INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTING

IJRC

International Journal of Research in Computing (IJRC)

Volume 03 Issue 01

July 2024

ISSN 2820-2139

Published by

Faculty of Computing, General Sir John Kotelawala Defence University,
Ratmalana, Sri Lanka

Tel: +94-11-2635268

E-Mail: editor@ijrcom.org & editorijrc@kdu.ac.lk
Website: http://ijrcom.org/

EDITORIAL COMMITTEE

MEMBERS OF THE EDITORIAL BOARD

EDITORIAL ASSISTANTS

MISCELLANEOUS TASKS

Proofreading

CONTENTS

Empowering the Captioning of Fashion Attributes from Asian Fashion Images

KVS Perera[1], DDA Gamini[1#]
[1]Department of Computer Science, University of Sri Jayewardenepura, Sri Lanka
[#]gamini@sjp.ac.lk

ABSTRACT Fashion image captioning, an evolving field in AI and computer vision, generates descriptive captions for fashion images. This paper addresses the prevalent bias in existing studies, which focus predominantly on Western fashion, by incorporating Asian fashion into the analysis. This paper describes developing more inclusive AI technologies for the fashion industry by bridging the gap between Western and Asian fashion in image captioning. We leverage transfer learning techniques, combining the DeepFashion dataset (primarily Western fashion) with a newly curated Asian fashion dataset. Our approach employs advanced deep learning methods for the encoder and decoder components to generate high-quality captions that capture various fashion attributes, such as style, color, and garment type, tailored specifically to Asian fashion trends. Results demonstrate the efficacy of our methods, with the model achieving accuracies of 93.63% for gender, 83.42% for article type, and 61.34% for base color on the training dataset, and 94.13%, 79.25%, and 59.71%, respectively, on the validation dataset. These findings highlight the importance of inclusivity and diversity in AI research, advancing the field of fashion image captioning.

INDEX TERMS Multi-label image captioning, deep learning, fashion image analysis, Asian fashion images, transfer learning

I. INTRODUCTION

The fashion industry is a dynamic and ever-evolving domain, with new trends and styles continually emerging. Fashion enthusiasts and industry professionals are constantly engaged in decoding these trends, understanding consumer preferences, and predicting future shifts. Traditionally, the categorization and analysis of fashion products have required extensive manual effort and expertise, a process both time-consuming and prone to human error.

The advent of computer vision and machine learning has brought transformative changes to this landscape, offering new possibilities for automating and enhancing the analysis of fashion products. These technologies enable the extraction of valuable insights from visual data, facilitating accurate categorization, classification, and prediction of various fashion attributes. This paper leverages these advancements to develop a robust system focused on gender, article type, and base color classification in fashion images.

Unlike traditional methods reliant on manual assessment or rule-based systems, our approach employs advanced machine learning algorithms to analyze and interpret fashion images. This method allows us to uncover underlying patterns, correlations and trends that may not be immediately discernible to human observers. By integrating computer vision and machine learning with fashion analysis, we aim to streamline the categorization process, providing invaluable insights to designers, retailers, and consumers. Beyond fashion analysis, our research holds potential applications in e-commerce, retail merchandising, and trend forecasting, revolutionizing the industry's approach to product analysis.

II. RELATED WORK

The domain of fashion image captioning has seen notable advancements, driven by the demand for AI solutions capable of generating accurate and descriptive captions for fashion content. Despite its growing importance, this area remains under-researched, with few studies focusing on specialized models for fashion-related content.

One significant contribution is the work by researchers who introduced "Accurate and Expressive Fashion Captioning: A Learning Framework" [1]. Their framework utilizes attribute-level semantic rewards and attribute embedding techniques to create precise and expressive descriptions for fashion items. Building on this foundation, the study integrates maximum likelihood estimation (MLE) and Reinforcement Learning (RL) to enhance caption quality further. The researchers developed the FAshion CAptioning Dataset (FACAD), a comprehensive collection of 993,000 fashion images paired with 130,000 diverse and enchanting descriptions. This dataset enables training models to effectively capture and convey fashion item attributes. Experiments on FACAD demonstrate the framework's effectiveness in generating high-quality captions, showcasing advancements in fashion captioning through innovative methodologies and robust dataset utilization. Building on this foundation, Moratelli et al. [2] proposed an approach integrating external memory retrieval with transformer-based neural networks for fashion captioning. Their method leverages transformer architectures with cross-attention mechanisms for reading and retrieving items from external textual memory, facilitated by k-nearest neighbor (kNN) searches. This design optimizes the flow of information from external sources using a novel fully attentive gate mechanism. The study achieved state-of-the-art performance on the FACAD fashion captioning dataset, demonstrating its ability to generate detailed and contextually rich descriptions of fashion articles. This approach highlights the effectiveness of incorporating external textual memory to enhance the quality and informativeness of image captions in fashion AI applications.

[3] presents an advanced approach to fashion synthesis using generative adversarial networks (GANs). The research focuses on generating new clothing designs on a wearer while preserving the wearer's body structure and pose, guided by a descriptive sentence about the desired outfit. To achieve this, the generative process is divided into two conditional stages. In the first stage, a semantic segmentation map is generated with a spatial constraint that respects the wearer's pose. This map serves as a latent spatial arrangement guiding the synthesis process. The second stage employs a novel compositional mapping layer within the GAN framework to render the final image with precise regions and textures based on the generated segmentation map. The researchers extended the DeepFashion dataset by annotating 79,000 images with descriptive sentences, enabling training and evaluation of their approach. Quantitative metrics and qualitative assessments demonstrate the effectiveness of the method in producing realistic and structurally coherent fashion images aligned with textual descriptions, highlighting advancements in generative fashion modeling.

A novel approach to clothing style detection and retrieval through fine-grained learning is introduced in [4]. They address challenges like clothing item variability and deformability by creating a detailed attribute vocabulary from human annotations on a specialized dataset. This vocabulary trains a visual recognition system capable of identifying complex stylistic attributes beyond basic features like color and pattern. The study validates its effectiveness with benchmark tests on the Women's Fashion Coat Dataset, demonstrating superior recognition and differentiation of nuanced stylistic elements. Furthermore, it explores the application of attribute-based multimedia retrieval in mobile interfaces, enhancing user experience through detailed image annotations and precise clothing item searches. This integration of human-derived attribute vocabularies with advanced visual recognition techniques provides valuable insights for enhancing the granularity and accuracy of clothing style detection and retrieval systems.

[5] provides large-scale clothes dataset with over 800,000 images annotated comprehensively with attributes, landmarks, and cross-scenario correspondences. They propose FashionNet, a deep learning model optimized iteratively to predict clothing attributes and landmarks simultaneously. FashionNet utilizes predicted landmarks to enhance feature learning through pooling or gating mechanisms. This work establishes DeepFashion as the largest annotated clothes dataset, facilitating advancements in clothes recognition and retrieval algorithms, with defined benchmark datasets and evaluation protocols.

An automated system [6] is designed to generate detailed semantic descriptions of clothing from images. The system extracts low-level features in a pose-adaptive manner, ensuring that attributes are accurately identified regardless of the position of the person in the image. By combining these features, the system trains attribute classifiers to recognize various clothing characteristics. It further improves attribute predictions by using a Conditional Random Field (CRF) to model the dependencies between attributes, which allows for more accurate and contextually relevant descriptions. The system's performance was validated on a challenging clothing attribute dataset, demonstrating its ability to generate precise and meaningful clothing descriptions. This research provides valuable insights into the use of pose-adaptive feature extraction and CRF for enhancing the accuracy of semantic attribute recognition in clothing.

In their study [7], an advanced approach to image captioning that goes beyond the traditional encoder-decoder models, which typically only recognize objects and their relationships in a given image is explored. The authors identify a critical gap in existing methodologies, particularly when applied to fashion images, where it's essential not only to describe items but also to capture intricate details such as texture, fabric, shape, and style. To address this gap, the researchers propose an innovative model that integrates an attention mechanism within the conventional encoder-decoder framework, enabling it to generate more comprehensive and nuanced descriptions of fashion items. This model leverages spatial attention to dynamically adjust the sentence generation context across multiple layers of feature maps, ensuring that both the items and their detailed attributes are effectively covered in the generated captions. The efficacy of this approach is validated through experiments on the Fashion-Gen dataset, a leading dataset in the field of fashion image analysis. The results demonstrate significant improvements, with the model achieving impressive scores on key metrics like CIDEr, ROUGE-L, and BLEU-4, surpassing baseline methods on the same dataset. This research is particularly relevant to our study as it highlights the importance of incorporating detailed attribute recognition and spatial attention mechanisms in fashion image captioning, offering a robust framework that enhances the descriptive quality and relevance of generated captions.

[8] addresses the challenge of maintaining captioning quality when input data diverges from the training distribution, crucial in fashion's nuanced garment descriptions. By employing a pre-training strategy with noise generation, the study enhances system generalization. It integrates GPT-2, Vision Transformer, and BERT, showing competitive results even with limited fine-tuning. Colombo emphasizes user-centric evaluation via a user study, confirming improved captioning quality. This work suggests transfer learning's efficacy in adapting captioning systems to varied data, vital for reliable outputs across applications.

Recent research has increasingly emphasized the importance of inclusivity and diversity in artificial intelligence (AI) technologies, addressing the under-representation of various cultural and regional fashion styles in existing datasets and models. Hacheme and Sayouti [9], applying the "Show and Tell" model introduced by Vinyals et al. in 2015 [10], highlighted the lack of representation of African fashion styles in current datasets. In response, a pioneering study introduced the InFashAIv1 dataset, comprising nearly 16,000 African fashion item images with titles, prices, and general descriptions. This dataset, alongside the well-known DeepFashion dataset, serves as a critical resource for training AI models in fashion image captioning. Captions are generated using the Show and Tell model, leveraging a CNN encoder and RNN decoder architecture. The research demonstrates that joint training on both datasets significantly enhances caption quality, particularly for African style fashion images, demonstrating effective transfer learning from Western style data. The release of the InFashAIv1 dataset on GitHub aims to foster research that promotes diversity and inclusion in fashion AI applications.

Recent advancements in fashion image captioning have explored various techniques to enhance the accuracy and richness of descriptions for fashion items, employing technologies such as deep learning models, external memory retrieval, and semantic segmentation. Methodologies encompass attribute-level semantic rewards, reinforcement learning, attention mechanisms, and generative adversarial networks (GANs). Table 1 presents a summary of key findings and contributions from recent research in fashion image captioning. Current fashion image captioning research predominantly focuses on Western fashion, resulting in a significant gap in the representation and understanding of non-Western styles, particularly Asian fashion. Existing datasets and models lack the diversity needed to accurately capture and describe the unique attributes of these underrepresented fashion styles. This limitation leads to poor performance and limited applicability of existing captioning systems for non-Western fashion. So, as the research gap highlights, there is a need for diverse datasets that include Asian fashion styles and specialized models that can effectively address this gap. Our research aims to fill this void by curating a comprehensive dataset of Asian fashion and developing models that improve captioning accuracy and inclusivity for global fashion audiences.

Table 1. Summary of related works

Reference	Technology Used	Key Findings
[1]	Attribute-level semantic rewards, RL, MLE	Enhanced caption quality for fashion images
[2]	Transformer-based NNs, External memory retrieval, Cross-attention mechanisms	Achieved good performance on FACAD dataset, generating detailed and context-rich captions
[3]	GANs, Semantic segmentation	Produced realistic and structurally coherent fashion images, preserving body structure and pose
[4], [5]	Visual recognition system, Human annotations	Highlighted the capability of recognizing and retrieving fine-grained clothing styles with a visual recognition system. Demonstrated the system's capability to distinguish intricate stylistic details in clothing items
[6], [9], [10]	CNN, RNN, Semantic attributes, Neural image captioning	Generated detailed descriptions of clothing items. Enhanced model performance for under-represented fashion styles. Highlighted advancements in neural image caption generation
[7]	Attention mechanisms, Deep learning models	Improved the relevance and richness of fashion image captions
[8]	Transfer learning, NNs	Revealed the effectiveness of transfer learning for improving captioning models

III. METHODOLOGY

This section details the methodology adopted for developing our fashion image captioning model with a focus on Asian fashion styles. The process involves several key steps: dataset integration and preprocessing, creation of a multi-label model, and the subsequent training and testing phases. Initially, a specialized Asian fashion dataset is curated and integrated with an existing benchmark dataset to form a comprehensive repository. Following this, a multi-label classification model is developed, leveraging advanced neural network architectures and tailored loss functions. The model is then trained and validated using carefully selected hyperparameters, with performance metrics calculated at each stage. Finally, the model is tested on unseen data to evaluate its accuracy and robustness. Figure 1 provides a concise summary of the methodological steps and the technologies employed in this research.

