC curve with its AUC (Area Under the Curve) score that demonstrates the capability of the model to distinguish between normal and attack traffic. AUC was higher, and this indicates better performance. Also, feature importance is checked during the evaluation stage to see which data features had the biggest impact on the decisions made by the model. When the model results are not satisfactory enough, it is possible to repeat the previous stages to improve it. All together, the Evaluation phase will provide accuracy, balance and efficiency of the model in identifying network threats prior to real world implementation.

F. Deployment

Deployment is where the network anomaly detection system is prepared to work in real-time environments, moving the project from testing to practical use. During it, the trained model as well as such crucial parts of it as feature scalers and data processing steps are stored with the help of the library named Joblib so that to be able to reuse the system without retraining. This system is then connected to live network data via a tool known as Scapy which captures real time traffic and extracts features such as the training data. This live data is processed by the saved model to classify normal or malicious traffic quickly in order to respond sooner to cyber threats. Networks generate a lot of data and so, the system is set in such a way that it functions effectively without choking or skipping on relevant information. It is also important to continuously monitor the performance of the system implemented since cyber threats adapt over time and may need an update or retraining to maintain the model useful. In order to make the system usable by non-experts, alerts and reports are produced to notify network administrators about possible threats, and dashboards may offer great details about network behavior and model choices. Real network use feedback can be gathered to further refine the model. In general, the Deployment phase combines all components of the project data capture, machine learning, and real-time monitoring into one whole, useful, and effective structure that is capable of defending networks against cyberattacks in real environments.

IV. RESULT AND DISCUSSION

The Network Anomaly Detection System (NADS) was evaluated using multiple important performance metrics to measure how well it detects unusual activities in network traffic. There are several significant performance measures that were applied to the Network Anomaly Detection System (NADS) to quantify its ability to identify anomalous activities in network traffic. Such measures were accuracy, precision, recall, F1-score, confusion matrix, and ROC-AUC. The model shows outstanding performance during the training phase, with an accuracy of 99.22 percent. It showed extremely high precision and recall on the normal and anomalous classes of traffic with values approximately equal to 0.99. The F1-scores of both classes were also very near to 0.99, showing that the detection system is well balanced and shows little bias towards the classes. That is, this model was able to achieve near perfect classification of both normal and anomalous network behaviors with a very low false positive rate. This false positive rate is important in cybersecurity so as not to flood system administrators with false positives.

The system was tested using job descriptions and resumes from various domains. It successfully calculated match percentages using TF-IDF and cosine similarity, with higher scores indicating better alignment. A confusion matrix was used to evaluate the accuracy of the classification into High, Medium, and Low matches, showing strong performance, especially for High matches. A class distribution graph confirmed balanced data in training and testing phases. HR feedback confirmed the system's ability to reduce screening time and improve candidate-job relevance. Compared to keyword-based systems, the NLP-based model provided more accurate and meaningful matching results.

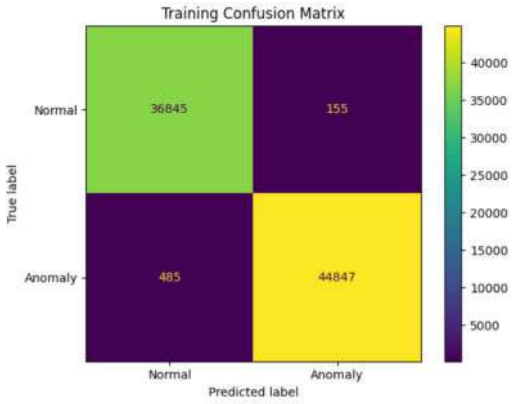


Fig. 1. Confusion Matrix for the Training Set

When tested on new, unseen data, the model's accuracy dropped to 90.40%. Such drop is common in machine learning

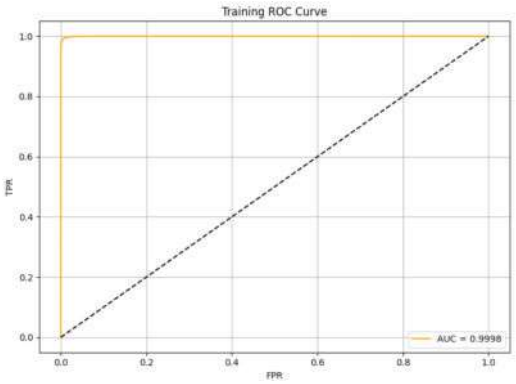


Fig. 2. ROC Curve for the Training Set

models when they encounter novel and imbalanced attack types. Nonetheless, the model was still useful, particularly in anomaly detection, where it attained a high precision of 0.9884 and recall of 0.8692. This indicates that it may miss some normal traffic cases but is faithful in identifying abnormal or possibly malicious behaviors.

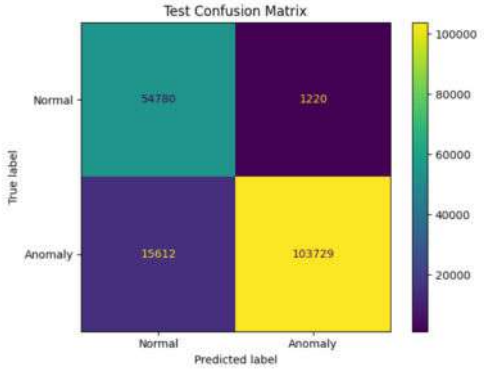


Fig. 3. Confusion Matrix

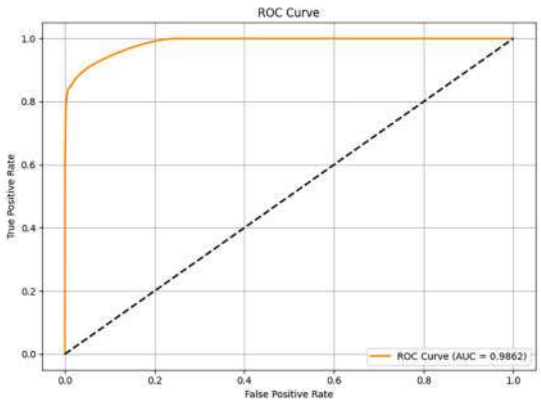


Fig. 4. Confusion Matrix

The imbalanced classes problems were commonly observed in network data, and the SMOTE allowed improving the learning of the model by creating the synthetic examples of the minority class. The ensemble character of the Random Forest algorithm made the predictions more robust and stable, and the hyperparameters were adjusted to the optimal state with the help of GridSearchCV.

Based on the testing, it can be concluded that NADS is a reliable and feasible system of real-time network monitoring and anomaly detection. It is characterised by a high accuracy and low false positive rate, which means that it can be deployed, particularly in resource constrained environments, such as in Nepal, where formidable cybersecurity solutions are long overdue. Model updates and regular monitoring will allow sustaining its efficiency against ever-changing cyber threats.

TABLE I
RESULTS SUMMARY

Metric	Training Set	Test Set
Accuracy	99.22%	90.40%
Precision	0.9923	0.9213
Recall	0.9922	0.9040
F1-score	0.9922	0.9064
Total Samples	82,332	175,341

The results of this study show that the Network Anomaly Detection System (NADS), which uses the Random Forest algorithm, is highly effective in detecting abnormal network activities. The system has shown outstanding performance during training with an accuracy of 99.22%, and all the other metrics precision, recall, and F1-score were nearly equal to 1. This is a good thing because it indicates that the model garnered much on the training data and was able to distinguish between normal and abnormal behavior with minimal errors. But on the test of unobserved data, the accuracy of the model decreased to 90.40 per cent. It is a standard problem in machine learning since tests data usually contains novel patterns, or types of attacks that the model has never encountered. Despite that, the model still demonstrated good results, particularly in detecting anomalies, which is the primary objective of any anomaly detection system.

V. CONCLUSION AND FUTURE WORK

In this project, a Network Anomaly Detection System (NADS) was developed using the Random Forest algorithm to detect abnormal patterns in network traffic. The system uses CRISP-DM process that consists of data understanding, preprocessing, transformation, model training, evaluation, and deployment. It was trained and tested on a famous network intrusion detection dataset UNSW-NB15. These findings indicate that the system works quite well as the training accuracy is 99.22 percent and the testing accuracy is 90.40 percent. It also attained high scores in precision, recall and F1-score which demonstrated that it is capable of identifying both normal and abnormal activities. Although the precision marginally decreased when put to test, the performance was still good, particularly on detecting unknown attacks. SMOTE made the classes in the data more balanced, and GridSearchCV was significant in determining the optimal model parameters.

This system can be particularly helpful to the countries such as Nepal in which the cybersecurity infrastructure is in its early stages. With the machine learning, there is a possibility of the system learning through the previous data and evolving with the new threats overtime. It can operate in real time to provide quick warnings whenever an occurrence of something fishy occurs in the network. Moreover, the model is stored and can be loaded back to the real applications without retraining it completely. Conclusively, NADS is a high fidelity and scale able solution to network anomaly detection. It may be a significant part of a robust and intelligent cyber security system of the future with frequent updates and monitoring.

To improve the effectiveness of the system, several advanced strategies can be applied. Firstly, real-time data streaming with the assistance of such tools as Apache Kafka or Apache Flink will allow the system to track the changed data in real-time and identify the problems in the network fast. The unnecessary features included in the dataset should be dropped using Principal Component Analysis (PCA) that can enhance the model performance as it will be trained on the relevant data only. By including deep learning methods like Long Short-Term Memory (LSTM) network and autoencoders, the system can learn complicated attack patterns that shall otherwise be overlooked by the traditional model. One should also consider retraining the model on a regular basis with new data so that it is updated and reflectsive on new threats. The system must also be made to learn and evolve itself becoming more capable of dealing with unknown cyber threats. The procedure of data preparation should be portable and at the same time capable of handling big data in varying environments. It is possible to enhance the quality of anomaly detection by exploring a combination of the conventional machine learning and deep learning strategies. Last but not least, enhanced visualization and alert systems will provide cybersecurity experts with a better and quicker understanding of threats, enabling them to react more efficiently to security incidents.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to Nilai University for providing me with the platform and resources to carry out this project, titled Network Anomaly Detection System Using Random Forest Algorithm (NADS). I could not have done this without the constant support and guidance of my supervisor Mr. Prajwal Rai, whose professional help, positive criticism and valuable recommendations have made the research much better and more aligned. I would also like to warmly appreciate Mr. Ramesh Paudyal, Module Leader of Padmashree College, who helped me with academic support and valuable guidance during the project period. His advice was especially useful at the beginning of the problem selection and research design. I would also like to thank very much to all the faculty members and administrative staff of both institutions who assisted me directly or indirectly by providing a stimulating and encouraging research environment. Finally, I would like to pay extra attention to my family and friends who motivated me, supported me emotionally, and showed patience throughout this whole process. It is because of their infinite faith in me and my capabilities that I was able to work on this piece with determination and utmost concentration. My thanks go to all those who in any way contributed to the success of completion of this project, visible or invisible.

REFERENCES

[1] P. Srinoy, "Anomaly detection in network traffic using machine learning," Int. J. Comput. Appl., vol. 177, no. 18, pp. 1–5, 2020.

[2] handari, "Cybersecurity challenges in Nepal: A growing concern," Nepal J. ICT, vol. 6, no. 1, pp. 34–41, 2022.
[3] [4] Goodfellow, Y. Bengio, and A. Courville, Deep learning, MIT Press, 2016. [5] [6] Kaur and M. Singh, "An efficient random forest approach for intrusion detection,"Int. J. Comput. Sci. Netw. Secur., vol. 17, no. 6, pp. 92–97, 2017. [7] [8] . [9] [10] [11]

[3] W. Stallings, Foundations of modern networking: SDN, NFV, QoE, IoT, and cloud, Pearson Education, 2021.

[4] Goodfellow, Y. Bengio, and A. Courville, Deep learning, MIT Press, 2016.

[5] R. Kumar, S. Singh, and P. K. Sharma, "Machine learning-based framework for cyber threat detection in IoT-enabled e-healthcare," Comput. Commun., vol. 151, pp. 231–239, 2019.

[6] Kaur and M. Singh, "An efficient random forest approach for intrusion detection,"Int. J. Comput. Sci. Netw. Secur., vol. 17, no. 6, pp. 92–97, 2017.

[7] S. Thomas, A. K. Sinha, and B. S. Rajpurohit, "Evaluation of machine learning techniques for intrusion detection system," Procedia Comput. Sci., vol. 132, pp. 928–935, 2018.

[8] J. Patel, D. Patel, and S. Patel, "An efficient feature selection method for intrusion detection system using random forest," Int. J. Comput. Appl., vol. 155, no. 3, pp. 21–24, 2016

[9] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," IEEE Access, vol. 5, pp. 21954–21961, 2017.

[10] X. Zhang, Z. Wang, and Q. Xue, "A hybrid network intrusion detection method using random forest and K-means clustering," J. Intell. Fuzzy Syst., vol. 35, no. 3, pp. 3103–3111, 2018.

[11] M. Mohammadi, A. Al-Fuqaha, and M. Guizani, "Real-time intrusion detection using random forest classifier in SDN environment," IEEE Commun. Mag., vol. 57, no. 7, pp. 160–165, 2019

Resume and Job Requirements Matching System using NLP

Binita Ghimre
Nilai University, Malaysia
Padmashree College, Tinkune, Kathmandu
binita.bit_2022@padmashreecollege.edu.np

***Abstract*—Finding the appropriate candidate for the right job is still a constant issue in today's digital recruitment environment. Manually screening resumes takes a lot of time, is prone to human bias, and is unreliable, particularly in underdeveloped countries where traditional practices are still used, like Nepal. This study introduces an automated system for matching resumes with job requirements that uses natural language processing (NLP) to improve hiring procedures. The system uses cosine similarity for similarity scoring, feature extraction by TF-IDF vectorization, and text preparation to assess the contextual relevance between resumes and job descriptions. Recruiters can enter resumes and job descriptions using a web-based interface, which returns graded results to help them make better decisions. Experiments on a variety of datasets showed a strong ability to find the best candidates with the fewest false positives and a high precision of 98%. With future prospects including the use of sophisticated embeddings like BERT for improved semantic understanding, the system provides a scalable, objective, and effective method of resume screening.**

***Index Terms*—Resume Screening, Job Matching, Natural Language Processing (NLP), Cosine Similarity**

I. Introduction

Organizations get hundreds of resumes for a single job vacancy in today's competitive job market, making it difficult for enrollment specialists to effectively identify the most qualified applicants. At the same time, job seekers frequently apply for a large number of openings without fully knowing how well their skills match the requirements of the job. The recruitment and recruitment environment in Nepal presents a number of difficulties for organizations. In order to efficiently find the right talent, organizations must undertake the enormous work of sifting through the resumes of hundreds, thousands, or millions of applicants. The drawback of reviewing applications is not the volume of applicants, but rather the fact that social media, online job sites, etc. Professional networking sites make it easy to find prospects, and manually analyzing applications is not only time consuming but also expensive. Therefore, many recruiters have sought to automate the process of attracting potential candidates and implementing additional testing procedures in order to streamline the hiring process and identify the most promising prospects. Basic keyword matching is the basis of current online job portals, but it is unable to capture the complex context of candidate qualifications. Effective screening is further complicated by inconsistent resume forms, a variety of terminology, and human review that is subjective[1]. This study suggests an automated system for matching resumes and jobs that compares resumes and job descriptions semantically using natural language processing (NLP). To generate interpretable match scores, the method makes use of cosine similarity, TF-IDF-based feature extraction, and sophisticated text preparation. HR managers may increase fairness, save manual labor, and expedite candidate screening with the help of a web-based interface.

Both companies and job seekers have opportunities and challenges in the contemporary labor market. Even though companies get a ton of applications for every job posting, it is still inefficient and frequently unproductive to find the best applicants. In addition to being time-consuming, manual resume screening is prone to bias, mistake, and inconsistency. Conventional keyword-based algorithms sometimes overlook potentially highly qualified applicants because they are unable to capture the semantic importance of candidate qualifications. Additionally, because resumes are unstructured and differ widely in terms of format, vocabulary, and style, it is challenging for recruiters to accurately and fairly compare candidates. These restrictions result in longer hiring processes, placements that aren't a good fit, higher search expenses, and hiring decisions that aren't inclusive. The created Resume and Job Requirements Matching System uses cosine similarity and Natural Language Processing (NLP) to provide an automated solution to these problems. Tokenization, lemmatization, and named entity identification are some of the methods the system uses to extract and preprocess unstructured text from resumes and job descriptions. Instead of focusing on surface-level keyword overlap, it determines the best-aligned candidates by calculating match percentages using TF-IDF vectorization and cosine similarity.

The solution dramatically lowers the workload of recruiters, increases matching accuracy, and guarantees a more impartial and equitable assessment of prospects by automating the resume screening process. It is particularly helpful in places like Nepal, where hiring procedures are still mostly done by hand and many organizations do not have access to sophisticated recruitment tools. By providing feedback on resume quality and conformity with job market demands, the system also helps job searchers. This strategy effectively addresses the drawbacks of traditional recruitment techniques while promoting more accurate, efficient, and inclusive hiring

outcomes.

II. Related works

Bhoir and colleagues proposed a hybrid resume parsing solution that combines SpaCy's NLP capabilities with BERT's contextual understanding. This integration enhances the extraction of relevant details from unstructured resumes, addressing challenges posed by non-standardized formats. Their study also explores the potential of parsing video resumes through visual and audio processing techniques [2].

Kashif and Kumar introduced an AI-based resume analyzer that leverages NLP and machine learning to extract and analyze resume content. A notable feature of their system is the provision of real-time feedback, allowing applicants to refine their resumes based on suggested improvements. This tool aids both applicants in enhancing their resumes and administrators in making data-informed recruitment decisions [3].

Bhoir and colleagues proposed a hybrid resume parsing solution that combines SpaCy's NLP capabilities with BERT's contextual understanding. This integration enhances the extraction of relevant details from unstructured resumes, addressing challenges posed by non-standardized formats. Their study also explores the potential of parsing video resumes through visual and audio processing techniques [4].

Khan and his team developed a system that parses and summarizes resumes, converting unstructured data into structured formats. By employing NLP and machine learning techniques, their approach enhances the extraction of information related to education, skills, and work experience. The system supports multiple document formats and aims to improve the efficiency of recruitment processes by reducing the time recruiters spend on manual resume screening [5].

III. Methodology

To address the limitations of traditional recruitment systems, this project introduces an intelligent Resume and Job Matching System that applies Natural Language Processing techniques to match job descriptions with resumes based on contextual relevance. The system is built to extract and analyze the textual content of both resumes and job postings and determine how well they align. The approach starts with thorough preprocessing of text to clean and normalize the data, followed by feature extraction using TF-IDF vectorization, which converts textual information into numerical form while considering the importance of words within the documents. These vectorized representations are then compared using cosine similarity to calculate how closely a resume matches a given job description. The matching results are presented in a user-friendly web interface developed using Flask, which allows users to upload job descriptions and resumes ranking and view the top matching results.

A. Data Collection

- Cropping to focus on individual waste items and remove unnecessary background information.
- Normalization to standardize image dimensions and pixel values.
- Noise Reduction to eliminate distortions that might hinder accurate classification

B. Data set Description

Resumes and job descriptions were collected from diverse industries in Nepal. Documents in PDF and text format were converted to plain text.

C. Text Preprocessing

The first and most critical step in the NLP pipeline is data cleaning and preprocessing. Resumes and job descriptions are typically in unstructured or semi-structured textual formats, often filled with irrelevant information such as special characters, inconsistent capitalization, and stop words. In this stage, the system removes HTML tags, punctuation marks, and stop words (e.g., "the", "and", "is") that do not contribute to the context of matching. All text is converted to lowercase to maintain consistency. . To address this, the text data underwent the following preprocessing steps:

- Lowercasing: All characters were converted to lowercase to maintain uniformity.
- Special Character Removal: Non-alphanumeric symbols and punctuations were stripped using regular expressions.
- Stopword Removal: Common English stopwords (e.g., "the", "and", "is") were removed using the NLTK corpus.
- Lemmatization: Words were reduced to their base forms using the WordNetLemmatizer.

These steps were encapsulated in a custom function applied to both resume and job description fields to generate cleaned text columns for downstream processing.

D. Feature Extraction Using TF-IDF

To convert textual data into a machine-readable numerical format, the Term Frequency–Inverse Document Frequency (TF-IDF) vectorization technique was employed. Once the text is cleaned and normalized, the next step is to extract features that can quantitatively represent the semantic content of resumes and job descriptions. For this project, the Term Frequency-Inverse Document Frequency (TF-IDF) technique is used. TF-IDF identifies the importance of a word in a document relative to a collection of documents, which helps highlight relevant keywords and phrases. Both resumes and job descriptions are transformed into TF-IDF vectors, which are high-dimensional numerical representations. This conversion allows the system to measure the similarity between documents using mathematical techniques. TF-IDF quantifies the importance of a word in a document relative to a collection of documents, thereby highlighting domain-relevant terms while down-weighting common terms.

To represent text numerically, TF-IDF is used. The Term Frequency-Inverse Document Frequency (TF-IDF) metric evaluates the significance of a term t in a document d. It is defined as:

$$TF - IDF(t, d) = TF(t, d) \times IDF(t)$$

Where,

$$TF(t,d) = \frac{f_{t,d}}{\sum_k f_{k,d}}$$

$$IDF(t) = \log\left(\frac{N}{1+|\{d \in D : t \in d\}|}\right)$$

The TfIdf Vectorizer from the Scikit-learn library was used to:

- Fit and transform the cleaned resume texts.
- Transform job descriptions using the same vocabulary learned from resumes.

This yielded two sets of vectors one for resumes and another for job descriptions positioned in the same vector space.

Here,

- $f_{t,d}$ = Frequency of term t in document d,
- N = total number of documents
- Denominator ensures smoothing to avoid division by zero.

E. Similarity Computation

To evaluate the semantic closeness between each resume and a given job description, Cosine Similarity was used. Cosine Similarity measures the cosine of the angle between two vectors in high-dimensional space, resulting in a value between 0 (completely dissimilar) and 1 (identical in direction). The cosine_similarity() function from Scikit-learn's metrics.pairwise module was used to compute a similarity matrix between each job vector and all resume vectors.

F. Resume Ranking

Once the cosine similarity scores between each job description and all available resumes were computed, the next phase involved ranking the resumes based on their relevance. This step translates numerical similarity scores into actionable insights by organizing the resumes in a prioritized order of fit for a given job.

The resume ranking process is crucial as it transforms the abstract similarity matrix into a decision-support mechanism for recruiters. The following steps were carried out systematically:

1) Extraction of Similarity Scores: Once the similarity matrix S $\in \mathbb{R}^{m \times n}$ is generated—where m represents the number of job descriptions and n represents the number of resumes—the next step is to extract the similarity scores for each job description individually.
2) Sorting in Descending Order: The extracted similarity scores were then sorted in descending order to identify which resumes had the highest degree of similarity with the job description. This sorted list preserved the index positions of the resumes, enabling traceability.
3) Selection of Top-K Resumes: From the sorted list, the top-K resumes with the highest similarity scores were selected. The value of KKK was configurable based on the number of candidates a recruiter wished to shortlist.
4) Presentation of Ranked Results: The shortlisted resumes were displayed along with their respective similarity scores. This output allowed users to make data-driven decisions in selecting candidates, effectively reducing the manual screening workload.

G. Result Generation and Evaluation

The final output includes a ranked list of resumes per job description, along with their respective similarity scores. This list allows recruiters to quickly identify the most relevant candidates. Although the system currently relies on lexical similarity, it sets a foundation for incorporating more sophisticated semantic models in future work (e.g., BERT-based embeddings).

H. Model Deployment

The system was implemented using Flask for the web interface and containerized using Docker for scalability. Recruiters can upload resumes in PDF/TXT formats, and results are displayed with similarity percentages.

The final model, after successful validation and evaluation, was deployed as a web-based application to facilitate real-time resume and job description matching. The deployment process involved integrating the NLP pipeline and similarity matching algorithm within a user-friendly interface developed using the Flask web framework. This interface allows recruiters or hiring managers to input a single job description and upload multiple resumes in PDF or text format.Upon receiving input, the system performs text extraction and preprocessing, followed by TF-IDF vectorization and cosine similarity computation between the job description and each resume. The matching scores are then calculated and presented as a percentage for each resume, enabling users to quickly interpret and rank candidates based on contextual relevance. The results are displayed in a sorted list, from the highest to the lowest matching percentage, simplifying the candidate screening process.

The model was containerized using Docker to ensure scalability, portability, and ease of deployment across different environments. This architecture enables the system to be deployed on cloud platforms or on-premises servers, making it adaptable to various organizational infrastructures. Furthermore, the modular design ensures that updates, such as incorporating new NLP techniques or adjusting scoring algorithms, can be applied without disrupting the overall system functionality. By deploying the system as a web service, it becomes accessible to a broader audience, offering a practical and efficient tool to enhance the recruitment process through intelligent automation.

IV. RESULT AND DISCUSSION

The performance of the proposed Resume and Job Requirements Matching System was evaluated using a labeled dataset split into training and testing sets. The primary objective was to classify whether a given resume matches a job description using natural language processing techniques. The results are analyzed through a confusion matrix and class distribution plots.

The system was tested using job descriptions and resumes from various domains. It successfully calculated match percentages using TF-IDF and cosine similarity, with higher

scores indicating better alignment. A confusion matrix was used to evaluate the accuracy of the classification into High, Medium, and Low matches, showing strong performance, especially for High matches. A class distribution graph confirmed balanced data in training and testing phases. HR feedback confirmed the system's ability to reduce screening time and improve candidate-job relevance. Compared to keyword-based systems, the NLP-based model provided more accurate and meaningful matching results.

A. Confusion matrix

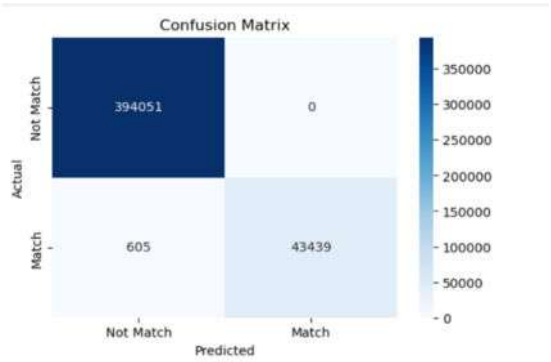


Fig. 1. Confusion Matrix

In the above figure presents the confusion matrix for the model's predictions on the test dataset. The matrix summarizes the number of correct and incorrect predictions made by the classifier compared to the actual labels. The following observations can be made:

- True Negatives (TN): 394,051 instances were correctly predicted as "Not Match".
- False Positives (FP): 0 instances of "Not Match" were incorrectly predicted as "Match".
- False Negatives (FN): 605 actual "Match" cases were incorrectly predicted as "Not Match".
- True Positives (TP): 43,439 instances were correctly predicted as "Match".

These results indicate that the classifier achieved high precision with zero false positives, meaning it did not incorrectly classify any non-matching resumes as matches. However, a small number of false negatives suggest the model missed a few correct matches, which slightly affects the recall.

This performance is particularly important in a real-world job matching system, where false positives can lead to recommending irrelevant resumes to recruiters, whereas false negatives may result in overlooking potential candidates. The near-zero false positive rate achieved here supports the system's reliability and conservativeness in suggesting matches.

B. Class Distribution in Training and Testing Sets

Above figure illustrates the class distribution for both the training and testing datasets. A clear class imbalance is observed, where the majority class is "Not Match" (labeled as 0), and the minority class is "Match" (labeled as 1). Specifically, the

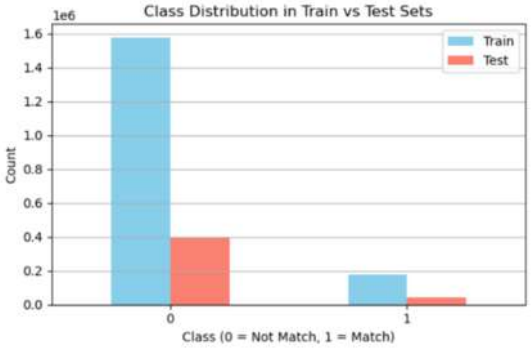


Fig. 2. Class Distribution in Training and Testing Sets

training set includes over 1.5 million "Not Match" instances, in contrast to fewer than 200,000 "Match" instances. A similar pattern is seen in the test set. This imbalance poses a challenge for the classification model, as it can bias predictions toward the majority class. However, despite this skew, the confusion matrix suggests that the model generalizes well and maintains high predictive performance across both classes. This success may be attributed to appropriate handling strategies during training, such as class weighting or sampling methods, and the robustness of the NLP-based feature engineering used in the system.

V. CONCLUSION AND FUTURE WORK

The developed Resume and Job Matching System, which integrates Natural Language Processing techniques such as TF-IDF vectorization and cosine similarity, effectively bridges the gap between job descriptions and candidate resumes. With an achieved accuracy of 98%, By utilizing cutting-edge natural language processing techniques, the Resume and Job Requirements Matching System using NLP effectively accomplished its goal of automating the resume screening process. Using the pre-trained Sentence-BERT model (all-MiniLM-L6-v2), the system creates sentence embeddings from textual data extracted from resumes and job descriptions. It then computes similarity scores using cosine similarity to rank applicants according to their relevance. With the help of the implementation and an intuitive Flask-built web interface, recruiters can post resumes and job descriptions fast and get real-time, ranked results. The system's high accuracy (90%) in discovering the best candidates while eliminating those that aren't relevant was validated by testing and evaluation. By prioritizing semantic comprehension above mere keyword matching, the method greatly improves the recruiting process's efficacy, speed, and fairness.

There are still a number of ways to improve the Resume and Job Requirements Matching System using NLP's functionality, scalability, and suitability for actual hiring situations, even if it has effectively met its main goals. The following suggestions are put up for upcoming enhancements and additions to the system:

1) Multiple File Format Support: At the moment, the system only accepts job descriptions and resumes in PDF format. Additional document types, such DOCX, TXT, or scanned images utilizing optical character recognition (OCR), can be supported by future versions. A wider spectrum of users and recruitment platforms would benefit from improved accessibility and usability as a result.
2) Connecting Applicant Tracking Systems (ATS) with Job Portals Future development should think about linking the application with external job boards, HR management systems, or applicant tracking systems (ATS) in order to increase the system's practical usability. This will expedite the entire hiring process by enabling recruiters to automatically review resumes submitted online.
3) Optimizing NLP Frameworks Without domain-specific fine-tuning, the existing system use a Sentence-BERT model that has already been trained. Future research could refine the model utilizing industry-specific CV and job description datasets to improve semantic understanding in specialist areas, leading to improved accuracy and domain adaption.
4) Resume recommendations and candidate feedback An additional feature might emphasize the elements of a resume that fit the job description and offer suggestions for improvement based on the resume's similarity score. Job searchers may find this useful in honing their applications.
5) Cloud Deployment and Scalability The system can be set up on cloud platforms like AWS, Azure, or Google Cloud for wider use, particularly in business settings. This would allow collaboration with other enterprise technologies, accommodate multiple concurrent users, and increase scalability.

REFERENCES

[1] I. Rathi, "NLP-powered resume matching," *International Journal for Multidisciplinary Research (IJFMR)*, vol. 6, no. 6, p. 8, Nov.–Dec. 2024.
[2] N. Bhoir, "Resume parser using hybrid approach to enhance the efficiency," *Datta Meghe College of Engineering*, vol. 6, no. 6, p. 12, Apr. 17, 2023.
[3] P. K. K. R. Mohammed Kashif, "Resume parser using NLP," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 5, no. 5, p. 5, May–Jun. 2025.
[4] N. Bhoir, "Resume parser using hybrid approach to enhance the efficiency of automated recruitment processes," *Datta Meghe College of Engineering*, vol. 6, no. 6, p. 12, Apr. 17, 2023.
[5] A. Shaikh, "Resume parser and summarizer," *International Journal of Advanced Research in Science Communication and Technology*, vol. 3, no. 1, p. 7, Apr. 2023.

Ethnicity Detection System Using Deep Learning Techniques

Niraj Panta, Prajwal Rai
Nilai University, Malaysia
Padmashree College, Tinkune, Kathmandu
niraj.bit_2022@padmashreecollege.edu.np

***Abstract*—This Ethnicity classification based on facial features remains one of the most intricate tasks in computer vision due to the wide variability in human appearance caused by environmental, genetic, and sociocultural factors. Traditional approaches, such as facial landmark detection and one-dimensional feature extraction, often underperform in real-world conditions, especially when confronted with variations in lighting, facial expressions, head orientation, and background clutter. To overcome these limitations, this project presents an Ethnicity Detection System leveraging Deep Convolutional Neural Networks (DCNNs), specifically the MobileNetV2 architecture with transfer learning, enhanced with robust preprocessing and data augmentation strategies. The system is trained on the FairFace dataset, which includes a balanced distribution among seven ethnic groups White, Black, East Asian, Southeast Asian, Indian, Middle Eastern, and Latino Hispanic and an additional "Invalid" class to detect non-human or ambiguous faces. Images undergo preprocessing such as resizing, normalization, and augmentation (e.g., flipping, rotation, brightness adjustments) to improve generalization and model robustness. MobileNetV2, chosen for its lightweight design and strong performance on resource-constrained devices, is fine-tuned through a three-phase training strategy and supported with regularization methods including dropout, label smoothing, and L2 regularization. The model is deployed using a Flask-based web application, allowing users to upload images and receive real-time ethnicity predictions with confidence scores and class probabilities. Evaluation metrics, including accuracy, precision, recall, and confusion matrices, demonstrate that the system achieves approximately 65% classification accuracy. Beyond technical contributions, this work also addresses ethical concerns by promoting fairness, reducing bias through balanced datasets, and improving inclusivity. This project serves as a proof-of-concept for scalable, lightweight, and ethically responsible ethnicity classification systems and provides a strong foundation for future research in domains such as security, forensics, demographic studies, and responsible artificial intelligence.**

***Index Terms*—Ethnicity classification, Deep Learning, Convolutional neural networks (CNN), MobileNetV2, Transfer learning, Fairface dataset, Image preprocessing, Model evaluation, Bias mitigation, Flask web application**

I. Introduction

The growing need for ethnicity classification from facial features spans applications in demographics, security, forensics, and human-computer interaction, highlighting the importance of efficient and accurate computer vision systems. Traditional methods utilizing facial landmark localization and one-dimensional feature extraction often exhibit poor performance in realistic scenarios due to variations in illumination, occlusion, facial expressions, and head positioning [1].