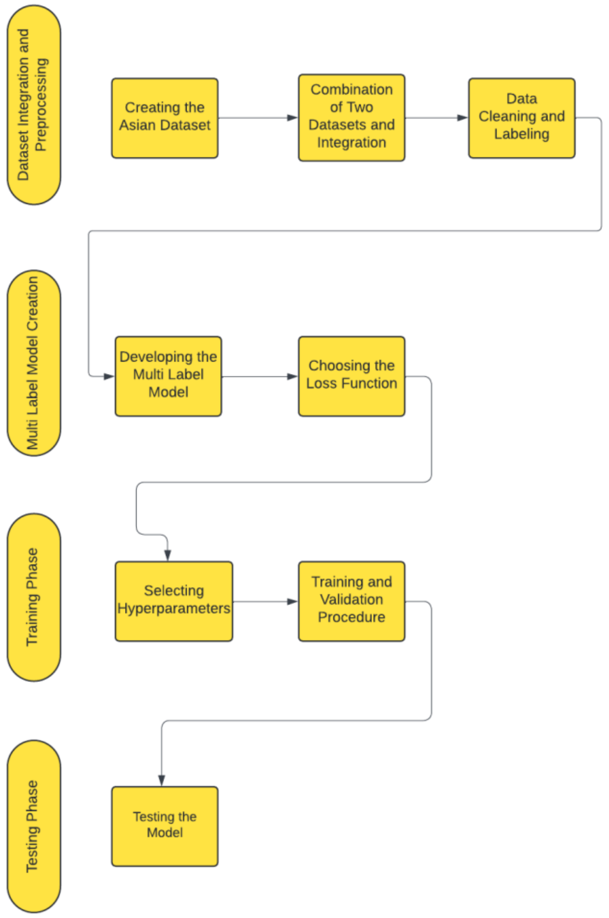

Dataset Integration and Preprocessing
Creating the Asian Dataset
Combination of Two Datasets and Integration
Data Cleaning and Labeling
Multi Label Model Creation
Developing the Multi Label Model
Choosing the Loss Function
Training Phase
Selecting Hyperparameters
Training and Validation Procedure
Testing Phase
Testing the Model

Figure 1. Methodological steps

A. *Dataset Integration and Preprocessing*

1) Asian Dataset Creation: In this research, a specialized Asian-focused dataset comprising 2,500 meticulously curated fashion images is developed, sourced primarily from platforms like Pinterest. Careful selection criteria ensure the dataset's breadth, encompassing traditional, contemporary, casual, and formal Asian fashion styles. Preprocessing steps, including image resizing and quality enhancement, prepare the dataset for annotation and model training. Special emphasis is placed on capturing a diverse palette of colors prevalent in Asian fashion, ensuring richness and inclusivity. Additionally, the dataset covers a wide array of clothing types commonly found in Asian fashion, guaranteeing versatility and accuracy in recognizing and describing various fashion items and styles.

2) Dataset Combination and Integration: In the process of dataset combination and integration, merging the curated Asian dataset with the benchmark Fashion dataset is a critical step aimed at enhancing the model's robustness and generalization capabilities. This fusion leverages the diverse attributes of both datasets, creating a more comprehensive training environment for the multi-label image classification model. Before merging, meticulous attention is paid to ensuring consistency and compatibility between the two datasets. This involves aligning column names and preserving their order across both data frames to facilitate seamless integration and avoid discrepancies during concatenation. Attribute names such as 'gender,' 'articleType,' and 'baseColour' are standardized to ensure uniformity and coherence, maintaining data integrity. Following the alignment process, the curated Asian dataset is merged with the benchmark Fashion dataset, resulting in a unified dataset that encompasses a broader spectrum of fashion styles, trends, and attributes. This combined dataset serves as a comprehensive repository, enabling the model to learn from a diverse range of fashion data and improve its ability to recognize and describe various fashion items accurately.

3) Data Cleaning and Labeling: In the dataset cleaning and labeling phase, ensuring the quality, consistency, and reliability of the dataset is crucial. This involves removing inconsistencies, missing values, and irrelevant data entries, and annotating the dataset with detailed attributes for multi-label classification and captioning tasks. A thorough cleaning process is conducted to eliminate discrepancies, duplicates, and missing values to ensure completeness and reliability. Subsequently, label dictionaries are created and mapped for categorical attributes, transforming textual labels into a numerical format for efficient model training and inference. This includes generating unique category mappings for gender, articleType, and baseColour attributes, with special handling for NaN values in baseColour.

B. *Multi Label Model Creation*

1) MultiHeadResNet50: In the multi-caption model development, the cornerstone is the MultiHeadResNet50 architecture, leveraging the ResNet50 backbone pretrained on the ImageNet dataset for its feature extraction capabilities. This model features three distinct fully connected layers, each serving as a classification head for specific fashion attributes. The gender classification head outputs predictions across five gender categories, including male, female, unisex, kids, and others, enabling effective categorization of gender representation in fashion images. The article type classification head predicts across 142 categories, covering a wide range of fashion items from shirts and dresses to specialized articles, providing a comprehensive understanding of fashion ensembles. The base color classification head outputs predictions across 47 color categories, capturing a diverse palette from traditional colors like black and white to nuanced shades like turquoise blue and fluorescent green, reflecting the richness and diversity of fashion colors.

2) Loss Function: Complementing the model architecture, a bespoke loss function is formulated to efficiently optimize the multi-label classification task. This function employs the Cross-Entropy Loss mechanism, a widely adopted approach for classification tasks, to compute the loss for each classification head. Mathematically, the loss function is defined as:

$$Loss_{total} = \frac{Loss_{gender} + Loss_{articleType} + Loss_{baseColour}}{3}$$

where:
$Loss_{gender}$, $Loss_{articleType}$ & $Loss_{baseColour}$ represent the Cross-Entropy Losses computed for gender, articleType and baseColour predictions, respectively.

C. Training Phase

1) Selecting hyperparameters: The training phase optimizes the MultiHeadResNet50 model using key hyperparameters and configurations. The learning rate is set to 0.001, determining the step size during optimization and affecting convergence and stability. The Adam optimizer is employed for its adaptive learning rate capabilities, efficiently handling large-scale datasets. A batch size of 32 is used, specifying the number of samples processed before updating the model's weights, which impacts memory utilization and learning stability. The model undergoes 40 training epochs, indicating the number of times the entire dataset is passed forward and backward through the neural network.

2) Training and Validation Procedure: The training and validation procedures are divided into two functions: train and validate. These functions compute loss and accuracy metrics for both datasets, facilitating model evaluation. Loss is calculated using the multi-head loss function, which combines Cross-Entropy losses for gender, article type, and base color classifications, resulting in an averaged overall loss. Accuracy is determined by the percentage of correct predictions across all categories, calculated using the formula:

$$Accuracy = \frac{Correct\ Predictions}{Total\ Predictions} \times 100$$

During each epoch, the model undergoes forward and backward passes on both training and validation datasets, with metrics such as average loss and accuracy tracked. Upon completion, the model, optimizer, and loss criterion are saved, and a visualization of training and validation loss evolution is generated for further analysis.

D. Testing Phase

The testing phase involves evaluating the model's performance on unseen data. This begins with initializing necessary configurations and loading the model. Proper environment configuration ensures efficient image processing to optimize memory usage, while device specification determines the computation device for inference. The MultiHeadResNet50 model is instantiated and loaded with trained weights from a saved checkpoint. Image processing and model inference form the core of the testing pipeline. Input images undergo resizing and normalization to match model requirements before being fed into the model. The model generates outputs corresponding to gender, article type, and base color classifications. Predicted indices with the highest label scores are extracted, representing predicted classes for each category.

Post-processing and visualization are the final stages of the testing phase. Predicted indices are mapped back to their respective labels using pre-loaded dictionaries, providing human-readable labels for each category. These labels are then annotated onto the original image using OpenCV, offering a visual representation of the model's predictions. Annotated images are saved to disk for future reference and analysis.

IV. RESULTS

A. Training and Validation Loss

Over the course of 60 epochs, the training loss consistently decreases, signaling the model's effective learning from the training data. This decreasing trend indicates that the model is converging towards an optimal solution, improving its ability to minimize the difference between predicted and actual outputs. The validation loss generally decreased during the initial training epochs, indicating that the model was learning from the data. However, around epoch 20, the decrease plateaued, showing minimal fluctuations until epoch 60. This stabilization suggests that the model might not benefit significantly from further training beyond this point, potentially hinting at the onset of overfitting. Figure 2 provides visual insight into the phenomenon of the training and validation loss.

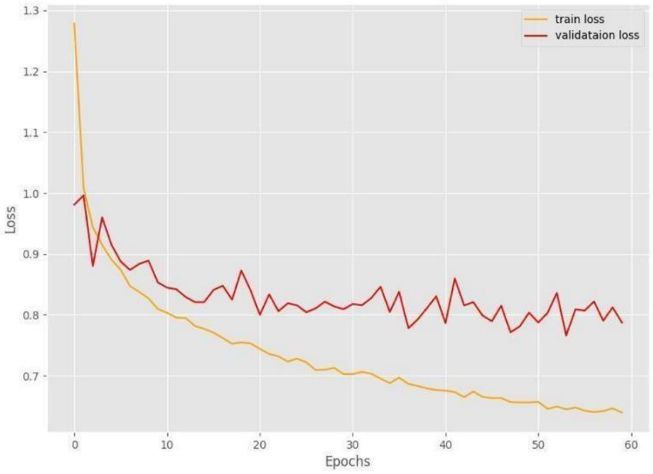


Figure 2. Training and validation loss learning curve

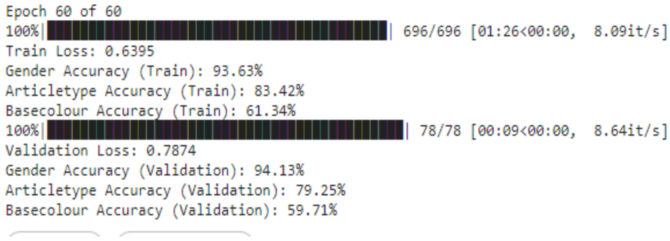
B. Training and Validation Accuracy

The accuracy of the model is calculated for both the training and validation datasets across 60 epochs. This metric is computed using the formula:

$$Accuracy = \frac{Number\ of\ Correct\ Predictions\ for\ the\ particular\ label}{Total\ Number\ of\ Predictions\ for\ the\ particular\ label}$$

At the conclusion of the 60 epochs, the model exhibited impressive accuracy rates. For the training dataset, the accuracies were 93.63% for Gender, 83.42% for Article Type, and 61.34% for Base Color. Similarly, for the validation dataset, the accuracies were 94.13% for Gender, 79.25% for Article Type, and 59.71% for Base Color. Gender and Article Type labels achieved notably high accuracies, indicating the model's strong ability to classify these categories effectively.

Figure 3 depicts the accuracy for each label category across both the training and validation datasets at the 60th epoch, providing insights into the model's learning progress and its ability to correctly classify each label throughout the training process.



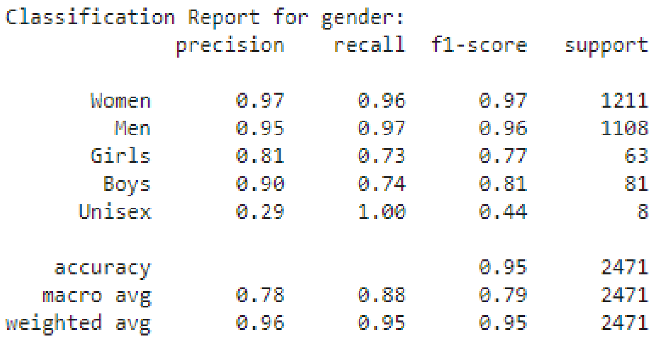
```
Epoch 60 of 60
100%|██████████████████████████████████████████████| 696/696 [01:26<00:00,  8.09it/s]
Train Loss: 0.6395
Gender Accuracy (Train): 93.63%
Articletype Accuracy (Train): 83.42%
Basecolour Accuracy (Train): 61.34%
100%|████████████████████████████████████████████████| 78/78 [00:09<00:00,  8.64it/s]
Validation Loss: 0.7874
Gender Accuracy (Validation): 94.13%
Articletype Accuracy (Validation): 79.25%
Basecolour Accuracy (Validation): 59.71%
```

Figure 3. Training and validation accuracies

C. *Classification Reports for Validation Dataset*

Classification reports were generated for the validation dataset to evaluate the model's performance across different label categories comprehensively. These reports offer detailed insights into the model's precision, recall, and F1-score for each label, enabling a thorough examination of its performance on a category-by-category basis. Tables 2, 3, and 4 present comprehensive evaluations of the model's performance across various classification tasks, providing a granular understanding of its capabilities within each category.

Table 2. Gender classification report

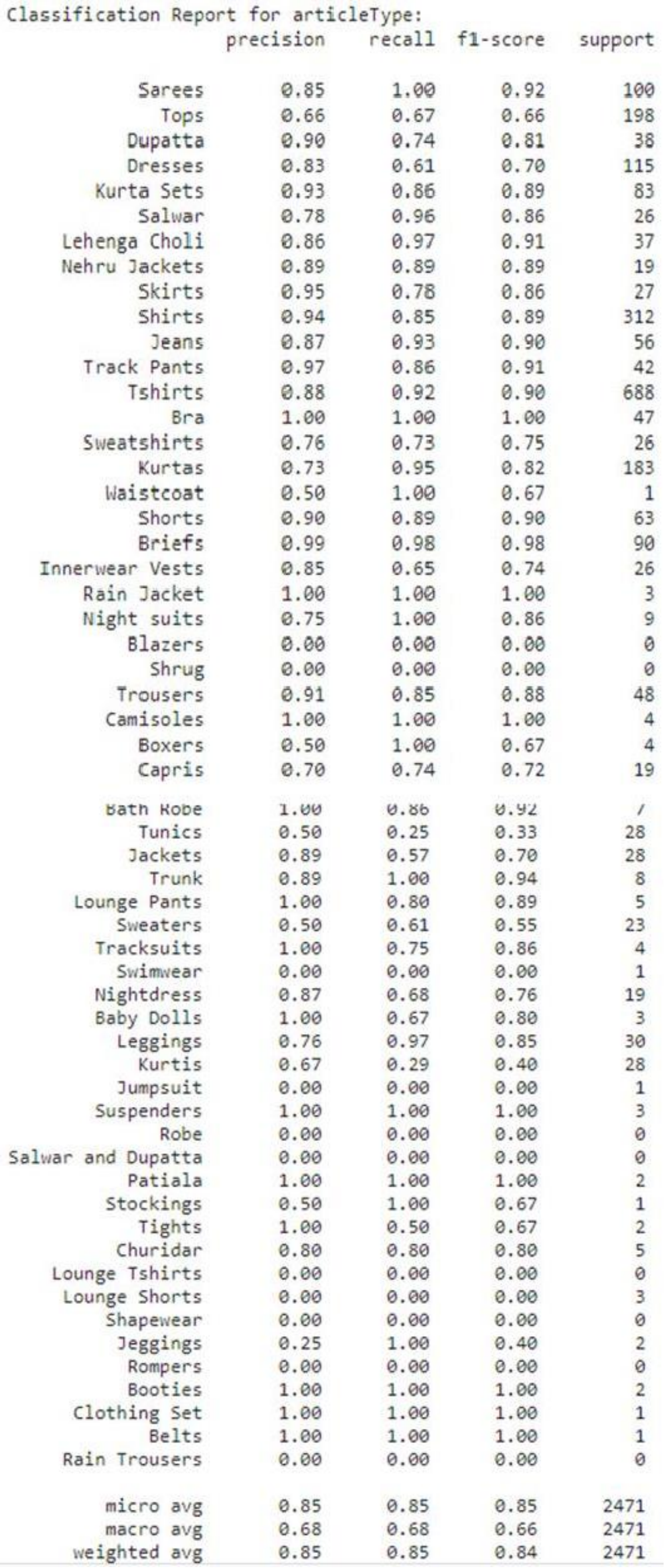


```
Classification Report for gender:
              precision    recall  f1-score   support

       Women       0.97      0.96      0.97      1211
         Men       0.95      0.97      0.96      1108
       Girls       0.81      0.73      0.77        63
        Boys       0.90      0.74      0.81        81
      Unisex       0.29      1.00      0.44         8

    accuracy                           0.95      2471
   macro avg       0.78      0.88      0.79      2471
weighted avg       0.96      0.95      0.95      2471
```

D. *Results on Unseen Images*

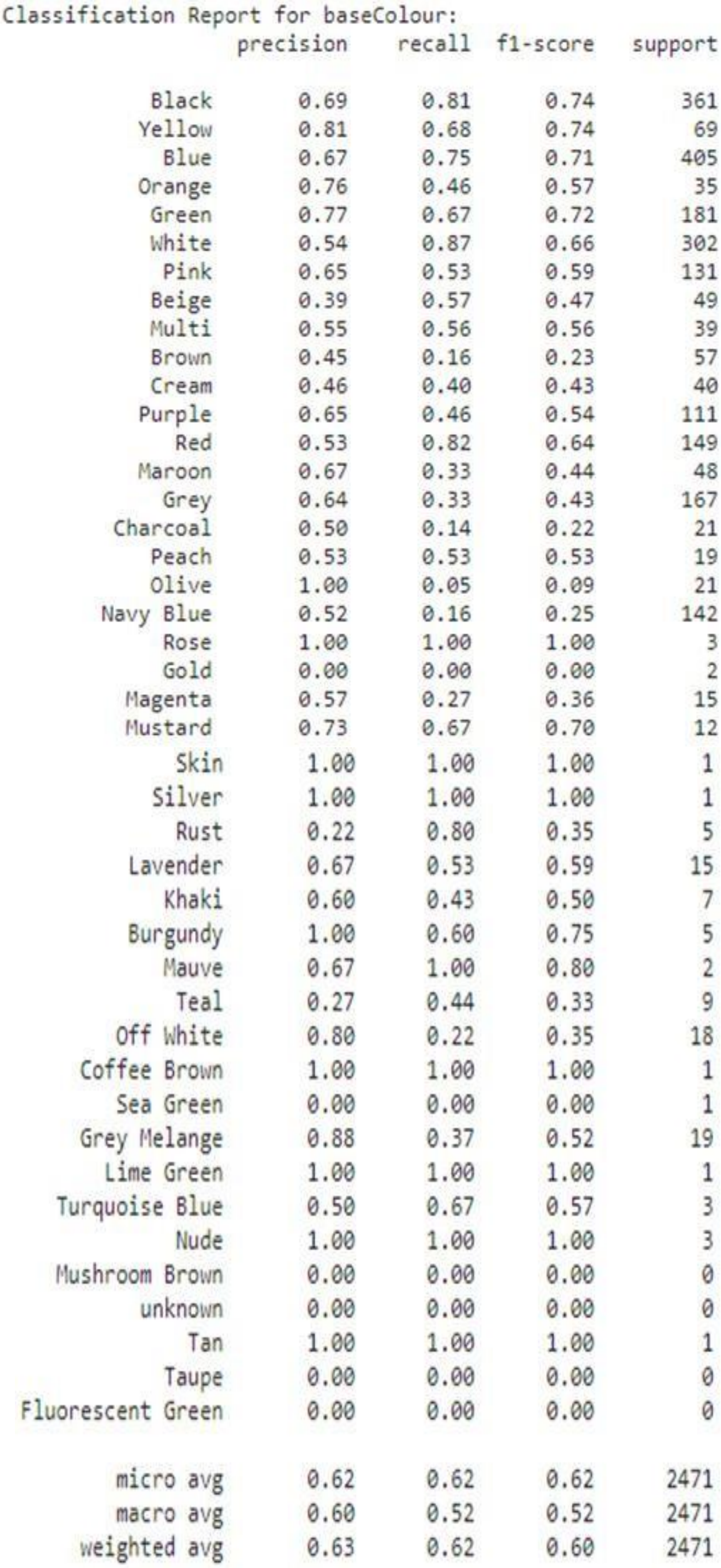
In this section, we present the results of our model's predictions on a diverse set of fashion images, as shown in Figures 4 through 9. Each image was processed by our multi-label classification model, which predicts three key attributes: gender, base color, and article type. The predicted labels for each image were then annotated directly onto the original images using OpenCV, providing a clear visual representation of the model's output.

Table 3. Article type classification report



```
Classification Report for articleType:
                    precision    recall  f1-score   support

            Sarees       0.85      1.00      0.92       100
              Tops       0.66      0.67      0.66       198
           Dupatta       0.90      0.74      0.81        38
           Dresses       0.83      0.61      0.70       115
        Kurta Sets       0.93      0.86      0.89        83
            Salwar       0.78      0.96      0.86        26
      Lehenga Choli       0.86      0.97      0.91        37
      Nehru Jackets       0.89      0.89      0.89        19
            Skirts       0.95      0.78      0.86        27
            Shirts       0.94      0.85      0.89       312
             Jeans       0.87      0.93      0.90        56
       Track Pants       0.97      0.86      0.91        42
           Tshirts       0.88      0.92      0.90       688
               Bra       1.00      1.00      1.00        47
       Sweatshirts       0.76      0.73      0.75        26
            Kurtas       0.73      0.95      0.82       183
         Waistcoat       0.50      1.00      0.67         1
            Shorts       0.90      0.89      0.90        63
            Briefs       0.99      0.98      0.98        90
    Innerwear Vests       0.85      0.65      0.74        26
       Rain Jacket       1.00      1.00      1.00         3
       Night suits       0.75      1.00      0.86         9
           Blazers       0.00      0.00      0.00         0
             Shrug       0.00      0.00      0.00         0
          Trousers       0.91      0.85      0.88        48
         Camisoles       1.00      1.00      1.00         4
            Boxers       0.50      1.00      0.67         4
            Capris       0.70      0.74      0.72        19

         Bath Robe       1.00      0.86      0.92         7
            Tunics       0.50      0.25      0.33        28
           Jackets       0.89      0.57      0.70        28
             Trunk       0.89      1.00      0.94         8
      Lounge Pants       1.00      0.80      0.89         5
          Sweaters       0.50      0.61      0.55        23
        Tracksuits       1.00      0.75      0.86         4
          Swimwear       0.00      0.00      0.00         1
        Nightdress       0.87      0.68      0.76        19
        Baby Dolls       1.00      0.67      0.80         3
          Leggings       0.76      0.97      0.85        30
            Kurtis       0.67      0.29      0.40        28
          Jumpsuit       0.00      0.00      0.00         1
        Suspenders       1.00      1.00      1.00         3
              Robe       0.00      0.00      0.00         0
Salwar and Dupatta       0.00      0.00      0.00         0
           Patiala       1.00      1.00      1.00         2
         Stockings       0.50      1.00      0.67         1
            Tights       1.00      0.50      0.67         2
          Churidar       0.80      0.80      0.80         5
     Lounge Tshirts       0.00      0.00      0.00         0
      Lounge Shorts       0.00      0.00      0.00         3
         Shapewear       0.00      0.00      0.00         0
          Jeggings       0.25      1.00      0.40         2
           Rompers       0.00      0.00      0.00         0
           Booties       1.00      1.00      1.00         2
      Clothing Set       1.00      1.00      1.00         1
             Belts       1.00      1.00      1.00         1
      Rain Trousers       0.00      0.00      0.00         0

         micro avg       0.85      0.85      0.85      2471
         macro avg       0.68      0.68      0.66      2471
      weighted avg       0.85      0.85      0.84      2471
```

Table 4. Base color classification report



```
Classification Report for baseColour:
                  precision    recall  f1-score   support

           Black       0.69      0.81      0.74       361
          Yellow       0.81      0.68      0.74        69
            Blue       0.67      0.75      0.71       405
          Orange       0.76      0.46      0.57        35
           Green       0.77      0.67      0.72       181
           White       0.54      0.87      0.66       302
            Pink       0.65      0.53      0.59       131
           Beige       0.39      0.57      0.47        49
           Multi       0.55      0.56      0.56        39
           Brown       0.45      0.16      0.23        57
           Cream       0.46      0.40      0.43        40
          Purple       0.65      0.46      0.54       111
             Red       0.53      0.82      0.64       149
          Maroon       0.67      0.33      0.44        48
            Grey       0.64      0.33      0.43       167
        Charcoal       0.50      0.14      0.22        21
           Peach       0.53      0.53      0.53        19
           Olive       1.00      0.05      0.09        21
       Navy Blue       0.52      0.16      0.25       142
            Rose       1.00      1.00      1.00         3
            Gold       0.00      0.00      0.00         2
         Magenta       0.57      0.27      0.36        15
         Mustard       0.73      0.67      0.70        12
            Skin       1.00      1.00      1.00         1
          Silver       1.00      1.00      1.00         1
            Rust       0.22      0.80      0.35         5
        Lavender       0.67      0.53      0.59        15
           Khaki       0.60      0.43      0.50         7
        Burgundy       1.00      0.60      0.75         5
           Mauve       0.67      1.00      0.80         2
            Teal       0.27      0.44      0.33         9
       Off White       0.80      0.22      0.35        18
    Coffee Brown       1.00      1.00      1.00         1
       Sea Green       0.00      0.00      0.00         1
    Grey Melange       0.88      0.37      0.52        19
      Lime Green       1.00      1.00      1.00         1
  Turquoise Blue       0.50      0.67      0.57         3
            Nude       1.00      1.00      1.00         3
  Mushroom Brown       0.00      0.00      0.00         0
         unknown       0.00      0.00      0.00         0
             Tan       1.00      1.00      1.00         1
           Taupe       0.00      0.00      0.00         0
Fluorescent Green       0.00      0.00      0.00         0

       micro avg       0.62      0.62      0.62      2471
       macro avg       0.60      0.52      0.52      2471
    weighted avg       0.63      0.62      0.60      2471
```