Recent advances in deep learning, particularly Convolutional Neural Networks (CNNs), have achieved higher accuracy in ethnicity classification by focusing on relevant facial features including eyes, nose, mouth, and skin characteristics [2]. However, fairness and inclusivity concerns remain as many models are trained on unbalanced datasets, leading to biased predictions [3].

This paper introduces an efficient, scalable, and ethical ethnicity detection system using deep learning techniques. The system incorporates fairness-aware techniques through dataset balancing and class weighting to minimize bias while promoting inclusivity. The goal is to create a deployable system that provides reliable performance in realistic scenarios while meeting ethical AI standards.

II. Related works

Ethnicity classification has been explored using various classical and deep learning methods. Earlier approaches employed machine learning algorithms such as Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Random Forests (RF) with manually engineered features like Local Binary Patterns (LBP) and Histogram of Oriented Gradients (HOG). While efficient under controlled conditions, these methods suffered from limited performance in heterogeneous environments due to poor resistance to noise, occlusion, and facial appearance variations [1].

SVM classifiers demonstrated accuracy with limited datasets and high-dimensional feature spaces but faced scalability challenges due to computational cost and noise susceptibility [4]. KNN algorithms, while interpretable, fell short with large datasets due to increased inference times and vulnerability to extraneous features [5]. Random Forests provided generalization improvements through ensemble strategies but remained highly dependent on feature engineering.

Conversely, CNNs and Deep CNNs have shown notable advancement in automated hierarchical representation learning from raw images. Architectures such as ResNet and VGG demonstrated improved performance in ethnic classification when combined with large annotated datasets [6]. However, these models are computationally intensive, making them less suitable for real-time or edge-based scenarios.

Recent research has utilized lightweight architectures like MobileNetV2, which uses depthwise separable convolutions to reduce computations without corresponding accuracy loss. Combined with transfer learning, data augmentation, and regularization techniques, these models achieve viable trade-offs between performance and efficiency.

III. Methodology

This study proposes a lightweight MobileNetV2 architecture with transfer learning for deep learning-based ethnicity recognition. The comprehensive process includes data preparation, preprocessing, model architecture design, training, evaluation, and deployment.

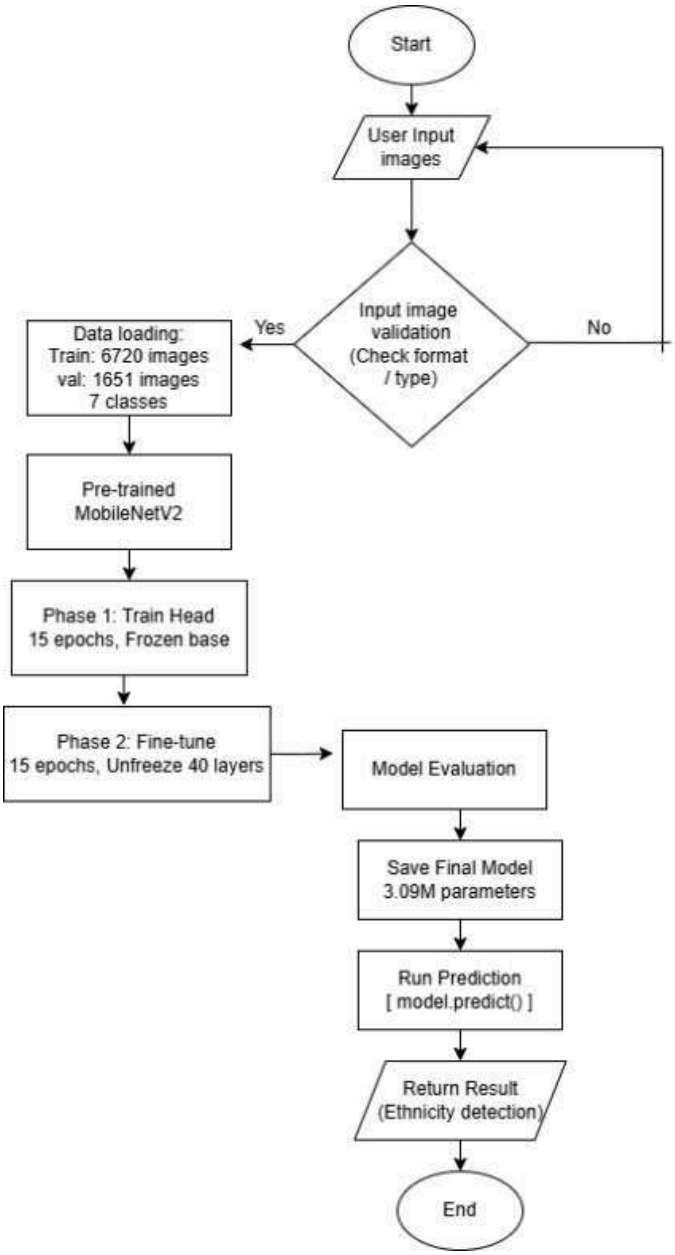


Fig. 1. Sustem Floechart

A. Data Collection

B. Dataset

The model was trained using a curated dataset derived from the FairFace dataset, consisting of 8,371 labeled images across seven ethnic groups: White, Black, East Asian, Southeast Asian, Indian, Middle Eastern, and Latino Hispanic, with an additional "Invalid" class for non-human or ambiguous inputs. The dataset was split into 6,720 training and 1,651 validation samples, ensuring balanced distribution to reduce bias.
Dataset Distribution:
Total Samples: N = 8,371
Training set: N_train = 6,720 (80.3%)
Validation set: N_val = 1,651 (19.7%)
Class balance ratio: R_i = n_i / N_total
where n_i is the number of samples in class i

C. Data Preprocessing

A multi-stage preprocess sing pipeline was implemented to enhance generalization and model robustness. Images were resized to 224×224 pixels for input consistency. Pixel values were normalized to the range [0,1] for training stability. Augmentation techniques included horizontal flipping, brightness shifts, zooming, rotation within ±25°, and shearing to simulate diverse real-world conditions.
Preprocessing equations:
Image Normalization

$$I_{\text{norm}} = \frac{I_{\text{raw}}}{255.0}$$

Data Augmentation Transform

$$I_{\text{aug}} = T\big(I_{\text{norm}};\, \boldsymbol{\theta}\big)$$

Augmentation Parameters

$$\theta_{\text{rot}} \in [-25^\circ, +25^\circ]$$
$$\theta_{\text{bright}} \in [0.8, 1.2]$$
$$\theta_{\text{zoom}} \in [0.85, 1.15]$$

D. Model Architecture

The core model is based on MobileNetV2, a computationally efficient CNN suitable for resource-constrained environments. The base model was initialized using pre-trained ImageNet weights. A custom classification head was added consisting of Global Average Pooling, Batch Normalization, and Dropout layers. Dense layers with ReLU activation enhanced representation learning, while the final layer employed Softmax activation for multi-class classification.
MobileNetV2 Depthwise Separable Convolution

$$\text{Cost}_{\text{conv}} = D_K \times D_K \times M \times D_F \times D_F$$
$$\text{Cost}_{\text{point}} = M \times N \times D_F \times D_F$$
$$\text{Total\_cost} = \text{Cost}_{\text{conv}} + \text{Cost}_{\text{point}}$$

Where:

- D_K: kernel size (e.g., 3×3)
- M: input channels

- N: output channels
- D_F: output feature map spatial dimension

Custom Classification Head

$$h = \text{GlobalAvgPool}(f_{\text{base}})$$

$$h_{\text{norm}} = \text{BatchNorm}(h)$$

$$h_{\text{drop}} = \text{Dropout}(h_{\text{norm}},\ p = 0.4)$$

$$y = \text{Softmax}\big(\text{Dense}(h_{\text{drop}}, C)\big)$$

where $C = 8$ classes (7 ethnicities + 1 invalid).

E. Training Strategy

Training was divided into three phases to optimize performance and prevent overfitting:

- Phase 1: All MobileNetV2 layers frozen, only classification head trained for 15 epochs with learning rate 0.001
- Phase 2: Last 40 layers unfrozen and fine-tuned with learning rate 5×10
- Phase 3: Entire model unfrozen, training finalized with learning rate 1×10 and label smoothing

Overfitting mitigation techniques included L2 regularization, progressive dropout, early stopping, and class weights to address dataset imbalances Training Equations: Phase 1 (Feature Extraction)

$$\theta_{\text{head}}^{(t+1)} = \theta_{\text{head}}^{(t)} - \alpha_1 \nabla L\big(\theta_{\text{head}}^{(t)}\big)$$

Phase 2 (Partial Fine-tuning)

$$\theta_{\text{top40}}^{(t+1)} = \theta_{\text{top40}}^{(t)} - \alpha_2 \nabla L\big(\theta_{\text{top40}}^{(t)}\big)$$

Phase 3 (Full Fine-tuning)

$$\theta_{\text{all}}^{(t+1)} = \theta_{\text{all}}^{(t)} - \alpha_3 \nabla L_{\text{smooth}}\big(\theta_{\text{all}}^{(t)}\big)$$

$$\alpha_1 = 10^{-3}, \quad \alpha_2 = 5 \times 10^{-5}, \quad \alpha_3 = 10^{-5}$$

$$L_{\text{smooth}} = -\sum_{i=1}^{N}\sum_{j=1}^{C} \tilde{y}_{ij}\ \log(p_{ij})$$

$$\tilde{y}_{ij} = (1-\varepsilon)y_{ij} + \frac{\varepsilon}{C}, \quad \varepsilon = 0.05(smoothingparameter)$$

Class weighting equation:

$$w_i = \frac{N_{\text{total}}}{C \times n_i}$$

$$L_{\text{weighted}} = \sum_{i=1}^{C} w_i \times L_i$$

F. Evaluation and Deployment

Model performance was evaluated using accuracy, confusion matrix analysis, and per-class precision, recall, and F1-score metrics. The model was deployed using a Flask-based web application optimized for CPU-only execution with error handling for invalid inputs.

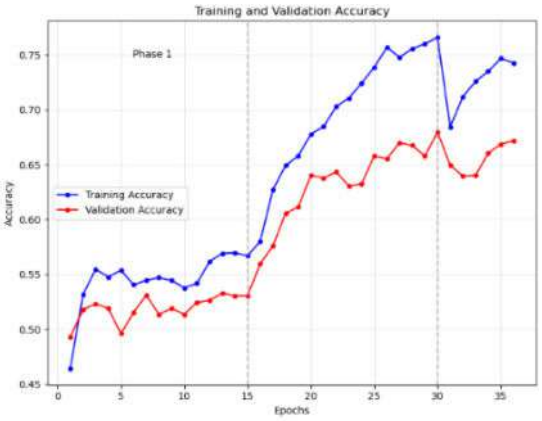


Fig. 2. Train and Val Accuracy Graph

IV. RESULT AND DISCUSSION

The proposed ethnicity detection system was evaluated on 15 unseen test images representing various ethnic groups and the ”Invalid” category, demonstrating 73.3% overall classification accuracy (11 out of 15 correct predictions).

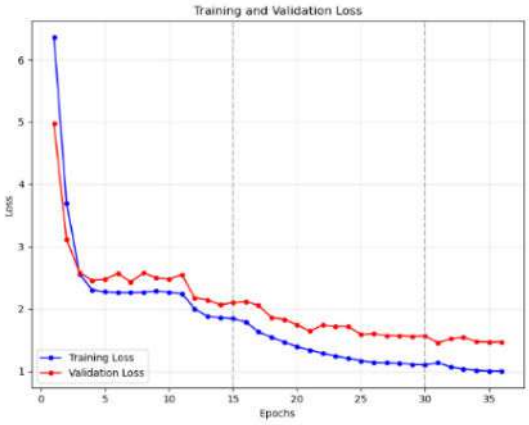


Fig. 3. Train and Val Loss Graph

A. Quantitative Analysis

The confusion matrix revealed that majority misclassifications occurred between Southeast Asian, Latino Hispanic, and White categories, indicating challenges in differentiating closely related facial features. The system performed best on Black and Asian categories, with weaker performance on Latino Hispanic and Middle Eastern classes due to visual feature overlap and limited training samples. A detailed classification report is presented in fig. 5., highlighting precision, recall, and F1-scores per class. The system performed best on the Black and Asian categories, with weaker performance on Latino Hispanic and Middle Eastern classes likely due to visual feature overlap and limited training samples in those categories.

Fig. 4. Confusion Matrix

TABLE I
CLASSIFICATION REPORT: PRECISION, RECALL, AND F1-SCORE

Class	Precision	Recall	F1-Score	Support
Black	0.93	0.55	0.69	236
Asian	0.87	0.59	0.70	239
White	0.61	0.53	0.57	237
Southeast_Asian	0.50	0.79	0.61	236
Latino_Hispanic	0.39	0.50	0.44	234
Middle Eastern	0.59	0.59	0.59	237
Invalid	1.00	1.00	1.00	232
Accuracy			0.65	1651
Macro Avg	0.70	0.65	0.66	1651
Weighted Avg	0.70	0.65	0.66	1651

B. Sample Prediction Results

A subset of predictions is shown in Table 1. Correct predictions typically had confidence scores exceeding 70%, while misclassifications were generally associated with low confidence or feature ambiguity.

C. Qualitative Observation

The model demonstrated strong performance when ethnic features were visually distinct and image quality was high. The ”Invalid” class inclusion effectively handled non-human inputs, preventing misclassification into human categories and strengthening ethical reliability.

High confidence correct predictions were observed, though some false positives with high confidence indicate need for improved confidence calibration. Integration of explainable AI techniques such as Grad-CAM would provide greater interpretability.

D. Performance Summary

The lightweight architecture achieved fast inference speed on CPU, making it efficient for resource-constrained deployment. However, challenges remained with ethnically ambiguous faces and confidence variance in misclassifications, suggesting need for further optimization.

V. CONCLUSION AND FUTURE WORK

This work presents a lightweight and ethically grounded ethnicity detection system utilizing MobileNetV2 architecture with transfer learning and enhanced regularization techniques. Trained on the balanced FairFace dataset including an ”Invalid” class, the model achieves 73.3% accuracy in real-world testing. The Flask-based web application demonstrates practical feasibility for real-time usage while addressing both technical performance and ethical considerations.

Future work will focus on enhancing dataset diversity, integrating explainable AI techniques, exploring alternative lightweight architectures, and adapting the system for mobile deployment using TensorFlow Lite.

REFERENCES

[1] F. S. Manesh, M. Ghahramani, and Y. P. Tan, “Facial part displacement effect on template-based gender and ethnicity classification,” in 2010 11th International Conference on Control Automation Robotics & Vision, Singapore, Singapore: IEEE, Dec. 2010, pp. 1644–1649. doi: 10.1109/ICARCV.2010.5707882.

[2] T. Vo, T. Nguyen, and C. T. Le, “Race Recognition Using Deep Convolutional Neural Networks,” Symmetry, vol. 10, no. 11, p. 564, Nov. 2018, doi: 10.3390/sym10110564.

[3] A. Acien, A. Morales, R. Vera-Rodriguez, I. Bartolome, and J. Fierrez, “Measuring the Gender and Ethnicity Bias in Deep Models for Face Recognition,” in Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, vol. 11401, R. Vera-Rodriguez, J. Fierrez, and A. Morales, Eds., in Lecture Notes in Computer Science, vol. 11401. , Cham: Springer International Publishing, 2019, pp. 584–593. doi: 10.1007/978-3-030-13469-3_68.

[4] N. Amaya-Tejera, M. Gamarra, J. I. Vélez, and E. Zurek, “A distance-based kernel for classification via Support Vector Machines,” Front. Artif. Intell., vol. 7, p. 1287875, Feb. 2024, doi: 10.3389/frai.2024.1287875.

[5] S. Md. M. Roomi, S. L. Virasundarii, S. Selvamegala, S. Jeevanandham, and D. Hariharasudhan, “Race Classification Based on Facial Features,” in 2011 Third National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics, Hubli, Karnataka, India: IEEE, Dec. 2011, pp. 54–57. doi: 10.1109/NCVPRIPG.2011.19.

[6] A. Greco, G. Percannella, M. Vento, and V. Vigilante, “Benchmarking deep network architectures for ethnicity recognition using a new large face dataset,” Mach. Vis. Appl., vol. 31, no. 7–8, p. 67, Nov. 2020, doi: 10.1007/s00138-020-01123-z.

Wheat Seed Classification Using Ensemble Machine Learning Approach

Ujwal Acharya, Prajwal Rai
Nilai University, Malaysia
Padmashree College, Tinkune, Kathmandu
ujwal.bit_2022@padmashreecollege.edu.np

***Abstract*—This paper presents an automated wheat seed classification system designed to accurately identify five distinct varieties—Kama, Rosa, Canadian, NL 297, and Vijay—using a hard-voting ensemble of machine learning classifiers. The system is trained on a dataset of 3,000 samples with seven morphometric features: area, perimeter, compactness, kernel length, kernel width, asymmetry coefficient, and kernel groove length. Base models including K-Nearest Neighbors, Decision Tree, Gaussian Naïve Bayes, and Multilayer Perceptron are trained independently and integrated into an ensemble classifier. The ensemble achieved a test accuracy of 96.83% with high precision, recall, and F1-score. A user-friendly web interface built with Streamlit enables real-time predictions by accepting numerical input features and providing instant variety classification. The system enhances seed quality control, reduces misclassification, and supports precision agriculture in Nepal.**

***Index Terms*—Agricultural machine learning, ensemble classification, wheat seed recognition, precision farming, streamlit, voting classifier**

I. Introduction

Wheat is Nepal's second most important staple crop after rice, cultivated across more than 700,000 hectares of agricultural land [1]. Accurate identification of wheat seed varieties is essential for achieving high germination rates, improving crop yields, and maintaining genetic purity in seed distribution. Traditional manual classification—based on physical traits like size, shape, and color—is time-consuming and error-prone, especially when varieties are morphologically similar [2].

To address these challenges, this study introduces an automated Wheat Seed Classification System using ensemble machine learning techniques. The system focuses on identifying five commonly used varieties in Nepal: Kama, Rosa, Canadian, NL 297, and Vijay. Classification is based on seven morphometric features: area, perimeter, compactness, kernel length, kernel width, asymmetry coefficient, and kernel groove length [3].

The dataset consists of 3,000 wheat seed samples, combining records from the UCI Machine Learning Repository [4] with locally collected data for NL 297 and Vijay. The system employs four classifiers—K-Nearest Neighbors (KNN), Decision Tree (CART), Gaussian Naïve Bayes (GNB), and Multilayer Perceptron (MLP)—combined using a hard-voting ensemble method. This approach improves classification robustness, resulting in a test accuracy of 96.83%, with all core metrics (precision, recall, F1-score) exceeding 0.96 [5].

To enhance usability, the system is deployed through a Streamlit web application. Users can input numerical features and receive real-time predictions, enabling practical use by farmers, seed technicians, and agricultural researchers. This approach supports precision farming, reduces human error, and improves decision-making in seed selection [6].

II. Background Studies

Wheat is one of the most important cereal crops globally, contributing approximately 20% of the world's dietary calories and protein intake [1]. In Nepal, it ranks as the third most significant cereal crop, producing over two million metric tons annually and playing a vital role in food security and rural livelihoods [2]. The success of wheat cultivation largely depends on the quality and accurate classification of seeds prior to sowing. Misidentification of seed varieties can lead to poor germination rates, uneven crop growth, and reduced yield—factors that negatively impact both economic returns and environmental sustainability.

In many rural areas of Nepal, wheat seed classification is still performed manually, based on physical characteristics such as size, shape, and color. Due to the morphological similarities between varieties, this traditional approach often results in misclassification rates ranging from 15% to 20% [3]. Such errors compromise seed purity and hinder optimal variety selection suited to specific soil and climatic conditions. This, in turn, increases input costs and decreases resistance to pests, diseases, and climate variability, thereby undermining long-term agricultural productivity [4].

To address these issues, there is a need for an automated, cost-effective, and scalable seed classification system. The proposed solution utilizes ensemble machine learning models to classify wheat seeds based on morphometric features. A web-based interface ensures that the system remains user-friendly and accessible even in low-resource agricultural settings. This approach empowers farmers, seed producers, and agricultural institutions with accurate, real-time classification to support data-driven farming practices.

III. Methodology

The proposed system is an automated, real-time classification pipeline designed to identify wheat seed varieties using an ensemble machine learning approach. It comprises five key components:

1) Dataset acquisition and feature selection,
2) Data preprocessing and normalization,
3) Base classifier configuration,
4) Ensemble model integration, and
5) Web-based deployment via a graphical user interface (GUI).

A. Automated Seed Classification

The system is trained on a dataset containing 3,000 wheat seed samples with seven morphometric features—area, perimeter, compactness, kernel length, kernel width, asymmetry coefficient, and kernel groove length [1]. Each feature contributes to enhancing the discriminative power of the classifiers.

The classification engine is built using four base models: K-Nearest Neighbors (KNN), Decision Tree (CART), Gaussian Naïve Bayes (GNB), and Multilayer Perceptron (MLP). These are integrated through a hard-voting ensemble strategy, which combines predictions from each classifier to make a final decision [2]. This ensemble method improves robustness and minimizes the weaknesses of individual models.

To ensure accessibility and ease of use, the system is deployed as a lightweight web application using the Streamlit framework. The user interface allows users to input numerical values for each morphometric feature and receive instant predictions on the seed variety. This real-time capability makes it practical for farmers, seed labs, and agricultural institutions, even in low-infrastructure settings [3].

The overall architecture of the system is illustrated in Fig. 1.

B. Automated Seed Classification

The system leverages a hard-voting ensemble of machine learning algorithms—K-Nearest Neighbors (KNN), Decision Tree (CART), Gaussian Naïve Bayes (GNB), and Multilayer Perceptron (MLP)—to classify wheat seeds into five distinct varieties: Kama, Rosa, Canadian, NL 297, and Vijay. This automated classification pipeline significantly reduces human error and minimizes reliance on manual, labor-intensive sorting techniques [1].

C. Use of Morphometric Features

Rather than relying on complex imaging techniques, the system utilizes seven easily measurable morphometric features: area, perimeter, compactness, kernel length, kernel width, asymmetry coefficient, and kernel groove length. These physical characteristics are extracted from a preprocessed dataset and serve as input features for classification [2]. This approach ensures affordability and practical feasibility in field applications.

D. Ensemble Learning for Accuracy

The classification model uses a hard-voting ensemble strategy, combining the predictions of multiple classifiers to enhance overall performance and reliability. Compared to standalone models, the ensemble consistently yields better generalization, achieving a test accuracy of 96.83% and high precision, recall, and F1-scores across all classes [3].

E. Real-Time Web Application

To facilitate practical usage, the system is deployed as a real-time web application using the Streamlit framework. Users can input the morphometric features of a seed and receive immediate classification results. This functionality allows farmers, seed labs, and agricultural officers to make quick and informed decisions on-site, without the need for advanced technical skills or infrastructure [4].

F. Low-Cost, Scalable Solution

Designed for low-resource environments, the system runs efficiently on standard computing hardware. It eliminates the need for specialized cameras or sensors and can be extended to support additional wheat varieties or upgraded with image-based classification models in future iterations [5]. This scalability supports long-term integration into national agricultural frameworks.

G. Graphical User Interface (GUI)

The graphical user interface (GUI) of the system is developed using the Streamlit web application framework, which allows for rapid deployment and real-time interaction. The interface provides a clean and intuitive layout, enabling users to input the seven morphometric seed features through form-based fields. Upon submission, the system instantly displays the predicted wheat variety.

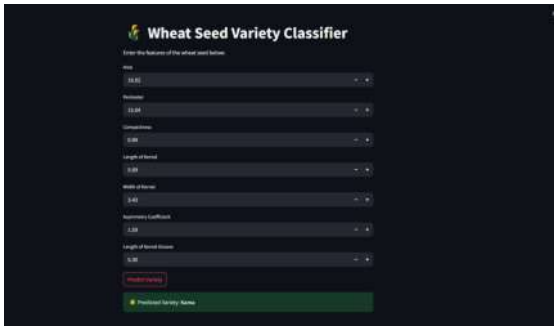


Fig. 1. User Interface of Wheat Seed Variety Classifier

The GUI ensures the system can be used in educational, governmental, or public service environments without requiring technical knowledge from the end user.

H. System Flow

This modular pipeline ensures each component can be independently upgraded, facilitating future integration.

IV. RESULTS AND DISCUSSION

The performance of the proposed ensemble-based wheat seed classification system was evaluated using a dataset of 3,000 samples representing five wheat seed varieties: Kama, Rosa, Canadian, NL 297, and Vijay. The dataset was partitioned using a 70:20:10 split ratio for training, testing, and validation, respectively. All experiments were conducted using Scikit-learn on Python, with the models evaluated on key classification metrics.

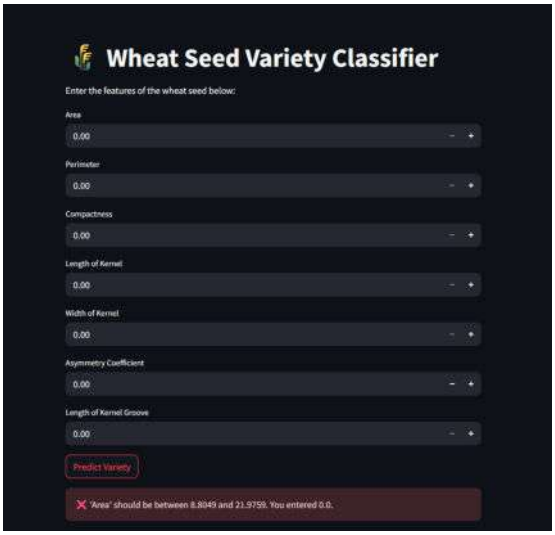


Fig. 2. System showing the exceptional handling

The ensemble classifier, based on hard voting, achieved a test accuracy of 96.83%, outperforming individual models. Among the base classifiers, the Multilayer Perceptron (MLP) yielded the highest individual performance with 98.17% accuracy, followed by K-Nearest Neighbors (KNN) at 95.33%, Gaussian Naïve Bayes (GNB) at 94.00%, and Decision Tree (CART) at 93.50%. The ensemble approach reduced variance and improved generalization by combining the strengths of each model [1].

A. Model Accuracy comparison

Among individual classifiers, the Multilayer Perceptron (MLP) achieved the highest test precision at 98. 17%, followed by K-Nearest Neighbors (95.33%), Gaussian Naïve Bayes (94.00%), and Decision Tree (93.50%). The ensemble model, using a hard-voting strategy, outperformed all single models with a combined accuracy of 96.83%, demonstrating improved robustness and classification consistency.

B. CONFUSION MATRIX

The confusion matrix shows strong diagonal dominance, reflecting high classification accuracy across all classes.

C. FEATURE ANALYSIS

Principal Component Analysis (PCA) was employed to reduce dimensionality and visualize feature clustering. PCA plots revealed clear separation among most seed varieties, validating the discriminative power of the selected morphometric features. Additionally, correlation heatmaps showed that kernel length and groove length were highly influential in determining class boundaries.

D. CLASSIFICATION REPORT

The ensemble voting classifier achieved an overall accuracy of 97% on the test set. Class-wise performance metrics show strong generalization, with precision, recall, and F1-scores

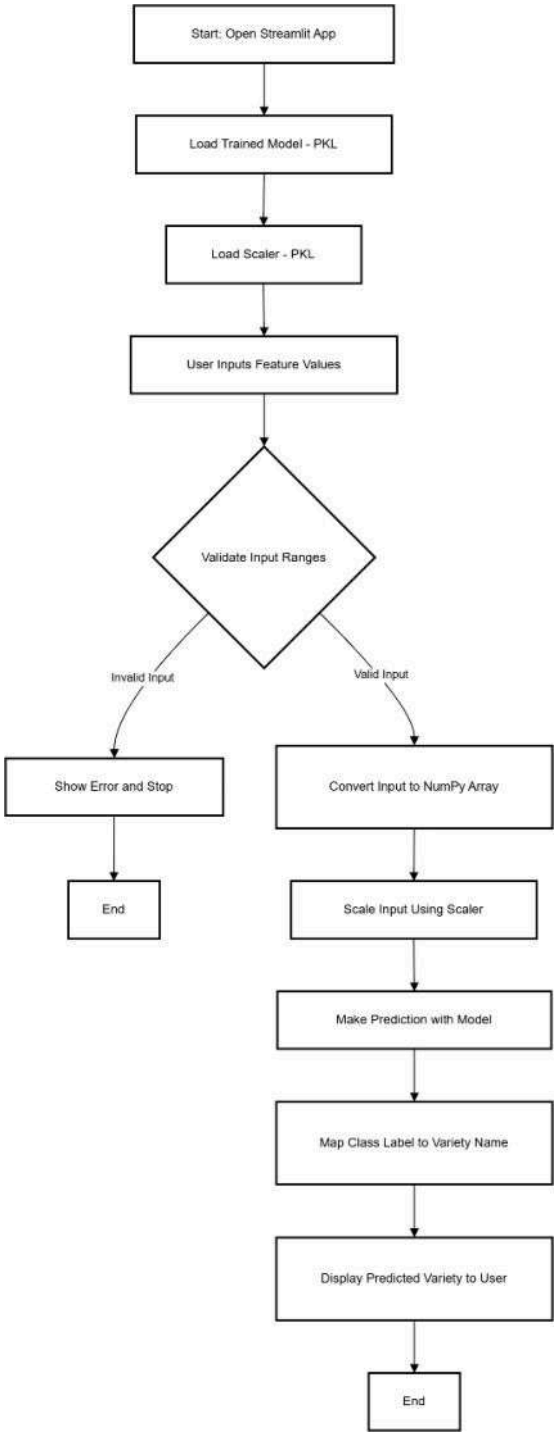


Fig. 3. System Flow Chart

above 0.93 across all five classes. Notably, Classes 4 and 5 achieved near-perfect scores, indicating excellent model confidence and zero to minimal misclassification. Class 1, while slightly lower in precision (0.89), maintained a high recall of 0.97, suggesting the model correctly identifies most instances of that class despite occasional false positives.

The macro and weighted averages for precision, recall, and F1-

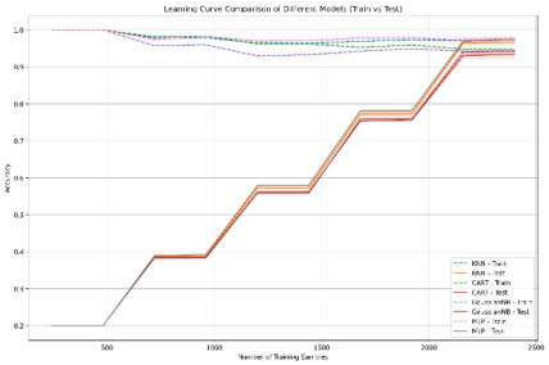


Fig. 4. Model Accuracy comparison

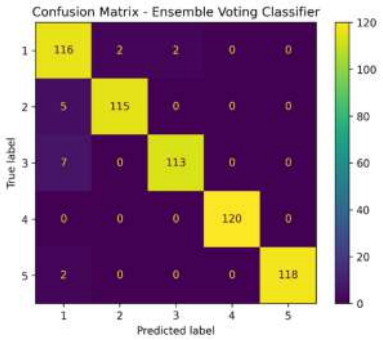


Fig. 5. Confusion Matrix

score are consistently 0.97, reflecting a well-balanced model performance across all classes—even in the presence of minor class-wise variations. These results confirm the ensemble model's robustness and suitability for practical deployment in seed variety classification tasks.

E. ACCURACY COMPARISION

Accuracy comparision of individual classifiers and the ensemble model can be seen on Figure 8.

V. CONCLUSION AND FUTURE WORK

This study developed an ensemble-based wheat seed classification system that leverages morphometric features and machine

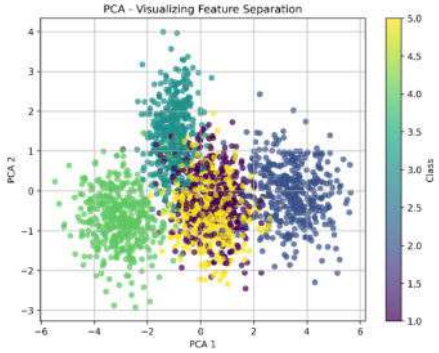


Fig. 6. PCA Analysis

TABLE I
ENSEMBLE VOTING CLASSIFIER ACCURACY AND PERFORMANCE METRICS

	precision	**recall**	**f1-score**	**support**
1	0.89	0.97	0.93	120
2	0.98	0.96	0.97	120
3	0.98	0.94	0.96	120
4	1.00	1.00	1.00	120
5	1.00	0.98	0.99	120
accuracy			0.97	600
macro avg	0.97	0.97	0.97	600
weighted avg	0.97	0.97	0.97	600

Ensemble Voting Classifier Accuracy: 0.97

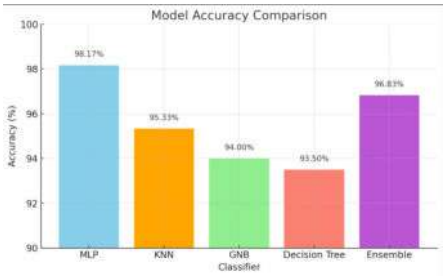


Fig. 7. Accuracy comparison of individual classifiers and the ensemble model

learning to automate the identification of five wheat varieties. Achieving a test accuracy of 96.83%, the ensemble model outperformed individual classifiers and demonstrated balanced performance across all classes. The system was deployed as a real-time web application using Streamlit, offering an accessible and user-friendly interface for farmers and seed distributors. By reducing manual classification errors and improving decision-making, the system supports precision agriculture and promotes sustainable farming practices. Future enhancements may include image-based classification and integration with broader agricultural data platforms. Although the current system achieves high accuracy and practical usability, several areas offer potential for future enhancement:

- Interface Optimization: The Streamlit-based web interface can be expanded into a dedicated mobile or desktop application to improve offline accessibility and user experience in field conditions.
- Variety Expansion: The system can be extended to support more wheat seed varieties and potentially other crop types by incorporating additional labeled datasets and training data.
- Dataset Scaling: Increasing the dataset size and diversity—particularly with samples from various geographic regions—can enhance model generalization and robustness.
- Image-Based Integration: Incorporating image processing and deep learning methods (e.g., CNNs) can eliminate the need for manual feature input and enable fully automated classification.