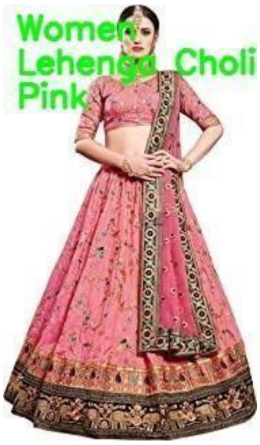


Figure 4. Image of a woman wearing a pink color lehenga choli

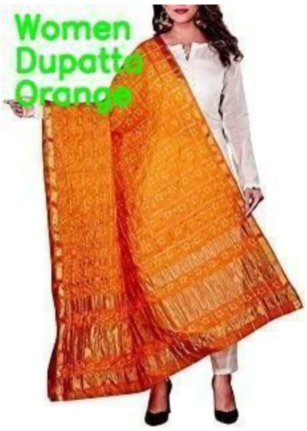


Figure 5. Image of a woman wearing an orange color dupatta

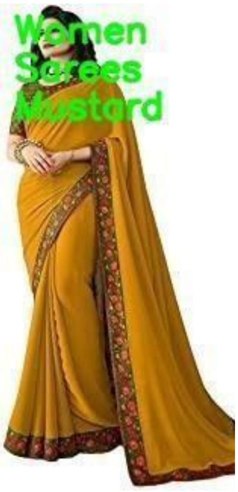


Figure 6. Image of a woman wearing a mustard color saree

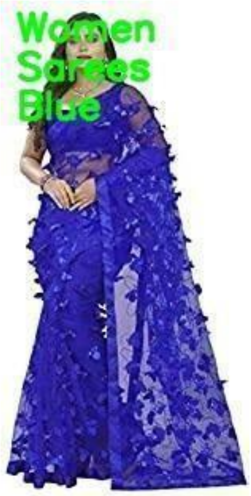


Figure 7. Image of a woman wearing a blue color saree

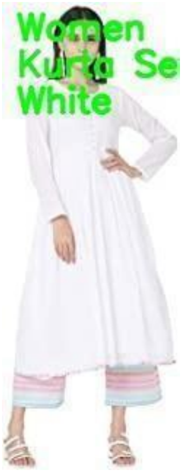


Figure 8. Image of a woman wearing a white color kurta set

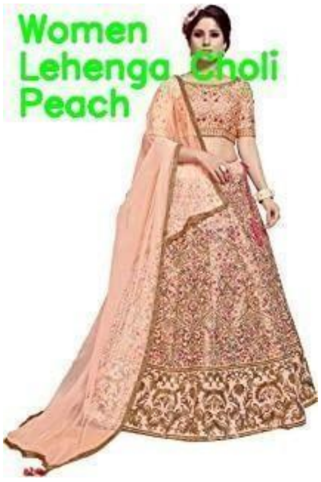


Figure 9. Image of a woman wearing a peach color lehenga choli

V. DISCUSSION AND CONCLUSION

This paper focused on developing and evaluating a Multi-Head ResNet50 model tailored for multi-label classification in the Asian fashion domain, specifically predicting gender, article type, and base color labels. The model demonstrated commendable performance in capturing the nuanced characteristics of fashion items prevalent in Asian markets, achieving high accuracy for gender and article type predictions. Specifically, our model attained an epoch-wise accuracy of 93.63% for gender and 83.42% for article type on the training dataset, which is particularly noteworthy considering the complexity and variety of the fashion items in the dataset. Validation results were similarly impressive, with accuracies of 94.13% for gender and 79.25% for article type, indicating strong generalization capabilities and minimal signs of overfitting.

However, the accuracy for base color on the training dataset was 61.34%, and 59.71% on the validation dataset, which is relatively low compared to other attributes. This discrepancy highlights a significant challenge in predicting base color accurately. The lower accuracy for base color can be attributed to several factors, including the high variability and subtlety of color shades in fashion items, especially in the context of Asian fashion where intricate and nuanced color palettes are common. Additionally, the dataset's inherent imbalance, with fewer instances of certain base colors, may have contributed to the lower performance in this category.

To address the issue of lower accuracy for base color, several strategies could be explored. First, implementing techniques to mitigate class imbalance, such as oversampling underrepresented classes, undersampling overrepresented ones, or employing weighted loss functions, could help the model learn more effectively from the available data. Additionally, enhancing the feature extraction process by incorporating more sophisticated color recognition techniques or leveraging additional data sources for more comprehensive color annotation might improve the model's accuracy. Incorporating advanced augmentation techniques specifically designed to highlight color variations could also prove beneficial.

In conclusion, we introduced and implemented a Multi-Head ResNet50 model tailored for multi-label classification within the fashion domain, specifically targeting gender, article type, and base color. The model's robust performance in predicting gender and article type underscores its potential in the fashion industry for categorizing diverse fashion items with high accuracy. However, the relatively lower accuracy for base color suggests areas for further improvement. Addressing these challenges through targeted strategies such as handling class imbalance and refining feature extraction techniques could enhance the model's overall performance and applicability. Future work will focus on these areas to develop more inclusive and accurate fashion image captioning systems, ensuring better representation and recognition of diverse fashion styles.

VI. FUTURE RESEARCH DIRECTIONS

This study highlights several avenues for future research to enhance model performance and applicability. Firstly, refining techniques for base color prediction and expanding the dataset with more diverse color samples could address the current lower accuracy in this area. Secondly, implementing methods to handle class imbalance, such as oversampling, undersampling, or weighted loss functions, can improve accuracy across all attributes. Additionally, incorporating multimodal approaches by integrating image data with textual descriptions and contextual information can enhance the richness and precision of fashion item captions. Exploring transfer learning and domain adaptation can help the model generalize better across various fashion styles, extending its applicability beyond Asian fashion. Expanding research to include underrepresented fashion categories, such as African or Latin American styles, will promote diversity and inclusivity in fashion image captioning systems. Lastly, addressing ethical implications by ensuring models are trained on diverse, representative datasets is crucial for developing fair and equitable AI systems for a global audience.

REFERENCES

[1] X. Yang, H. Zhang, D. Jin, Y. Liu, C.-H. Wu, J. Tan, D. Xie, J. Wang, and X. Wang, "Fashion captioning: Towards generating accurate descriptions with semantic rewards," in Proceedings of the 2020 European Conference on Computer Vision (ECCV), Aug. 2020,
pp. 1-17.

[2] N. Moratelli, M. Barraco, D. Morelli, M. Cornia, L. Baraldi, and
R. Cucchiara, "Fashion-Oriented Image Captioning with External Knowledge Retrieval and Fully Attentive Gates," Sensors (Basel), vol. 23, no. 3, p. 1286, 2023, doi: 10.3390/s23031286.

[3] S. Zhu, S. Fidler, R. Urtasun, D. Lin, and C. C. Loy, "Be Your Own Prada: Fashion Synthesis with Structural Coherence," in Proceedings of the International Conference on Computer Vision (ICCV), Oct. 2017.

[4] W. Di, C. Wah, A. Bhardwaj, R. Piramuthu, and N. Sundaresan, "Style Finder: Fine-Grained Clothing Style Detection and Retrieval," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 11, pp. 2670-2683, November 2013, doi: 10.1109/TPAMI.2013.78.

[5] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich Annotations," International Journal of Computer Vision, vol. 124, no. 1, pp. 74-95, November 2016, doi: 10.1007/s11263-016-0932-3.

[6] H. Chen, A. Gallagher, and B. Girod, "Describing Clothing by Semantic Attributes," in Proceedings of the ACM Multimedia Conference, October 2019.

[7] B. T. Nguyen, O. Prakash, and A. H. Vo, "Attention Mechanism for Fashion Image Captioning," IEEE Transactions on Multimedia, vol. 23, no. 5, pp. 1567-1580, May 2021, doi: 10.1109/TMM.2021.3069988.

[8] X. Colombo, "Transfer Learning Analysis of Fashion Image Captioning Systems," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2021.

[9] G. Hacheme and N. Sayouti, "Neural Fashion Image Captioning: Accounting for Data Diversity," arXiv preprint arXiv:2106.12154v2, Jun. 2021.

[10] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015, pp. 3156–3164.

Artificial Neural Network Based Grey Exponential Smoothing Approach for Forecasting Electricity Demand in Sri Lanka

D. M. K. N. Seneviratna[1#]

[1]Department of Interdisciplinary Studies, Faculty of Engineering, University of Ruhuna, Galle, Sri Lanka

[#]seneviratna@is.ruh.ac.lk

ABSTRACT The electricity supply of the country has greatly impacted the economy and the nation's standard of living; an accurate forecast of electricity demand is essential for any country to enhance industrialization, farming, and residential requirements and to make proper investment decisions. Therefore, most countries have been allocating and spending significant amounts from their annual budgets on power generation. This current study proposes an Artificial Neural Network (ANN) based approach to forecast electricity demands in Sri Lanka. For model validation, GM (1, 1), Moving Average, and Grey Exponential Smoothing models were used based on electricity gross generation data from 2000 to 2022. The empirical results suggest that the hybrid Grey Exponential Smoothing model is highly accurate under the non-stationary framework.

INDEX TERMS: Electricity Demand, ARIMA, ANN Algorithms and GM (1, 1).

I. INTRODUCTION

Electricity is one of the main essentials for various sectors of a country, such as residential, commercial, industrial, public, agricultural, and transportation to maintain services under high-quality standards [1]. Rapidly growing population and their different needs, extensive urbanization, and industrialization have directly contributed to the increase in the country's electricity demands [2]. Especially, the industrial sector has a direct effect on the country's economic development, job creation, technological advancement, increased productivity, higher incomes, and improved living standards for human beings [3].

Due to the high cost of charcoal, inadequate irregular power supply, unreliable power quality, and lack of energy efficiency, Industrial sectors in Sri Lanka have been facing some unexpected challenges since 2010[4]. For example, during year 2021-2022 period, there were prolonged power cuts across the country to manage the limited power supply [4,5]. This study aims to forecast long-term monthly electricity demand in industrial sectors in Sri Lanka using Grey System Theory (GST) and Machine Learning Approaches (ML).

Based on the literature, various researchers have conducted a range of research endeavours aimed at forecasting electricity demands; Among them, Geometric Brownian Motion [6,7], robust statistical models[8], Machine Learning (ML) [9,10], and Artificial Neural Networks (ANN) approach are significant.

Based on the Univariate Time Series Analysis and Forecasting methods such as Autoregressive Integrated Moving Average (ARIMA) approach [11, 12], Rathnayaka et.al conducted a study to forecast the electricity production and consumption in Sri Lanka [9]. By using different ANN Algorithms, Wang et.al carried out a study to estimate Short-Term Electric power demand [13]. In 2022 Hus et.al carried out a study to forecast electricity by using a self-tuned ANN-based adaptable predictor [14].

Based on the fuzzy logic approach, Xiaohuaa et.al carried out a study to predict energy demands in China[15].In the same time, based on the Grey Markov model with a rolling mechanism, Jinjin et.al carried out a study to forecast energy resources such as natural gas consumption , crude petroleum, electricity and coal in China [16]. Wang et al have done study to forecast electricity consumption of Jiangsu province, china using Dynamic grey models [17]. Liu et.al developed a neural network hybrid approach to solve energy consumption in China [18]. Yao et.al exposed a new automated monitoring control system to forecast short–term electricity power demands in Taiwan [19]. In the modern literature, Recursive Neural Networks (RNNs) based Long short-term memory (LSTM) networks with hybrid algorithms have been widely applied under three different categories to forecast power demand [20].

The current study mainly focused on forecasting Electricity demand in Industrial Sectors in Sri Lanka. The remainder of the paper is structured as follows: Section II provides a concise summary of the methodologies. Section III examines and contrasts the results of Sri Lankan electricity consumption. Finally, Section IV concludes with a discussion and outlines future research directions.

II. METHODOLOGY

In the modern world, the ability to solve complex problems under the rapidly advancing technological landscape has become increasingly dependent on computational technologies; especially, with the numerous interdependent variables and their intricate relationships, solving modern problems surpasses the analytical capabilities of traditional problem-solving methods. These technologies enable us to process vast amounts of data, identify patterns, and generate innovative and practical solutions.

By leveraging advanced tools such as artificial intelligence (AI), machine learning (ML), and optimization algorithms, these technologies enable us to process vast amounts of data, identify patterns, and generate innovative and practical solutions.

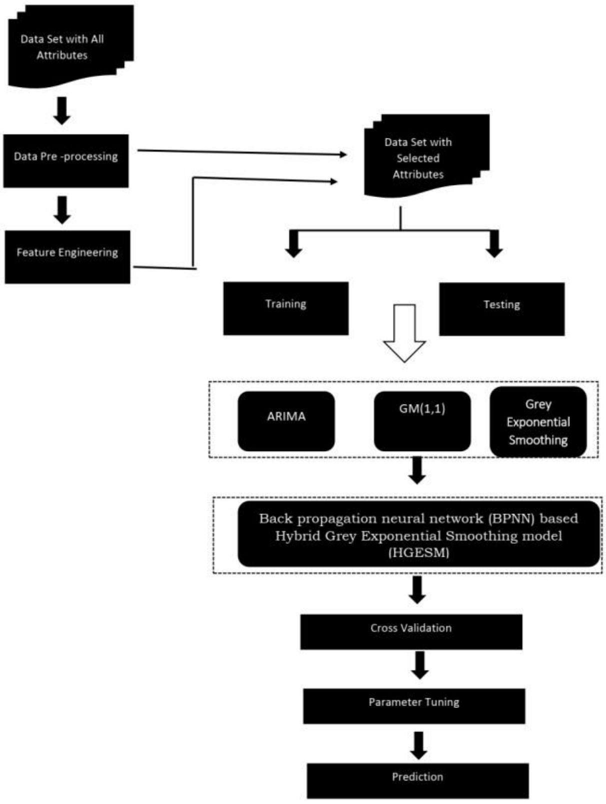

Data Set with All Attributes
Data Pre -processing
Feature Engineering
Data Set with Selected Attributes
Training
Testing
ARIMA
GM(1,1)
Grey Exponential Smoothing
Back propagation neural network (BPNN) based Hybrid Grey Exponential Smoothing model (HGESM)
Cross Validation
Parameter Tuning
Prediction

Figure 1: Implementation Methodology

The purpose of this study is to forecast electricity demand in Sri Lanka. The research process involves data extraction, pre-processing, feature extraction, feature selection, and classification stages to achieve effective sentiment analysis. Figure 1 explains the proposed methodology is processed throughout the five major steps as follows [22].

Step 1: Problem Definition and Understanding

Step 2: Data Collection, Preparation and Feature Engineering

Step 3: Algorithm Selection, Model Training and Validation

Step 4: Model Evaluation

Step 5: Documentation and Reporting.

By systematically following these steps, complex problems can be effectively tackled using computer algorithms, leading to robust and reliable solutions [23].

III. RESULT AND DISCUSSION

The electricity demand exceeding its supply is a particular problem faced by most developing countries including Sri Lanka today. Hydroelectricity has played a major role in the power-generating industry in Sri Lanka since the early 1990s [24].

Due to the various types of environmental and maintenance issues, hydropower generation in Sri Lanka has been going down seriously since the 1990s [24]. As a result, the Sri Lankan government has taken different steps. For example, electricity generation has transitioned to mixed hydro-thermal [20, 21].

As a result, the electricity generation of the country has transitioned to mixed hydro-thermal since 1998[20, 21]. In our work, annual power demand data from 2000 to 2022 is investigated. The observations were made from 2000 -2019 and are used for model fitting and 2020-2022 reserved ex-post testing.

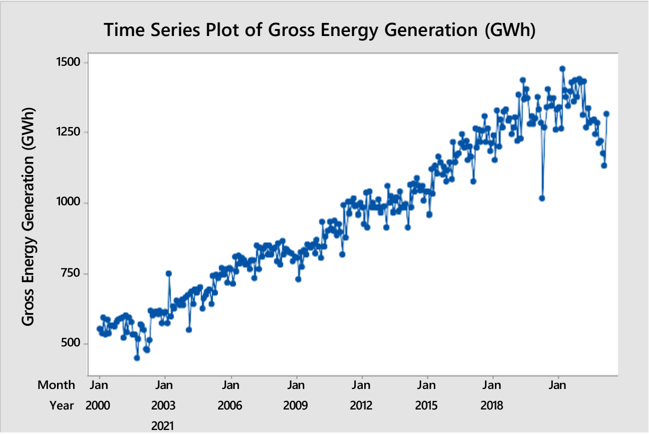


Figure 2: electricity _ Gross generation (MONTHLY)
Sources: Annual Report 2022[22]

According to figure 2, gross energy generation (GWT) of the country has increased rapidly with a significant positive trend with seasonal fluctuations.

A. *Autoregressive Integrated Moving Average (ARIMA) Approach for Forecasting and Evaluation of Electricity Demand in Sri Lanka*

The stationary levels were measured and summarized as Table 1[7,8].

Table 1: Unit Root Test Result _ Level Data

Method	Level data	1st difference	
	t-statistic	Prob*.	Prob.
ADF statistic	2.364504	0.999	0.0001*
Test critical	1% level	-4.057910	

values:	5% Level	-3.119910	
	10% Level	-2.701103	
PP test statistic	0.243138	0.966	0.00*
Test critical values:	1% level	-3.920350	
	5% Level	-3.065585	
	10% Level	-2.673459	
KPSS test	0.542287		0.224

Table results suggested that the first difference data stationary under the 0.05 level of significance. As a next step, the most appropriate ARIMA model was fitted.

The minimum Akaike info criterion (AIC), Schwarz criterion (SC) and Hannan-Quinn criterion (HQC) criterions suggested that, ARIMA (1, 1, 3) (AIC (0.915704), SBIC (1.104517), and HQIC (8.553372)) is a best model.

B. *Hybrid Grey Exponential Smoothing Model (HGESM)*

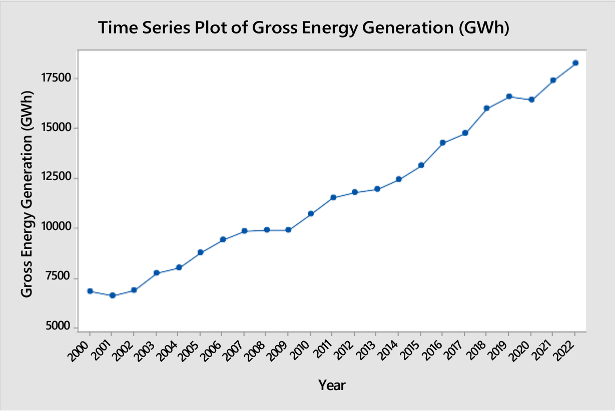


Figure 3: electricity _ Gross generation (Yearly)
Sources: Annual Report 2022[22]

According to the in Figure 3, exponential trend can be seen for electricity gross generation during 2000 -2022.

As a next step, the grey exponential smoothing model runs under the following steps.

Step I

Step I: Original row data series listed as.

$$^{(0)}(k) = [\, 6802.8, 6615.2, 6892, \; \dots, \dots] \quad (1)$$

Based on the row data series, the accumulated generating sequence (AGS) is obtained as equation (2).

$$^{(1)}(k) = [\, 6802.8, 13418, 20310, 28025.3, \dots] \quad (2)$$

$$^{(1)}(k) = [10110.4, 16864, 24167.65, \dots \dots .]^T \quad (3)$$

Step II

The GM (1,1) model can be estimated as equation (4);

$$\hat{x}^{(1)}(k+1) = (x^{(0)}(1) - 399{,}448)e^{-0.02k} + 399{,}448$$
$$k = 1,2, \dots, n \quad (4)$$

Where:Step III

$$[\hat{a}\,\hat{b}]^T = (B^T\,B)^{-1}B^TY = [-0.045 \;\; 6626.34]^T \qquad \text{and}$$

$$B^TB = \begin{bmatrix} 3612250121338.5 & -12436049.0 \\ -12436049.0 & 56.0 \end{bmatrix}$$

As a next, Grey double exponential smoothing (GDEM) model estimated as;

$$s'(\mathrm{k}) = 1.163\, x^{(r)}(k) + 0.10562 s'(k-1) \quad (5)$$

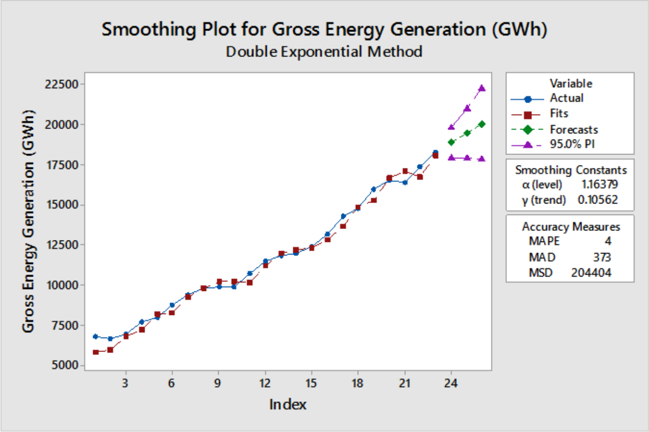


Figure 4: Double Exponential Method

Step IV

Due to the considerable error estimation, the Hybrid Grey Exponential Smoothing model (HGESM) is estimated as equation 06;

$$\hat{X}_e^{(0)}(k) = \hat{X}^{(0)}(k) + \hat{\varepsilon}^{(0)}(k) \quad (06)$$

Where; $\hat{\varepsilon}^{(0)}(\mathrm{k})$ is error estimation of GM(1,1) and HGESM, respectively.

B. Model Comparison

To find the best out-of-sample forecasting performance, three error accuracy measures namely MAD, MSE, and MAPE (%), are used and summarized in Table 3. Two samples from 2018 to 2020 (S_ 01) and 2020 to 2022 (S_02) were considered.

Table 2 : Model accuracy results

Model Accuracy	**Forecasting Accuracy (%)**							
	Moving Average		**Double Exp.**		**GM(1,1)**		**HGESM**	
	S_1	S_2	S_1	S_2	S_1	S_2	S_1	S_2
MAD (%)	0.24	0.218	0.321	0.111	0.132	0.154	0.062	0.050
MSE (%)	8.17	5.847	12.132	2.034	2.270	3.234	0.480	0.290
RMSE	2.85	2.418	3.483	1.426	1.506	1.798	0.692	0.538
MAPE (%)	0.34	0.312	0.452	0.159	0.187	0.220	0.088	0.071

*denotes the model with the minimum error values

According to the Table 2, the newly proposed Hybrid Grey Exponential Smoothing model is highly accurate (less than 10% MAD and MAPE) than double exponential smoothing models; especially, with non-stationary patterns.

IV. CONCLUTION

Since the introduction of the Lakshapana Wimalasurendra power station in 1965, the Power sector of Sri Lanka has become heavily dependent only on hydropower. It nearly covered more than 50% of the total grid capacity in the national installed power capacity [25,26]. However, the unexpected rising demands in the industrial sector in Sri Lanka have created a serious issue with a large increase in demand for electricity [26]. Furthermore, limited generation potential availabilities of existing hydro plants due to the limited rainfall have been helping to create this problem more seriously. As a result, Sri Lanka has decided to move towards renewable and non-renewable energy sources such as Coal-fired power stations, thermal wind, and solar power for electricity generation since 2010[26].

The current study predicts and analyzes the electricity demand in Industrial Sectors in Sri Lanka. The forecasting results show that the annual demand for electricity in Sri Lanka is expected to increase by 4.9 percent over the next decade.

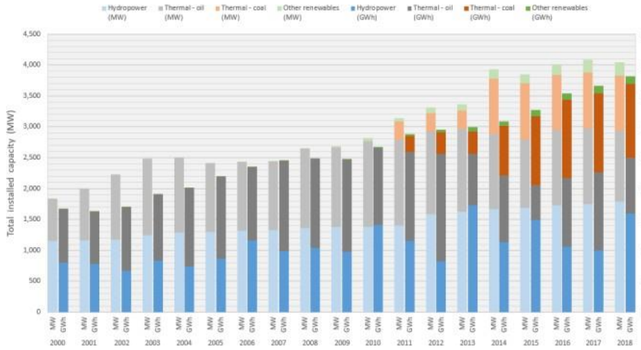


Figure 5: electrical capacity and production of Sri Lanka
Sources: Annual Report 2022[26]

The finding coincides with Ceylon Electricity Board (CEB) statistics published in 2023[26]. Hence, the government should pay urgent attention to adding additional interventions and alternative renewable and nonrenewable energy sources for the national power grid over the next 10 years as early as possible.

REFERENCES

[1] D. C. Idoniboyeobu, A. J. Ogunsakin, and B. A.Wokoma, “Forecasting of Electrical Energy Demand in Nigeria using Modified Form of Exponential Model,” Am. J. Eng. Res. AJER, vol.7, 2018, pp. 122–135.

[2] L. Hernandez et al., “A survey on electric power demand forecasting: Future trends in smart grids, microgrids and smart buildings,” IEEE Commun. Surv. Tutorials, vol. 16, no. 3,2014, pp. 1460–1495.

[3] H. Sangrody, N. Zhou, S. Tutun, B. Khorramdel, M.Motalleb, and M. Sarailoo, “Long term forecasting using machine learning methods,” in 2018 IEEE Power and Energy Conference at Illinois (PECI),2018.

[4] Dutta, Anurag & Chandra Voumik, Liton & Kumarasankaralingam, Lakshmanan & Kaya, Funda & Raihan, “Forecasting the Economic Crisis of Sri Lanka: Application of Machine Learning Algorithms for Time Series Data”, 2023, 10.13140/RG.2.2.32003.07209.

[5] Kayacan, E., Ulutas, B., & Kaynak, O., “Expert Systems with Applications Grey system theory-based models in time series prediction”, Expert Systems With Applications, Vol 37, No 2, 2010, pp. 1784–1789. doi:10.1016/j.eswa.2009.07.064.