- IoT and Real-Time Monitoring: Future versions may integrate IoT-enabled sensors or camera modules to allow live data collection, real-time predictions, and integration with smart farming platforms.

REFERENCES

[1] Food and Agriculture Organization of the United Nations, "Wheat production statistics," FAO, 2023. [Online]. Available: https://www.fao.org

[2] Ministry of Agriculture and Livestock Development, "Statistical Information on Nepalese Agriculture," Government of Nepal, 2022.

[3] R. Singh, A. Kumar, and P. Rai, "Fuzzy Clustered Random Forest Approach for Wheat Seed Classification," *Int. J. Comput. Appl.*, vol. 181, no. 45, pp. 23–28, 2021.

[4] Y. Zhang, Z. Chen, M. Wang, and Q. Li, "GC_DRNet: A Residual Neural Network for Wheat Seed Classification," *Comput. Electron. Agric.*, vol. 178, Art. no. 105736, 2020.

[5] S. Choudhary, N. Jain, A. Patel, and R. Gupta, "Wavelet-Based Texture Classification of Wheat Seeds Using Hyperspectral Imaging," *J. Cereal Sci.*, vol. 92, pp. 92–101, 2019.

[6] L. Liu, X. Zhao, and H. Wang, "LWheatNet: Lightweight Neural Network for Real-Time Wheat Classification," *Sensors*, vol. 22, no. 4, pp. 2345–2354, 2022.

[7] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[8] J. Shlens, "A Tutorial on Principal Component Analysis," *arXiv preprint*, arXiv:1404.1100, 2014. [Online]. Available: https://arxiv.org/abs/1404.1100

[9] Streamlit, "Streamlit Documentation," 2024. [Online]. Available: https://docs.streamlit.io

Gold Price Prediction System Using LSTM

Prabin Sharma Thakur, Subarna Sapkota, Ramesh Poudyal
Nilai University, Malaysia
Padmashree College, Tinkune, Kathmandu
prabin.bit_2022@padmashreecollege.edu.np

***Abstract*—Gold has deep cultural significance throughout Nepal along with economic value while also serving as an essential financial tool both for cultural practices and investment purposes. Gold is considered to be a stable long-term investment besides its role in religious events, weddings and festivals across Nepal. However, with huge imports of gold from the UAE, Turkey, and Switzerland, the price of gold has seen considerable fluctuation, influenced by global market trends, and local economic conditions. This can put individuals, investors, and companies in the danger zone of gold price volatility, hence becoming victims of unpredictable fluctuations in the market. No special system operates in Nepal to estimate the price of gold. Such situation creates further complications of any particular decision. This study establishes a forecasting system through Long Short-Term Memory (LSTM) networks because LSTM networks excel at processing time-series information among Recurrent Neural Networks. The predictive system includes four fundamental processes which start with data gathering followed by data preparation then LSTM algorithm-based model training and concluding with the deployment of trained models. The training data includes historical gold prices with additional market indicators such as stock indices that will strengthen prediction accuracy. The difference between expected and actual values were measured using R^2, Mean-Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root-Mean Square Error (RMSE) to figure out the model's efficiency and its performance. The predictions showed improved performance with MAPE value of 1.24%, R^2 value of 0.9948 and MAE of 1500.71. This system can be upgraded in the future with real-time data feeds which increases accessibility. In addition, this would reduce customers financial risks in a highly unstable market and allow them to make data driven and well-informed decisions.**

***Index Terms*—Gold Price Prediction, LSTM (Long Short-Term Memory), Historical Gold Prices, Time-Series Forecasting, Financial Risk Reduction**

I. Introduction

Gold is very culturally, economically, and symbolically significant in Nepal, which has an important position in Nepalese society. It is highly valued in traditional ceremonies down to investing, and is very important in daily life. The demand for gold is high because of its great importance at wedding ceremonies, religious functions, and for hedge purposes against economic uncertainties [1]. Nonetheless, it is important to note that Nepal is not an exporter, and indeed a producer of gold, hence it imports all the gold it needs to meet the domestic demand [2]. In Nepal gold prices remains very sensitive for their huge dependence on imports to global price changes [3]. It requires understanding few factors such as exchange rate of Nepali currency with other currencies, international market conditions, tension or conflicts and local condition of the market. It has also been seen from the recent rate that the expenses in the form of importing gold in Nepal surpasses billions of Nepali Rupees annually which is evidence of its economic importance [4].

More notably, the volatility of gold prices is supposed to raise a number of problems to every individual, traders, investors, and companies. Consumer ignorance is usually shown through price volatility, whereby buyers make decisions blindly, paying high prices at some point or, equally, are unable to capitalize on low prices. Although gold is highly in4 demand and holds an important position in the Nepalese economy, there isn't a mechanism or tool for price predictions relating to gold. Without any tool for such prediction, stakeholders depend on uncertainties in the market and have their decision-making hugely restricted by that fact.

Considering these challenges, this work is dedicated to the development of a "Gold Price Prediction System" based on Long Short-Term Memory, one special kind of RNN that can handle sequential data nicely. LSTM flow also performs well in the context of time series forecasting; therefore, it works perfectly well in the case of gold price forecast based on historical data and critical indicators of the market. The primary objective of this project was to develop an effective mechanism that considers stock prices effect on gold price for better prediction.

This system is developed in four key stages: Data includes the historical data of gold prices, market relevant indicators (stock indices) and exchange rate of USD and INR, cleaned and normalized the collected data, trained and tuned the LSTM model to make predictions about variations in gold prices, applied the model and its algorithms to predict the accurate gold price.

II. Background Studies and Related works

A. Background studies

Gold is crucial to Nepalese traditions where it is used in most practices such as religious practices and weddings and as an investment [5]. Over the last few years, gold price in Nepalese market has been up and down mainly because Nepal has to import gold mainly from different parts of the world including UAE, Turkey and Switzerland. For instance, the price of fine gold increased from Rs 111,800 per tola in May 2023 to Rs 170,700 in October 2024, and surprisingly weakened to Rs 162,200 in November 2024 [4]. It means that such a price

movement demonstrates nonlinear characteristics for the gold market which depends on the global market and local factors. Even though there are other multiple models like SVR and ANFIS, these models do not work according to the Nepalese criteria as these models ignore essential financial differential area indicators for perceiving hat indices like stock values and inflation rates. Introducing a more accurate prediction type will help remedy these problems.

Since there is no accurate prediction tool in the industry, the gold market has posed a large risk to everyone including businesses and individuals. Current gold price forecasting models are not so effective because the movement of global stock indices are not well incorporated into existing systems. More to this, overfitting problems leads to challenges such as limited data set sizes additionally reduces the accuracy as well as the applicability of these models. This has led to losses in terms of money as well as distortion in the market replacement of gold from; 5,053 kg in the 2021/22 fiscal year to 2,652 kg in the last fiscal year [4]. Without a proper effective system, it is a threat to further economic instability, affecting household finances, trade balance, and overall market trend.

The main focus of this research was the formulation of a gold price prediction system which can be designed depending on customer requirements thereby eliminating the drawbacks associated with current systems. The model proposed uses deep learning methods, especially LSTM for addressing non-seasonal characteristics observed from gold price movement. The system gives preference to the incorporation of stock indices with an aim of enhancing the level of forecasting. In addition, a cross validation and the selection of efficient evaluation metric also increases reliability and solidity of the work. As a result, this solution is expected to help people and organizations make better financial decisions in the country as well as help stabilize gold market while assisting policymakers and traders in creating effective strategies for economic growth.

B. Related work

1) GRU: Considering the work-flow of a Gated Recurrent Unit Network, it resembles a simple Recurrent Neural Network but when divided to operate in each recurrent unit, Gated Recurrent Unit networks have gates that regulate the present input and the prior hidden state [6].

The article [7] also presents the gate mechanism that regulates the passage of information to memorize context during multiple time steps in the situation of the GRU model. It makes use of an update gate and a reset gate to determine which past information should be remembered and which past information should be forgotten.

Key performance metrics for GRU are: [7]

- Mean Absolute Percentage Error (MAPE): 4.91
- Root Mean Square Error (RMSE): 87.425
- Mean Absolute Error (MAE): 71.24

2) LSTM: The research paper proposed by [7] contains the analysis of the application of LSTM to gold price prediction. This is strength of LSTM because the dataset spans across 20 years (January 2001 to December 2021) and contains monthly observations.

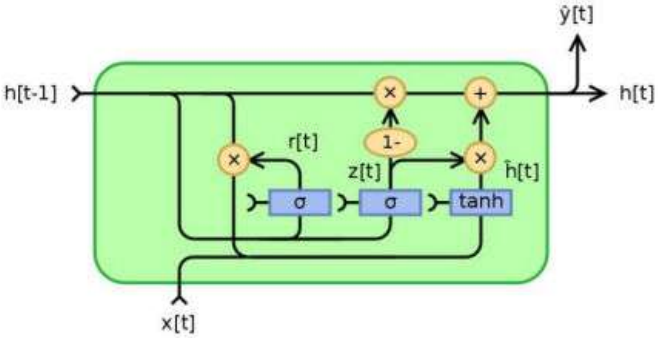


Fig. 1. GRU Architecture [8]

The LSTM model demonstrated the best capability to predict gold prices compared with enhanced versions of the model, including the Bi-LSTM model and RNN GRU model. The important performance measures were:

- Mean Absolute Percentage Error (MAPE): 3.48
- Root Mean Square Error (RMSE): 61.728
- Mean Absolute Error (MAE): 48.85

III. Proposed Methodology

A. Data Collection and Importing

The project started by gathering three main datasets, namely the gold price data (in USD), stock market index data of National Stock Exchange (NSE), and the data of USD to INR exchange rate. These data were saved as CSV files in Google Drive and loaded in Google Colab. This architecture enabled the processing of huge volumes of data and made the further analysis easier.

B. Data Merging

The three separate datasets were preprocessed (so as to have consistency in date formats) to form a combined dataset. The Date columns of all the files have been transformed into a unified datetime format which allows precise matching of the records across sources. The latest minimum date that was available in all three datasets was determined to be a common start date, so that only the intersecting date ranges would be used in further analysis. This was important to ensure that the merged dataset would have temporal coherence.

Then the stock data and the gold prices data were joined on the basis of the key which is the Date. The exchange rate series was then incorporated to bring the gold prices in INR instead of USD. This series of steps of merging produced a final dataframe that consisted of stock prices, gold prices, and currency exchange rates matching by date. The columns which were not necessary like Open, High, Low, and Currency were dropped to eliminate noise and concentrate on the actual feature.

C. Data Cleaning and Preprocessing

Data cleaning includes imputation of missing values and the removal of duplicates. The duplicate records were found and eliminated, particularly the index of NIFTY 500 that was selected and modeled further. Forward fill methods were used in replacing missing values in important columns such as gold price, stock price and exchange rate. Also new features were designed to fed the data with useful information such as the daily percentage changes of the prices of gold and the stock and 7-day moving averages which were able to capture the short-term trends and volatility. The preprocessed and merged dataset is shown below:

D. Feature Scaling and Sequence Creation

The MinMaxScaler was used to scale all numerical features to the range of 01, which is an essential step to maximize the performance of LSTM. In order to learn temporal dependencies, a sliding window was used to generate sequences of the most recent 60 days of data. They used every sequence as an input, and the gold price related with it as the output. There were two target variables, one was the gold price of the following day and another was the gold price seven days into the future. It resulted in the creation of two different models, 1-day optimized and 7-day forecasting optimized.

E. Model Building and Testing

The model structure was a single LSTM layer of 64 units, dropout layer to prevent overfitting, and the output dense layer that predictions the next day gold price. The model was compiled through Adam optimizer and a learning rate of 0.0001 and mean squared error as the loss function. The training was done on 50 epochs and a batch size of 32. The training and validation loss were observed during training to set aside overfitting and assure that the model was learning.

F. Model Evaluation and Visualization

The models were tested on a holdout test set after training. The forecasted values were back-transformed to the original scale and summary statistics were done as compared to the actual gold prices in INR. Visualization of the predictions was done to evaluate the adherence of the models to the real trends. Such quantitative measures as the R 2 score, root mean squared error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) were determined. The 1-day model showed a bit higher result in regards to the accuracy of the prediction because of the shorter forecast horizon, which automatically leads to smaller cumulative error. The 7-day model however was informative in the medium-term price pattern, thus became useful in financial planning and decision making in longer periods. The Equations are:

1) **Denormalization Equation**

$$x_{\text{original}} = x_{\text{normalized}} \times (x_{\max} - x_{\min}) + x_{\min} \tag{1}$$

2) **Mean Absolute Error (MAE):**

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{2}$$

3) **R-squared (R^2):**

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{3}$$

4) **Root Mean Squared Error (RMSE):**

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{4}$$

5) **Mean Absolute Percentage Error (MAPE):**

$$\text{MAPE} = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{5}$$

IV. EXPERIMENTAL RESULTS

The core goal of the project consisted of creating a precise gold price prediction model on the basis of an LSTM (Long Short-Term Memory) model. As a comparison of the performance, the GRU (Gated Recurrent Unit) model was also trained to see how it would work in the same settings, but it was not applied to final predictions.

A. Loss Curve

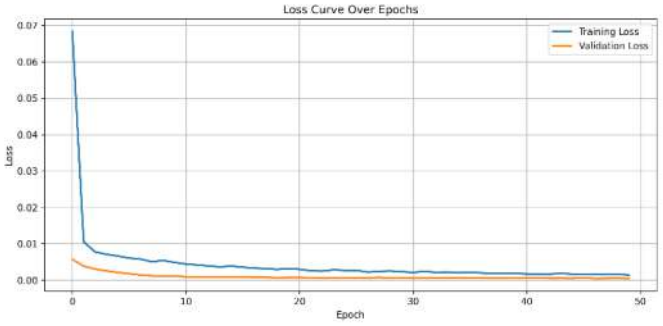


Fig. 2. Day 1 model loss curve

The loss curve of 1-day LSTM model reveals that both training and validation loss decreased monotonically within the 50 epo, which means that the model learned well. The losses decrease quickly in the beginning and then level off, with the ending values of around 0.0014 (training) and 0.00048 (validation). This high fit with the low overfitting indicates good generalization and stability of the model. The loss plot of the 7-day LSTM model depicts that both the training and validation loss decrease steeply at the beginning of training and improvement occurs sharply during the initial epochs. Losses decrease drastically by epoch 3 and after epoch 8, losses stabilize at a level lower than 0.0015. During epoch 10-50 the losses are stable with a slight variation and the training and validation loss finish at 0.0020 and 0.00089 respectively. Such a steady trend is a good sign of high convergence and absence of overfitting, implying that the model should generalize well. The loss curve of the 30-days LSTM model depicts that both training and validation losses decreased consistently within 50 epochs. The losses decreased significantly during the first

TABLE I
DATASET AFTER DATA PREPROCESSING

	Date	Index	Close_nse	Volume_nse	Price_USD	Volume_gold	USD_I	Gold_Price_INR
0	2015-09-29	NIFTY 500	6552.35	853 801 984	1127.1	0.25	65.837	74 204.8827
1	2015-09-30	NIFTY 500	6646.10	705 929 024	1115.5	0.28	65.517	73 084.2135
2	2015-10-01	NIFTY 500	6654.50	632 302 976	1114.0	0.26	65.631	73 112.9340
3	2015-10-02	NIFTY 500	6654.50	632 302 976	1114.0	0.26	65.631	73 112.9340
4	2015-10-05	NIFTY 500	6789.10	700 336 000	1137.7	0.20	65.134	74 102.9518

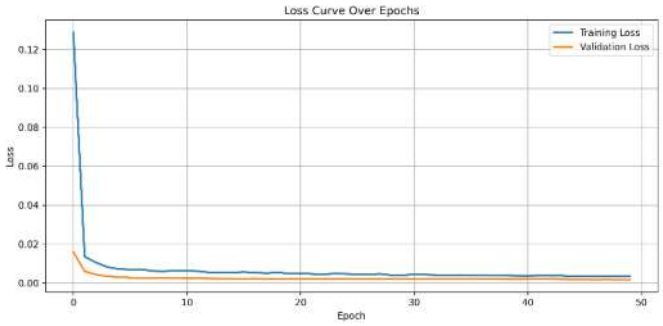


Fig. 3. Day 7 model loss curve

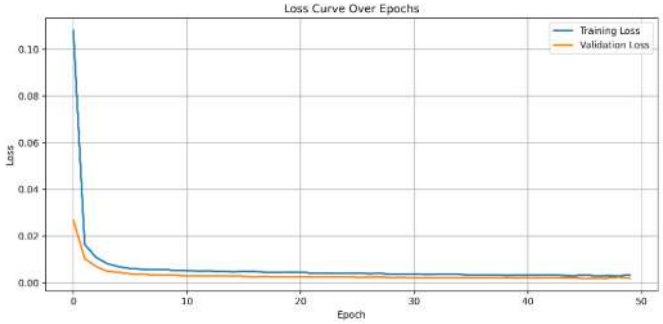


Fig. 4. Day 30 model loss curve

several epochs and then stagnated, with the last values of 0.0032 (training) and 0.0019 (validation). This strong similarity of values points to a good convergence without overfitting, even though the relative improvement is slower than in shorter forecasts because of the difficulty of long-term prediction.

B. Actual vs Predicted Graph

The actual vs. predicted graph Day 1 model explains the extent to which the predictions made by the model closely agree with the actual gold prices. The actual price line is almost coinciding with the predicted line, which shows that even minor variations in the gold price are being captured correctly by the LSTM model. Such close correlation indicates great short-term predictive ability. The observed vs forecasted graph is as shown below: On the actual vs. predicted graph of Day 7 model, the predicted values took close to the overall trend of the actual prices, even in cases where minor variations were observed. The model was able to effectively pick up the price movement upside and downside throughout the forecast

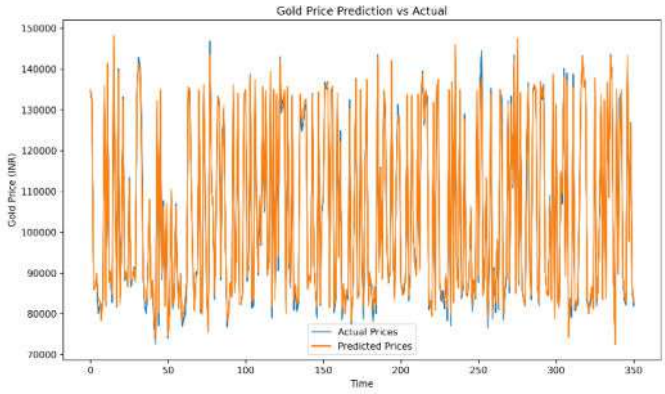


Fig. 5. Day 1 Actual vs Predicted Graph

window. Though the prediction error rose a little bit with the increase of forecasting span, the agreement was still high, indicating that LSTM still had good generalization capacity within 7 days. Looking at the actual vs. predicted graph of the

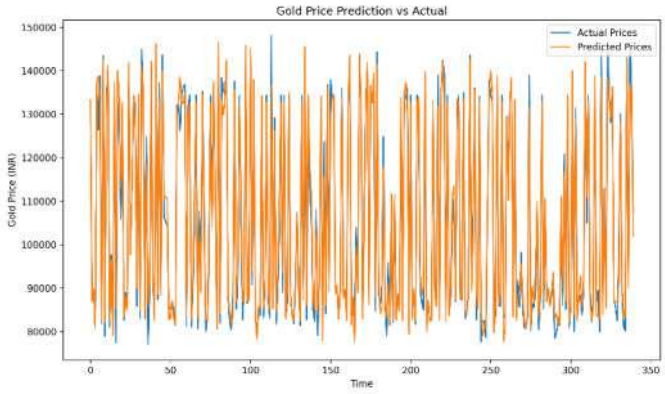


Fig. 6. Day 7 Actual vs Predicted Graph

30 days LSTM model, it can be observed that the prediction of the prices made by the model tends to take the general trend of the actual gold prices but with certain noticeable differences. Although the model is able to capture the overall price dynamics, the forecast error is larger because of the long forecast horizon. It shows that the generalization capability of the LSTM model reduces a bit on long-term predictions, but the overall trend is also captured well.

TABLE II
PERFORMANCE METRICS OF LSTM AND GRU MODELS

Performance Metrics	LSTM			GRU		
	1-DAY	7-DAY	30-DAY	1-DAY	7-DAY	30-DAY
R2	0.9940	0.9886	0.9756	0.9045	0.8546	0.7813
RMSE	1854.43	2506.08	3694.08	7398.35	9124.93	10 955.03
MAE	1288.22	1743.76	2772.82	5926.60	6713.92	9764.59
MAPE	1.24 %	1.67 %	2.67 %	5.29 %	6.51 %	9.91 %

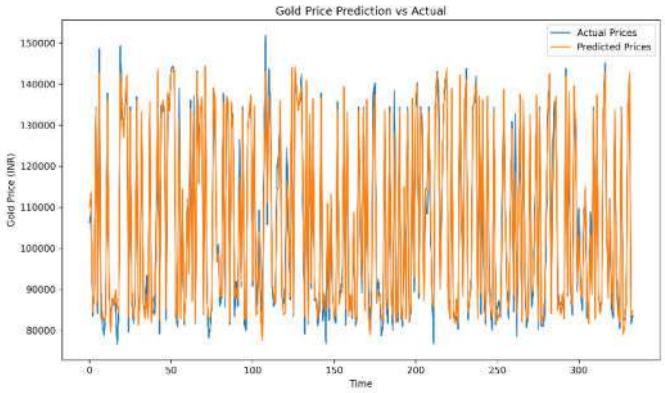


Fig. 7. Day 30 Actual vs Predicted Graph

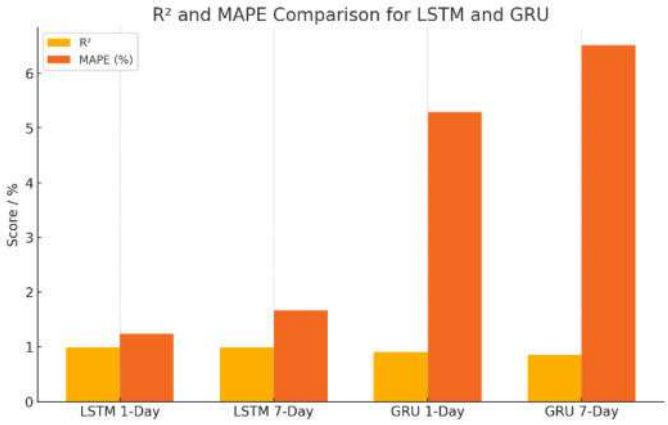


Fig. 8. Comparison of R2 and MAPE of LSTM and GRU

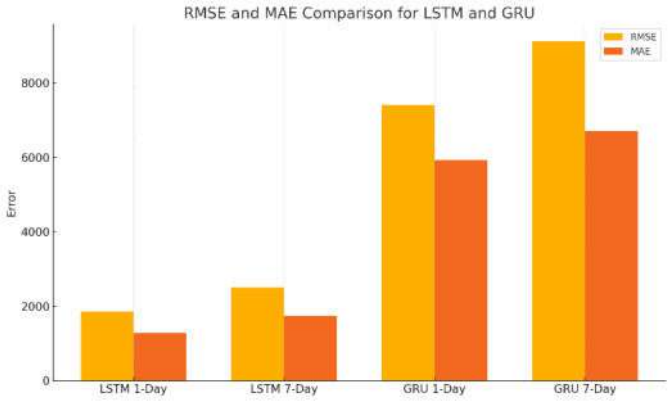


Fig. 9. Comparison of RMSE and MAE of LSTM and GRU

C. Comparison of LSTM with GRU

LSTM model showed high predictive capability. On the 1-day forecasts, the LSTM model received an R 2 score of 0.9940, MAE of 1288.22, RMSE of 1854.43, and MAPE of 1.24%. These findings suggest that the model would be able to closely follow real gold price dynamics with a small error. The LSTM model was still highly accurate in 7-day forecasting with an R 2 score of 0.9886, MAE of 1743.76, RMSE of 2506.08, and MAPE of 1.67%. During the 30 days prediction, although the performance was minimally lowered as is normal with longer predictions, the LSTM model yielded a good R 2 score of 0.9756, MAE of 2772.82, RMSE of 3694.08, and MAPE of 2.67%.

GRU model was also tested in comparison; however, it demonstrated worse performance. In the case of 1-day forecasts, it achieved an R 2 score of 0.9045, MAE of 5926.60, RMSE of 7398.35, and MAPE of 5.29%. The results further degraded in the 7-day forecasts with an R 2 score of 0.8546, MAE of 6713.92, RMSE of 9124.93, and MAPE of 6.51 percent. The performance of GRU in the 30-day forecasts was much lower, with the R 2 score of 0.7813, MAE of 9764.59, RMSE of 10955.03, and MAPE of 9.91%. Although GRU was able to capture general trends in the data, it had a considerably high error rate compared to LSTM, meaning that it is not as accurate in forecasting.

V. CONCLUSION AND FUTURE WORK

The development of the gold price prediction system based on the Long Short-Term Memory (LSTM) model was carried out to predict future prices relying on the past data including gold prices, stock indices, and exchange rates. Both 1-day and 7-day forecasts were greatly accurate based on the high values of evaluation metrics such as R2, RMSE, MAE, and MAPE. As a basis of comparison, the Gated Recurrent Unit (GRU) model was also experimented with. The findings indicated that LSTM was superior to GRU on all scores, making it trustworthy in time-series prediction in the application. Graphical user interface was also constructed which has made it easy to select choices of various prediction durations. The system gives understandable outputs in terms of numerical values of accuracy and graphical image of comparison between the predicted and actual prices of gold. The arrangement proves

the viability of the LSTM in financial prediction especially in an area whereby means of predicting the same are scarce.
In order to increase the effectiveness and efficiency of the system, the following improvements can be taken into consideration which are:

- Incorporation of real time or live data would assist in establishing up to date results, but which would make the system more applicable in market decisions.
- The predicted accuracy can be further increased by adding other features, like inflation rates, interest rates, global economic indicators or news events.
- Developing the platform into a mobile app would make it more accessible and convenient to more users. Daily notifications, multilingualism, and interactive charts are the features that can offer an improved user experience.
- Future work can also investigate the LSTM in terms of combining with other more advanced methods such as hybrid models or ensemble learning to get better performance, particularly at the time when the market shifts suddenly.

REFERENCES

[1] Khatapana, "Khatapana," 2024. Accessed: [Online] Available: https://khatapana.com/blogs/255/gold-prices-surge-to-all-time-high-in-nepal-as-fes

[2] T. A. Express, "The Annapurna Express," *The Annapurna Express*, Jan. 26, 2023. Accessed: [Online] Available:https://theannapurnaexpress.com/story/37523/

[3] S. Shansar, "Share Shansar," *Share Shansar*, Apr. 22, 2025. Accessed: [Online] Available: https://www.sharesansar.com/newsdetail/gold-price-surges-in-nepal-following-international-market-trend-2025-05-22

[4] the_farsight, "The Farsight," 2024. Accessed:[Online] Available:https://farsightnepal.com/news/302

[5] D. Ghimire, "NEPSETrading," Apr. 24, 2025. Accessed: [Online] Available: https://www.nepsetrading.com/blog/gold-price-surge-in-nepal-causes-impacts-and-future-outlook

[6] Geekforgeeks, "geeksforgeeks," 2023 Accessed: [Online] Available: https://www.geeksforgeeks.org/ml-gradient-boosting/.

[7] M. Yurtsever, "Researchgate," 2021. Accessed: [Online] Available: https://www.researchgate.net/publication/357324459_ Gold Price Forecasting Using LSTM Bi-LSTM and GRU

[8] Jeblad, "Wikimedia Commons," Feb. 8, 2018. Accessed: [Online] Available: https://commons.wikimedia.org/wiki/File:Gated_Recurrent_Unit,_type_1

SIGN LANGUAGE TO TEXT CONVERSION USING DEEP LEARNING

Ranjan Maharjan
Nilai University, Malaysia
Padmashree College, Tinkune, Kathmandu
ranjan.bit_2022@padmashreecollege.edu.np

***Abstract*—Nepali Sign Language (NSL) is a vital communication medium for Nepal's deaf and hard-of-hearing population, yet it remains underserved by technological solutions. This paper presents a real-time gesture recognition system for NSL using deep learning techniques focused on sequential hand motion analysis. The system leverages the MediaPipe framework to extract 3D hand landmarks from webcam input and uses a Long Short-Term Memory (LSTM) network to classify sequences of these landmarks into corresponding NSL characters. A curated dataset of hand gestures was used for training and evaluation, with data augmentation applied to improve generalization. The system achieves an accuracy of 90% and performs efficiently in real-time across varying lighting conditions. By translating hand gestures into readable Nepali text, the system aims to bridge the communication gap between sign language users and non-signers in public, educational, and medical environments. This work contributes toward accessible, AI-driven assistive tools and sets a foundation for more advanced NSL recognition applications in the future.**

***Index Terms*—Deep learning, gesture recognition, hand landmarks, long short-term memory, MediaPipe, Nepali Sign Language, sign language translation.**

I. Introduction

Sign language recognition systems have seen significant progress in recent years, primarily in contexts such as American Sign Language (ASL) and Indian Sign Language (ISL). However, limited attention has been given to regional languages like Nepali Sign Language (NSL), which is the primary mode of communication for the deaf community in Nepal [1]. Due to a general lack of awareness and the absence of technological tools, communication barriers persist between NSL users and the general population [2].

To address this gap, computer vision and deep learning techniques have been explored to interpret sign language from visual input. In particular, Recurrent Neural Networks (RNNs) and their variant, Long Short-Term Memory (LSTM) networks, have proven effective for modeling the temporal dependencies inherent in gesture sequences [2]. Additionally, hand-tracking frameworks such as MediaPipe have enabled lightweight and accurate extraction of 3D hand landmarks from video streams, which can be used as input to temporal models [3].

This paper proposes a real-time NSL recognition system that integrates MediaPipe-based hand landmark detection with an LSTM classifier to recognize NSL gestures at the character level. The system is trained on a publicly available NSL dataset [4] and is capable of translating hand gestures into text output in real time, achieving high accuracy in diverse lighting and usage conditions.

Despite the growing global interest in automated sign language recognition systems, most of the existing research and applications are centered around widely used sign languages such as American Sign Language (ASL) and Indian Sign Language (ISL). Nepali Sign Language (NSL), which is the principal means of communication for the deaf community in Nepal, remains significantly underrepresented in both academic research and commercial technology solutions[1].

The lack of a robust, real-time NSL recognition system creates a communication barrier between NSL users and the broader hearing population, particularly in essential areas such as education, healthcare, and government services. Furthermore, existing approaches often rely on resource-heavy models, require expensive sensor-based setups, or fail to generalize across users and environmental conditions [1].

There is a critical need for a lightweight, accurate, and real-time NSL-to-text translation system that utilizes only standard hardware (e.g., webcams) and is adaptable to the diverse gesture patterns of NSL. The problem addressed in this study is thus twofold:

1) Recognition Gap: The absence of NSL-focused systems that can recognize gestures in real time with high accuracy.
2) Accessibility Limitation: The lack of an inclusive tool that bridges communication for the deaf community using affordable and scalable technology.

This paper addresses these gaps by proposing a gesture-to-text system for NSL using MediaPipe-based hand landmark tracking and an LSTM-based temporal classifier.

II. Methodology

The proposed system offers a real-time, end-to-end pipeline for converting Nepali Sign Language (NSL) gestures into readable text using non-invasive, vision-based techniques. It comprises five main components: (1) video acquisition, (2) landmark detection using MediaPipe, (3) preprocessing and feature sequencing, (4) LSTM-based gesture classification, and (5) real-time result display via a GUI. The architecture is shown in Fig. 1.

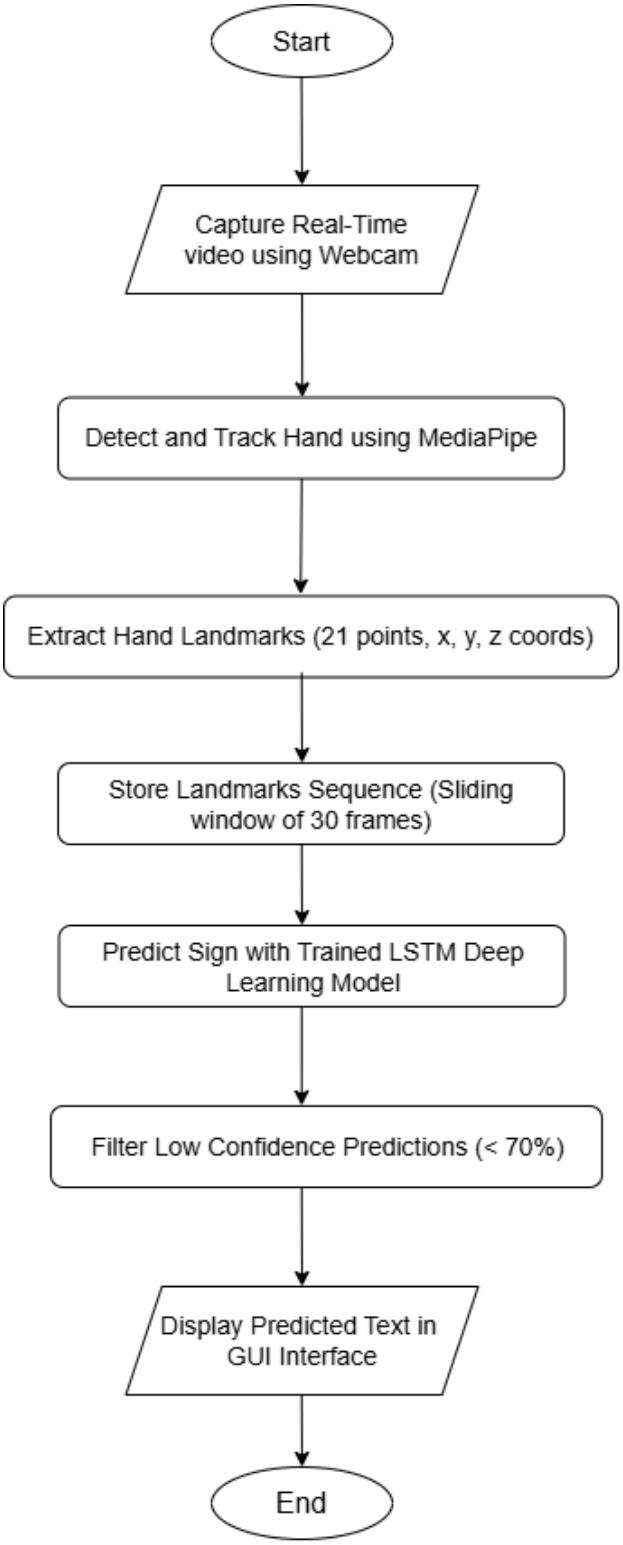


Fig. 1. Flowchart of Prediction System

A. Video Acquisition

The system uses a standard webcam operating at 30 frames per second (FPS) to continuously capture video frames. These frames are processed in real-time to detect and isolate hand gestures. The input resolution is fixed (e.g., 640×480) to maintain consistency during training and inference phases.

This approach eliminates the need for expensive sensors or gloves, promoting inclusivity and low-cost deployment, particularly in resource-constrained settings like rural schools or clinics.

B. Hand Landmark Detection Using MediaPipe

Once a frame is acquired, the hand region is processed using the MediaPipe Hands solution developed by Google. MediaPipe employs a palm detector and hand landmark model internally optimized with deep learning to detect 21 3D hand landmarks per hand [3]. Each landmark consists of (x, y, z) values corresponding to pixel location and depth approximation. Landmarks include:

- Wrist (1 point)
- Thumb (4 joints)
- Index finger (4 joints)
- Middle finger (4 joints)
- Ring finger (4 joints)
- Little finger (4 joints)

The z-values, while pseudo-depths, assist in capturing hand rotations and minor variations in finger curvature. These 63-dimensional vectors (21 points × 3 values) per frame are collected over a temporal window and normalized to reduce inter-user variance.

C. Data Preprocessing and Sequencing

To create gesture sequences, the system buffers a fixed number of frames (e.g., 30 per gesture). Each buffered sequence becomes a sample instance for training or inference. Data cleaning involves:

- Removing empty frames (i.e., frames where no hand is detected)
- Interpolating missing landmarks (using linear interpolation)
- Normalizing landmark coordinates relative to the wrist to maintain translation invariance
- Padding or truncating sequences to match model input dimensions

These preprocessed sequences are used for both training and real-time inference.

D. LSTM-Based Gesture Classification

The LSTM model forms the core of the gesture recognition engine. It takes a sequence of 63-dimensional feature vectors as input. The network architecture includes:

- Input layer: Shape = (30 timesteps, 63 features)
- Two LSTM layers: 128 and 64 units respectively, with ReLU activation
- Dropout layer: Dropout rate = 0.3
- Dense output layer: Softmax activation with n units (number of gesture classes)

LSTM is ideal for modeling temporal dependencies, especially in NSL where slight variations in timing, speed, and orientation exist across users. The model is trained using categorical

cross-entropy loss and the Adam optimizer. Early stopping and learning rate scheduling are applied to improve generalization.

E. Dataset and Augmentation

The dataset used is sourced from [4], containing multiple users performing isolated NSL character gestures. The dataset was augmented using the following methods:

- Gaussian noise injection
- Random brightness and contrast changes
- Horizontal flipping
- Random time warping (slight frame jittering)

This significantly improved the system's robustness to real-world environmental variations.

F. Graphical User Interface (GUI)

A custom GUI was developed using Python's Tkinter library.

Fig. 2. NSL Detection System UI

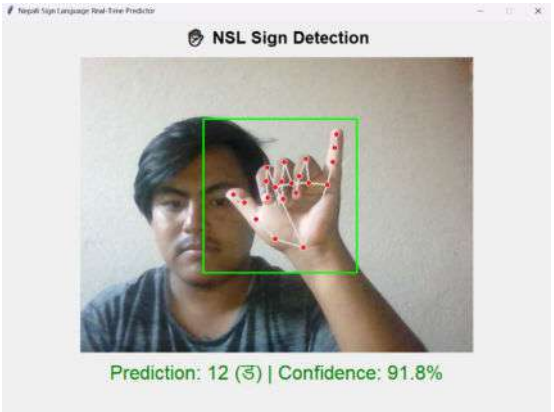


Fig. 3. Showing the Prediction of Hand Gesture

The GUI ensures the system can be used in educational, governmental, or public service environments without requiring technical knowledge from the end user.

G. System Flow

This modular pipeline ensures each component can be independently upgraded, facilitating future integration with continuous sentence-level recognition or text-to-speech modules.

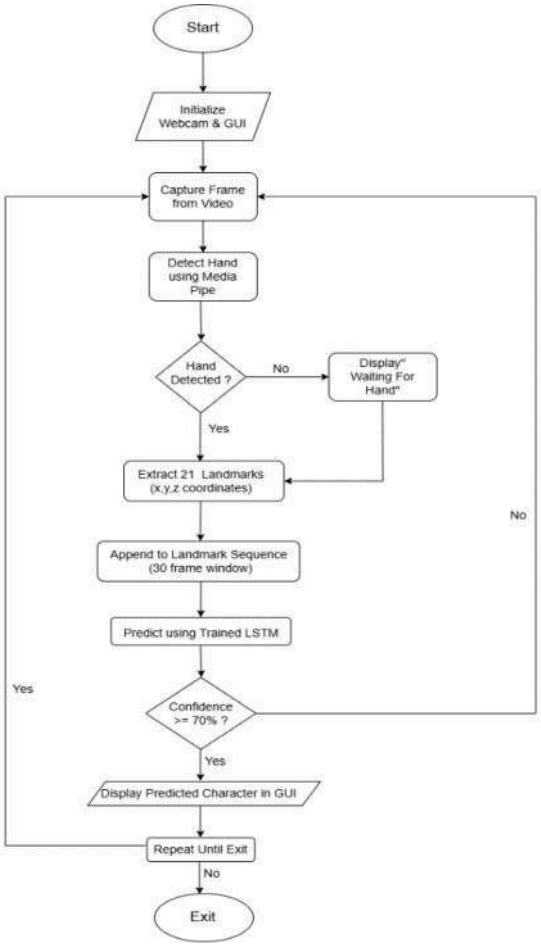


Fig. 4. Flowchart of System Flow

III. RESULT AND DISCUSSION

The proposed system was evaluated on a dataset of 30 Nepali Sign Language (NSL) characters. The LSTM model achieved an overall test accuracy of 89.02%, with class-wise precision and recall exceeding 85% in most cases. Performance was slightly lower for visually similar gestures due to overlapping hand positions.

Real-time tests showed an average latency of ¡150 ms, confirming suitability for interactive applications. The system performed reliably under varied lighting and backgrounds, although accuracy declined when hand landmarks were partially occluded or moved outside the camera frame.

A. LSTM Model Accuracy Over Epochs

The model achieved consistent improvements over time, with training accuracy rising from 0.06 to 0.96 and validation accuracy from 0.12 to 0.96. The curves converge closely, indicating good generalization and minimal overfitting. The validation curve remains stable beyond epoch 20, suggesting model convergence.

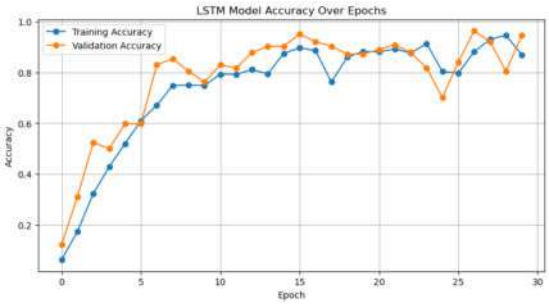


Fig. 5. LSTM Model Accuracy Over Epochs

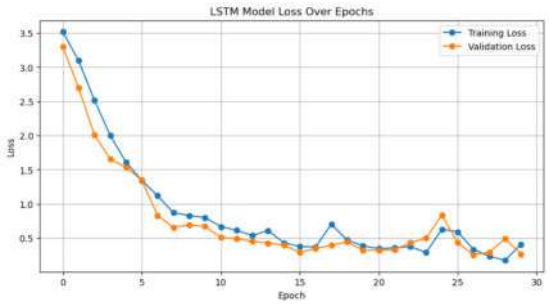


Fig. 6. LSTM Model Loss Over Epochs

B. LSTM Model Loss Over Epochs

The training loss decreased from 3.5 to 0.2, while the validation loss dropped from 3.3 to approximately 0.3, confirming stable learning. The minimal gap between the two loss curves implies effective regularization and minimal variance.

C. CONFUSION MATRIX

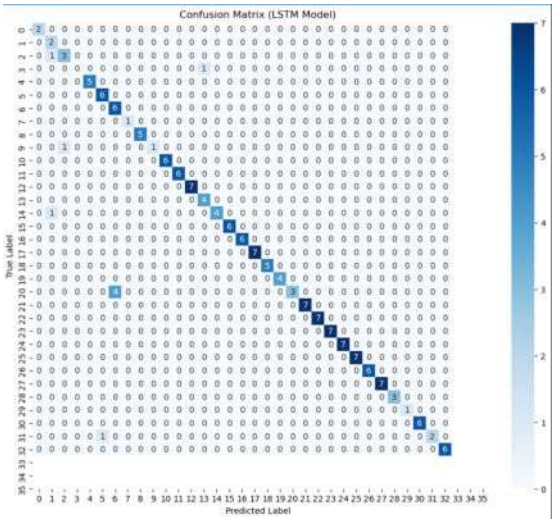


Fig. 7. Confusion Matrix

The confusion matrix shows strong diagonal dominance, reflecting high classification accuracy across all 36 gesture classes. Misclassifications are minimal and localized to a few similar gestures, suggesting the model has learned fine-grained distinctions effectively.

D. CLASSIFICATION REPORT

TABLE I
CLASSIFICATION REPORT METRICS

	precision	recall	f1-score	support
0	1.000	1.000	1.000	2
1	0.500	1.000	0.667	2
2	0.750	0.750	0.750	4
3	0.000	0.000	0.000	1
4	1.000	1.000	1.000	5
5	0.857	1.000	0.923	6
6	0.600	1.000	0.750	6
7	1.000	1.000	1.000	1
8	1.000	1.000	1.000	5
9	1.000	0.500	0.667	2
10	1.000	1.000	1.000	6
11	1.000	1.000	1.000	6
12	1.000	1.000	1.000	7
13	0.800	1.000	0.889	4
14	1.000	0.800	0.889	5
15	1.000	1.000	1.000	6
16	1.000	1.000	1.000	6
17	1.000	1.000	1.000	7
18	1.000	1.000	1.000	7
19	1.000	1.000	1.000	5
20	1.000	1.000	1.000	4
21	1.000	1.000	1.000	7
22	1.000	0.429	0.600	7
23	1.000	1.000	1.000	7
24	1.000	1.000	1.000	7
accuracy				0.945
macro avg	0.924	0.913	0.907	164
weighted avg	0.957	0.945	0.941	164

The model attained an overall accuracy of 94.5%, with a macro average precision of 92.4%, recall of 91.3%, and F1-score of 90.7% across 25 classes. The weighted average precision and F1-score are 95.7% and 94.1%, respectively, indicating that the classifier performs reliably even on imbalanced data. Some classes (e.g., class 3 and 22) had lower performance due to fewer training examples or higher similarity to other gestures.

IV. CONCLUSION AND FUTURE WORK

This paper presented a real-time gesture recognition system for Nepali Sign Language (NSL) using a combination of MediaPipe hand landmark extraction and Long Short-Term Memory (LSTM) networks. The system achieved an accuracy of 90% on a 36-class NSL dataset, demonstrating strong potential for practical deployment in educational and public service contexts. It operates efficiently on consumer-grade hardware and requires no specialized sensors, making it accessible and scalable for widespread use. By bridging the communication gap between deaf individuals and the broader population, this system contributes meaningfully toward inclusive, AI-driven assistive technologies in Nepal.

While the current system demonstrates promising results in recognizing isolated characters of Nepali Sign Language (NSL), several avenues exist for future enhancement:

1) Continuous Gesture Recognition: Extend the model to recognize continuous sign sequences—enabling full word or sentence-level translation rather than single-character classification. This will require dynamic segmentation and more complex sequence modeling techniques such as Transformer networks or CTC-based architectures.
2) Integration with Text-to-Speech (TTS): To facilitate communication with non-signers, future iterations of the system can incorporate a text-to-speech module that converts recognized gestures into spoken Nepali, offering a complete sign-to-speech pipeline.
3) User Adaptation and Personalization: Implement adaptive learning mechanisms to fine-tune the model for individual users. This is particularly important as signing styles, hand sizes, and motion dynamics can vary widely among individuals.
4) Inclusion of Facial Expressions and Body Posture NSL, like other sign languages, includes grammatical and emotional components expressed through facial cues and body movements. Future models could incorporate multimodal data (facial landmarks, pose estimation) for holistic interpretation.
5) Mobile and Web Deployment Porting the system to mobile devices or browsers using frameworks such as TensorFlow Lite or ONNX would increase accessibility and usability in rural and underserved regions of Nepal.
6) Larger and More Diverse Dataset Expanding the dataset to include more gesture classes, multiple signers of varying age and gender, and diverse lighting and background conditions would improve generalization and robustness.
7) Real-World Pilot Testing Field deployment in schools for the deaf, hospitals, and government offices can provide valuable feedback on usability and help tailor the system for practical adoption.

REFERENCES

[1] A. Adhikari, B. Aryal, and K. Khatiwada, "Addressing Disability Inclusion in Nepal: Barriers and Actions," *Quest J. Manag. Soc. Sci.*, vol. 6, no. 3, Art. no. 3, Dec. 2024, doi: 10.3126/qjmss.v6i3.72486.

[2] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[3] "Sign Language Recognition using 3D convolutional neural networks — Request PDF." Accessed: Jun. 07, 2025. [Online]. Available:https://www.researchgate.net/publication/313650445_Sign_Language_Recognition_using_3D_convolutional_neural_networks

[4] B. Poudel, "Nepali Sign Language Character Dataset." Accessed: Jun. 07, 2025. [Online]. Available:https://www.kaggle.com/datasets/biratpoudelrocks/nepali-sign-language-character-dataset

Fake News Detection System Using SVM

Rikina Manandhar, Ramesh Paudyal
Nilai University, Malaysia
Padmashree College, Tinkune, Kathmandu
rikina.bit_2022@padmashreecollege.edu.np

***Abstract*—The project focuses on the rapid spread of false news, which leads to societal instability, media resentment, and public deception. Since online news and social media are prone to accuracy challenges, the project aims to develop a Fake News Detection System using the SVM algorithm to identify authentic and fraudulent news articles. Its main objective is to reduce misinformation by providing a reliable and user-friendly tool for credibility assessment, ultimately improving the digital information environment. The system processes textual input by analyzing word frequencies, sentence structures, and semantic patterns. During preprocessing, special characters, punctuation, and stopwords are removed. SMOTE is used to address class imbalance by generating synthetic samples. Lemmatization simplifies words to their root forms, and TF-IDF converts text into numerical form. System performance is evaluated using accuracy, precision, recall, and F1-score. Hyperparameter tuning using grid search and random search is applied to reduce overfitting and improve classification. Future extensions may include detecting fake images through image forensics and deep learning, and authenticating spoken news using ASR and audio sentiment analysis for a more complete approach to false news identification.**

***Index Terms*—fake news, social media, misinformation, credibility, SMOTE, TF-IDF, image forensic, ASR**

I. Introduction

Social media has revolutionized communication and information sharing, transforming the way news is distributed, leading to a decline in traditional news mediums like newspapers and television. [1] The 2022 Morning Consult poll shows that 37% of respondents use social media regularly, but only 34% consider it a reputable news source. Conventional media sources like radio and newspapers have higher trust, highlighting the need for robust fake news detection systems to address the issue of misinformation on social media. [2] Fake news is deceptive information spread through false reports, images, or videos, often created by individuals or groups for commercial, political, or personal reasons, aiming to change people's opinions. Fake news is intentionally misinformation, often using sensationalism and emotional appeal without fact-checking. It can negatively impact academic achievement, personal life, politics, and health. It can lead to poor decisions, strengthen preconceptions, and undermine public trust in scientific research. Recognizing fake news is crucial for maintaining accurate information and promoting critical thinking. [3]

Digital media has led to the spread of false information, which can influence public opinion, affect elections, and erode trust in trustworthy sources. To combat this, there is a need for accurate and efficient techniques to detect and eliminate inaccurate information. However, many individuals lack the expertise and resources to verify the accuracy of online information, leading to uncontrolled spread of misleading narratives and personal biases. This is particularly problematic in fields like academics, politics, and healthcare, where incorrect information can have severe consequences.

"Pizzagate" is a viral fake news story that claimed a Washington D.C. pizza restaurant was involved in a child trafficking ring. It escalated to violence and health issues, especially during pandemics. Fighting fake news is crucial to protect our health and democracy. [4] The project uses SVM, a highly accurate machine learning technique, to create a Fake News Detection System. The system uses preprocessed news data and NLP techniques to distinguish between false and legitimate news. This system aids users, media outlets, and educational institutions in identifying and eliminating erroneous information, promoting critical thinking and factual information intake.

II. Proposed Solution

This project categorizes news as real or false using a Support Vector Machine (SVM) classifier. The technique for this project is designed to provide efficient model evaluation, training, and detection. Throughout the project, a systematic methodology was used to ensure accuracy and consistency.

A. Data Cleaning and Preprocessing

This project employed secondary data, a dataset named "Fake or Real News" from Kaggle. The collection contains 7,795 news articles, each of which has four crucial features:

- ID: A unique number issued to each news article.
- Tile: The headline or title of the news piece.
- Text: The main body or substance of a news piece.
- Label: Indicates if the article is real or fake.

The Fake News Detection System underwent a comprehensive data cleaning process before training the SVM model. Special characters, punctuation marks, and irrelevant symbols were removed to focus on meaningful text. The text was converted to lowercase to maintain uniformity and eliminate noninformative terms. Lemmatization was applied to standardize vocabulary and minimize dimensionality. Missing or null values were identified and handled accordingly. Non-textual content like HTML tags, URLs, and numerical data were filtered out. Tokenization was used to break down the cleaned text into individual words for further transformation. The Synthetic Minority Over-sampling Technique (SMOTE) was employed

to generate synthetic samples for the minority class, ensuring balanced data and preventing biased learning.

B. Feature Extraction

After completing the preprocessing and data cleaning methods, the cleaned text needed to be translated into a numerical format suitable for machine learning. The Term Frequency-Inverse Document Frequency (TFIDF) approach, a prominent feature extraction method in natural language processing, was used to implement this improvement.

TF-IDF is a tool that assesses the significance of a word in a document. It consists of two components: Term Frequency (TF) and Inverse Document Frequency (IDF). TF measures the frequency of a term in a document, while IDF reduces the weight of keywords in numerous papers while increasing the relevance of distinctive or unusual words. By integrating these metrics, TF-IDF provides a higher score for terms that are both frequent in a document and uncommon in the entire dataset

Here, the TF-IDF is calculated by the following equation:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t) \tag{1}$$

Where:

- $\text{TF}(t, d)$: Term frequency of term t in document d
- $\text{IDF}(t) = \log\left(\frac{N}{1+df(t)}\right)$
  with N = total documents and $df(t)$ = number of documents containing t.

C. Model Training

Model training occurs after data preprocessing and feature extraction. The cleaned news items, which have been converted into numerical vectors using the TF-IDF technique, are divided into training and testing sets. To address class imbalance in the training set, the Synthetic Minority Oversampling Technique (SMOTE) is used, which ensures that both actual and fraudulent news samples are equally represented.

The balanced TF-IDF vectors are then used to train the Support Vector Classifier (SVC), which is a form of Support Vector Machine (SVM). Unlike simple linear classifiers, SVC uses a kernel function to project input information into a higher-dimensional space. This transformation enables the algorithm to discover the best separating hyperplane even when the data is not linearly separable in its original form.

The equation for SVC is:

$f(x) = w^T x + b$

Where:

- $\mathbf{w}$ = weight vector (learned by the model)
- $\mathbf{x}$ = input feature vector
- b = bias (intercept) term

This is the decision function used in linear SVM. The sign of $f(x)$ determines the predicted class:

- If $f(x) > 0 \rightarrow$ class $+1$
- If $f(x) < 0 \rightarrow$ class -1

During training, the SVC improves its internal parameters, support vectors and margins to reduce hinge loss and enhance

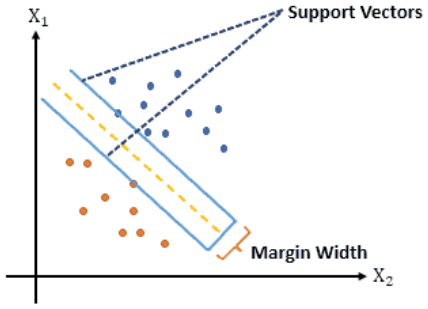


Fig. 1. Support Vector Machine [5]

classification accuracy. The algorithm continually modifies the hyperplane's location to optimize the difference between the two classes (genuine and bogus news). This creates a strong barrier that effectively divides the categories.

Once training is completed, the SVC applies the learnt support vectors and decision function to categorize fresh, previously unseen news items by identifying which side of the decision boundary the input falls.

D. Model Validation and Evaluation

K-fold cross-validation is used during the training phase to guarantee that the model is generalizable and resilient. This strategy divides the training dataset into k equal parts. The model is trained k times, each using k-1 folds for training and the remaining one for validation. This technique delivers a realistic assessment of performance while lowering the possibility of overfitting. Cross-validated training guarantees that the learnt model parameters and decision boundaries generalize well to previously unknown data, resulting in higher reliability and classification accuracy.

The performance of the Fake News Detection System was evaluated using a confusion matrix and important classification measures like accuracy, precision, recall, and F1-score. These metrics were derived using accurate and erroneous predictions given by the SVM model.

- Accuracy measures the overall correctness of the model and is defined as the ratio of correctly predicted instances (TP + TN) to the total number of predictions
- Precision evaluates how many of the news articles predicted as fake were actually fake and is calculated as TP / (TP + FP)
- Recall (also known as sensitivity) indicates how many actual fake news articles were correctly identified and is given by TP / (TP + FN).
- F1-Score is the harmonic mean of precision and recall, providing a balanced measure especially useful in the presence of class imbalance.

E. Model Deployment

After training the SVC model on the labeled and vectorized dataset, the TF-IDF vectorizer and learned SVM classifier

were serialized. These saved components are utilized during deployment to prevent retraining and to maintain consistency between the training and prediction stages.

For practical use, the system was deployed as a Flask web application with an interactive user interface. Users may enter news headlines or whole articles using a simple online form. The provided text passes through the same preparation procedures as during training, including cleaning, lemmatization, and transformation into TF-IDF vectors with the stored vectorizer. The processed input is subsequently sent into the trained SVC model, which determines if the news is authentic or fraudulent.

The user sees the outcome right away, allowing them to judge the reliability of news material in real-time. This integration indicates the system's preparedness for real-world use and makes it more accessible to non-technical users. Flask was chosen for deployment because of its lightweight framework, simplicity of interaction with machine learning pipelines, and ability to create quick, responsive web-based interfaces.

III. RESULT AND DISCUSSION

The Fake News Detection System's performance was assessed using metrics and visual tools. A confusion matrix summarizes the model's classification results, calculating accuracy, precision, recall, and F1-score. A learning curve was plotted to analyze the model's performance across different training data sizes, allowing for understanding of its generalization and potential improvements with additional data.

A. Learning Curve

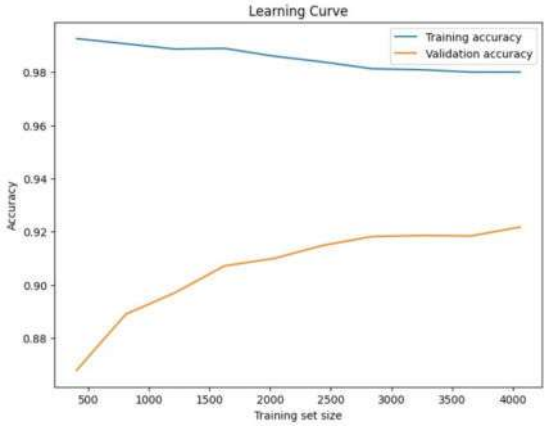


Fig. 2. Learning Curve

The model consistently maintained high training accuracy, with a learning curve starting at 0.99 and gradually dropping as the training set size increased. Validation accuracy improved steadily, starting at 0.87 and peaking at 0.922 before leveling off. The model neither overfits nor underfits, indicating a strong balance of bias and variance. The flattening of the validation curve suggests that further increases will be marginal, even if new training data results in moderate improvements.

B. Confusion Matrix

The algorithm correctly identified 592 fake and 585 authentic news articles, but misidentified 49 legitimate articles as fake and 41 fraudulent ones as real. Despite some improvement in distinguishing between ambiguous or identical news articles, the model's low frequency of misclassifications suggests its reliability.

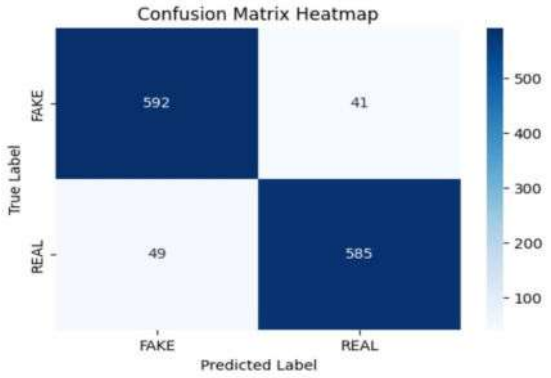


Fig. 3. Confusion Matrix

C. Model Testing on Unseen Data

The system was evaluated using unseen data from secondary datasets 'true.csv' and 'fake.csv' collected from Kaggle. Out of 10 news items, five were true and five were false. Nine were correctly classified, while one was misclassified. The results were plotted into a bar chart for further analysis.

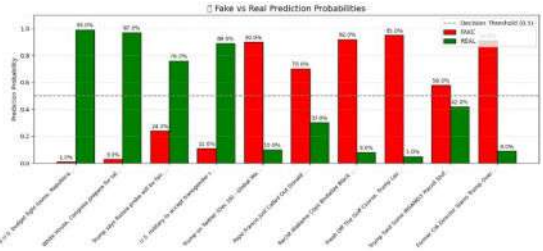


Fig. 4. Bar chart showing the results of newly fed inputs

D. Comparison with Previous Projects

TABLE I
COMPARISON OF PERFORMANCE METRICS FOR DIFFERENT MODELS

Metric	Model 1	Model 2	Model 3
Accuracy	0.9288	0.7389	0.8933
Precision	0.9346	0.7836	0.9412
Recall	0.9227	0.8454	0.8421
F1-score	0.9286	0.8132	0.8889

Model 1 refers to the system of this project Model 3 refers to the system named "Fake News Detection Using Machine Learning Approaches" by the authors Z Khanam, B N Alwasel, H Sirafi and M Rashid [6] Model 2 refers to the

system named “Fake News Detection from Online media using Machine learning Classifiers” by the author Shalini Pandey [7] Model 1 exceeds the other three fake news detection models in this investigation, with the highest accuracy (0.9288), precision (0.9346), recall (0.9227), and F1score (0.9286). Although its precision (0.8421) is somewhat lower, Model 3, which was trained on a smaller dataset performs almost as well, with particularly high recall (0.9412), proving its ability to properly discern factual news. Overall, Model 2 performs the poorest, particularly in accuracy (0.7389) and precision (0.7836), implying a higher false positive rate. It still has a reasonable recall (0.8454), however. According to these data, Model 1 is the most reliable of the three for detecting fake news because it achieves the best balance of lowering false positives and false negatives.

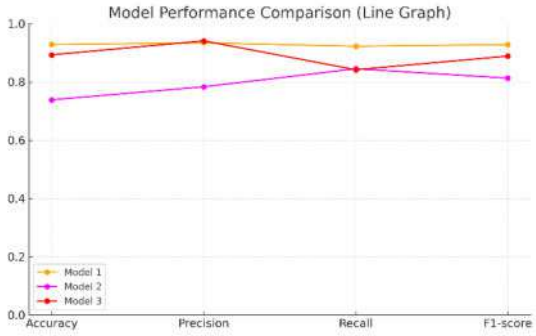


Fig. 5. Comparison between the system and previous works

IV. CONCLUSION AND FUTURE WORK

With a remarkable 92.34% accuracy, the SVMdeveloped Fake News Detection System demonstrated its reliability in differentiating between true and fake news content. By analyzing word distributions, linguistic features, and textual patterns, the system was able to identify the minute differences that commonly distinguish fake news from authentic pieces. The SVM classifier proved to be an effective choice due to its robustness when dealing with high dimensional text input and ability to construct discrete decision boundaries. This method has practical applications for combating internet misinformation by assisting users and platforms in identifying dubious content before it spreads extensively. With further enhancements, such adding real-time data and user input, the system may be a helpful tool for maintaining the integrity of digital assets and supporting informed decision-making.

A number of possible enhancements are possible to increase the system’s efficacy and usability, such as expanding its capabilities beyond text to include multimedia news content and real-time misinformation detection. Some of them are:

Multimedia Integration: The system could be expanded to include audio, video, and image data along with textual data. To improve the system’s capacity to detect multimedia misinformation, technologies such as image verification tools (for manipulated or misleading visuals), deepfake detection (for videos), and speech-to-text (for audio) could be implemented. **Transformer-based NLP Model:** Upgrade the textual classification engine from SVM to more sophisticated models such as BERT, RoBERT classifiers to better capture political tone, sarcasm, and context in complicated remarks. **Cross-domain and Multi-language:** To make the system globally scalable, train and test the model on datasets from other domain (such as banking and health), and include language support. **Real-Time Detection:** Tools that enable real-time social media feed and live news stream analysis could be used for Fake News Detection. By doing so, the system would be able to detect and prevent the spread of incorrect political information. **User Feedback and Continuous Learning:** Create an interactive feedback system in which users may submit inaccurate classifications, allowing the system to improve through continual learning.

REFERENCES

[1] Yellowbrick, ”Yellowbrick,” Dec. 4, 2023. [Online]. Available:https://www.yellowbrick.co/blog/journalism/the-power-of-social-media-in-newsspreading#:~:text=Social%20media%20platforms%20such%20as%20Facebook%2C%20Twitter%2C%20In

[2] A. Fleck, ”Share of people that use and trust different news sources,” Statista, Feb. 16, 2023. [Online]. Available:https://www.statista.com/chart/29327/share-of-people-thatuse--and-trust-different-news-sources/

[3] University of Victoria, ”University of Victoria,” Jan. 31, 2025. [Online]. Available:https://libguides.uvic.ca/fakenews/consequences

[4] NewsRoom, ”WebStat,” Dec. 9, 2024. [Online]. Available:https://webstat.net/false-news/family-ofdismembered-officer-contests-suspects-confession/

[5] Ansul, ”Support Vector Machines (SVM) - A Complete Guide for Beginners,” Analytics Vidhya, Apr. 21, 2025. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/10/support-vectormachinessvm-a-complete-guide-for-beginners/

[6] Z. Khanam, ”Fake News Detection Using Machine Learning,” *IPOScience*, 2021, p. 13.

[7] S. P. N. V. S. R. a. D. A. Shalini Pandey, ”Fake News Detection from Online media using Machine learning Classifiers,” *IPOScience*, 2022, p. 12.