[6] Rathnayaka, R. M. K. T., Seneviratna, D. M. K. ., & Jianguo, W. , “Grey system based novel approach for stock market forecasting”, Grey Systems: Theory and Application, Vol 5, No 2, 2015, pp.178–193. doi:10.1108/GS-04-2015-0014

[7] R. M. K. T. Rathnayaka, W. Jianguo and D. M. K. N. Seneviratna, "Geometric Brownian Motion with Ito's lemma approach to evaluate market fluctuations: A case study on Colombo Stock Exchange," 2014 International Conference on Behavioral, Economic, and Socio-Cultural Computing (BESC2014), Shanghai, China, 2014, pp. 1-6, doi: 10.1109/BESC.2014.7059517

[8] Liu, S., Lu, N., Shang, Z. and Rathnayaka, R.M.K.T. , "A new grey relational analysis model of cross-sequences", Grey Systems: Theory and Application, Vol. 14 No. 2, 2024, pp. 299-317. https://doi.org/10.1108/GS-10-2023-0098

[9] Rathnayaka, R.M.K.T.and Seneviratna, D.M.K.N., “Artificial Neural Network based New Hybrid Approach for Forecasting Electricity Demands in Sri Lanka”. Kelaniya International Conference on Advances in Computing and Technology (KICACT - 2017), Faculty of Computing and Technology, University of Kelaniya, Sri Lanka. 2017, pp 13-18.

[10] Rathnayaka, R.M.K.T., Seneviratna, D.M.K.N. and Jianguo, W., "Grey system based novel approach for stock market forecasting", Grey Systems: Theory and Application, Vol. 5 No. 2, 2015, pp. 178-193. https://doi.org/10.1108/GS-04-2015-0014.

[11] Rathnayaka, R.M.K.T. and Seneviratna, D.M.K.N. , "Taylor series approximation and unbiased GM(1,1) based hybrid statistical approach for forecasting daily gold price demands", Grey Systems: Theory and Application, Vol. 9 No. 1,2019, pp. 5-18. https://doi.org/10.1108/GS-08-2018-0032.

[12] Rathnayaka, R.M.K.T. and Seneviratna, D.M.K.N., "Predicting of aging population density by a hybrid grey exponential smoothing model (HGESM): a case study from Sri Lanka", Grey Systems: Theory and Application, 2024, https://doi.org/10.1108/GS-01-2024-0002.

[13] Wang, J., Ma, X., Wu, J., & Dong, Y., “Optimization models based on GM (1, 1) and seasonal fluctuation for electricity demand forecasting”, International Journal of Electrical Power and Energy Systems, Vol 43, No.1, 2018, pp.109–117. doi:10.1016/j.ijepes.2012.04.027.

[14] Hsu, L., “A genetic algorithm based nonlinear grey Bernoulli model for output forecasting in integrated circuit industry”, Expert Systems with Applications, Vol. 37 No. 1, 2022, pp.4318-4322.

[15] Wang Xiaohuaa, Dai Xiaqingb and Zhou Yuedongb, "Domestic energy consumption in rural China: A study on Sheyang County ofJiangsu Province", Biomass and Bioenergy, Vol. 22, 2022, pp.251 – 256.

[16] Jinjin Wang, Zhengxin Wang, Qin Li, “Export injury early warning of the new energy industries in China: A combined application of GM(1,1) and PCA method”, Grey Systems: Theory and Application,2017, 7(2), 272-285. https://doi.org/10.1108/GS-02-2017-0003

[17] Wang, J., Ma, X., Wu, J., & Dong, Y., “Optimization models based on GM (1, 1) and seasonal fluctuation for electricity demand forecasting”, International Journal of Electrical Power and Energy Systems, Vol 43, No.1, 2012, pp.109–117. doi:10.1016/j.ijepes.2012.04.027

[18] Hsu, L., “A genetic algorithm based nonlinear grey Bernoulli model for output forecasting in integrated circuit industry”, Expert Systems with Applications, Vol. 37 No. 1, 2010, pp. 4318-4322.

[19] Yao, A.W.L. and Chi, S.C., “Analysis and design of a Taguchi-Grey based electricity demand predictor for energy management systems”, Energy Conversion and Management, Vol. 45 No. 7, 2004, pp. 1205-1217.

[20] Sifeng Liu, Y. Yang, Naiming Xie, and J. Forrest, New progress of Grey System Theory in the new millennium, Grey Systems: Theory and Application, 2016, 6 (1): 2-31. https://doi.org/10.1108/GS-09-2015-0054

[21] Julong, D., “Introduction to Grey System Theory”, The Journal of Grey System, Vol1, No.1, 1989, pp.1–24

[22] Lifeng Wu, Sifeng Liu, Yingjie Yang, Grey double exponential smoothing model and its application on pig price forecasting in China, Applied Soft Computing, Volume 39, 2016, 117-123, https://doi.org/10.1016/j.asoc.2015.09.054.

[23] Rathnayaka, R.M.K.T., Seneviratna, D.M.K.N., Jianguo, W. and Arumawadu, H.I., "An unbiased GM(1,1)-based new hybrid approach for time series forecasting", Grey Systems: Theory and Application, Vol. 6 No. 3, 2016, pp. 322-340. https://doi.org/10.1108/GS-04-2016-0009

[24] Rajaratnam Shanthini, "Could Sri Lanka afford sustainable electricity consumption practices without harming her economic growth?", Paper presented at the 9th Asia Pacific Roundtable for Sustainable Consumption and Production, June 2022.

[25] Ceylon Electricity Board, Annual Report, 2022.

[26] Ceylon Electricity Board, Annual Report, 2021.

ABBREVIATIONS AND SPECIFIC SYMBOLS

- ANN: Artificial Neural Network
- GESM : Grey Exponential Smoothing model
- HGES:Hybrid Grey Exponential Smoothing
- ML: Machine Learning
- GST: Grey System Theory
- RNN: Grey System Theory
- LSTM:Long short-term memory
- ARIMA: Autoregressive Integrated Moving Average

AUTHOR BIOGRAPHY/IES

D.M.K.N. Seneviratna is working as a Senior Lecturer in the Department of Interdisciplinary Studies, Faculty of Engineering, University of Ruhuna, Sri Lanka. Her research interests include the fields of Time Series Modelling, Data Mining, and Machine Learning.

Automatic Bug Priority Prediction using LSTM and ANN Approaches during Software Development

D.N.A. Dissanayake[1], R.A.H.M. Rupasingha[2#], and B.T.G.S. Kumara[3]
[1,2]Department of Economics and Statistics, Sabaragamuwa University of Sri Lanka, Belihuloya, Sri Lanka
[3]Department of Computing and Information Systems, Sabaragamuwa University of Sri Lanka, Belihuloya, Sri Lanka

[#]hmrupasingha@gmail.com

ABSTRACT The process of manually assign a priority value to a bug report takes time. There is a high chance that a developer may allocate the wrong value, and this can affect several important software development processes. To address this problem, the objective of this research incorporates three unique feature extraction approaches to create a model for automatically predicting the priority of bugs using the Long Short-Term Memory (LSTM) deep learning algorithm and Artificial Neural Network (ANN) algorithm. First, we collected approximately 20,500 bug reports from the Bugzilla; bug tracking system. Followed preprocessing, created models using two classifiers and feature vectors including Global Vectors for Word Representation (GloVe), Term Frequency-Inverse Document Frequency (TF-IDF), and Word2Vec used individually. The final classification results were determined by comparing the all results of the different models, which were integrated into an ensemble model. For evaluating the models, accuracy, recall, precision, and f-measure were used. The ensemble model produced the highest accuracy of 92% than other models as ANN model's accuracy was 80.28%, LSTM GloVe model's accuracy was 89.58%, LSTM TF-IDF model's accuracy was 88.94%, LSTM W2V model's accuracy was 84.84%. And also, higher recall, precision, and f-measure results were found in the ensemble model. Using the proposed model by LSTM-based ensemble approach we could automatically find the bug priority level of bug reports efficiently and effectively. In the future studies, intend to gather data from sources other than Bugzilla, such as JIRA or a GitHub repository. Additionally, try to apply other deep algorithms to improve the accuracy.

INDEX TERMS Bug Priority Prediction, Ensemble Model, LSTM

I. INTRODUCTION

A crucial step in the software development process is software maintenance which stands for changing, tweaking, and updating software and its features to produce a better version of it [1]. Developers and other responsible parties maintain software for a variety of purposes, including enhancing general software performance and fixing bugs after the software is released.

A bug repository is one of the most crucial software repositories and the most significant database in the software development process [2]. For updating and keeping information about problems that emerge or suggestions for improving the project, many software projects establish and maintain bug repositories. The people generate, store, update, and research every software defect in the software repositories. As a result, developers have to continuously update and produce different bug reports to aid in the creation and maintenance of software.

The most crucial task of the software that is being improved is bug fixing. To improve software systems, developers and project managers collect bug reports and look at Bug Tracking Systems (BTS) [3], sometimes referred to as issue tracking systems, such as JIRA [4] and Bugzilla [5], which assist developers in handling bug triaging and bug reporting [6].

The performance and quality of software systems may decline as a result of the numerous defects that exist in them. It is impossible to produce error-free software and many projects will be delivered with flaws because bugs are a regular occurrence [7]. Software creators enable users to submit defects in the BTS to enhance the upcoming version of the program. The following pre-defined fields are included in a bug report: the bug ID, content ID, title, error description, owner/author, status, priority, version, and severity [8]. The urgency of a defect's remedy is determined by bug priority.

Assigning a bug priority or bug prioritization is a very important task due to several reasons [9]. It facilitates a deeper comprehension of the bug and identifies potential solutions. After finding the bug, we can improve the program architecture to prevent it from becoming a greater issue. The bug that is creating the most issues is determined to have the highest priority. The priority of the bug determines the sequence in which the developer or project manager should fix it. With P1 denoting the highest priority and P5 denoting the lowest priority, a bug report's priority is assigned on a scale of P1 to P5. Bug prioritization is a manual process that requires a lot of time because there are so many bug reports. When a defect is submitted, a developer looks into it and manually assigns priority to the pertinent bugs. The term "bug triaging" refers to this time-consuming manual process carried out by humans [10]. As a result, the likelihood of improper bug prioritizing is considerable. There may be a high possibility of incorrect bug prioritization as well. Automating the process of prioritizing bug reports is crucial for avoiding this serious problem. In this study, we suggested to build a model as a solution to the issue of identifying bugs with the highest priority.

This study's primary goal is to develop a model for automatically predicting the prioritization of bugs using ANN algorithm and LSTM deep learning algorithm by combining three feature extraction methods as a solution for above mentioned problem.

A bug priority prediction model can be useful in several ways. The machine learning and deep learning classifiers used for classifying the text of the bug reports when it comes to prioritizing bugs. After collecting data, they should pre-process. Then feature extraction is carried out utilizing three various techniques including TF-IDF, Word2Vec, GloVe with the LSTM algorithm and TF-IDF with ANN as algorithm. Three LSTM results were combined into ensemble model to take the final classification results with the comparison of individual model results of ANN and LSTM Models. Accuracy, recall, precision, and f-measure were used for measure the evaluation of the models.

The following is a summary of expected contributions of this paper.

I. Ensemble approach based on LSTM algorithm is proposed to automatic priority prediction of bug reports into five priority levels namely P1, P2, P3, P4 and P5.
II. The suggested strategy is based on analysing bug reports. The proposed methodology for bug priority prediction provides correct automatic priority levels for analysing and improving software systems on time.
III. LSTM three individual models, LSTM ensemble approach and ANN model are compared with each other to evaluate the performance of the proposed approach.

This paper is organized as follows. In Section 2, review existing literature. Section 3 explains the proposed methodologies. Research finding and evaluation of the results shown in Section 4. Finally, in Section 5 concludes the paper and discusses the recommendation.

II. LITERATURE REVIEW

A. Related Work

To identify the uniqueness of our research, it is important to review the existing literature in knowledge. The majority of current studies have used deep neural techniques, and relied on machine learning algorithms to forecast the priority levels in a bug report. We perform a critical analysis of the preceding works to show the originality of our study.

There were basically two main paths in early studies under the topic of bug report such as priority prediction and severity prediction [11]. Priority prediction of bug reports was recently carried out [12], using a CNN-based technique. Utilizing Natural Language Processing (NLP) techniques, done preprocessing on data bug descriptions, and created a classification model utilizing TCN, CNN, and SVM algorithms. Accuracy, Recall, Precision, and F1-score were used to evaluate how well the generated models performed. This study used a deep neural network-based algorithm, NLP techniques, and feature extractions to anticipate the priority levels of bug reports. And research findings state that CNN is best for priority prediction according to their study.

In order to eliminate manual bug prioritizing in [10], a software engineering domain repository was utilized to train and calculate the emotion value using emotion analysis. Based on input data, the CNN classifier makes a priority suggestion. The priority suggestions for the reports gathered from the Bugzilla and Eclipse projects were made using the CNN prioritization method. On average, proposed approach improves the F1 value by more than 6%. As well, some researchers Qasim Umer et al. [13], propose an automated approach for bug prediction of each issue report obtained using Eclipse data from the Bugzilla database. This method is based on emotion words. They coupled NLP techniques with machine learning algorithms like SVM, Naive Bayes classifier (NB), and Linear Regression (LR) to overcome the issue. As we select the LSTM approach for our study, Hani Bani Salameh et al. [14] constructed a deep learning RNN- LSTM network with five layers and compared the results with SVM and KNN for issue prediction based on more than 2000 JIRA bug reports. Results indicate that for performance-based accuracy, AUC, and f-measure, LSTM scored best. As in values, accuracy was 0.908, AUC was 0.95, and F measure was 0.892.

When it comes to severity prediction, the aim of previous studies is to investigate automated severity prediction because manual prioritization is time-consuming and tedious. The study [15], used NLP as a preprocessing strategy after extracting information from open-source project data and is based on the textual description that is under a deep neural network. Deep learning techniques such as CNN, LSTM, RF, and MNB were used for training and prediction, with CNN having the highest accuracy of all techniques. On average, it improves the F- score by 7.90% according to the results of the study. Additionally, severity prediction on data gathered through Bugzilla in [16], was carried out using the Bagging ensemble approach and the C4.5 classifier. The outcomes of comparing the two approaches indicate that the C4.5 classifier performs better at predicting severity of issue reports for cross component context and closed source software. According to the results J48 classifier gain 79.82% of accuracy while bagging classification algorithm become highest accuracy among them while representing 81.27% accuracy.

The approach [17], organizes Mozilla and Eclipse bug reports into severity categories based on topics, then extracts features from each topic. Then, by assimilating traits from the LSTM and CNN algorithms, forecast the severity. In order to estimate the severity, they feed the CNN with extracted features as its input, and it uses its output to feed the LSTM. The performance of the suggested model was assessed by comparing it to the baseline in order to make better predictions.

Instead of single priority and severity prediction, there were some areas of researches under hybrid approach in both priority prediction and severity prediction. According to [18], they build a hybrid model for predict the defective areas of source code named CBIL. First using source code, they extracted the Abstract Syntax Tree (AST) tokens as vectors. Then CNN extracted the semantics of AST tokens. After that Bi- LSTM track the key vectors and reject other features to improve the accuracy of the model. Used dataset were seven open-source Java projects. According to their results, RNN accomplished the top performance. Not only that, but also in [19], they proposed a hybrid model for software defect prediction which combined SVM and RBF with MRMR feature selection. According to their results, MRMR gives better performance compared to SVM. According to other researchers Tanujit and Ashis proposed a novel hybrid methodology in their study [20], for improvement of defect prediction for software. In their study, they prove the theoretical consistency of their proposed model under more than ten NASA SDP datasets while showing the superiority of their proposed method.

As well as priority prediction there were some researchers done severity prediction under hybrid approach such as [21]. In their study thy proposed an approach for severity prediction based on the feature selection algorithm of the severity of each topic of data from Eclipse and Mozilla open-source projects. In the process they conducted, first classify issue reports by topic-based severity and extracted features from it. Then severity was predicted by learning characteristics from the LSTM and CNN algorithms. The comparison of summary of reviewed papers is shown in Table 1.

Table 1. Summary of existing studies

Ref. No	Data	Methodology	Objective/s	Limitations	Overcome limitations
[7]	JIRA	LSTM, SVM, KNN	Provides a framework for automate predict priority	2000 of small dataset	Use more than 20500 of data
[14]	Eclipse project & Bugzilla	CNN	To end manual prioritization of bug reports	Limited only one feature extraction	Apply three feature extraction methods
[12]	4 open-source projects	CNN	Predict the bug report's priority automatically	Limited only one feature extraction	
[13]	Bugzilla	Emotional Analysis, SVM	By avoiding manual Prioritization, predict priority that help developers to focus bugs resolution	Only consider emotional analysis	Considering bug prioritization using deep learning algorithms
[15]	Bugzilla	CNN, LSTM, RF, MNB	Automate prioritizing the severity	Only consider severity prediction	Based on bug priority prediction of bug reports
[16]	Bugzilla	J48 algorithm, Bagging algorithm	Find a new technique to avoid assigning incorrect severity using bagging ensemble method	Small dataset Only consider severity prediction	Increase data set into 20,500 & Focus on bug priority prediction
[17]	Eclipse and Mozilla projects	CNN, LSTM	Use topic-based feature selection and CNN-LSTM to predict severity	Only consider severity prediction	Based on bug priority prediction of bug reports
[18]	open-source Java projects	CNN & Bi-LSTM	Extract the semantics of source code for software defect prediction	Limited only one feature extraction	Apply three feature extraction methods
[19]	NASA Metrics Data Program	SVM & RBF	Build a model for effective to deal with the imbalance datasets in software defect prediction	Limited only one feature extraction	
[20]	NASA SDP dataset	-	Proves the theoretical consistency of Hellinger net	Provide only theoretical base	Use algorithms to evaluate each algorithm performance, use statistical measurements to evaluate models' effectiveness

[22]	Eclipse, Mozilla, Apache, and NetBeans datasets	SMOTE & IFSM	Predict the severity and priority of software bugs using the IFSM	Small dataset	Increase data set into 20,500
[23]	Mozilla, Eclipse, NetBeans, GCC	CNN, SVM, Random Forests & Logistic Regression, HAN	Build a Hierarchical Attention Network (HAN) model for prioritizing software bug reports	Use GloVe word embeddings only	Apply three feature extraction methods
[24]	JIRA deployed by Apache	Empirical study	Empirical study to explore the phenomenon of bug priority changes	Conduct only empirical study	Conduct a full approach using algorithms

B. Research Gap

The majority of existing researchers built a model for bug prioritization using just one feature vector. Some research used two feature extraction methods for result comparison and found a better one. There were no studies conduct combining more than two feature vectors. Also, there were no any studies which were combining unique feature extraction techniques with LSTM and ANN for bug priority prediction.

To cover this gap in the existing literature, we were able to apply a variety of three feature extraction techniques of TF-IDF, GloVe, and Word2Vec with an ensemble approach based on the LSTM deep learning algorithm. Then the result was compared with the individual results including the ANN algorithm result.

III. METHODOLOGY

This study automated the priority levels determined by bug reports using an algorithmic technique. During the scope of this research, a further four-step process of inquiry has been carried out. The first and second processes were collecting bug reports and pre-processing the data. In third step, features were retrieved from the pre-processed data using various extraction methods, such as TF-IDF, GloVe, and Word2Vec. As a fourth step, determine the bug reports related to their priority level using the LSTM and ANN algorithms by combining the three results taken from the three feature vector generations.

The research approach is described in detail in the below Figure 1.

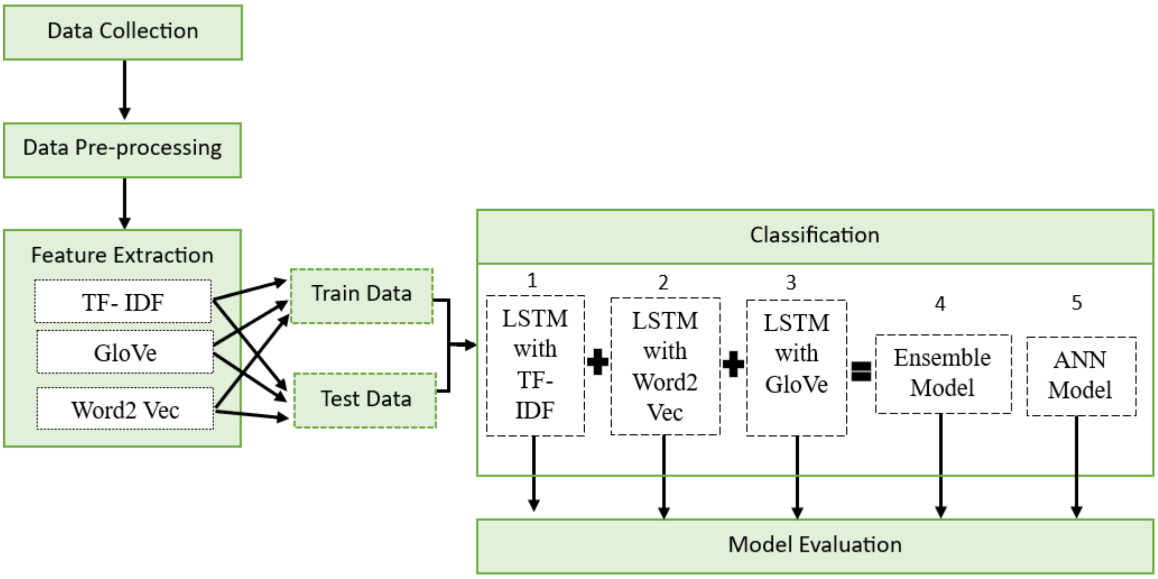


Figure 1. Research approach

A. Data Collection & Labeling

Software flaws also referred to as bugs, are found and tracked during software testing using bug tracking. Tracking issues or tracking defects can mention as other names for this process. There are many options for issue-tracking software, including Jira, Mantis, Redmine, Bugzilla, Backlog, and Bugnet. Software bug reports are submitted to bug-tracking systems every day. Through Bugzilla, we collected over 21,000 bug reports under four open-source applications, including Firefox, Eclipse, Netbeans, and Open Office. The total bug reports we collected are shown in following Table 2.

Table 2. Total Bug Reports

Project	No of bug reports
Mozilla	4165
Eclipse	8478
Netbeans	4305
Open Office	4553
Total	21,501

The source of the bug reports was Bugzilla Bug tracking system. The bug ID (bugID), description (sd), classification (cl), product (pd), component (co), platform (rp), operating system (os), bug status (bs), resolution (rs), priority (pr), and severity (bsr) are the 11 columns in the CSV formatted data files that were created from the collected data. As a major feature, we select description (sd), which establishes the priority level. The remainder of the columns were removed. According to the data, description is the main variable which is affecting the bug priority level. While description act as an independent variable, priority level considers as a dependent target variable.

The data in Table 3 below show the 10 selected examples for the priority values P1, P2, P3, and P4, respectively.

Table 3. Data Examples

Description		Priority
1	Scroll Up Down using Ctrl UP DOWN stopped to work	P1
2	Unable to create a new project	
3	Unable to update existing plugins and unable to install new plugins	P2
4	Can’t display thumbnail images for plugins added since	
5	Copy and paste for an aux file do not work	P3
6	UI freezes in Plugins View	
7	Dead Link from Plugins window	P4
8	Error when uninstall IDE Win	
9	Allow only certain pages to modify fonts	P5
10	Intermittent session restores many windows timeout	

B. Data Pre-processing

The crucial phase in the machine learning process is data pre-processing [25]. The bulk of bug reports contains extraneous and meaningless details. For the classification model to produce better results, the data must be in the correct format. Pre-processing will improve the quality of the data set by eliminating those unnecessary data. It is the process of transforming unstructured data into a more understandable format. After handling the missing values, the text in the bug reports will be cleaned up using different pre-processing techniques as shown in the Figure 2.

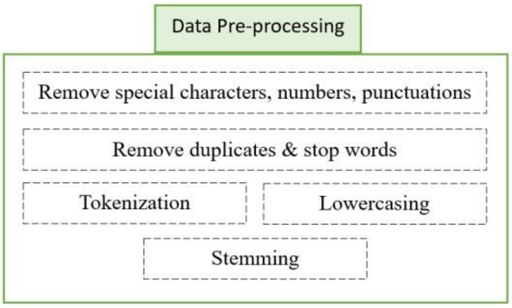

Data Pre-processing
Remove special characters, numbers, punctuations
Remove duplicates & stop words
Tokenization
Lowercasing
Stemming

Figure 2. Steps of data pre-processing

Remove special characters, numbers, punctuations – Remove unnecessary special characters, and numbers, including punctuation from the data, and working with data that contains unnecessary special characters, numbers can be difficult. For this reason, we got rid of some punctuation, special letters, numbers (0–9), and symbols. (! " ” # $ % & \ ' ( ) * + , . / : ; < = > ? @ [\ \ ] ^ _ ` { | } ~).

Table 4. Examples for Remove special characters, numbers, punctuations

Description before remove special characters, numbers, punctuations	Description after remove special characters, numbers, punctuations
PDE quickfix creates invalid @Since tag	PDE quickfix creates invalid Since tag
OpenJ9: Git failing on master, builds blocked	OpenJ Git failing on master builds blocked
Use setUseHashlookup in internal org.eclipse.e4.ui.dialogs.filteredtree.FilteredTree	Use setUseHashlookup in internal org eclipse e ui dialogs filteredtree FilteredTree

Remove duplicates and stop words – Eliminate duplication and stop words since redundant data will result from duplicate data. Duplicate values in the data set were therefore eliminated. Moreover, often used stop words that lack precise definitions alone include "in," "the," "a," "our," "is," and "that." These stop-words will be eliminated during pre-processing.

Table 5. Examples for Remove duplicates & Stop words

Description before remove duplicates & stopwords	Description after remove duplicates & stopwords
PDE quickfix creates invalid Since tag	PDE quickfix creates invalid tag
OpenJ Git failing on master builds blocked	OpenJ Git failing master builds block
Use setUseHashlookup in internal org eclipse e ui dialogs filteredtree FilteredTree	Use setUseHashlookup internal org eclipse ui dialogs filteredtree FilteredTree

Tokenization – Tokenization is the process of breaking down the text into individual words, phrases, and clauses [26]. Unneeded symbols will be eliminated. To put it simply, tokenization eliminates all symbols from the text and divides it into tokens. By doing this, incoming data is divided into useful chunks that can be embedded in a vector space.