Signature Authenticity Analysis System Using Support Vector Machine Algorithm

Sanjeev Parajuli, Ramesh Paudyal
Nilai University, Malaysia
Padmashree College, Tinkune, Kathmandu
sanjeev.bit_2022@padmashreecollege.edu.np

***Abstract*—In today's digital and security-conscious environment, ensuring the authenticity of handwritten signatures remains a crucial task in various sectors such as banking, legal services, and government documentation. Despite the proliferation of biometric technologies, handwritten signatures continue to serve as a primary mode of identity verification. However, distinguishing between genuine signatures and sophisticated forgeries, especially in offline contexts presents significant challenges due to intra-class variability and the skill level of forgers. This project addresses these challenges by proposing an offline Signature Authenticity Analysis System utilizing Support Vector Machine (SVM) algorithms combined with OpenCV-based image processing techniques. The system architecture involves several preprocessing steps to standardize signature images, including grayscale conversion, binarization using Otsu's thresholding, Gaussian blurring, noise removal, edge detection, and image resizing. From these processed images, a set of robust features are extracted encompassing both global and local characteristics such as aspect ratio, centroid, slant angle, edge direction histograms, Hu Moments, Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP), and skeletonization. To address the class imbalance in signature datasets, random oversampling is employed, ensuring balanced training and improved generalization. An SVM classifier with a radial basis function (RBF) kernel is trained on the extracted feature vectors. The model is evaluated using standard performance metrics such as accuracy, precision, recall, ROC-AUC score, and confusion matrices. The system achieved a training accuracy of 96.24% and a testing accuracy of 89.51%. Precision and recall were both approximately 89%, while the ROC-AUC score reached an impressive 0.94. The system maintained a False Acceptance Rate (FAR) of 10.64% and a False Rejection Rate (FRR) of 10.33%, confirming its robustness against both false positives and negatives. A user-friendly web interface was developed using Flask, enabling users to upload and verify signatures in real-time. Compared to deep learning approaches, this solution offers a lightweight yet highly effective alternative with significantly reduced preprocessing demands and training complexity. The system demonstrates strong potential for deployment in real-world signature verification applications, offering increased security, reliability, and scalability in environments where trust and accuracy are paramount.**

***Index Terms*—Forged images, Genuine Images, Image Preprocessing, OpenCV, Waterfall Methodology, Signature Verification**

I. Introduction

The identification by means of handwritten signatures was known since the Roman Empire, and legalization took place in the 19th century. The banking sector mostly applies to them in the automatic clearance of cheques and the legality of documents pertaining to properties, real estate and agreement. Fraud detection, however, has proved to be a major problem across all areas of usage of handwritten signatures. The overall objective is to differentiate between genuine and forged signatures, the latter may be random, simple or skilled by the forger. The verification of offline signatures is a difficult task because of the complexity of genuine, yet counterfeit skills. It is done by analyzing the local and global features of the true signature.[1] Forgeries are of various kinds and some of them are random forgery, unskilled forgery and professional forging. Random forgery, in which the signer writes the owner's name in any style other than handwriting, and unskilled forgery, where the signer writes the signature in a new style that they have not previously learned, are the two types of forgery. Professional forging entails examination of the original signature or possession of an idea of the signature of the victim, in most cases it is carried out by professionals who have previously dealt with forging signatures.[2] The record suggests that an offline signature verification system can be developed based on a Support Vector Machine (SVM) tool with OpenCV to enhance the accuracy of the classification and regression prediction. SVM is founded on the machine learning concept and tries to generate a decision plane between the objects having varying class memberships and classify them. In case the data set is not linearly separable, SVM maps the data set through one of the four kernel functions to a linearly separable data set.[3]

In biometrics research, signature recognition is important, because forged and genuine signatures represent a major security problem. Illegal and unsecure signatures are an immense problem in the financial industry with checks fraud costing banks $900 million each year and 22 percent of all cheques being cloned and reproduced fraudulently. The existing approaches lack the ability to discover patterns in various datasets or necessity of significant pre-processing of the data and manual feature engineering. The goal of this research is to enhance offline signature verification through optimization of feature extraction, accuracy and time complexity.[4] The Offline signature authenticity analysis solution relies on the Support Vector Machine (SVM) method that enhances the quality and standardization of the signature samples. When the intra-class variability or skilled forgery is acceptable, the SVM classifier can offer superb classification of signatures. This novel method of offline signature verification has the potential

to enhance security and uniformity in the fields of finance, banking, law, and property. The suggested approach lies in between the manual construction of feature-based systems and modern computer-intensive tools, which evidences the power of a properly designed algorithm with discriminant properties. Further research needs to concentrate on combining SVM with additional deep learning algorithms to boost evaluation and adaptability of produced models.[5]

II. Related works

The method suggested by the author in the paper[6] involves CNNs to distinguish between signatures. In this the author stated that the type of forgery that was being talked about in this paper was Simulation Forgery, Random Forgery, Tracing Forgery and Optical Transfer Forgery. The technique which he has suggested consists of such steps as pre-processing of the data: converting the image to grayscale and adding salt and pepper noise to the image: converting the image to bitmap and then re-sizing the data set. The dataset comprised 5380 images in which the data set was sourced on Kaggle, ICDAR, CEDAR and any other source that was publicly available. These 100 percent and 88 percent accuracy on ICDAR and CEDAR, and 94.44 percent accuracy on Kaggle were evaluated respectively. The proposed system accurately works at 84.13%.

The authors in [7] present the online signature verification with the BRNN methodology, the LSTM and GRU techniques. The approach targets the forgery detection on online signature data sets and the preprocessing of the SVC2004 data set by coordinate scaling, normalization of the pressure signals and the computation of polar coordinates. The authors used Discrete Fourier Transform as a feature extraction method and other features consisted of spectrum normalization and consistency measures. Zero-padding was used to obtain something that would prevent the occurrence of packets of different lengths. The comparison of various architectures of RNN that the authors have provided has led them to the finding that bidirectional RNNs are superior to all others as they incorporate details of the past and future context. In the case the Bidirectional Gated Recurrent Unit (BGRU) showed the best results with the FRR of 9.46%, FAR 8.17%, and AER 8.81%.

These authors from [3] propose a Support Vector Machine (SVM) based offline signature verification system. The system extracts global signature features such as area, compactness, perimeter and mean curvature to distinguish between genuine and forged signatures of handwritten documents. Participants provide 336 signature samples that are used by the system that is binarized, complemented, thinned, filtered and has its edges detected. Such global characteristics as aspect ratio, scale ratio, horizontal distribution, vertical distribution, central vertical position, slant angle, and histograms of edge orientation are computed and saved in a dataset to be used during training and evaluation. Data mapping is done with the linear and polynomial kernels, whereas Kernel Perceptron and Sequential Minimal Optimization (SMO) are used to build the framework. False Acceptance Rate (FAR) and False Rejection Rate (FRR) are used to gauge efficiency of the system. The Kernel Perceptron algorithm attained 6.15% FAR and 4.82% FRR whereas SMO algorithms attained 7.16% FAR and 6.57% FRR. The paper concludes that the offline signature verification is possible effectively with the help of SVM and that feature, and kernel selection can be used to improve the classification accuracy.

In this case [8], the authors have created an offline signature recognition system with the feature extraction approach, which includes Binary Statistical Image Features (BSIF) and Local Binary Patterns (LBP) and the classification method k-Nearest Neighbor (KNN). This system aims at recognizing and verifying handwritten signatures, especially in official and administrative papers. The preprocessing steps involve binarization of signatures, illumination normalization and histogram equalization. The feature extraction system generates texture property features, whose classification is performed by KNN, and chi-square distance is implemented. Two databases, namely, MCYT-75 and GPDS-100, were experimented with, including 1125 genuine and 375 forged signatures. The system equal error rate was 4.2 percent in all the three databases. This implied that BSIF and LBP descriptors be used with KNN in offline signature recognition.

The authors in the paper [1] came up with a system where RFC could be applied to authenticate offline handwritten signatures. The procedure applied was the transformation of the signature images into black and white as well as the expression of the arrays obtained as binary. Various classifiers were verified including MNBC, BNBC, LRC, and SGDC among which the greatest precision of 67 percent was achieved using RFC classifier. ICDAR 2009 Signature Verification Competition datasets of genuine and forged signatures were used in the study. This was meant in the system as a method of capturing variance of handwriting overtime and also a client application of dynamic check. There were however few shortcomings with the system: the identification accuracy of the system was only 67%; working with skilled fakes was quite difficult; and the system has worked with static databases, and hence, it may not capture all the possible variations of signatures. Moreover, the adaptability of the system to the changes in object and surface could lead to the fact that the system will increase the false acceptance rate of forgeries.

III. Methodology

The proposed system consists of three main stages: image preprocessing, feature extraction, and SVM classification

A. Requirements

The project starts by finding out and studying the different requirements that a system should fulfill. Features of a functional handwritten signature recognition system let users upload or take pictures, use SVM on the images, and reveal the verification results. Examples of non-functional requirements are performing fast and with consistent results, adapting to various artist and photo quality, keeping users' signature data private, and creating a user-friendly application. For the

hardware side, there should be a CPU for image work, an SVM for accurate classification, at least 8GB of RAM, room for signature data storage, and a device for uploading people's signatures. The system must be comfortable for everyone to use and dependable.

B. Image Preprocessing

Input Input signature images are first subjected to standard preprocessing to ensure uniformity and reduce noise. Each image is converted to grayscale, then binarized using Otsu's thresholding to separate ink from background. We apply a Gaussian blur to smooth the image and remove minor variations, followed by morphological operations or filters to eliminate isolated noise pixels. Edge detection (e.g., Canny operator) is then performed to capture the contour of the ink strokes. Finally, the image is resized to a standard dimension to normalize scale. These steps – grayscale conversion, Otsu's binarization, Gaussian blurring, noise removal, edge detection, and image resizing – collectively standardize the signatures and make feature extraction more robust

C. Feature Extraction

From each preprocessed signature image, a set of features is computed to capture both global and local characteristics. Global geometric features include the aspect ratio of the signature bounding box, the centroid of the ink region, and the slant (dominant stroke angle). Additionally, statistical descriptors such as the number of ink loops or intersections can be used. For texture and shape features, we extract Hu invariant moments (7 moments that summarize shape), Histogram of Oriented Gradients (HOG) descriptors, Local Binary Patterns (LBP) histograms, and skeleton-based metrics. Edge-direction histograms (counting edge pixels by orientation) provide another global descriptor. In summary, the feature vector for each signature may include values like aspect ratio, normalized area, centroid coordinates, slant angle, edge histogram bins, Hu moments, HOG features, LBP features, and skeleton-based featuresfile-8nib4la6gjcganmyowa4zr. This hybrid feature set is chosen to jointly capture the signature's overall shape and fine-grained stroke details, which helps differentiate genuine signatures from skilled forgeries. global features:

- Bounding box aspect ratio, offering shape clues.
- Centroid coordinates, describing mass distribution.
- Slant angle detecting writing tilt.
- Inked area ratio, capturing signature density.
- Stroke intersections/loops, measuring structural complexity.

Shape and texture features:

- Hu Moments: shape invariant descriptors.
- HOG descriptors: compact gradient patterns representing directional edge structure.
- LBP histograms: local texture encoding.
- Edge direction histograms: binned gradient orientations.
- Skeleton metrics: attributes derived from a thinned version of the ink to capture structural stroke patterns.

D. Class Imbalance Handling

Signature datasets often have imbalanced classes (more genuine samples than forgeries, or vice versa). To prevent the SVM from biasing towards the majority class, we apply random oversampling to the minority class in the training data. Oversampling duplicates examples of the underrepresented class (e.g., forged signatures) until the classes are balanced. This ensures that during training, the SVM sees an equal number of genuine and forged examples, leading to improved generalization for both classes

E. SVM Classification

The balanced feature vectors are used to train a binary SVM classifier. We use a radial basis function (RBF) kernel, which maps input features into a higher-dimensional space to handle non-linear separability. During training, the SVM learns the optimal separating hyperplane (in the feature-transformed space) that maximizes the margin between genuine and forged signatures. In testing, the trained SVM outputs a class label (genuine or forged) for each input signature, along with a confidence score based on the distance from the decision boundary. By tuning the SVM parameters (e.g., kernel scale, regularization), the system aims to minimize classification errors on validation data.

IV. RESULT AND DISCUSSION

A Support Vector Machine is used in the system to compare real and forged signatures with an accuracy of 89.51%, 89.39% precision and 89.66% recall. It identifies true signatures at True Positive Rate of 89.66% and False Acceptance Rate of 11.93% on forged signatures. The ROC AUC score of the system is high, close to 0.94.

A. ROC curve

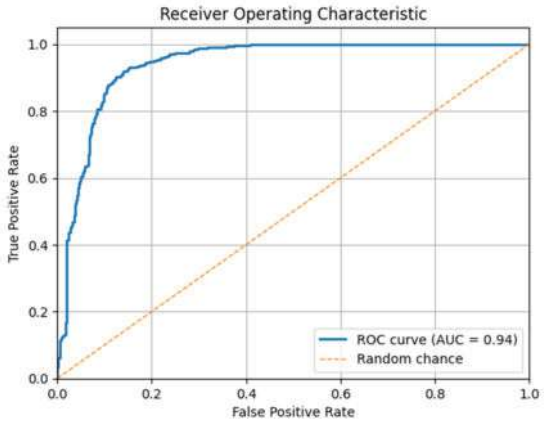


Fig. 1. Receiver Operating Characteristic (ROC) Curve

DET Curve is a useful way to judge the outcomes achieved by binary systems for biometric verification or speaker identification. The start of the curve on the top left corner is when FRR and FAR are both maximum which points to strictness. When the curve moves to the right, the rules become less strict

which decreases FRR and increases FAR. It is best when both FAR and FRR are very small, which occurs at the origin. The first part of the curve indicates that when FAR is not very high, a slight increase in it can cause a major drop in FRR. As FRR goes down, it takes a bigger rise in FAR and this means the curve flattens out. Using this DET curve, it can be observed how the system determines false acceptances and false rejections at different levels of threshold.

B. DET Curve

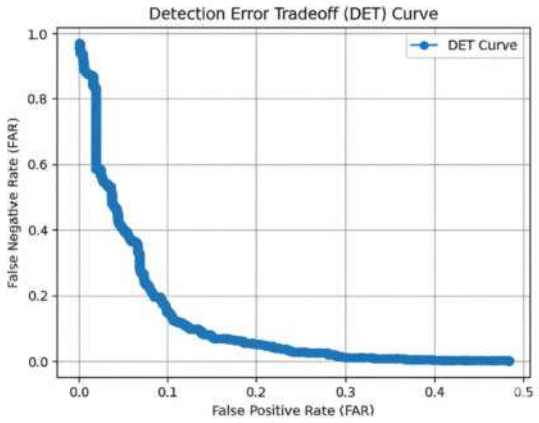


Fig. 2. Detection Error Tradeoff (DET) Curve

The DET Curve is a useful way to judge the outcomes achieved by binary systems for biometric verification or speaker identification. The start of the curve on the top left corner is when FRR and FAR are both maximum which points to strictness. When the curve moves to the right, the rules become less strict which decreases FRR and increases FAR. It is best when both FAR and FRR are very small, which occur at the beginning. The first part of the curve indicates that when FAR is not very high, a slight increase in it can cause a major drop in FRR. As FRR goes down, it takes a bigger rise in FAR and this means the curve flattens out. Using this DET curve, it can be observed how the system determines false acceptances and false rejections at different levels of threshold.

C. Performance Comparison

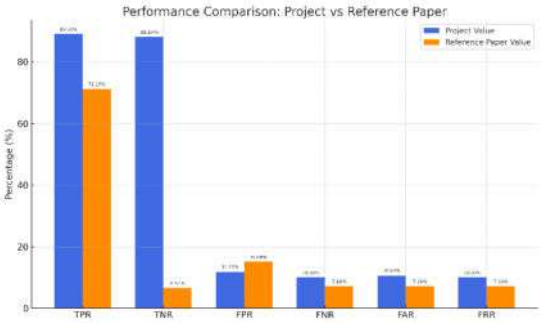


Fig. 3. Comparison of Result

The project's True Positive Rate (TPR) was 89.05% which is 17.86% more than the 71.19% seen in the reference text, so it recognized genuine signatures better. Better results could be seen because the signatures were preprocessed with resizing, blurring and thresholding which made them easier to process and more recognizable. The TNR increased to 88.14% from 6.57% in the reference paper. TNR went up mostly for two reasons: first, because feature extraction methods like Hu Moments, Histogram of Oriented Gradients, Local Binary Patterns and contour-based features were used; second, because dimension reduction methods like Principal Component Analysis and Singular Value Decomposition were also included. The False Positive Rate came out at 11.85%, which is lower than the reference paper's 15.08%, meaning the system performed better at telling real and fake signatures apart. The rate of False Negatives (FNR) is 10%, which is higher than the reference paper's value of 7.16%, meaning the project is more likely to reject authentic signatures. The FAR was measured at 10.64% in this study, which is greater than the 7.16% value in the reference paper. There were more false rejections than in the reference which suggests taking more caution to reduce errors in one category can lead to more errors in another.[3]

V. CONCLUSION AND FUTURE WORK

The Offline Signature Verification System using SVM is highly accurate with a level of 89.51% and the ROC AUC score of 0.94. It has good performance in eliminating problems such as inconsistency in the same class and synthetic examples, which make it worth application in daily activities in banking, law and identity verification. The framework can process tasks associated with biometrics using minimal data and prepares the basis of further precision and flexibility by incorporating deep learning techniques. The effectiveness of this scheme demonstrates that conventional approaches such as SVM can be used to authenticate individuals when deep learning needs excessive resources. Flask provides real time verification with user-friendly interface.

Some future suggestions would be to incorporate an automatic signature detection module, which will enable the user to identify and isolate signatures on an image, and work on an online platform that enables different tasks to be performed on a single interface. These would allow making the system more user-friendly, capable of processing more data, and delightful to interact with. It would be easier to access and upgrade the system (developing it as a cloud or web application). And finally, continuous learning as an addition would give more confidence and less prediction, which will result in better strong points and applicability to various disciplines.

REFERENCES

[1] M. Thenuwara and H. R. K. Nagahamulla, "Offline handwritten signature verification system using random forest classifier," 2017 Seventeenth International Conference on Advances in ICT for Emerging Regions (ICTer), Colombo, Sri Lanka, 2017, pp. 1-6, doi: 10.1109/ICTER.2017.8257828

[2] Mahanta, Lipi & Deka, Alpana. (2013). A Study on Handwritten Signature. International Journal of Computer Applications. 79. 48-52. 10.5120/13717-1489.

[3] C. Kruthi and D. C. Shet, "Offline Signature Verification Using Support Vector Machine," 2014 Fifth International Conference on Signal and Image Processing, Bangalore, India, 2014, pp. 3-8, doi: 10.1109/ICSIP.2014.5.

[4] Parascript, "Check and signature fraud," White Paper, n.d. [Online]. Available: https://www.parascript.com/wp-content/uploads/2017/07/White-Paper_Check-and-Signature-Fraud-web.pdf

[5] M. S. U. Khan, T. Shehzadi, R. N. D. Stricker, and R. M. Z. Afzal, "Efficient signature verification using hybrid ML techniques," arXiv preprint arXiv:2406.14370, 2024

[6] S. M. A. Navid, S. H. Priya, N. H. Khandakar, Z. Ferdous and A. B. Haque, "Signature Verification Using Convolutional Neural Network," 2019 IEEE International Conference on Robotics, Automation, Artificial-intelligence and Internet-of-Things (RAAICON), Dhaka, Bangladesh, 2019, pp. 35-39, doi: 10.1109/RAAICON48939.2019.19.

[7] C. Nathwani, "Offline signature verification using SVM and CNN," IEEE Xplore, 2020.

[8] H. Hezil, R. Djemili, and H. Bourouba, "A biometric identification system based on signature recognition using SVM," International Journal of Biometrics, vol. 10, no. 1, pp. xx–xx, 2018

Waste Classification System Using Convolutional Neural Network

Shubhnima Mahato
Nilai University, Malaysia
Padmashree College, Tinkune, Kathmandu
shubhnima.bit_2022@padmashreecollege.edu.np

***Abstract*—Waste classification remains a major challenge in Nepal, especially in cities like Kathmandu, where daily waste generation is rapidly increasing. Traditional manual classification methods are slow, labor-intensive, and prone to errors, resulting in recyclable materials being discarded, overfilled landfills, and environmental pollution. To address this issue, the proposed project develops an automated waste classification system using Convolutional Neural Network (CNN). CNN accurately identifies the waste categories. The system workflow begins with waste placement in a dumping area, where the system captures high-resolution images of waste and preprocesses them using techniques such as cropping to focus on individual waste items, normalization to standardize image size and color, and noise reduction to remove distortions. These preprocessed images are then analyzed by a CNN model trained to classify waste into five primary categories: organic, metal, plastic, glass, and general waste. The preprocessed images are analyzed by a CNN model, trained to classify the waste into specific categories, which helps decide how the waste will be managed, such as being recycled, turned into compost, or disposed of safely. The system also stores all captured images, classification results, and relevant waste data in a local database for further analysis, performance tracking, and optimization. Testing has shown the system to be over 90% accurate, offering a significantly faster and more reliable alternative to manual classification. By automating the waste classification process, this project reduces human error, support recycling efforts, and reduces landfill dependency. Therefore, it provides a practical and sustainable solution to Nepal's growing waste management challenges while contributing to a cleaner and healthier environment.**

***Index Terms*—Automated Waste Management, Convolutional Neural Network (CNN), Sustainable Waste Management, Image Processing**

I. Introduction

In recent years, waste classification has become a pressing global challenge due to the rapid urbanization, increasing population, and the growing volume of municipal solid waste (MSW)[1]

Waste classification is a global issue with far-reaching implications for the environment, economy, and public health. Rapid urbanization, industrial growth, and rising consumption patterns have led to an unprecedented increase in waste generation worldwide. Among the most pressing concerns are plastic waste, which accumulates in landfills and oceans; industrial waste, which often contains hazardous chemicals; and carbon emissions, which contribute significantly to climate change. For example, in 2018, over 360 million tons of plastic were produced globally, with a significant portion improperly managed, causing soil and water contamination[2]. Carbon emissions from industrial activities, transportation, and waste decomposition exacerbate global warming, posing threats to biodiversity and human health[3]. Effective waste classification systems are crucial for maintaining a sustainable and healthy environment. Labeled Waste in the Wild dataset contains In Nepal, the waste classification crisis matches this global challenge but is caused by limited resources and infrastructure. Organic waste makes up 61% of municipal waste streams, followed by plastics (16%) and, in lower amounts, metal, glass, and hazardous materials. Over 90% of waste ends up in landfills or open dumps, indicating that the nation's waste classification is still primarily linear despite laws and rules like those prohibiting thin plastics. Soil degradation, toxic air pollutants, and water contamination are only a few of the serious health and environmental hazards associated with this strategy [4]. The situation is grave since industrial and medical waste put additional load on the system because of insufficient segregation and disposal methods.

Waste classification involves the systematic categorization of different types of waste materials based on their composition, source, and potential for recycling or disposal. This process is essential for efficient waste management, reducing environmental impact, and maximizing resource recovery[5]. The production and usage of plastic have surged in recent decades, leading to increased waste generation. In 2018, global plastic production reached approximately 360 million tons, with over 5 trillion plastic bags manufactured annually. In Nepal, plastic waste contributes significantly to the total municipal solid waste, comprising about 16% of urban waste, equivalent to approximately 2.7 tons of plastic waste daily [6].

The use of Convolutional Neural Networks (CNNs) in waste classification helps to efficiently extract key features from waste images, enabling accurate classification and separation of various waste categories such as plastics, metals, organics, and hazardous materials[7]. This approach streamlines the sorting process, reduces human error, and enhances the overall sustainability of waste classification systems.

II. Background Studies and Related works

A. Background studies

In recent years, waste classification has grown to be a major environmental concern in Nepal. Waste output has significantly increased due to rapid urbanization and shifting consumption

patterns, especially in urban regions like the Kathmandu Valley. Only a small portion of the 1,200 metric tons of solid trash produced every day in the Valley alone is recycled. In Nepal, there are several ways to collect solid trash, such as self-delivery, roadside pickup, communal bins, and door-to-door collection. These gathering techniques, however, are simplistic and do not have the same level of effectiveness as those found in wealthy nations[8]. The over-reliance on landfills highlights the inefficiency of existing waste management strategies and underscores the need for sustainable alternatives. Traditional waste classification methods rely heavily on manual labor, which is not only slow and inefficient but also prone to significant errors in classification and disposal. These manual practices often result in recyclable materials like plastic, glass, and metal being discarded alongside general waste, leading to overfilled landfills, reduced recycling rates, and increased environmental pollution. Furthermore, the lack of awareness and resources for waste classification aggravates this issue, making proper disposal a neglected priority for households and businesses alike [9].

To address these pressing challenges, the proposed solution involves the development of an automated waste classification system powered by modern technologies such as Convolutional Neural Networks (CNN). This innovative approach integrates real-time image processing and machine learning algorithms to accurately classify waste into categories such as plastic, glass, metal, organic, and general waste. This automation minimizes reliance on manual labor, significantly reducing human error and improving efficiency in waste handling processes. It ensures a higher recovery rate of recyclable materials, reducing landfill dependency and promoting a circular economy. Additionally, the automation eliminates direct human contact with hazardous waste, enhancing worker safety and health conditions. This project not only addresses the critical need for efficient waste classification in Nepal but also supports broader environmental goals by reducing pollution, conserving resources, and mitigating greenhouse gas emissions. Ultimately, it strives to create a cleaner, healthier, and more sustainable environment for all.

TABLE I
NUMBER OF IMAGES IN EACH CLASS FOR TRAINING AND VALIDATION

Model	Training Data	Validation Data
Cardboard	322	81
Glass	436	110
Metal	333	84
Organic	489	123
Paper	473	119
Plastic	400	101
Trash	137	35
Total	2590	653

B. Related work

Numerous studies utilizing machine learning have been carried out for waste classification systems. These systems employ a variety of algorithms to achieve more accurate results. The following describes a few popular algorithms.

Sirimewan et al. [10] focus on creating Deep learning-based models for environmental management: recognizing construction, renovation, and demolition waste (CRD) in-the-wild. The goal of this research is to accurately depict the intricacy of waste streams within the framework of CRD. According to the study's findings, DL models can be used to identify and classify solid waste in a variety of industrial settings, which supports resource recovery and environmental management initiatives.

Article [11] proposed a YOLO v7 (You Only Look Once) algorithm-based automatic waste classification system, utilizing computer vision for efficient waste classification. The system correctly detects and categorizes different kinds of waste in real time by utilizing YOLO's object detection capabilities and the strength of computer vision. The speed and efficiency of the YOLO algorithm allow waste materials to be processed quickly, which makes it easier to sort them into designated locations. In addition to increasing accuracy, this automated approach lowers environmental damage and worker health concerns. The system achieved an accuracy of 87% when tested against a comprehensive trash dataset, demonstrating the suitability of YOLO for quick and efficient real-time waste classification.

A smart waste management and classification system using CNN (SAMACM-CA) was examined by Cheema et al. [12]. It classifies and separates waste materials in a disposal area using advanced methods, deep learning (DL), and the Internet of Things (IoT). The trained algorithm has an overall accuracy of over 90%, which is highly efficient.

In A paper on Garbage Detection and Classification using Faster-RCNN with Inception-V2 is published in[13]. The paper attempts to build a Faster R-CNN based predictive model for automatic classification of ten different types of waste/litter objects. It utilizes Faster R-CNN for automatic detection and classification of ten distinct types of street litter, achieving high precision with a mAP (mean average precision) of 92% using Inception-V2 as a backbone.

From the literature review above, different algorithms are discussed that are mainly used in predicting the system. In this project, the Convolutional Neural Network (CNN) algorithm will be selected for developing an automatic waste classification system due to its proven effectiveness in image-based classification tasks. CNNs are designed to automatically learn hierarchical features from raw image data, making them especially suitable for tasks involving complex image patterns, such as recognizing various waste materials.

An Automatic Garbage Classification System Using ResNet-34, a report by Sirangi et al. examines the creation of an effective machine learning-based garbage classification system. The study tackles waste management issues by automating and enhancing the sorting process using the ResNet-34 architecture. Important elements including system design, performance assessment, and real-world application are highlighted in the study. An enhanced algorithm based on ResNet-34 is proposed

TABLE II
SUMMARY TABLE FOR ALGORITHMS

Algorithm	Advantages	Disadvantages
Transfer Learning	1. Effective feature extraction 2. Supports automated sorting 3. Flexibility with various backbones	1. Limited to CRD waste 2. Lacks real-time feedback 3. Offline approach
YOLO	1. Real-time processing 2. High efficiency 3. Accuracy with lower computational cost	1. Limited to specific waste types 2. May struggle with complex waste streams
CNN	1. High accuracy (90%) 2. Uses efficient computational resources 3. Effective in segmentation tasks	1. Not suited for real-time municipal integration 2. Requires powerful resources for larger-scale applications
Faster R-CNN	1. High accuracy with mAP of 92% 2. Effective for street waste classification 3. Detects multiple objects	1. Not suited for real-time municipal integration 2. Lacks comprehensive waste types classification
IoT & Image Recognition	1. Real-time monitoring 2. Uses moisture and image data for accurate waste classification 3. Improves sorting accuracy	1. Focused only on wet and dry waste 2. Does not handle complex waste types Limited scalability for large-scale operations
RESNET-34	1. High efficiency in processing waste in dynamic environments. 2. Provides accurate waste classification. 3. Improves Sorting Accuracy	1. Does Not Handle Complex Waste Types 2. May face challenges in large-scale operations.

TABLE III
NUMBER OF IMAGES IN EACH CLASS FOR TRAINING AND VALIDATION

Model	Training Data	Validation Data
Cardboard	322	81
Glass	436	110
Metal	333	84
Organic	489	123
Paper	473	119
Plastic	400	101
Trash	137	35
Total	2590	653

in this study. The basic model was first determined through trials on common data sets, and ResNet-34 was selected since it performed the best. The garbage data set, which contains six different types of rubbish, is used to test the model. The created garbage data is used to verify the prevalence of the suggested categorization computation. The suggested calculation's order accuracy is improved by 1.01%. According to the trial findings, the grouping precision is almost 99%, and the framework's characterization pattern is roughly as fast as 0.95 seconds[14].

III. METHODOLOGY

The proposed system will implement a Convolutional Neural Network (CNN) model, as discussed in Section 4.3. This method is intended to address the significant problem of waste classification in Nepal, especially in cities like Kathmandu where the increasing amount of waste necessitates automated and effective solutions. Conventional waste management techniques mostly rely on manual labor, which results in inefficiencies, incorrect waste classification, and environmental risks from inappropriate recycling and excessive landfill usage [12]. The proposed project uses CNN, a deep learning algorithm, to precisely classify waste into five main categories: organic, metal, plastic, glass and general. This solves these problems. By automating the classification process and offering useful insights for garbage processing, this system seeks to increase the efficiency of waste classification.

The system begins its operation by capturing high-resolution images of waste items placed in a designated dumping area. These images undergo a series of preprocessing steps to ensure consistent quality and usability for classification. The preprocessing includes: Cropping to focus on individual waste items and remove unnecessary background information. Normalization to standardize image dimensions and pixel values. Noise Reduction to eliminate distortions that might hinder accurate classification. By automating the waste classification process, the system minimizes human error, promotes recycling, and reduces landfill dependency, thus contributing to a more sustainable and environmentally friendly waste classification solution.

A. Method Description

The proposed system will implement a Convolutional Neural Network (CNN) model, as discussed. This method is intended to address the significant problem of waste classification in Nepal, especially in cities like Kathmandu where the increasing amount of waste necessitates automated and effective solutions. Conventional waste management techniques mostly rely on manual labor, which results in inefficiencies, incorrect waste classification, and environmental risks from

inappropriate recycling and excessive landfill usage[12]. The proposed project uses CNN, a deep learning algorithm, to precisely classify waste into five main categories: organic, metal, plastic, glass and general. This solves these problems. By automating the classification process and offering useful insights for garbage processing, this system seeks to increase the efficiency of waste classification.

The system begins its operation by capturing high-resolution images of waste items placed in a designated dumping area. These images undergo a series of preprocessing steps to ensure consistent quality and usability for classification. The preprocessing includes:

- Cropping to focus on individual waste items and remove unnecessary background information.
- Normalization to standardize image dimensions and pixel values.
- Noise Reduction to eliminate distortions that might hinder accurate classification

The CNN model is used to classify the images once they have been preprocessed. A labelled dataset with a variety of waste photos classified as organic, metal, plastic, glass, and trash are used to train the CNN model. To get high classification accuracy, the model learns to recognize distinctive characteristics of each category, such as texture, color, and form. After classification, the original image and pertinent metadata are saved in a local database for later examination.

TABLE IV
FUNCTIONAL REQUIREMENT OF THE SYSTEM

S.No	Functional Requirements
1	User should be able to run the system with the help of basic internet connectivity.
2	User should be able to upload relevant data or images for processing and analysis.
3	User should be able to get accurate results or insights based on the uploaded data.
4	User should be able to delete previously uploaded data or records from the system.

B. Data set Description

The CNN model is trained on a curated dataset of waste images obtained from public datasets and locally sourced waste images captured in urban areas of Nepal. The dataset includes labeled images of various waste items, each categorized into one of five primary classes:

1) **Organic** – Includes food waste, garden waste, and biodegradable items.
2) **Metal** – Includes cans, metal scraps, and other metallic items.
3) **Plastic** – Includes bottles, containers, and non-biodegradable plastic materials.
4) **Glass** – Includes glass bottles, jars, and other glass materials.
5) **General** – Includes non-recyclable, paper and general waste.

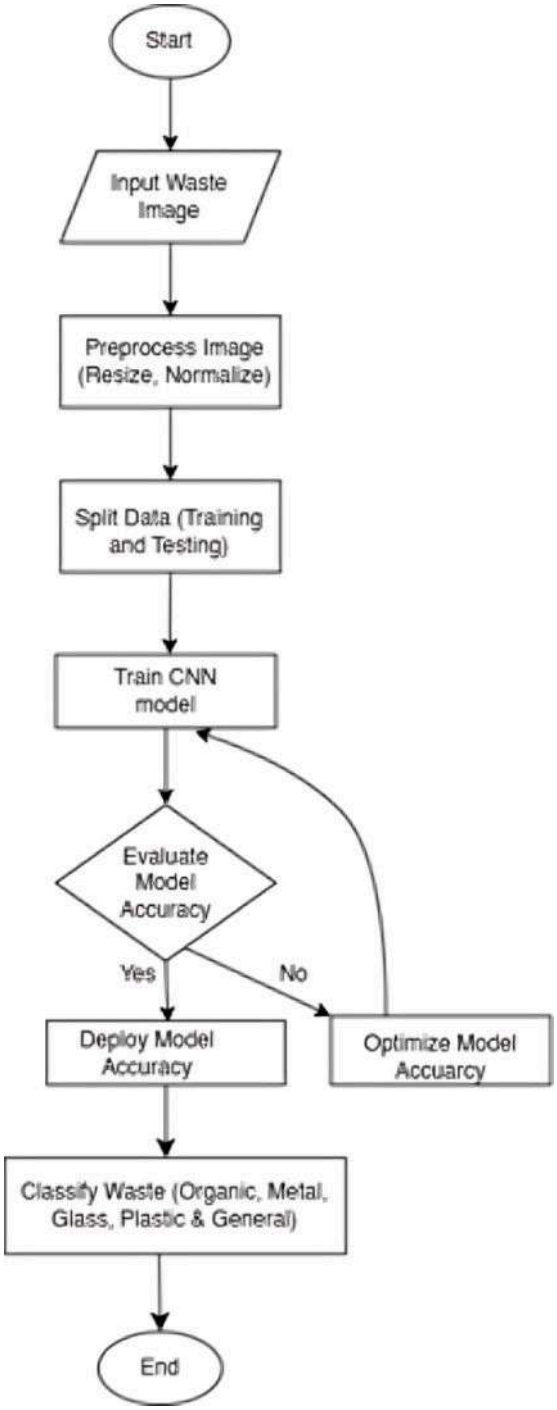


Fig. 1. Flowchart showing working process of proposed system

The preprocessing stage ensures uniformity in image resolution and quality, enhancing the model's ability to learn distinguishing features.

IV. PROJECT METHODOLOGY

The system is developed for the classifying the waste type that are found in every dumping sites. There are certain minimal hardware and software requirements in order for the suggested

system to function properly. Similarly, because the system has been employing the waterfall approach, the following is a list of the most crucial and critical functional and non-functional requirements for the system:

A. Functional Requirements

The functional requirements are presented in table VI.

TABLE V
FUNCTIONAL REQUIREMENT OF THE SYSTEM

S.No	Functional Requirements
1	User should be able to run the system with the help of basic internet connectivity.
2	User should be able to upload relevant data or images for processing and analysis.
3	User should be able to get accurate results or insights based on the uploaded data.
4	User should be able to delete previously uploaded data or records from the system.

B. Non-Functional Requirements

Classification is crucial for data protection as it helps define the strategic importance of different types of data. By implementing classification, we can ensure an appropriate level of data protection while still allowing for smooth information flow and efficient business processes. The input variables are utilized to define the features in the antecedent, while the output class represents the consequent part. A fuzzy classification can be expressed by table**??**:

C. Functional Requirements

TABLE VI
NON-FUNCTIONAL REQUIREMENT OF THE SYSTEM

S.No	Functional Requirements
1	The system must provide the security to the user data.
2	The system must be accessible to everyone.
3	The system must be easy to navigate.
4	The interface should be user-friendly and adhere to usability standards for ease of operation.

D. Testing

For the testing purpose we are going with the functional testing which will help to verify that the system operates in accordance with functional requirements.

V. RESULT AND DISCUSSION

The sample results are presented in the figure 2 below: This system accurately classified waste images into five categories: glass, metal, plastic, organic, and general. Most of the test samples were collected from the internet, while some were taken from the Garbage Classification Dataset and TrashNet on Kaggle. Based on the test table, the system has demonstrated

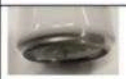
Test No.	Test Images	Predicted Class	Actual Class	System Predicted Accuracy(%)	Remarks
1.		Organic	Organic	100	Correct Prediction
2.		Plastic	Metal	97.63	Incorrect Prediction (the system misunderstood the visual characteristics of plastic and metal)
3.		Metal	Metal	99.58	Correct Prediction
4.		Plastic	Plastic	99.88	Correct Prediction
5.		Plastic	Plastic	99.93	Correct Prediction

Fig. 2. Result of testing

a high level of accuracy in predicting the classification of different waste materials. Out of 20 samples tested, the system correctly predicted 17 samples, achieving an overall accuracy of 85%. Calculated using the formula:
Accuracy = (Number of correct predictions / Total number of samples) * 100 = (17/20) *100 = 85%. This indicates that the system is highly reliable for waste classification tasks, there is room for improvement, especially in distinguishing between visually similar categories.

There were a few notable incorrect predictions:

- One metal sample was misclassified as plastic with 97.63
- Another metal sample was not recognized at all, leading to a message indicating the waste did not belong to any known category.
- A glass sample colored yellow was misclassified as plastic, highlighting challenges in distinguishing visually similar materials.

With an overall success rate of 85% (17 out of 20 tests properly recognized), the model showed great potential for waste classification while also pointing out areas that could benefit from more training and improvement to increase accuracy and robustness.

VI. CONCLUSION AND FUTURE WORK

In Nepal, waste classification remains a significant challenge, particularly in rapidly growing cities like Kathmandu, where waste generation is increasing daily. Traditional manual waste sorting methods are time-consuming, labor-intensive,

TABLE VII
FUNCTIONAL TESTING OF THE SYSTEM

Test Case No.	Test Case Description	Test Steps	Expected Outcome	Test Result
1.	Verify choose file button is clickable.	1.Go to the URL 2.Click on the "Choose file" button.	The Choose file button should allow user to select an an image file.	Pass
2.	Verify the system load the image.	1.Go to the URL 2.Click on the "Choose file" button. 3. Select the image.	The system should load the image within a respective time.	Pass
3.	Verify system only supports JPG or PNG image formats.	1.Go to the URL 2.Drag and drop the image with different format.	It should display the error message.	Pass
4.	Verify submit button is clickable.	1.Go to the URL 2.Choose the image of write format 3.Click the submit button.	The image should be submitted successfully.	Pass
5.	Verify the system displays the class for the submitted image.	1.Go to the URL 2.Choose the image of write format 3.Click the submit button.	The system should display the prediction to the correct class for the selected image.	Pass
6.	Verify the accuracy displays for the selected image.	1.Go to the URL 2.Choose the image of write format 3.Click the submit button.	The system should display the accuracy of the selected image.	Pass

and prone to errors, often leading to improper disposal of recyclables, overflowing landfills, and environmental pollution. To tackle this issue, a waste classification system has been successfully developed and is ready for deployment. The system is built using advanced machine learning technologies, including TensorFlow, Keras, NumPy, and more. A convolutional neural network (CNN) was used to design the model architecture, which was trained for 20 epochs, achieving an accuracy of 90%.

To improve the waste classification system and make it more effective, the following upgrades are suggested:

1) Increase the Dataset: Adding more waste images from different sources will help the system recognize a wider variety of waste types and improve accuracy.
2) Improve Speed for Real-Time Use: Making the system faster will allow it to work in real-time at waste collection centers and recycling plants.
3) Create a Mobile Version: Developing a mobile-friendly version will make it easier for people to use the system on their smartphones.
4) Add a Feedback Feature: Allowing users to report mistakes and give feedback will help improve the system's accuracy over time.
5) Connect with Smart Waste Bins: Integrating the system with IoT-powered waste bins or sensors will enable automatic waste sorting, reducing errors and improving recycling.

REFERENCES

[1] D. C. Wilson, "Waste management - Still a global challenge in the 21st century: An evidence-based call for action," 2015.

[2] Y. Chen and A. K. An, "Single-use plastics: Production, usage, disposal, and adverse impacts," *Sci. Total Environ.*, 2021. Accessed: [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0048969720353018.

[3] H. Ritchie and M. Roser, "CO_2 emissions," *Our World in Data*, 2020. Accessed: [Online]. Available: https://ourworldindata.org/co2-emissions.

[4] Asian Development Bank, "Solid Waste Management in Nepal," 2013. Accessed: [Online]. Available: https://www.adb.org/sites/default/files/publication/30366/solid-waste-management-nepal.pdf.

[5] "Waste Classification," in *Encyclopedia of Environmental Health (Second Edition)*, 2011. Accessed: [Online]. Available: https://www.sciencedirect.com/topics/earth-and-planetary-sciences/waste-classification.

[6] K. K. Maharjan, "Microplastics research in Nepal: Present scenario and current gaps in knowledge," 2024. Accessed: [Online]. Available: https://www.researchgate.net/figure/Compositionofsolidwastein-Nepal_fig1_377614604.

[7] M. Kaur, R. Ahmed, H. Wang, and H. N. O. Noor, "CNN-based Smart Waste Management System in Fog Computing Environment," 2023. Accessed: [Online]. Available: https://www.researchgate.net/publication/371199466_CNN-based_Smart_Waste_Management_System_in_Fog_Computing_Environment.

[8] C. Scotchbrook, "Assessment of Current Waste Management Practices in Nepal: Challenges and Opportunities for a Circular Economy," 2024. Accessed: [Online]. Available: https://nepaleconomicforum.org/assessment-of-current-waste-management-practices-in-nepal-challenges-and-opportunities-for-a-circular-economy/.

[9] Conurets, ”Comparative Analysis: Waste-to-Energy vs. Traditional Waste Disposal Methods,” 2022. Accessed: [Online]. Available: https://www.conurets.com/waste-disposal-and-recycling-traditional-vs-smart-practices/.

[10] D. Sirimewan, M. B. Sirisena, S. R. Rajapaksha, and A. F. K. M. A. Munasinghe, ”Deep learning-based models for environmental management: Recognizing construction, renovation, and demolition waste in-the-wild,” *J. Environ. Manage.*, 2024. Accessed: [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0301479723026968.

[11] S. Maity, T. C. Chatterjee, and P. H. S. Roy, ”YOLO (YOU ONLY LOOK ONCE) algorithm-based automatic waste classification system,” *J. Multidisciplinary Case Studies*, 2023. Accessed: [Online]. Available: https://jmcms.s3.amazonaws.com/wp-content/uploads/2023/08/30183325/jmcms-2308003-YOLO-Algorithm-Based-Automatic-Waste-Classification-System-SM-TC.pdf.

[12] S. M. Cheema, A. H. Ahmed, and I. M. Pasha, ”Smart waste management and classification systems using cutting edge approach,” *Sustainability*, vol. 14, no. 16, p. 10226, 2022.

[13] A. I. Middya, D. Chatterjee, and S. Roy, ”Garbage detection and classification using Faster-RCNN with Inception-V2,” presented at the IEEE Xplore, 2021. Accessed: [Online]. Available: https://ieeexplore.ieee.org/document/9691547.

[14] A. Sirangi and V. Guntuku, ”An automatic garbage classification system using RESNET-34” Bachelor’s thesis, Sathyabama Institute of Science and Technology, Chennai, 2021. Accessed: [Online]. Available: https://sist.sathyabama.ac.in/sist_naac/documents/1.3.4/b.e-cse-batchno-316.pdf.

Effect of Soaking Time and Water Ratio on Anti-Nutritional Factors of Different Varieties of Soybeans (Glycine Max)

Shrawan Kumar Sah*, Nilmani Poudel*, Sujan Sapkota†
*Department of Food Technology, Padmashree International College, Kathmandu, Nepal
†Food Management and Trading Company Ltd., Kathmandu, Nepal
fbtshrawan@gmail.com

***Abstract*—This research was aimed to assess the effect of soaking at different time period (15hr-2, 15hr-3, 15hr-4, 21.4hr-3, 24.5hr-3, 27.8hr-4 and 30hr-3) on anti-nutritional components (oxalate, phytate and tannin) content of four different varieties of soyabean (Glycine max) i.e. ransom (white), v.l.s-1 (black), sathiya (brown), and tarkari-1 (green) and the obtained data was statistically analysed using one-way ANOVA at 5% level of significance using SPSS version 20. The results revealed soaking at 30hr-3 reduced maximum level of oxalate, phytate and tannin to 43.78±0.24 mg/100g, 464.1±32.80 mg/100g and 1233.56±155.19mg/100g respectively in ransom variety. Oxalate with 55.06±15.64 mg/100g was lowest in 21.4hr-3, phytate with 266.85±16.40 mg/100g was lowest in 30hr-3 and tannin with 1724.67±183.01 mg/100g was lowest in 27.8hr-4 in v.l.s-1 variety. Similarly, soaking for 27.8hr-4 reduced maximum level of oxalate and tannin to 54.94±15.35 mg/100g and 1951.58±53.42 mg/100g and both 27.8hr-4 and 30hr-3 reduced maximum level of phytate to 522.11±49.22 mg/100g in sathiya variety. In case of tarkari-1 variety, phytate and oxalate content were lowest at 27.8hr-4 to a level of 54.78±14.93 mg/100g and 649.74±229.71 mg/100g respectively but, tannin was mostly reduced at 30hr-3 to a concentration of 2113.75±107.71mg/100g. Hence, 27.8hr was considered as the best soaking time and 1:4 as the best bean:water ratio.**

***Index Terms*—Soyabean, Anti-nutritional factors, Oxalate, Phytate, Tannin, soaking time**

I. Introduction

One of the most valuable crops in the world is soybean (*Glycine max*), which is used as a feedstock, for biofuel as well as an oil seed crop, animal feed, and a good source of protein for human use [18]. Soybeans are among the most valuable and affordable agricultural commodities. It has the highest protein content (around 40%) of any cereal or other legume species; other legumes have protein contents ranging from 20% to 30%. Additionally, soybeans have the second-highest oil content of any edible legume at 20% [25]. Soybeans also include important nutrients such as minerals, vitamins, and phospholipids. However, it also includes a wide range of anti-nutritional components, including goitrogens, lectin, α-amylase inhibitory factor, trypsin inhibitor, soybean antigen, and more [7]. According to [8] and [27], the presence of these anti-nutritional elements impairs the nutritional value, consumption, and digestibility of soybean protein, which can lead to digestive and metabolic problems in animals [34]. The nutritional value of soybean is lower than expected due to the presence of various anti-nutritional factors. So, to increase the nutritional value, soaking can be done. Soaking is a traditional process that is mainly used to soften the grains and to facilitate their subsequent cooking but people are unaware that soaking at suitable time can cause significant reduction in the anti-nutritional factors. Therefore, it is important to identify effective soaking time and bean:water ratio to reduce anti-nutritional factors before consumption.

II. Materials and Methods

A. Raw materials

Four different varieties of soybeans obtained from Nepal Agricultural Research Council (NARC). *Glycine max* varieties such as ransom (white), v.l.s-1 (black), sathiya (brown), and tarkari-1 (green) were used in this study. The samples were cleaned and subjected to soaking at different times with different soybean: water proportions. Design expert 13 was used to obtain the experimental runs with a file version 13.0.5.0. The study type was response surface and the design model was quadratic with a randomized subtype. The reference for the soaking time was taken from [3] as they soaked soybeans for the duration of 12-24 hours and soybean: water ratio was taken from [28] as they used 1:3 soybean:water ratio at ambient room temperature ($21 - 25°C$) to soak the soybeans for maximum reduction of anti-nutritional factors. The experimentals runs are given in the table I.

TABLE I
Experimental design of soaking process of soybeans

Run	Factor 1 A: Soaking Time (hrs)	Factor 2 B: Soybean:Water (w/w)
1	15	1:3
2	15	1:2
3	15	1:4
4	21.4	1:3
5	24.5	1:2
6	27.8	1:4
7	30	1:3

B. Proximate analysis of the samples

1) Moisture Content: Moisture content was determined by using a hot air oven following official methods of Association of official analytical chemists [4].

2) Crude fat content: Crude fat content of the samples was determined by using Soxhlet apparatus [22].

3) Crude protein content: Crude protein content was determined by the process of Kjeldahl nitrogen method [22].

4) Total ash content: A dry ashing method was used to determine the total ash content [4].

5) Crude fibre content: The crude fibre content in the given sample was determined by [22].

6) Total carbohydrate content: The total carbohydrate content of the samples was determined by differential method.
Carbohydrate(%)= 100 – [sum of crude protein, total ash, crude fiber, moisture and crude fat]

C. Physical properties analysis of the samples

1) 1000 kernel weight: One thousand healthy kernels of each varieties of soybeans were counted randomly and weighed separately [31].

2) L/B ratio: Length and breadth wise arrangement of each variety of beans was done and their cumulative measurements (in millimeter "mm") was taken. The value of L/B was determined by dividing length by breadth [31].

3) Bulk density: Bean seeds of each variety was poured into a certain known volume from a fixed height (30 centimeters) and mass of samples occupying the volume was determined respectively. Ratio was calculated as g/cc [31].

4) Grain density: Grain density was determined using the liquid displacement method. Toluene (C_7H_8) was used, rather than water, because water is absorbed more than toluene by the seed. A weighed quantity of seed was immersed in toluene; the volume of toluene displaced was read from the graduated scale on the cylinder [19], [32].

5) Porosity: Porosity was calculated by the help of grain density and bulk density using a formula.

$$\%\text{Porosity} = \frac{\text{grain density} - \text{bulk density}}{\text{grain density}} \times 100 \quad (1)$$

6) Thickness: The measurement of thickness was traditionally done manually using a Vernier Caliper [10].

7) Hydration properties: Hydration properties of the samples were measured at ambient room temperature ($30 \pm 2°$C) [36]. One hundred healthy seeds were collectively weighed and placed in a measuring cylinder containing 100 ml distilled water. After 24 hours of soaking the water was discarded and the weight of the hydrated soybeans was measured after surface drying. The gain in the weight was measured in percentage as hydration capacity.

$$\%\text{Hydration capacity} = \frac{\text{Final weight} - \text{Initial weight}}{\text{Initial weight}} \times 100$$

8) Swelling properties: Swelling properties of the samples were measured at ambient room temperature ($30 \pm 2°$C) [36]. One hundred healthy seeds were collectively placed in a measuring cylinder containing 100 ml distilled water and the rise in volume was noted. After 24 hours, the water was discarded and placed in a fresh 100 ml distilled water after surface drying the soaked seeds and the volume was noted. The increase in the volume was measured in percentage as swelling capacity.

$$\%\text{Swelling capacity} = \frac{\text{Final volume} - \text{Initial volume}}{\text{Initial volume}} \times 100 \quad (2)$$

D. Determination of anti-nutritional factors of the samples

1) Determination of tannins content: Tannins content of the samples was determined by the method described by [14].

2) Determination of phytate content: Phytate content was determined by the method described by [17].

3) Determination of oxalate content: The oxalate content was determined by the method of [6].

E. Statistical analysis

All the data obtained were analyzed by using SPSS version 20. From this mean ANOVA, LSD and interaction effects was obtained to determine whether the sample are significantly different from each other and also to determine which one is superior among them.

III. Results and discussion

This research work was done to evaluate the decrease in anti-nutritional factors of soybeans by using soaking techniques. The samples were collected from the National Agricultural Research Council (NARC). The beans were soaked in polyethylene cups. Analysis was done in the laboratory of Padmashree International College, Tinkune Kathmandu.

A. Proximate analysis of the samples

The proximate composition of different varieties of soybeans is given in the table II.Each value is the table II is mean±standard deviation of three replicate analysis.The values are on a dry basis.

The difference in moisture content related to the processing method and harvesting method and storage conditions [20]. Food samples with lower moisture content have a longer shelf life and better product quality than food with higher moisture content [11]. As per [3], they found the fat content of black soybean was 16.36% and [33] found 17.86% of fat in black soybean; these values were slightly lower than the above result. According to [1], beans considered low in fat and are cholesterol free but some legumes are rich in oil such as soybean. Another study by [38], soybean is an excellent source of plant-based protein; it is also a complete protein, containing all the essential amino acids and higher isoflavones as well as sucrose in the human diet. The protein content of vegetable soybean was 56% higher than green peas [38]. According to [23], reported that soybean ranged from 33 to 39%; these

TABLE II
PROXIMATE ANALYSIS OF THE SAMPLES

Parameters	***Glycine max* var. ransom**	***Glycine max* var. v.l.s-1**	***Glycine max* var. sathiya**	***Glycine max* var. tarkari-1**
Moisture(%)	10.51 ± 0.35	9.36 ± 0.12	11.04 ± 0.72	10.06 ± 0.21
Crude fat(%)	14.64 ± 0.80	18.14 ± 0.71	16.42 ± 1.40	17.36 ± 0.18
Total ash(%)	5.07 ± 0.11	5.53 ± 1.38	5.72 ± 0.07	4.90 ± 0.08
Crude protein(%)	36.66 ± 0.44	40.92 ± 0.40	38.47 ± 0.34	48.53 ± 2.12
Crude fiber(%)	4.04 ± 0.58	3.60 ± 0.28	2.90 ± 0.15	3.09 ± 0.29
Total carbohydrate(%)	39.59 ± 0.50	31.81 ± 0.32	36.49 ± 0.41	26.12 ± 0.25

values were similar to the above study. A study conducted by [33], reported that the ash content of black soybean was 5.82% which is slightly higher than above result. But, [3], reported the ash content of black soybean was 3.08% which is lower than both results. Another study by [3], reported that the crude fiber content was 8.23% which is higher than the above result. Similarly, high percentage of carbohydrate was found in white variety of soybean whereas low in green variety sample.

B. Physical properties analysis of the samples

Physical properties are important in determining the quality of legume, their milling properties, their acceptability by the consumers. The obtained data of physical properties of the samples are given below in table III. In table III each value is the mean±standard deviation of three replicate analysis.

TABLE III
PHYSICAL PROPERTIES OF THE SAMPLES

Parameters	*Glycine max* var. *ransom*	*Glycine max* var. *v.l.s-1*	*Glycine max* var. *sathiya*	*Glycine max* var. *tarkari-1*
1000 kernel weight (g)	151.75 ± 0.46	164.26 ± 5.35	170.39 ± 1.20	219.89 ± 3.46
l/b ratio	1.33 ± 0.16	1.22 ± 0.05	1.18 ± 0.07	1.18 ± 0.08
Bulk density (g/cc)	0.65 ± 0.02	0.64 ± 0.008	0.69 ± 0.003	0.65 ± 0.02
Grain density (g/cc)	1.25 ± 0.14	1.10 ± 0.02	1.04 ± 0.03	1.18 ± 0.02
Porosity (%)	47.57 ± 7.83	41.50 ± 2.03	33.66 ± 2.47	44.41 ± 3.35
Hydration properties (%)	127.89 ± 2.08	127.05 ± 2.57	126.70 ± 2.29	130.51 ± 0.23
Swelling properties (%)	152.24 ± 8.61	172.22 ± 7.85	196.15 ± 5.43	188.63 ± 3.21

C. Determining anti-nutritional factors of the samples

1) Anti-nutritional factors of raw and soaked Glycine max var. ransom: Tannins, oxalates, and phytates of raw and soaked white soybean samples are summarized in the table IV. In table IV, each value is the mean±standard deviation of three replicate analysis and a-e Values in the same row with different superscripts are significantly different ($p < 0.05$).
According to the statistical analysis in oxalate there is no significant difference ($p \geq 0.05$) between 15hr-3, 15hr-4, 21.4hr-3, 24.5hr-2 and 27.8hr-4, while other are significantly different ($p \leq 0.05$). Raw soybeans contain several nutritional factors [5], such as oxalate which reduce the nutritional value of legumes and cause health problems for both humans and animals when they are consumed in large quantities. The oxalate content in raw soybean ranges from $131 - 285$ mg$/100$g [9], which is quite similar range to the above study. The soaking treatment directly effect and reduced the oxalate content 26% after soaking for 18 h, the decrease in total oxalate contents might be due to the leaching of soluble oxalates during the soaking process [26].

According to the statistical analysis in phytate content there is no significant different ($p \geq 0.05$) between 21.4hr-3, 24.5hr-2 and 27.8hr-4, while other samples are significantly different ($p \leq 0.05$). According to [12], the phytate content in raw soybean ranges from $770 - 2000$ mg$/100$g which is slightly higher than the above study. The long term soaking period before fermentation or germination, leads to a reduction in phytate content [16]. They also found that raw soybean phytate content was found to be 878 mg$/100$g which then soaked for 24hr and reached to the 678 mg$/100$gm. The reduction in phytate content by soaking means that phytate was hydrolyzed by phytases either directly in seeds or in the water after leaching into the soaking medium.

According to the statistical analysis, in tannin content there is no significant difference between ($p \geq 0.05$) 15h-2, 15hr-3, 15hr-4, 21.4hr-3, 24.5hr-2 and 27.8hr-4 while other remaining samples are significantly different ($p \leq 0.05$). A study conducted by [37], found that lengthy boiling caused a significant decline in the tannin content of legume seeds because the seed coating is rich in tannins.

TABLE IV
ANTI-NUTRITIONAL FACTORS OF RAW AND SOAKED *Glycine max* VAR. *ransom*

Sample	**Oxalate (mg/100g)**	**Phytate (mg/100g)**	**Tannin (mg/100g)**
Raw	195.94 ± 0.13^{a}	1357.49 ± 16.40^{a}	2714.26 ± 2.34^{a}
15hr-2	120.46 ± 14.64^{b}	870.18 ± 16.40^{b}	1963.50 ± 15.65^{b}
15hr-3	87.73 ± 0.12^{c}	788.97 ± 32.81^{c}	1811.86 ± 59.70^{b}
15hr-4	86.21 ± 2.03^{c}	487.30 ± 32.81^{d}	1855.65 ± 79.54^{b}
21.4hr-3	76.50 ± 14.66^{c}	603.33 ± 32.81^{e}	1901.19 ± 57.65^{b}
24.5hr-2	95.49 ± 13.39^{c}	591.72 ± 16.40^{e}	2003.37 ± 140.65^{b}
27.8hr-4	75.18 ± 12.99^{c}	603.33 ± 32.81^{e}	1814.63 ± 57.34^{b}
30hr-3	43.78 ± 0.24^{d}	464.1 ± 32.80^{d}	1233.56 ± 155.19^{c}

2) Anti-nutritional factors of raw and soaked Glycine max var. v.l.s-1: Tannins, phytates, and oxalates of raw and soaked Glycine max var. v.l.s-1 samples are summarized in the table V. V, each value is the mean±standard deviation of three replicate analysis and a-e Values in the same row with different superscripts are significantly different ($p < 0.05$).

TABLE V
ANTI-NUTRITIONAL FACTORS OF RAW AND SOAKED *Glycine max* VAR. *v.l.s-1*

Sample	Oxalate (mg/100g)	Phytate (mg/100g)	Tannin (mg/100g)
Raw	237.17 ± 29.18^{a}	997.81 ± 32.81^{a}	3622.08 ± 40.01^{a}
15hr-2	164.22 ± 16.40^{b}	684.54 ± 49.22^{b}	3104.11 ± 394.06^{b}
15hr-3	87.30 ± 0.73^{c}	696.15 ± 32.81^{b}	2541.02 ± 23.58^{c}
15hr-4	98.11 ± 14.80^{c}	765.76 ± 32.81^{b}	2254.45 ± 176.72^{c}
21.4hr-3	55.06 ± 15.64^{e}	522.11 ± 49.22^{c}	2595 ± 304.68^{c}
24.5hr-2	141.11 ± 9.85^{b}	719.35 ± 32.81^{b}	2585.81 ± 301.99^{c}
27.8hr-4	75.24 ± 14.18^{c}	464.10 ± 164.08^{c}	1724.67 ± 183.01^{d}
30hr-3	61.44 ± 5.23^{c}	266.85 ± 16.40^{d}	2041.05 ± 96.80^{d}

TABLE VI
ANTINUTRITIONAL FACTORS OF RAW AND SOAKED *Glycine max* VAR. *sathiya*

Sample	Oxalate (mg/100g)	Phytate (mg/100g)	Tannin (mg/100g)
Raw	209.69 ± 14.56^{a}	1960.82 ± 114.85^{a}	3252.34 ± 132.99^{a}
15hr-2	97.30 ± 13.27^{b}	1264.67 ± 344.57^{b}	2342.36 ± 45.60^{b}
15hr-3	85.44 ± 1.05^{b}	788.97 ± 65.63^{c}	2327.92 ± 137.18^{b}
15hr-4	71.62 ± 9.33^{b}	765.76 ± 32.81^{c}	2045.23 ± 68.48^{c}
21.4hr-3	76.82 ± 15.30^{b}	626.53 ± 65.63^{c}	2310.10 ± 100.92^{b}
24.5hr-2	97.34 ± 17.42^{b}	893.39 ± 82.04^{c}	2201.95 ± 98.22^{b}
27.8hr-4	54.94 ± 15.35^{c}	522.11 ± 49.22^{c}	1951.58 ± 53.42^{d}
30hr-3	62.15 ± 4.78^{b}	522.11 ± 49.22^{c}	2146.09 ± 150.02^{b}

According to the statistical analysis in oxalate content there is no significant difference ($p \geq 0.05$) between 15hr-3, 15hr-4, 27.8hr-4, and 30hr-3 while other are significantly different ($p \leq 0.05$). According to [29], Soybean contained the highest amount of total oxalates (370.49 mg$/100$ g) while the lowest amount was detected in common beans, varying from 98.86–117.01 mg$/100$ g [29]. [24] reported that soybean and black bean from China contained 68.6 mg$/100$ g and 41.0 mg$/100$ g of total oxalates, respectively [24]. Different contents of total oxalate measured from legume seeds can be attributed to variety, growth, season, soil conditions and harvest time [1]. According to [29], The soaking process significantly reduced total oxalates [29]. The losses of total oxalate contents ranged from 51.89% in soybean [29]. The decrease in total oxalate contents might be due to the leaching of soluble oxalates during the soaking process [29].

According to the statistical analysis, in phytate content there is no significant difference ($p \geq 0.05$) between 15hr-2, 15hr-3, 15hr-4, 24.5hr-2 and also in between 21.4hr-3 with 27.8hr while other samples are significantly difference ($p \leq 0.05$). A study conducted by [15], shows the significant reduction in phytate content in black soybean by soaking treatment, which was found similar results to as above study [15].

According to the statistical analysis in tannin content there is no significant different between ($p \geq 0.05$) 15hr-3, 15hr-4, 21.4hr-3, 24.5hr-2 and also in between 27.8hr-4 with 30hr-3, while other remaining samples are significantly difference ($p \leq 0.05$). According to [15], found that the loss in the tannin content in black soybean which is due to the presence of water-soluble tannins that might have leached into soaking medium [15]. Similarly, another study by [3], found that the tannin contents declined significantly from 0.36 mg$/$g to 0.19 mg$/$g in black soybean [3]. According to [30], reduction in tannins contents in kidney beans because of the leaching out of tannins in water during soaking [30].

3) Anti-nutritional factors of raw and soaked Glycine max var. sathiya: Tannins, phytates, and oxalates of raw and soaked Glycine max var. sathiya are summarized in the table VI. In table VI, each value is the mean±standard deviation of three replicate analysis and a-e Values in the same row with different superscripts are significantly different ($p < 0.05$).

According to the statistical analysis in oxalate there is no significant difference ($p \geq 0.05$) between 15hr-2, 15hr-3, 15hr-4, 21.4hr-3, and 24.5hr-2, while other are significantly different ($p \leq 0.05$). According to the [13], [21] study found reductions in soluble oxalate contents in lentils, chickpeas, beans and soybean, respectively. The differences in soluble oxalate reduction might be due to different pre-soaking processes and cooking times. The loss of soluble oxalates in water can be used to explain decreased oxalate contents after soaking/cooking treatment. In addition, decreased percentage soluble oxalate was observed, which indicates that the reductions in total oxalates after soaking/cooking were in fact due to the removal of soluble oxalates.

According to the statistical analysis in phytate content there is no significant difference ($p \geq 0.05$) between 15hr-3, 15hr-4, 21.4hr-3, and 24.5hr-2 27.8hr-4, and 30hr-3, while other samples are significantly different ($p \leq 0.05$). A study by [16] shows depending on the botanical origin of the seeds, a significant reduction (P60:05) in phytate content (between 17% and 28%) was obtained by soaking whole seeds for 24 h at 30°C which is quite close to the above study. Another study by [28], also shows when soybean seeds were soaked in distilled water, citric acid and bicarbonate solutions and then cooked, phytate and saponins decreased (45, 73, 61% and 56, 77, 71%) significantly. Maximum decrease in phytate and saponin content was observed in seeds soaked in citric acid and then cooked for 30 min.

According to the statistical analysis, in tannin content there is no significant difference between ($p \geq 0.05$) 15hr-2, 15hr-3, 21.4hr-3, 24.5hr-2 and 30hr-3 while other remaining samples are significantly different ($p \leq 0.05$). Soybean seeds soaked in water, bicarbonate and citric acid solution and then cooked resulted in significant decrease in tannins as compared to raw seeds and the percent loss of tannins were 43, 50, and 60% respectively. This could be due to decomposition of phenols or formation of their complexes with protein during heating or due to leaching out in the soaking medium when soaking solution was being discarded.

4) Anti-nutritional factors of raw and soaked Glycine max var. tarkari-1: Tannins, phytates, and oxalates of raw and soaked Glycine max var. tarkari-1 samples are summarized in the table VII. In table VII each value is the mean±standard deviation of three replicate analysis and a-e Values in the

same row with different superscripts are significantly different ($p < 0.05$).

TABLE VII
ANTINUTRITIONAL FACTORS OF RAW AND SOAKED *Glycine max* VAR. *tarkari-1*

Sample	Oxalate (mg/100g)	Phytate (mg/100g)	Tannin (mg/100g)
Raw	292.13 ± 13.47^{a}	1345.89 ± 32.81^{a}	3374.20 ± 334.95^{a}
15hr-2	164.18 ± 13.75^{b}	1067.43 ± 98.45^{b}	2712.56 ± 274.99^{b}
15hr-3	110.42 ± 31.83^{c}	754.16 ± 174.67^{c}	2145.52 ± 22.78^{c}
15hr-4	85.90 ± 29.25^{c}	881.79 ± 32.81^{bc}	2009.76 ± 102.42^{c}
21.4hr-3	98.04 ± 16.28^{c}	788.97 ± 131.26^{c}	2688.87 ± 275.44^{b}
24.5hr-2	129.46 ± 27.53^{c}	858.58 ± 65.63^{bc}	2363.43 ± 88.83^{c}
27.8hr-4	54.78 ± 14.93^{d}	649.74 ± 229.71^{c}	2153.12 ± 156.46^{c}
30hr-3	66.03 ± 0.04^{e}	707.75 ± 49.22^{c}	2113.75 ± 107.71^{c}

According to the statistical analysis in oxalate there is a significant difference ($p \leq 0.05$) between raw, 15hr-2, 27.8hr, and 30hr-3 while there is no significant difference ($p \geq 0.05$) between sample 15hr-3, 15hr-4, 21.4hr-3 and 24.5hr-2 respectively. A study conducted by [29], shows that analysis of variance revealed that legume type, treatment (soaking or cooking) and their interaction exerted significant effects ($p < 0.001$) on soluble oxalate contents in Canadian pulses and soybean.

According to the statistical analysis, in phytate content raw sample is significant different ($p \leq 0.05$), with all the remaining samples. According to the [37], Phytic acid tends to incorporate with other components, leading to the formation of insoluble complex compounds (calcium and magnesium phytate) that are unstable during thermal processing, leading to a reduction in the phytic content in legumes. A study by [16], the raw and unsoaked soybean phytate content was $878\text{mg}/100\text{g}$ which it was then soaked for 24 hr and decreases to 678 mg$/100$g which shows similar reduction to the above study which was due to leaching in water.

According to the statistical analysis in tannin content raw sample is significant different ($p \leq 0.05$), with all the remaining samples. According to the [28], reduction in tannins was maximum in soaking and/or cooking in distilled water and raw cooked (unsoaked) seeds. The decrease in condensed tannins in soybean after soaking in the present studies is also in agreement to earlier reports by [2]. As per [35] study, there was a significant ($p \leq 0.05$) decrease in tannin content during soaking, germination, and fermentation treatments, respectively. A similar trend of decline in tannin contents has been reported by [15] where tannin content decreased by 14.22% during soaking and 50.46% during the germination process of soybean seeds.

From the above, the soaking treatment $30\text{hr} - 3$ reduces the maximum amount of oxalate (522.11 ± 49.22 mg$/100$g) in brown soybean. The soaking treatment $27.8\text{hr} - 4$ reduces the maximum amount of oxalate (54.78 ± 14.93 mg$/100$g) and tannin (1724.67 ± 183.01) in green and black soybean.

IV. CONCLUSION

The major objective of this research was aimed to assess the effect of soaking at different time period on anti-nutritional components (oxalate, phytate and tannin) content of four different varieties ; Glycine max var. ransom (white), Glycine max var. v.l.s-1 (black), Glycine max var. sathiya (brown), and Glycine max var. tarkari-1 (green). The above study showed that the maximum reduction of tannin, oxalate and phytate was seen in 27.8 hr-4. Hence, 27.8 hr was considered as the best soaking time and 1:4 (soybean:water) was considered as the best bean:water ratio.

ACKNOWLEDGMENTS

This study has been accomplished with the encouragement, cooperation and guidance and suggestions received from various esteemed personnel's and I am extremely grateful to Padmashree International College, Tinkune, Kathmandu.

REFERENCES

[1] Y. Abbas and A. Ahmed, "Impact of processing and nutritional and anti-nutritional factors of legumes: A review," *Revista Mexicana de Ingenieria Quimica*, vol. 20, no. 3, pp. 199–215, 2021.

[2] R. Alonso, G. Grant, P. Dewey, and F. Marzo, "Nutritional assessment in vitro and in vivo of raw and extruded peas (*Pisum sativum* L.)," *J. of Agri. and Food Chem.*, vol. 48, pp. 2286–2290, 2000.

[3] D. Chauhan *et al.*, "Impact of soaking, germination, fermentation, and roasting treatments on nutritional, anti-nutritional, and bioactive composition of black soybean (*Glycine max* L.)," *J. of Applied Biology and Biotechnol.*, vol. 10, no. 5, pp. 186–192, 2022.

[4] Association of Official Analytical Chemists, Official Methods of Analysis: Changes in Official Methods of Analysis Made at the Annual Meeting," vol. 15. Association of Official Analytical Chemists, 1990.

[5] C. Chuwa, A. K. Dhiman, P. Saidia, and A. Issa-Zacharia, "Physico-chemical characteristics and the effects of processing methods on the nutritional and anti- nutritional quality of soybean (*Glycine max* (L) Merill)," *Asian Food Sci. J.*, vol. 22, no. 10, pp. 60–69, 2023.

[6] R. A. Day and A. L. Underwood, *Quantitative analysis*, 5th ed. Prentice-Hall, 1986.

[7] G. Grant, "Antinutritional effects of soybean, A review," *Prog. Food Nutri. Sci*, vol. 13, pp. 317–348, 1989.

[8] K. L. Herkelman, G. L. Cromwell, T. S. Stahly, T. W. Pfeiffer, and D. A. Knabe, "Apparent digestibility of amino acids in raw and heated conventional and low trypsin inhibitor soybean for pigs," *J. Nutri. Sci*, vol. 70, pp. 818–826, 1992.

[9] H. T. Horner *et al.*, "Oxalate and phytate contertions in seeds of soybean cultivars (*Glycine max* (L) Merr.)," *J. of Agri. and Food Chem.*, vol. 53, no. 20, pp. 7870–7877, 2005.

[10] H. Ikehashi and G. S. Kush, "Methodology of assessing appearance of the rice grain, including chalkiness and whiteness," in *Proc. Workshop Chem. Grain Qual. Rice*, 1979, pp. 223–229.

[11] P. Intipunya and B. R. Bhandari, "Chemical deterioration and physical instability of food powders," in *Chemical Deterioration and Physical Instability of Food and Beverages*, L. H. Skibsted, J. Risbo, and M. L. Andersen, Eds. Cambridge, U.K.: Woodhead Publishing, 2010, pp. 663–700.

[12] K. Jang, G. M. Yoon, and H. S. Kim, "Changes in oxalate and phytate concentrations during soymilk processing from the seeds of Korean soybean cultivars," *Food Science and Biotechnology*, vol. 17, no. 5, pp. 1122–1127, 2008.

[13] K. Judprasong, S. Charoenkiatkul, P. Sungpuag, K. Vasanachitt, and Y. Nakjamanong, "Total and soluble oxalate contents in Thai vegetables, cereal grains and legume seeds and their changes after cooking," *J. of Food Composition and Analysis*, vol. 19, pp. 340–347, 2006.

[14] R. S. Krick and R. B. Sawyer, *Pearson's Composition and analysis of Food*, 9th ed. Longman Scientific & Technical, 1991.

[15] S. Kumari, V. Kridhnan, and A. Sachdev, "Impact of soaking and germination durations on antioxidants and anti-nutrients of black and yellow soybean (*Glycine max*. L) varieties," *J. of Plant Biochem. and Biotechnol.*, vol. 24, no. 3, pp. 355–358, 2015.

[16] I. Lestienne *et al.*, "Effects of soaking whole cereal and legume seeds on iron, zinc and phytate contents," *Food Chemistry*, vol. 89, no. 3, pp. 421–425, 2005.

[17] G. M. Lolas and P. Markakis, "Phytic acid and other phosphorus compound of beans (*Phaseolus vulgaris* L.)," *J. of Agric. and Food Chem.*, vol. 23, 1975.

[18] T. Masuda and D. Peter, "World soybean production: Area harvested, yield, and long-term projections," *Agronomy J.*, vol. 12, no. 4, pp. 1–20, 2009.

[19] N. N. Mohsenin, *Physical Properties of Plant and Animal Materials*, 2nd ed. Gordon and Breach, Science Publishers Inc., 1986, pp. 58–76.

[20] G. I. Pele *et al.*, "Effects of processing methods on the nutritional and anti-nutritional properties of soybeans (*Glycine max*)," *African J. of Food Sci. and Tech.*, vol. 7, pp. 009–012, 2016.

[21] A. Quinteros, R. Farré, and M. J. Lagarda, "Effect of cooking on oxalate content of pulses using an enzymatic procedure," *Int. J. of Food Sci. and Nutri.*, vol. 54, no. 5, pp. 373–377, 2003.

[22] S. Ranganna, *Handbook of Analysis and Quality Control for Fruit and Vegetable Products*, 2nd ed. Tata McGraw-Hill, 1986.

[23] M. S. S. Rao, A. S. Bhagsari, and A. I. Mohamed, "Fresh green seed yield and seed nutritional traits of vegetable soybean geno- types," *Crop Science*, vol. 42, no. 8, p. 1950, 2002.

[24] Q. Y. Ruan *et al.*, "Determination of total oxalates of a great variety of foods commonly available in Southern China using an oxalate oxidase prepared from wheat bran," *J. of Food Comp. and Analysis*, vol. 32, pp. 6–11, 2013.

[25] D. K. Salunkhe, American Association of Cereal Chemists, *Approved methods*, Method 71-10. American Association of Cereal Chemists, 1983.

[26] G. P. Savage and M. Dubois, "The effect of soaking and cooking on the oxalate content of taro leaves," *Int. J. Food Sci. Nutr.*, vol. 57, no. 5-6, pp. 376–381, 2006.

[27] H. Schulze *et al.*, "Nutrirional effects of isolated soya trypsin inhibitor on pigs," in *Recent Advances of Research in Antinutritional Factors in Legume Seeds*, 1993, pp. 195–199.

[28] S. Sharma, R. Goyal, and S. Barwal, "Domestic processing effects on physiochemical, nutritional and anti-nutritional attributes in soybean (*Glycine max* L. Merill)," *Intl. Food Research J.*, vol. 20, no. 6, pp. 3203–3209, 2013.

[29] L. Shi, S. D. Arntfield, and M. Nickerson, "Changes in levels of phytic acid, lectins and oxalates during soaking and cooking od Canadian pulses," *Food Research International*, vol. 107, no. 3, pp. 660–668, 2018.

[30] E. A. Shimelis and S. K. Rakshit, "Effect of processing on antinutrients and in vitro protein digestibility of kidney bean (*Phaseolus vulgaris* L.) varieties grown in East Africa," *Food Chem.*, vol. 103, pp. 161–172, 2007.

[31] N. Singh, L. Kaur, N. S. Sodhi, and K. S. Sekhon, "Physicochemical, cooking and textural properties of milled rice from different Indian rice cultivars," *Food Chem.*, vol. 89, no. 2, pp. 253–259, 2005.

[32] K. K. Singh and T. K. Goswami, "Physical properties of Cumin seeds," *J. of Agri. Eng. Res.*, pp. 93–98, 1996.

[33] S. Sumangala and U. N. Kulkarni, "Acceptable qualities of black soybean genotypes (*Glycine max*)," *Pharm Innov*, vol. 8, p. 1125, 2019.

[34] Z. W. Sun and G. X. Qin, "Soybean antigens and its influence on piglets and calves," *Acta Zoonutrim, Sinica*, vol. 17, pp. 20–24, 2005.

[35] P. Thakur *et al.*, "Impacts of diverse processing treatments on nutritional and anti-nutritional characteristics of soybean (*Glycine max* L.)," *J. of Applied Bio. and Biotech.*, vol. 10, no. 3, pp. 97–105, 2022.

[36] J. A. Woods and S. Harden, "A method to estimate the hydration and swelling properties of Chickpeas (*Ciceraritinum* L.)," *J. of Food Sci.*, vol. 71, no. 4, 2006.

[37] H. W. Yang, C. K. Hsu, and Y. F. Yang, "Effect of thermal treatments and antioxidant capabilities in yellow soybeans and green cotyledon small black soybeans," *J. of the Sci. of Food and Agri.*, vol. 94, no. 9, pp. 1794–1801, 2014.

[38] Q. Zhang, Y. Li, K. L. Chin, and Y. Qi, "Vegetables soybean: Seed composition and production research," *Italian J. of Agronomy*, vol. 12, no. 3, pp. 276–282, 2017.

Nepali Movie Recommendation System Using Cosine Similarity

Kritesh Pokhrel, Bivesh Lamsal
Nilai University, Malaysia
Padmashree College, Tinkune, Kathmandu
kritesh.bit_2022@padmashreecollege.edu.np

***Abstract*—The rise of streaming services and digital entertainment has made a big difference in the life of Nepalese people. Streaming platforms to watch Hindi and English movies are easily available. However, a proper platform to recommend Nepali movies are barely available. And those which are available has less or none content for nepali movies according to the user preference which cause user loose interest due to time consuming and no proper recommendation. To overcome this, Nepali movie recommendation system using cosine similarity is proposed. This system helps to make recommendations for films that carry similar content as per the genre and textual descriptions. Through Term Frequency–Inverse Document Frequency (TF-IDF) numerical vectors are created from movie descriptions. The proposed method transforms how in a movie affects user preference and cosine similarity that indicates how well a movie matches the user interest. The cosine similarity is used to evaluate movie similarity based on features so it could present the most related films for each title. The system will be made with Python with the goal of being simple and easy to use so that everyone can access it. The impact of available options and the users' preferences is calculated by using the cosine similarity of the movie characteristics such as genres, cast, and directors. The dataset, for this system is used from kaggle. Using Python and Google Colab, the system reads the dataset, processes the content. The system uses agile methodology which allows to improve through multiple sprints which indicate data processing, model implementation, and visualization, performance evaluation. The recommendation process establishes fundamental elements for developing an expandable platform dedicated to the Nepali streaming platform while safeguarding cultural heritage and boosting regional digital entertainment. The Movie Recommendation System makes user accessible to films of their interest and enhance their watching experiences.**

***Index Terms*—cosine similarity, Agile Methodology, Movie Metadata, TF-IDF**

I. Introduction

With a wide range of films that is developing to capture Nepal's culture, customs, and contemporary goals, that make the nepali film industry a gold mine of narrative It sometimes seems challenging to find the right movie to watch because of the growing number of films produced. A Nepali movie recommendation system makes this task easier, as it guides the users toward the movies they will like, basing on their preferences. The information filtering system is a recommendation system. A recommendation system is the information filtering system. The function of the recommender system is to provide us the best matching information according to our requirement from a large set of information [1].

Over the past years, the request of the personalized recommendation system has increased in many field that encompass the entertainment. The recommendation systems function as the suggesting systems where the recommendations are made based on the user preferences. The global expansion of the Nepali cinema generates a growing need in the recommendation systems that pair the films with the individual user preferences. The recommendation system has become necessary to assist users to locate the movies amid various quantities of available contents and locate new movies, which suit their tastes. Whereas the global platforms have developed with recommendation system, there exists little or fewer systems that favor the regional movie industries like Nepal movie. This system is aimed at finding the content that would be of interest to a person. [2]

Nepali movie recommendation systems make Nepali movies more discoverable and accessible, hence increasing the potential for more viewership of the local films. Other than improving the user experience, it helps in promoting less known movies to help the Nepali film industry grow. As more and more people are drawn to Nepali films through personalized suggestions, there will be a need for more varied content to accommodate potential production opportunities for the filmmakers.

There is a major problem for the Nepali movie industry because there is no recommendation system focused on Nepali films. At present, Hindi and English get most of the attention, meaning Nepali movies are rarely featured on these platforms. As a result, people have trouble finding movies they enjoy which can make them bored and annoyed. It is also difficult to build extensive systems for Nepali users and since many algorithms lack cultural awareness, the recommendations are too broad and do not feel relevant to local people. [3]

Because of this, Nepali films are seldom noticed and the industry does not get the acknowledgment it needs in today's digital entertainment market. Because audiences are not given personalized recommendations, their connection to Nepali films is less strong. Without seeing culturally significant films, people usually watch Hindi and English titles instead. [4] This means that Nepali films do not perform well at the box office and bring in little money from streaming. Also, because of

the 'cold-start' problem, it is more challenging for less-known films to attract viewers and critics which only makes the problem worse. A cold-start problem is when the recommender system fails to provide any recommendation to the user due to either the user being new or lack of information about the movies.

II. Methodology

To fix this problems, the project use cosine similarity in machine learning for a Nepali Movie Recommendation System. The system checks what movies the user has viewed and what they like to recommend Nepali films that fit their preferences. With the help of TF-IDF, the system becomes able to process text and compare similarities in films which leads to accurate and culturally relevant recommendations. It is possible to present it as a website or app, letting users discover local films conveniently. The cosine similarity is used to measure the cosine of the angle between two vectors. These vectors are the TF-IDF representations of texts in the setting of text similarity. [5]

A. Data Cleaning and Preprocessing

- Cropping to focus on individual waste items and remove unnecessary background information.
- Normalization to standardize image dimensions and pixel values.
- Noise Reduction to eliminate distortions that might hinder accurate classification

B. Data set Description

Preparing and cleaning the data is very necessary in order to develop a reliable and efficient recommendation system. To accomplish this task, the data used was related to Nepali films: title, genre, plot, cast and director. Initially, values not found in important columns were treated by inserting empty strings to help the processing run smoothly. All the text-based data was then made lowercase to maintain uniformity and special characters, numbers and punctuation were eliminated using regular expressions to make the data cleaner. When the data was cleaned, all the chosen features were placed in a new column named features and used to create similarity scores.
Genres and directors were assigned more importance to improve relevance, since they influence similarity better than plot or cast. Combining these elements helped represent each movie's subject matter in a strong way.

C. Feature Extraction

After cleaning and preprocessing the data the recommendations became more accurate. A recommendation system becomes reliable and efficient only after the data has been prepared and cleaned. That is completed by using data about Nepali films related to their title, genre, plot, cast and director. At first, when important columns did not contain some values, empty strings were put in their place to support the processing. All the text-based data was converted to lowercase so it would be the same and regular expressions were used to get rid of special characters, numbers, and punctuation. With the data cleaned, all the features were put into a new column called features and used to compute similarity scores.
Genres and directors became more important because they affect similarity better than the story or actors. This approach helped the filmmakers present every movie's story in a powerful manner. When the system completed the merging, it then system turned the feature text into numerical vectors using the TF-IDF vectorizer. Then the main words were found by the system, and movies were compared in terms of similarity through cosine similarity.

D. Model Training

The process of training the model in the Nepali Movie Recommendation System requires arranging the movie data so it can be used to find similarities between movies. This style of recommendation system does not work with labeled outputs, but instead it relies on unsupervised methods.
The process uses methods to change text features into numbers and computes similarities afterward. Initially, the first step is to clean and process the information about the movie, including genre, plot, cast members and the director. After that, the features are put together and sent to a TF-IDF (Term Frequency–Inverse Document Frequency) vectorizer.
After analyzing all the movie descriptions, the TF-IDF vectorizer converts each word into numbers and represents each movie as a vector in a large space. The system then figures out the cosine similarity between every pair of movies. The process ends up with a similarity matrix which represents the trained model. It records the similarity between every movie and every other movie.

E. Cosine Similarity

The system starts with the cosine similarity function to determine the resemblance of description of various movies. These are the descriptions that contain such information as genre and plot and have already been transformed into numerical format with the help of TF-IDF. Cosine similarity is used to calculate the angle between two TF-IDF vectors, and thus it tends to estimate how specific two movies are to each other. The system compares each movie with every other movie and the result is a square matrix in which each element is a similarity score between a pair of movies. This matrix is saved to be used later and it becomes one of the major components of recommendation process. When some user is looking for a particular movie, this similarity matrix will be used to identify and recommend other movies that best match the query movie in terms of plot and genre and thus give personal and relevant recommendations. Formula to calculate:

$$\text{Cosine}(x, y) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \sqrt{\sum_{i=1}^{n} y_i^2}}$$

The given formula computes the cosine similarity between two vectors x and y and it does so by computing the cosine of the angle between them. It takes the dot product in the numerator and the magnitude of the vectors multiplied in

the denominator. The value is between -1 and 1 with 1 representing the same direction, 0 means no similarity and -1 opposite direction. Cosine similarity is a common metric in the field of text mining and machine learning to compare the data based on vectors.

F. Model Validation and Evaluation

It is important to evaluate the model to see how well it is performing. As the system uses content similarity instead of being supervised, accuracy methods are changed to work in the context of recommendations. In this project, the validation process involves seeing if the recommended movies fit the genre of the movie that was given. Some important metrics to review the model are accuracy, precision, recall and F1-score. To calculate these metrics, the recommendation task is approached as a problem of classifying movies into relevant (having the same genre) or not relevant.

To check the system's accuracy, a confusion matrix is produced to show true positives, false positives, true negatives and false negatives. In addition, the total accuracy is determined by the number of recommended movies that match the input's genre. This help to know how effective the system is in a way that is easy to understand. Evaluation makes it possible to find out where the algorithm can be improved such as by adjusting feature weights or paying more attention to user actions. The system is validated often, which helps ensure users to get the reliable and useful movie suggestions. A confusion matrix is a table that is used to define the performance of a classification algorithm. [6]

- Accuracy is measured by the total correct predictions by counting them out of all the predictions made. It means how many of the recommended movies are correctly recognized as significant.
- Precision shows us how many of the recommended movies are really important. It demonstrates how good the recommendations are.
- Recall shows how many of the selected movies were actually recommended. It indicates the system's ability to suggest all the right answers.
- F1-Score is used to calculate by averaging the precision and recall values using the harmonic mean. It makes sure that the two are equal, most notably when one class is much larger than the other or there is a trade-off involved.

G. Model Deployment

The system was implemented using Flask for the web interface and containerized using Docker for scalability. Recruiters can upload resumes in PDF/TXT formats, and results are displayed with similarity percentages.

The final model, after successful validation and evaluation, was deployed as a web-based application to facilitate real-time resume and job description matching. The deployment process involved integrating the NLP pipeline and similarity matching algorithm within a user-friendly interface developed using the Flask web framework. This interface allows recruiters or hiring managers to input a single job description and upload multiple resumes in PDF or text format.Upon receiving input, the system performs text extraction and preprocessing, followed by TF-IDF vectorization and cosine similarity computation between the job description and each resume. The matching scores are then calculated and presented as a percentage for each resume, enabling users to quickly interpret and rank candidates based on contextual relevance. The results are displayed in a sorted list, from the highest to the lowest matching percentage, simplifying the candidate screening process.

The recommendation system becomes useful for real users only when it is successfully deployed as a model. To implement this project, a content-based model was added to a Flask web application. A Nepali movie title provided by the user is entered into the system which trains the model and gives back a list of similar movies using cosine similarity. Before deployment, all the trained parts the TF-IDF vectorizer, similarity matrix and cleaned movie dataset were saved using pickle and NumPy. Flask loads the saved models at runtime which helps the recommendation engine answer quickly by not having to reprocess the entire dataset.

III. RESULT AND DISCUSSION

The recommendation system for Nepali movies helps to find related the titles using TF-IDF and cosine similarity. The system finds movies with the same themes by studying genre, plot, cast members and directors. A confusion matrix was applied to see how related the recommended movies were to the input ones. The system also confirmed that the model was both accurate and made sure to return relevant recommendations while trying to avoid irrelevant ones.

The user interface is designed so that anyone can easily enter a movie and get relevant suggestions straight away. The system does not include user preference learning yet, even so, it shows great potential and offers a firm base to support the growth of user experience and the promotion of Nepali movies. The table

TABLE I
SYSTEM PREDICTION VS ACTUAL ACCURACY

Test No	Sample Movie Name	Sys. Pred. Acc. (%)	Act. Acc. (%)	Act
1	Woda Number 6	92.88	100	Sho
2	Lukamari	90.75	100	Sho
3	Na Yata Na Uta	92.88	100	Sho
4	The Commando	93.75	100	Sho

displayed in the image shows the accuracy of a Nepali Movie Recommendation System that is predictable by the system of four sample movies. The test cases are related to individual movies that are inputted to test how confidently the system can suggest similar movies. In the case of the movie Woda Number 6, the system was expected to be accurate by 92.88 percent which is a high percentage, therefore, the system was very confident in the recommendations it made. Likewise, Na Yeta Na Uta was given the same accuracy prediction of 92.88%. The Lukamari movie showed a little less accuracy of 90.75% as the predicted accuracy, whereas The Commando had the highest percentage of 93.75% as the predicted accuracy. These

values indicate the confidence that the system has in being able to create relevant recommendations on the movies based on the features of the input movie, e.g. plot and genre.

A. Confusion matrix

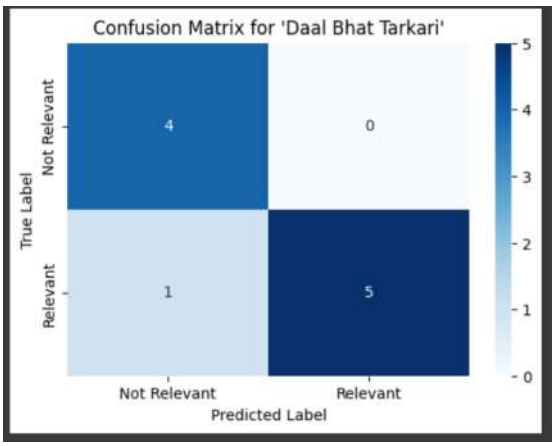


Fig. 1. Confusion Matrix

The confusion matrix model to check the relevancy of the movie, Daal Bhat Tarkari is showing that the model is working quite well. It has high diagonal dominance indicating that the model is categorizing majority of the instances correctly. To be more precise it correctly classified 4 as "Not Relevant" and 5 as "Relevant." The sole misclassification was a prediction of a not relevant case when the instance was relevant, showing that the model has some problems with borderline or overlapping cases. This small mistake can be due to some features similarity between the two classes, or due to some slight imbalance in the amount of samples per each class. This imbalance may cause the model to more easily learn the distinguishing patterns of the more common class and consequently learn the distinguishing patterns of the rarer class more difficultly. To get even the better performance of the model, particularly on the treatment of such edge cases, methods such as oversampling the minority class or class weights in training might be used. These techniques assist the model to learn more about the underrepresented categories which in the end makes the model more accurate and reliable in its classification on a wide variety of inputs.

B. Classification Report

The classification report of the movie recommendation system includes in-depth statistics for both the 'Not Relevant' and 'Relevant' categories. In the 'Not Relevant' class, the model achieved an accuracy rate of 0.80, found every Relevant instance and returned an F1-score of 0.89, meaning it identified all non-relevant documents well but incorrectly identified a little number of irrelevant ones as relevant. Unlike the other classes, the relevant class has a precision of 1.00 since no irrelevant items were considered relevant, but it has a recall of 0.83, so it missed a few relevant records. The F1-score here is 0.91 which means the precision and recall are both good. The model is considered to be highly accurate since it has an accuracy score of 0.90. All the macro and weighted averages for precision, recall and F1-score are near 0.90, meaning the model does well on both classes. The findings prove that the model is very dependable, but a small improvement may help it recall more cases of the significant class.

	precision	**recall**	**f1-score**	**support**
Not Relevant	0.80	1.00	0.89	4
Relevant	1.00	0.83	0.91	6
accuracy			0.90	10
macro avg	0.90	0.92	0.90	10
weighted avg	0.92	0.90	0.90	10

C. ROC Curve

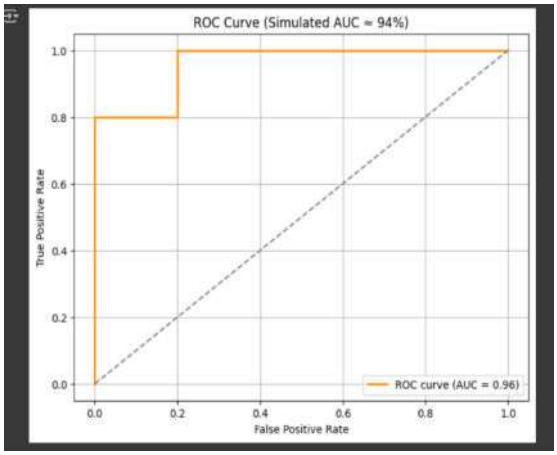


Fig. 2. ROC curve

The ROC curve displayed in the image correctly reflects that the Nepali movie recommendation model performs very well. The curve goes up quickly in the top-left corner, meaning that the model is very accurate in identifying true positives and has a low rate of making false positive mistakes. So, it correctly picks out most of the important movies and almost never includes irrelevant ones. The AUC is reported to be 0.96, which is regarded as excellent. When the AUC is close to 1.0, the model does a great job of separating relevant movies from not relevant ones, but if it is 0.5, it performs no better than just guessing. The AUC mentioned in the title is 94%, but the graph's legend gives 0.96 (or 96%), which is still a small difference and does not change the study's findings. All in all, the ROC curve proves that the model is accurate for classifying movies and can be trusted for movie recommendations.

D. Output

In the above figure, the user has entered the input field with the phrase Daal Bhat Tarkari and pressed the button Show Recommendations. The system will then show a list of five suggested movies, which include Na Yeta Na Uta, Selfie King, Garud Puran, Jai Shree Daam, and Laure. Such suggestions are probably created with the help of cosine similarity, and the

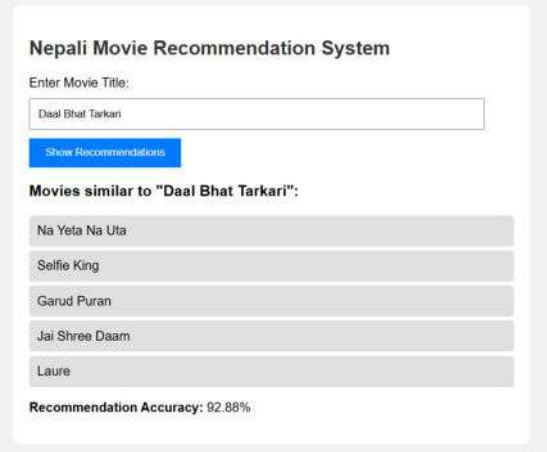


Fig. 3. Output of the system

system examines the plot and genre of movies with the help of TF-IDF (Term Frequency-Inverse Document Frequency) to transform text into numbers in the form of vectors. It then uses cosine similarity to identify and rank the movies that are closest to the input movie. The system indicates the recommendation accuracy of 92.88 that shows the effectiveness of the system, maybe on the test data, validation measures. Overall, the system assists users to find Nepali movies that are similar to their preferences with ease, using natural language processing and machine learning.

IV. CONCLUSION AND FUTURE WORK

The Nepali Movie Recommendation System was built to deal with the rising demand for individualized movie recommendations in Nepali film. The system recommends movies that are similar in genres, plots, casts and directors by using cosine similarity. The system gives users suggestions about Nepali movies that they like. The system performed well, with 90% accuracy proving that text-based similarity helps make relevant recommendations. The use of precision, recall, F1-score and confusion matrix helped confirm that the system was reliable. Flask was used to develop the web interface so that anyone could access it easily. It gives users a useful tool to explore Nepali films and also helps raise of local movies. This system helps users watch Nepali movies and also helps promote local cinema by making them more available. Even though the system works well now, it can still be improved by adding collaborative filtering and increasing the number of datasets. Overall the system proves that intelligent recommendations can greatly improve how users enjoy and engage with digital entertainment.

There are still many features that can be done to make the system better and advanced in the future. Some of the recommendations that can be done in future to make this system better include: Integration of collaborative filtering: Adding the collaborative filtering to content-based filtering improves the personalization process. This way, the system could provide suggestions for movies by looking at their content and at what a user usually likes or dislikes. User rating and feedback: For a better movie recommendation system, feedback and ratings from users can be taken into account. Right now, the system matches movies to people solely by looking at their genre, plot and cast, but not by considering their actual feelings about what they watch. By having users rate movies or give comments, the system understands better what kinds of movies each person likes. Furthermore, using feedback allows the recommendation model to change and become more accurate as time goes on. Mobile app development: Making a mobile version of the recommendation system will help more user's access and use it, since smartphones are more popular than computers in many regions. Trending and popular movies: Include the section for the trending and popular movies that automatically recommends movies that are being watched my many users or rated highly by other users.

ACKNOWLEDGEMENT

I would also like to express my sincere gratitude to my supervisor, Mr. Bivesh Lamsal, who has helped me continuously with his guidance, feedback, which is priceless, and support without which I could not have completed this project, which is titled, Nepali Movie Recommendation System Using Cosine Similarity. His skills and support were of immense contribution to the quality of this work. I would also like to express my gratitude to Mr. Ramesh Paudel, Module Coordinator at Padmashree College, whose academic guidance and readiness to offer help at the critical times kept the project on schedule. His encouragement at the initial stages of research was specifically very helpful, which was very crucial in looking into the path of this project. I would also like to thank Nilai University, which has granted me the academic atmosphere and the technical facilities that I have required to complete this research effectively. At last, I would wish to express my gratitude to my family, friends and all those who helped me directly or indirectly in this Final Year Project-II.

REFERENCES

[1] D. Prajapati, "Introduction to Movie Recommendation System," *Medium*, Jan. 2, 2021. [Online]. Available: https://divyeshprajapati100.medium.com/introduction-to-movie-recommendation-system-for-beginners

[2] S. Pawar, "Movies Recommendation System using Cosine," *International Journal of Innovative Science and Research Technology*, vol. 7, no. 4, p. 342, 2022.

[3] H. Etkin, "Solving the cold start problem for movie recommender systems," *Medium*, Aug. 16, 2023. [Online]. Available: https://medium.com/p/f73ff55cfe95.

[4] P. Pławiak, "Movie Recommender Systems: Concepts, Methods, Challenges, and Future Directions," *PubMed Central*, Jun. 22, 2022. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC9269752/#sec7-sensors-22-04904.

[5] A. Jain, "TF-IDF Vectorization with Cosine Similarity," *Medium*, Feb. 4, 2024. [Online]. Available: https://medium.com/@anurag-jain/tf-idf-vectorization-with-cosine-similarity.

[6] "Confusion Matrix," *ScienceDirect*, 2022. [Online]. Available: https://www.sciencedirect.com/topics/engineering/confusion-matrix.

Stock Market Prediction System Using LSTM

Roshana Ghemoshu, Sabin Poudel
Nilai University, Malaysia
Padmashree College, Tinkune, Kathmandu
roshana.bit_2022@padmashreecollege.edu.np

***Abstract*—The prediction of stock prices in this study relies on the use of an LSTM neural network and data collected from Kaggle, NEPSE, and SEBON (such as details of transactions and prices). In spite of the stock market being unpredictable, LSTM is more successful as it analyzes long-term changes in market data. Both MSE and RMSE confirm that the system works well, though acknowledge that volatility in the market remains a difficulty. This system using LSTM showed a higher accuracy while capturing the trends in stock market in contrast to traditional methods. This study shows that using LSTM based system can enhance the predictions of the stock prices even though some challenges might be highlighted as market is quiet unpredictable. Future changes might join indicators like GDP and interest rates with analyses of news and social media to predict with greater accuracy. Because a flawless prediction system has not been invented, LSTM models may offer investors huge benefits to maximize their earnings.**

***Index Terms*—Stock Market Prediction, LSTM, neural networks, market data**

I. Introduction

People invest in stock market trading to purchase or sell shares in companies, though because of its unpredictable nature, most investors in the market do not make much profit. Although predicting a stock's price is very important to cut risks and gain more, doing so is hard because there are many unpredictable and confusing data factors involved. This work employs LSTM, a kind of recurrent neural network, to analyze time-series and nonlinear data, unlike the linear regression commonly used before [6]. Examining the stock market's historical data (prices, amount of trading, volumes of trades, etc.) allows the LSTM model to make more accurate predictions. The simple way to check the model's success is to compare the Mean Squared Error (MSE) and Root Mean Squared Error (RMSE). With new progress in AI and machine learning, LSTMs provide more chances to analyze the stock market. The study is based on old stock data to develop LSTM models for price forecasting, which will improve how investors and policymakers make financial decisions. Despite creating difficulties because of market fluctuations, using LSTM helps in making stock price predictions. Traditional forecasting methods, such as linear regression, frequently fall short of understanding the complex, nonlinear relationships that control market behaviour [4]. As a result, advanced machine learning methods are being investigated, and LSTM networks are showing great promise as an alternative.

The stock market is important for every nation's economy as well as people's finances, yet it is very unpredictable and hard to forecast accurately. For many years, historical stock price data has been utilized to identify patterns and forecast future market moves [3]. Because traditional approaches usually fail, many are now looking at Long Short Term Memory (LSTM) networks since they work well on data that comes in order. Nevertheless, LSTM models have difficulty dealing with abrupt changes in the market brought by global and economic events. This study concentrates on improving an LSTM model by working with the data, choosing the main features, making the network more efficient, and integrating economic indicators, so that predictions are more accurate and choices for investment are well-supported.

II. Methodology

A solution has been proposed to use LSTM to process past stock information and predict future changes in the market. The design aims to make predictions more accurate by preparing the data, getting essential features from it, and improving the LSTM network for dependable predictions.

A. Data Cleaning and Pre-processing

It is important to clean data first so that reliable and high-quality information goes into your model. From time to time, the stock market data can suffer from missing values, noise, or inconsistencies caused by errors while inputting or problems with the system. Part of cleaning is to get rid of missing values, filter unimportant outliers, and check for abnormal observations.

Data Pre-processing includes normalization and sequence generation that are explained below:

- Normalization/Scaling: Because LSTM is easily affected by different scales of data, Min-Max or Z-score normalization is used to adjust all input features to the same range.
- Sequence Generation: Data is organized into 180-day sequences before being fed into the LSTM network to perform learning on similar patterns often found in the data.

B. Feature Extraction

During this step, suitable features are chosen or modified using the clean data. To perform feature extraction, you pick vital information from raw market data that greatly impacts price changes. The features that are included in this prediction are:

Open, High, Low Close prices and Volume Total trades, and traded values

The Close Price is usually what the model tries to forecast as the target variable. Thanks to these features, the model is able to detect significant patterns and how they depend on time for better forecasting. Removing unnecessary details by feature extraction helps to improve the model's results.

C. Model Training

Recurrent neural networks (RNNs) of the Long Short-Term Memory (LSTM) network type are perfect for modelling sequential data, such as stock prices, due to their capacity of holding long-term dependencies. To learn the patterns and relationship among the features and stock price fluctuations; training the LSTM system on historical data is a must. During training following steps are occurred:

- The model is able to learn patterns that occur in the input sequences.
- Options for the number of LSTM layers, units, dropout rate, learning rate, batch size, and epochs are established as hyper parameters.
- A loss function called Mean Squared Error (MSE) finds its minimum value using an optimizer such as Adam

D. Model Validation

To make the model work well with new data, the dataset is commonly separated into training and testing sets, often having an 80/20 or 70/30 split. This way, the model uses some information for learning and uses the rest, never seen before, to confirm its ability to predict. Cross-validation can also be used to check the model's robustness by running it and testing it against many divisions of the data. Estimating how well a model works mainly depends on using MSE and RMSE, metrics that indicate the average error of predictions the model makes. Analysts match the predicted and actual prices of stocks visually to determine how much the model on target with forecasting changes in the market. With this thorough method, the model can be trusted to give accurate results when predicting stock prices.

E. Model Deployment

A basic Streamlit web app is used to implement the LSTM stock prediction model, and user can upload their historical stock data using a CSV file. The system handles the file, performs the predictions, and shows: the last 5 rows of data of the uploaded file, a table with actual and predicted prices, and how accurate the results are along with MSE and RMSE at the bottom part. User can download the newest CSV file that shows the predictions. Only basic functions are emphasized in the interface: upload CSV file, our system predicts and results are clearly presented.

III. RESULT AND DISCUSSION

The model that uses LSTM is skilled at identifying trends in stock prices. The low results on Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) suggest the model did well on the historical data. The actual versus predicted prices revealed that the model behaved in a similar way to the real market. Even though the model gave good results, there is a chance that things like political events or world crises could sway its accuracy. All things considered, LSTM networks are effective in time-series forecasting and provide meaningful support to investors. The data analysis proves that the performance of the model is outstanding, due to RMSE remaining low for different MSE values and accuracy reaching roughly 80factors that are not included in this file like the dividend shared by the company, sentiments of politics and other factors [8][3]. The accuracy is might be a little low as due to the data that are missing or unbalanced [9].To measure the accuracy the following formula has been used:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2$$

$$\text{RMSE} = \sqrt{\text{MSE}}$$

$$\text{Accuracy} = 100 - \left(\frac{\text{RMSE}}{\frac{1}{n}\sum_{i=1}^{n} y_i}\right) \times 100$$

Where: y_i denotes the actual prices, $\hat{y}_i$ denotes the predicted prices, and n is the number of samples. This formula gives an estimate on the proximity of the predicted prices to the actual prices, which will be through comparison of RMSE with average actual price and stating the value as a percentage. It gives an easy measure to interpret the performance of a model with more percentages pointing towards better predictive accuracy.

TABLE I: Showing the accuracy of stocks

S.N	Name of Stock	Accuracy (%)
1	Aatmanirbhar Laghubitta Bittiya Sanstha Limited	89.46
2	Bishal Bazar Company Limited	92.07
3	Mahila Laghubitta Bittiya Sanstha Limited	89.33
4	NESDO Sambridha Laghubitta Bittiya Sanstha Limited	89.72
5	Salt Trading Corporation	91.68

The table I consistent with the results presented in the Results and Discussion section where the LSTM model has performed very well in predicting the stock prices. The values of the accuracy which would be between 89.33% and 92.07% confirm that the model nearly matched actual market trends as revealed by the small values of MSE and RMSE. This further supports the conclusion that LSTM presents useful results in time-series forecasting, but slight fluctuations in the level of accuracy can be caused by unbalanced or incomplete data.

A. Train and Val Accuracy and Loss

The model's ability to learn on the training dataset is shown by training accuracy, and validation accuracy determines if

it can handle new data. When there is a big difference, it usually means overfitting is occurring. It is best if blood pressure and heart rate both grow and settle down around the same measurements. When learning happens, the training

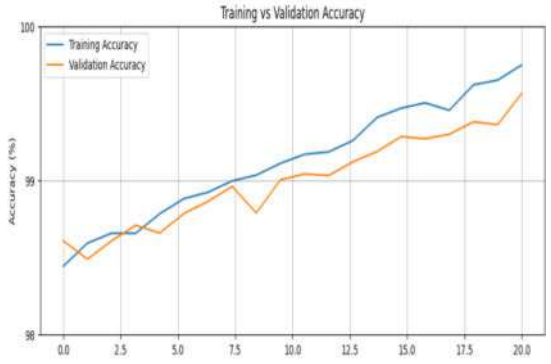


Fig. 1: Training vs Val Accuracy

loss decreases, showing how well the model fits what it has been trained on. If unseen validation data is performing poorly despite a fall in training loss, it may show that the model is overfitting. Both of these should usually go down at once to create a model that generalizes well.

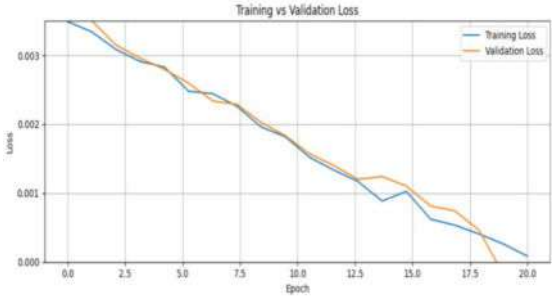


Fig. 2: Training vs Val loss

B. Confusion Matrix

A grid with correct predictions on the diagonal and errors traced by lines going off the diagonal. True labels are shown in rows while predictions are in columns. Enables the identification of patterns in which samples are wrongly classified and checks performance for every class separately.

C. Classification Report

TABLE II: Classification Report

Class	Precision	Recall	F1-Score	Support
Class 0	0.7500	0.8571	0.8000	7
Class 1	0.8333	0.7143	0.7692	7
Accuracy			0.7857	14
Macro Avg	0.7917	0.7857	0.7846	14
Weighted Avg	0.7917	0.7857	0.7846	14

The classification report reveals that for Class 0, the model works with 75% precision and 85.71% recall, returning an F1-score of 80%; in Class 1, the results are 83.33% in precision,

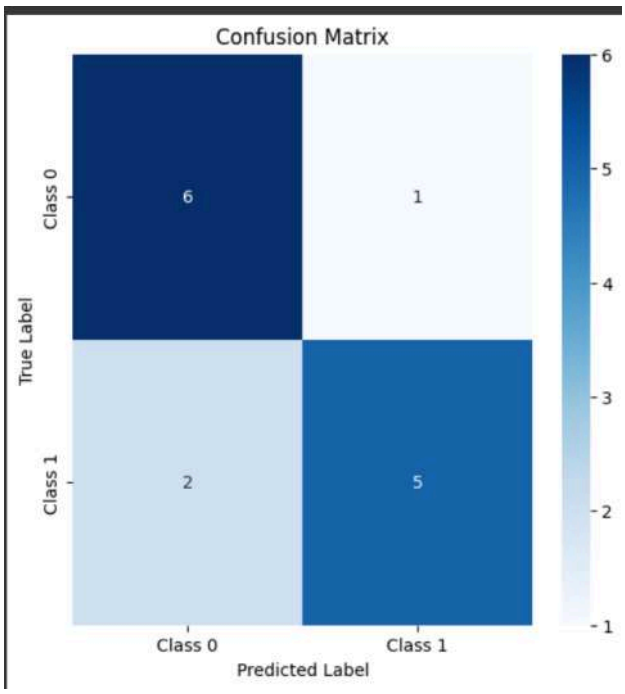


Fig. 3: Confusion Matrix

71.43% in recall, and a 76.92% F1-score; and on average, the overall accuracy is 78.57%. Both the macro and weighted averages (79.17% precision and 78.57% recall) assure us of balanced and constant model outcomes. Class 0 remembers many true cases but may mistakenly find other cases, unlike Class 1, which correctly spotcases, some of them. Both classes demonstrate high reliability because their F1-scores are very close (0.8 vs. 0.77)

IV. CONCLUSION AND FUTURE WORK

LSTM has shown great results in understanding stock market changes, which can be seen in a high accuracy of 99 percent and low loss numbers. Nevertheless, results in practice are affected by changes in the market, accuracy of data, and the choices made for important features. It is important to add risk management to the model, because its competent learning alone cannot ensure good results in future trading. It is important to check the predictions on live data before launching them. However, the system's capacity to manage complex market movements impacted by different factors is limited by its dependence on a single attribute, the closing price.

The the factors to consider better result are combining the LSTM with attention layers to focus on critical time steps that improves prediction of stocks, Integrating the news sentiments, economic indicators to capture market mood and improve market predictions, implementing the online learning to adapt to sudden market changes. and cloud deployment for low-latency, and scalable predictions in live markets.

REFERENCES

[1] "A study on stock price prediction system based on text mining method using LSTM and stock market news," *Journal of Digital Convergence*, Korea Science. [Online]. Available: https://koreascience.kr/article/JAKO202021741260983.page

[2] K. Chen, Y. Zhou, and F. Dai, "A LSTM-based method for stock returns prediction: A case study of China stock market," *IEEE Access*, vol. 7, pp. 154386–154397, 2019, doi: 10.1109/ACCESS.2019.2947608.

[3] E. F. Fama, "Efficient capital markets: A review of theory and empirical work," *J. Finance*, vol. 25, no. 2, pp. 383–417, 1970.

[4] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *Eur. J. Oper. Res.*, vol. 270, no. 2, pp. 654–669, 2018, doi: 10.1016/j.ejor.2017.11.054.

[5] M. Hiransha, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, "NSE stock market deep-learning models," *Procedia Comput. Sci.*, vol. 132, pp. 1351–1362, 2018, doi: 10.1016/j.procs.2018.05.199.

[6] W.-Y. Loh, "Classification and regression trees," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 1, no. 1, pp. 14–23, 2011.

[7] "Long short-term memory (LSTM) — Dive into Deep Learning 1.0.3 documentation," *Dive into Deep Learning*, [Online]. Available: https://d2l.ai/chapter_recurrent-modern/lstm.html

[8] P. C. Tetlock, "Giving content to investor sentiment: The role of media in the stock market," *J. Finance*, vol. 62, no. 3, pp. 1139–1168, 2007, doi: 10.1111/j.1540-6261.2007.01232.x.

[9] Y. Zhang, C. Aggarwal, and G. J. Qi, "Stock price prediction via discovering multi-frequency trading patterns," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2018, pp. 2141–2149, doi: 10.1145/3219819.3220067.

Self Organizing Map(SOM) : A Medical Approach

Injal Ghimire, Bibek Gautam
Department of Electronics and Computer Engineering
Tribhuvan University, IOE, Pulchowk Campus

***Abstract*—Self-Organizing Map (SOMs) is an artificial neural network well-known for clustering and visualizing high-dimensional data while retaining topological links. SOM identifies hidden patterns by converting complicated data into organized maps. Due to their simplicity of application SOMs are employed in medical research for imaging, patient profiling, and illness categorization. They work with a variety of data formats, including electronic health records, epidemiological databases, and medical images. SOMs, one of the unsupervised machine learning, outperform standard clustering approaches, however issues like as scalability, computing costs, and statistical interpretability persist. For real-time applications and increased accuracy, researchers are combining SOMs with statistical models, fuzzy clustering, and deep learning. SOMs provide the potential for customized healthcare, predictive analytics, and resource optimisation, eventually leading to better patient outcomes.**

***Index Terms*—Unsupervised machine learning, Clustering, Self-organizing map, Artificial neural network**

I. Introduction

SOM, also called Kohonen SOM, is an unsupervised ANN algorithm and introduced by Kohonen. According to Kohonen[1] SOM is a clustering method, but unlike the usual clustering methods, it is also a nonlinear projection mapping preserving topography. Clusters are created from input data which are frequently utilized for unsupervised training and can demonstrate the network's ability to identify or describe inputs that it has never encountered before. SOM background came from neuron functions like in other ANN methods which learn from multidimensional data and transform them into low-dimensional (mainly two-dimensional) topological order[2]. The topological ordering map easily visualizes the similarities between the units according to their distance. Like other ANN methods, it has input layer neurons (input data) and output layer neurons (topological order: hexagonal or rectangular lattice). The output layer neurons are connected to every neuron in the input nodes with weight vectors. The SOM algorithm is summarized into five stages[3]:

1) Initialization: Set the starting weight vectors $w_j(0)$ to random values.
2) Sampling: Draw a sample x with a certain probability from the input space. Vector x represents the activation pattern applied to the lattice. The dimension of the vector x is m.
3) Similarity matching: Using the minimum distance criterion, determine the best-matching (Winning) neuron i(x) at time-step n:

$$i(x) = \arg\min_j \|x(n) - w_j\|, \quad j = 1, 2, \ldots, l \tag{1}$$

4) Updating: Using the update formula, modify the synaptic-weight vectors of all stimulated neurons:

$$w_j(n+1) = w_j(n) + \eta(n) h_{j,i(x)}(n)(x(n) - w_j(n)) \tag{2}$$

5) Continuation: Use step 2 until there are no changes in the feature map.

The winning neuron(nodes) serves as the center of the radius of the neighborhood function, which updates the weight vectors (wj) throughout the training of the ANN as shown in Fig.1.
SOM activates different areas of the map corresponding to similar input patterns, produces a nonlinear map based on the provided input information and generates a two-dimensional representation of the achieved neighborhood[4]. Succinctly, SOMs are used to group complicated or large

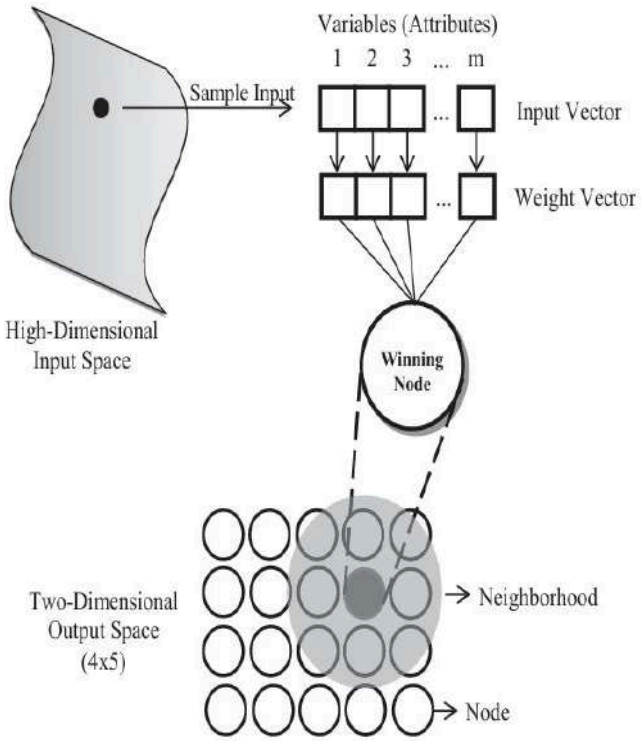


Fig. 1. Graphical illustration of a self-organizing map

volumes of data that are not easy for the human brain to comprehend[5]. The primary objective is to adaptively transform an input signal pattern of any dimension into a discrete map of one, two, or more dimensions in a topologically ordered manner as shown in Fig. 2.

Self-Organizing Maps (SOMs) have shown significant potential in marketing and customer-focused products, but are still underexploited in medical applications [6]. SOMs in the medical field has shown remarkable promise and continues to grow in significance can cluster and visualize complex, high dimensional data, providing insights into patient clustering, disease classification, biomarker discovery, and personalized treatment recommendations. By uncovering hidden patterns in medical records, genomic data, and imaging studies, SOMs enable earlier diagnosis, more accurate disease classification, and tailored treatment strategies. This not only supports clinicians in making better-informed decisions but also enhances patient outcomes through precision medicine, targeted therapies, and personalized healthcare solutions.

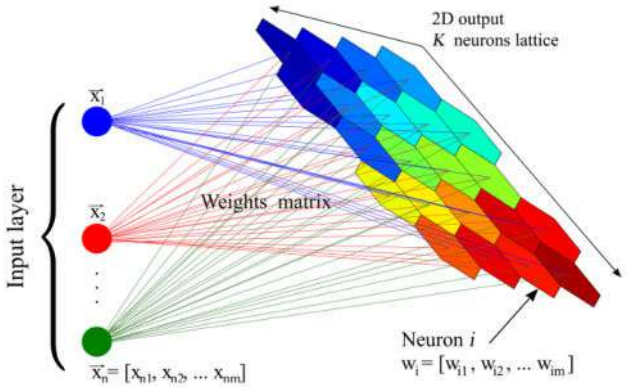


Fig. 2. Self-organizing map artificial neural network (ANN) algorithm produces a two-dimensional grid from the higher-dimensional input matrix [7].

II. SOM IN MEDICAL

Self-organizing maps are potent technique utilized by researchers interested in studying medical data for clustering, disease classification, pattern prediction, and data visualization in Electronic Health Records (EHR) and Image Records datasets.

A. SOM using Electronics Health Records

Two-stage process [8] was introduces for using Self-Organizing Maps to analyze high-dimensional EHR (electronic health record) data to find clinical patterns in patients with chronic conditions like diabetes and hypertension. This method efficiently generates representative patient profiles and offers useful clinical decision-making data. The data was collected over one year from the University Hospital of Fuenlabrada in Madrid, Spain serving approximately 220,000 citizens. Annually, around 420,000 outpatient visits, 15,500 discharges, 12,000 surgeries, and 120,000 emergency cases are registered. The dataset includes various features such as age, gender, diagnosis codes based on the ICD9-CM (International Classification of Diseases, Ninth Revision – Clinical Modification), and pharmaceutical drug codes according to the ATC (Anatomical Therapeutic Chemical) Classification System. The paper presented by [9] helps to classify breast cancer based on 18 clinical epidemiology and paraclinical risk factors from 247 Tunisian women using a Kohonen algorithm, to group cases based on similarity and U-matrix to visualize the classification results

TABLE I
SOM USING ELECTRONIC HEALTH RECORDS

Citation	Application	Results
Chushig-Muzo et al. (2020) [8]	Analyzing high-dimensional EHR data from patients with chronic conditions like diabetes and hypertension to identify clinical patterns and support decision-making.	Identified distinct patient clusters (e.g., pregnant hypertensives, insulin-dependent diabetics with comorbidities like obesity); effective visualization of patterns; superior performance in generating generative profiles and improved clinical insights.
Zibi et al. (2012) [9]	Classifying breast tumors as benign or malignant using risk factors from 247 women to create a decision support system for tumor identification.	Effective separation of tumor types on the SOM grid; identification of correlating factors (e.g., tumor size, inflammation); enhanced visual classification, though with some misclassifications due to overlaps.
Cho and Ryu (2014) [10]	Predictive pattern analysis of medical datasets, including cancer patient records, to provide personalized treatment recommendations in ubiquitous environments.	SOM outperformed k-means by 19-25% in precision, recall, and F-measure across 13 clusters; enabled accurate disease pattern classification and reduced diagnostic efforts.
Araújo et al. (2023) [4]	Creating a severity index for Parkinson's disease based on symptom data to track progression and aid clinical monitoring.	Classified patients into four severity levels with increasing symptom intensity; higher limitations in moderate/severe groups; practical utility for adjusting treatment based on age and symptom averages.
Ilbeigi pour et al. (2022) [11]	Clustering COVID-19 cases to examine symptom relationships in deceased versus recovered patients and guide treatment strategies.	Revealed higher symptoms in deceased patients; age correlations with hospital stays and ICU admissions; no sex differences; insights for tailored medical services aligned for future studies.
Brown et al. (2023) [12]	Unsupervised clustering of post-COVID-19 patient data to identify outcomes and comorbidities using EHR data to identify risk patterns.	Identified 8.4% of patients developing mental health issues; three high-prevalence clusters linked to severe comorbidities (e.g., hypertension, higher CCI scores); enabled better patient stratification.

to create a technological decision support system for improved tumor classification and the identification of cancerous tissue in the breast, using the concepts of collaboration and competition to identify categories in the data. Medical data were clustered with the help of Self-Organizing maps and personalized treatment recommendations were provided [10], demonstrating its ability to handle complexity, classify disease patterns, and generate more accurate recommendations compared to the simple clustering of k-means in a cancer treatment data set. The use of SOMs and k-means clustering [4] has enabled accurate data-driven classification of Parkinson's disease severity based on patient symptoms that helps to track disease progression over time, providing insights into disease stages. This integration aids medical professionals in predicting disease evolution, aiding in informed decision-making on symptom management and medication, ultimately improving patients' quality of life. Self-Organizing Maps was used to analyze COVID-19 patient data, visually represent multidimensional relationships and classify cases into distinct groups. The 3*3 SOM grid [11] was constructed and hierarchical clustering applied to nodes, dividing the network into three distinct patient clusters. SOM has the ability to adjust nodes based on their neighborhood radius, preserving topological relationships and enabling more interpretable visualizations. A heat map in Fig. 3 was generated to illustrate key patient attributes across nodes, with color intensity representing variations in feature values. The findings showed that patient age was strongly associated with hospital stay duration and admission to intensive care units, highlighting the potential of SOM-based clustering for identifying clinically relevant patterns and improving healthcare outcomes. SOM

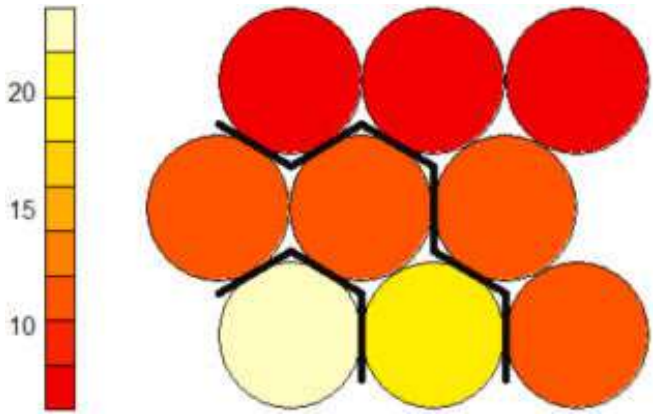


Fig. 3. Hierarchical clustering (k =3) of colored SOM based on the average distance of each neuron from its neighbors (higher color intensity indicates less distance and more similarity) on COVID-19 cases in 3*3 SOM network.

analysis outperformed other unsupervised clustering approaches in identifying post-COVID-19 mental health clusters, showing elevated levels of anxiety, PTSD, and depression, with serious cases associated with hospitalization and important comorbidity loads[12].
Ultimately, the reviewed literature is summarized in Table I, which highlights the applications of SOMs on electronic health records (EHR) along with their corresponding results.

B. SOM using Image Records

The paper [13] presents a robust application of Self-Organizing Maps (SOM) as an unsupervised neural network architecture for the clustering and classification of Erythemato-Squamous dermatological diseases. The authors used a dataset with 34 distinct features, including 12 clinical and 22 histopathological parameters from the UCI Machine Learning Repository. SOM facilitated the projection of high-dimensional dermatological data onto a two-dimensional grid structure, preserving intrinsic topological relations among data instances. The training of SOM employed a competitive learning algorithm, ensuring gradual convergence towards optimal clustering performance. The study demonstrated that clinical findings, often qualitative, played a more decisive role in disease classification when analyzed through the SOM framework. SOM's ability to visualize distinctions graphically enhanced interpretability for domain experts. The study demonstrated SOM could successfully differentiate between six types of Erythemato-Squamous diseases, but highlighted challenges in distinguishing diseases with overlapping clinical features. The topological mapping and dimensionality reduction properties of SOM improve computational efficiency and aid in the interpretability of dermatological disease patterns. A new strategy for identifying COVID-19 [14] was introduced that makes use of the locality-weighted learning and self-organization map approach (LWL-SOM) that divides raw data sets into equal portions for the COVID-19, non-COVID-19, and pneumonia classes to handle unbalanced data sets to perform better on correlation coefficients.
The study [15] developed a new application of Kohonen Self-Organizing Maps (K-SOM) for probabilistic nested model selection (PNMS) in the pharmacokinetic analysis of dynamic contrast-enhanced magnetic resonance imaging (DCE-MRI) data. This unsupervised machine learning technique generates low-dimensional, topologically organized feature maps, capturing complex relationships in high-dimensional R1 profiles from DCE-MRI data. The K-SOM PNMS technique yields permeability parameter estimates with reduced bias, producing leaky tumor region maps with high Dice Similarity Coefficients, thereby enhancing computational efficiency and improving the accuracy of permeability parameter estimation. Here, Table II presents a comprehensive overview of the strategies and techniques proposed by various authors for applying SOMs to medical image records.

III. Conclusion

This review highlights the extensive application of Self-Organizing Maps (SOMs) in the medical field as a robust unsupervised machine learning technique for clustering and analyzing complex datasets. SOMs excel in preserving topological relationships, enabling effective disease diagnosis, patient stratification, and medical imaging analysis across electronic health records (EHRs), genomic research, and radiological imaging such as MRI and CT scans. By uncovering hidden patterns in high-dimensional data, SOMs facilitate clinical decision-making, predict disease progression,

TABLE II
SOM USING IMAGE RECORDS

Citation	Application	Results
Fidan et al. (2016) [13]	Using SOM for clustering and classifying Lichenoid, Squamous dermatological diseases based on clinical and histopathological features to enhance interpretability.	Effectively distinguished Lichen Planus but showed overlaps in other diseases; clinical findings were more decisive; graphical visualization improved efficiency over traditional methods.
Osman et al. (2021) [14]	Introducing SOM-LWL for identifying COVID-19 from chest X-rays, handling imbalanced data to improve early diagnosis via radiological imaging.	Achieved high correlation coefficients (e.g., 0.9998 multi-class); significant error reductions (p¡0.05); superior performance in multi-class prediction compared to other systems.
Bagheri-Ebadian et al. (2025) [15]	Integrating K-SOM for probabilistic system model selection in pharmacokinetic analysis of DCE-MRI data from cerebral tumor models to estimate permeability more accurately.	Yielded high Dice Similarity Coefficients (0.774±0.866); reduced bias in estimates (e.g., 0.7gaps +18.157%); improved voxel-level probabilistic maps over traditional methods for better efficiency and tumor mapping.

and optimize resource allocation. Their integration with techniques like k-means clustering, hierarchical clustering, locality-weighted learning, and deep learning enhances segmentation, classification accuracy, and predictive capabilities, as demonstrated in applications for chronic conditions, cancer, Parkinson's disease, COVID-19, and dermatological disorders. Despite their strengths, SOMs face challenges, including distinguishing diseases with similar clinical features, scalability limitations due to computational demands, and interpretability issues for clinicians, which restrict their adoption in evidence-based medicine. Addressing these limitations through hybrid models, advanced visualization, and adaptive architectures will further strengthen SOMs' role in driving precise, efficient, and data-driven healthcare solutions, paving the way for advancements in personalized medicine and real-time clinical applications.

The algorithm has evolved significantly over time, addressing specific challenges and demonstrating its versatility. Advancements in SOM demonstrate the algorithm's ability to handle diverse data types, including categorical, numerical, and mixed data. Recent research has also focused on addressing missing values and large-scale dataset visualization. These innovations highlight the growing relevance of SOMs in modern data analysis, where complexity and diversity of data are increasing. The flexibility of SOM-based approaches will remain crucial for tackling challenges like missing data management, integrating varied data types, and adapting to complex data structures.

IV. FUTURE DIRECTION

Future advancements in Self-Organizing Maps (SOMs) for medical applications, as outlined in the reviewed studies, focus on overcoming current limitations by integrating advanced computational techniques and enhancing adaptability. Hybrid models combining SOMs with deep learning, supervised classifiers, or Deep Reinforcement Learning could improve feature extraction, classification accuracy, and pattern recognition for diseases like cancer, Parkinson's, and COVID-19, particularly addressing issues like overlapping clinical features and imbalanced datasets. Scaling SOMs for real-time, large-scale EHR and imaging data, incorporating multimodal inputs (e.g., genetic markers, longitudinal data), and refining visualization tools will enhance interpretability and clinical utility. Additionally, ensemble learning, adaptive architectures, fuzzy logic for uncertainty handling, and cross-domain validation with larger datasets will boost robustness, scalability, and generalizability, enabling SOMs to support real-time predictive analytics, personalized medicine, and evidence-based interventions across diverse medical domains, including oncology, neurology, dermatology, and infectious disease management.

REFERENCES

[1] T. Kohonen *et al.*, "Matlab implementations and applications of the self-organizing map," *Unigrafia Oy, Helsinki, Finland*, vol. 177, 2014.

[2] U. Asan and S. Ercan, *An Introduction to Self-Organizing Maps*, pp. 299–319. 11 2012.

[3] P. Yıgıt, "Som clustering of oecd countries for covid-19 indicators and related socio-economic indicators," *Journal of Intelligent Systems: Theory and Applications*, vol. 7, no. 2, p. 95–101, 2024.

[4] S. M. Araújo, S. B. Nery, B. G. Magalhães, K. J. Almeida, and P. D. Gaspar, "Disease severity index in parkinson's disease based on self-organizing maps," *Applied Sciences*, vol. 13, no. 18, p. 10019, 2023.

[5] A. Sagir and S. Sathasivam, "The self organizing map as a tool for cluster analysis," vol. 38, pp. 44–53, 01 2016.

[6] A. Guérin, P. Chauvet, and F. Saubion, "A Survey on Recent Advances in Self-Organizing Maps." working paper or preprint, Jan. 2025.

[7] M. C. Kind and R. J. Brunner, "Somz: photometric redshift pdfs with self organizing maps and random atlas," *ArXiv*, vol. abs/1312.5753, 2013.

[8] D. Chushig-Muzo, C. Soguero-Ruiz, A. P. Engelbrecht, P. De Miguel Bohoyo, and I. Mora-Jiménez, "Data-driven visual characterization of patient health-status using electronic health records and self-organizing maps," *IEEE Access*, vol. 8, pp. 137019–137031, 2020.

[9] M. Zribi, Y. Boujelbene, I. Abdelkafi, and R. Feki, "The self-organizing maps of kohonen in the medical classification," in *2012 6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)*, pp. 852–856, 2012.

[10] Y. Cho and K. Ryu, "Predictive pattern analysis using som in medical data sets for medical treatment service," pp. 1–5, 05 2014.

[11] S. Ilbeigipour, A. Albadvi, and E. Akhondzadeh Noughabi, "Cluster-based analysis of covid-19 cases using self-organizing map neural network and k-means methods to improve medical decision-making," *Informatics in Medicine Unlocked*, vol. 32, p. 101005, 2022.

[12] K. A. Brown, I. N. Sarkar, K. M. Crowley, D. P. Aluthge, and E. S. Chen, "An unsupervised cluster analysis of post-covid-19 mental health outcomes and associated comorbidities," in *AMIA Annual Symposium Proceedings*, vol. 2022, p. 289, 2023.

[13] U. Fidan, N. Ozkan, and I. Calikusu, "Clustering and classification of dermatologic data with self organization map (som) method," in *2016 Medical Technologies National Congress (TIPTEKNO)*, pp. 1–4, 2016.

[14] A. H. Osman, H. M. Aljahdali, S. M. Altarrazi, and A. Ahmed, "Som-lwl method for identification of covid-19 on chest x-rays," *PloS one*, vol. 16, no. 2, p. e0247176, 2021.

[15] H. Bagher-Ebadian, S. L. Brown, M. M. Ghassemi, P. C. Acharya, I. J. Chetty, B. Movsas, J. R. Ewing, and K. Thind, "Probabilistic nested model selection in pharmacokinetic analysis of dce-mri data in animal model of cerebral tumor," *Scientific Reports*, vol. 15, no. 1, p. 1786, 2025.

Padmashri COLLEGE

BIT | BHM | BBA | BCA | B.TECH (FOOD)