Table 6. Examples for Tokenization

Description before tokenization	Description after tokenization
PDE quickfix creates invalid tag	"['PDE', 'quickfix', 'creates', 'invalid', 'tag']"
OpenJ Git failing master builds block	"['OpenJ', 'Git', 'fail', 'master', 'builds', 'block']"
Use setUseHashlookup internal org eclipse ui dialogs filteredtree FilteredTree	"['Use', 'setUseHashlookup', 'internal', 'org', 'eclipse', 'ui', 'dialogs', 'filteredtree', 'FilteredTree']"

Lowercasing – Lowercasing refers to the process of changing all text and data to lowercase letters. Even though the terms "Home" and "home" have the same meaning, the vector space modeling recognizes them as two distinct words if they are not written in lowercase.

Table 7. Examples for Lowercasing

Description before lowercasing	Description after lowercasing
"['PDE', 'quickfix', 'creates', 'invalid', 'tag']"	"['pde', 'quickfix', 'creates', 'invalid', 'tag']"
"['OpenJ', 'Git', 'fail', 'master', 'builds', 'block']"	"['openJ', 'git', 'fail', 'master', 'builds', 'block']"
"['Use', 'setUseHashlookup', 'internal', 'org', 'eclipse', 'ui', 'dialogs', 'filteredtree', 'FilteredTree']"	"['use', 'setusehashlookup', 'internal', 'org', 'eclipse', 'ui', 'dialogs', 'filteredtree', 'filteredtree']"

Stemming – In the English language, a single sentence can take many different forms. These inconsistencies in a text cause data in machine learning models to become redundant. Due to this, it will be impossible to produce the desired results. Thus, stemmed terms in the data set ought to be eliminated. Words are being reduced to their stems in this process. As an illustration, the terms "take," "takes," "taken," and "took" can all be replaced with the one word "take."

Table 8. Examples for Stemming

Description before stemming	Description after stemming
"['pde', 'quickfix', 'creates', 'invalid', 'tag']"	"['pde', 'quickfix', 'create', 'invalid', 'tag']"
"['openJ', 'git', 'fail', 'master', 'builds', 'block']"	"['openJ', 'git', 'fail', 'master', 'build', 'block']"
"['use', 'setusehashlookup', 'internal', 'org', 'eclipse', 'ui', 'dialogs', 'filteredtree', 'filteredtree']"	"['use', 'setusehashlookup', 'internal', 'org', 'eclipse', 'ui', 'dialogs', 'filteredtree', 'filteredtree']"

After the data had been appropriately gathered and captured from the Bugzilla, we apply some data pre-processing techniques to clean the bug reports and eliminate the superfluous content. There were no longer any tags, URLs, links, or numbers. As a result, deleting them aids in shrinking the feature space.

C. Feature Extraction

After preprocessing, data must be converted into features for modeling. Raw text input data cannot be directly used to use machine learning techniques. The acquisition of contextual characteristics of the text is required in order to convert it into feature vectors. Symbolic and numeric characters are represented by features in machine learning and pattern recognition [27]. Three different feature vector approaches, TF-IDF, GloVe, and Word2Vec were utilized in this inquiry.

1) TF-IDF

Text input will simply be converted into a numerical format known as a vector form capable of machine learning techniques by using this feature vector. A statistical assessment called TF-IDF [28] looks at a word's relevance to each document in a set of documents. A word's TF-IDF is determined by multiplying two separate metrics.

Term Frequency (TF): The phrase "term frequency" refers to how frequently a word appears in a document. This is calculated by dividing the number of times a word appears in a document by the total number of words in the document. The calculation is shown in following (1).

$$TF = \frac{(\# \, of \, repetitionsof \, word \, in \, a \, document)}{(\# \, of \, words \, in \, a \, document)} \qquad (1)$$

Inverse Document Frequency (IDF): The inverse document frequency measures how frequently a word appears in a group of documents. This indicates how uncommon a word is over the full corpus of documents. This value will be close to 0, if the word is widely used and frequently found in a document, else it will be 1. The calculation is shown in following (2).

$$IDF = \frac{(\# \, of \, documents)}{(\# \, of \, words \, in \, a \, document)} \qquad (2)$$

TF-IDF: These two numbers are multiplied to provide the TF-IDF score of a word in a document. The word becomes more significant in that specific document the higher the score. The calculation is shown in following (3).

$$TF - IDF = TF * IDF \qquad (3)$$

2) GloVe

The GloVe [29],an unsupervised learning method for Word Representation, is a popular algorithm for generating word embeddings in machine learning. Word embeddings are dense vector representations of words that capture semantic and syntactic relationships between words based on their contexts in a given corpus of text. The primary goal of GloVe is to create word embeddings that capture the meaning of words by leveraging the statistics of word co-occurrence in a large corpus. It combines elements from two different approaches to word embeddings: count-based methods like Latent Semantic Analysis and predictive methods like Word2Vec.

3) Word2Vec

Word2Vec [30] is a popular algorithm in machine learning that is used for representing words as numerical vectors in a high-dimensional space. It was introduced by Tomas Mikolov et al. at Google in 2013 and has since become a fundamental tool for NLP tasks. The main idea behind Word2Vec is to capture the semantic and syntactic relationships between words by learning vector representations based on their contextual usage in a given corpus of text. The algorithm operates on the assumption that words appearing in similar contexts tend to have similar meanings. For example, in the sentence "The cat sat on the mat," the words "cat" and "mat" are more likely to be similar in meaning because they both appear in the context of a sentence about sitting on an object.

D. Classification

Then LSTM deep learning algorithm and ANN algorithms are used to forecast the priority levels in a bug report. Cleaning and feature extraction of the data resulted in the separation of the data into training and testing data, with training data accounting for 70% and testing data for the remaining 30% by the experiment. The different approaches generate for priority prediction by loading it into ANN and LSTM independently.

1) LSTM

LSTM is a type of recurrent neural network, (RNN [31]) architecture that is designed to handle and model long-term dependencies in sequential data. It overcomes the limitations of traditional RNNs by introducing memory cells and gating mechanisms. The key idea behind LSTM is the concept of a memory cell, which allows the network to remember information over long sequences. The memory cell is responsible for storing and updating information, selectively forgetting or retaining it based on its relevance to the current context. Overall, LSTM is a powerful RNN variant that enables the modeling of long-term dependencies and has been widely adopted in various domains of machine learning and natural language processing due to its ability to effectively handle sequential data.

2) ANN

Artificial neural networks, often known as neural networks, are models that use computer techniques to imitate the behavior of neuron-based biological systems [32]. ANN is with machine learning and pattern recognition capabilities. The central nervous system of animals serves as the model's inspiration. This network of "neurons" may compute values from input data. Three layers, including an input layer, a hidden layer, and an output layer, may be present in a neural network.

LSTM deep learning algorithm and the ANN algorithm were chosen due to their strengths in handling sequential data, learning complex relationships [33], and their established effectiveness in similar machine learning tasks based on the literature review.

In this study, TF-IDF, GloVe, and Word2Vec feature vectors were used to create three LSTM prediction model separately. Then, three distinct models that were in alignment with the LSTM algorithm were combined to make an ensemble model. When each model predicts the priority, we selected the majority by comparing three results generated by three models as below table 9 shows.

Table 9. Selecting majority value using three LSTM models

Description	GLOVE	TF-IDF	W2V	Majority	Ensemble Result
Performance loss in composite WM paint	1	1	1	1	1
Validator incorrectly flags conditionally declared class as a PHP error	2	3	3	3	3
Crash when opening a presentation macOS	4	4	5	4	4

Then assigning those majority value as the ensemble model result, we compared it with the individual algorithm results including ANN model to select the best one. The process of ensemble model creation is explained in Figure 3.

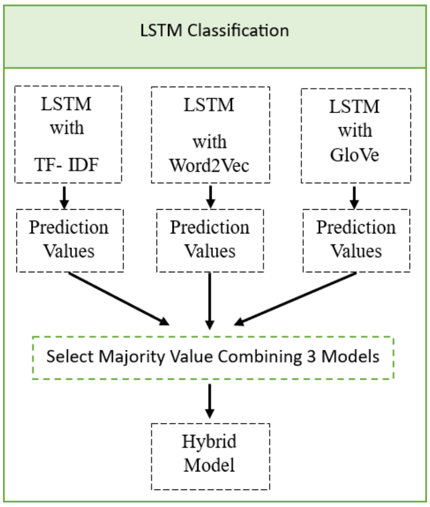


Figure 3. Process of making ensemble model

In parallel, ANN model also created using same three feature vectors and we were identified results were lower than the LSTM Model results. And TF-IDF is the only method which is generating highest accuracy among those three feature vector generations with ANN. So, we decided to use ANN with TF-IDF for further comparison.

After creation of all five models, considering actual priority values and predicted priority values under all models including ANN model, LSTM GloVe, LSTM Word2Vec, LSTM TF-IDF, and ensemble model we done the validation to find a better model using selected 5000 total data set.

IV. RESULTS & EVALUATION

The experimental platform used Microsoft Windows 11 on a PC with processor 8th Gen Intel(R) Core (TM) i3-8145U CPU @ 2.10 GHz 2.30 GHz, 4.00 GB RAM to training, testing, and implementing model.

The evaluation is based on three feature vector extraction methods with LSTM algorithm and ANN algorithm under collected bug reports via Bugzilla. The experiment is designed to classify the descriptions of bug reports into five priority levels using above feature vectors and algorithm. The model is being developed in Python programming language.

Accuracy, precision, recall, and f-measure were computed for ANN and LSTM algorithm under three feature vectors using below (4) – (7) respectively.

Following (4) used to evaluate the accuracy of the model. To do that, total number of correct predictions should divide by total number of predictions. Following (5) used for measure the actually correct proportion of positive identifications. We used recall and f-measure for the evaluation and calculated using (6) and (7) respectively.

$$Accuracy = \frac{No\ of\ Correct\ Predictions}{Total\ Prediction} \tag{4}$$

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (5)$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad (6)$$

$$F - Measure = \frac{2 * Precision * Recall}{Precision + Recall} \quad (7)$$

A. Results of Model Implementation

First, we implement the three LSTM models under different feature vectors and ANN model with TF-IDF. In this section shows the results based on above evaluation measurements on full dataset of 20,501 bug reports. Based on the performance of each model, Figure 4 shows the accuracy results for ANN and LSTM classifier under three feature extraction methods. According to the results, the accuracy of ANN model is 80%, LSTM TF-IDF model is 78.10%, LSTM GloVe model is 81.47% and LSTM Word2Vec model is 75%. Based on the results, LSTM GloVe model shows the highest accuracy among all four models.

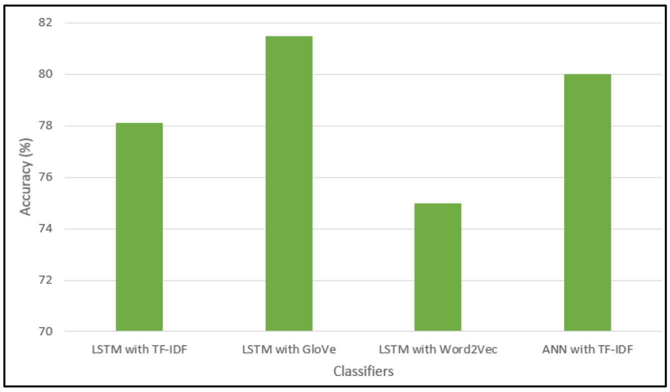


Figure 4. Model Implementation Accuracy of ANN and LSTM under TF-IDF, GloVe, Word2Vec

Table 9 shows the evaluation result for precision, recall, and f-measure of ANN model and LSTM model under three different feature vectors.

Table 10. Precision, recall, f-measure of ANN & LSTM under TF-IDF, GloVe, Word2Vec

Feature Vector	Precision	Recall	F- measure
LSTM TF-IDF	96%	81%	87%
LSTM GloVe	98%	83%	89%
LSTM Word2Vec	94%	77%	84%
ANN TF-IDF	93%	83%	83%

Furthermore, evaluation was consider using different percentages (57%, 67%, 77% and 87%) of training data. Accuracy values for ANN and LSTM under three feature vectors were as in Table 10. By considering the results of different training data sizes, we identified 77% of training data size as optimum value for our evaluation process. Then we divide dataset into 77% of training data and 33% of testing data.

Table 11. Evaluation of accuracy according to the different training data set

Classifier	Training data size			
	57%	67%	77%	87%
LSTM TF- IDF Accuracy (%)	78%	78%	78%	78%
LSTM GloVe Accuracy (%)	78%	77%	81%	78%
LSTM W2V Accuracy (%)	75%	74%	75%	74%
ANN TF-IDF Accuracy (%)	72%	76%	80%	78%

And also, hyperparameters of the ANN and LSTM classifiers were found, and the obtained optimal value for hyperparameters is provided in Table 11 as below and evaluation of accuracy based on those parameters shown in Figure 5 – Figure 8.

Table 12. ANN - LSTM Classifier Hyperparameters

Classifier	Optimum values discovered for the hyperparameters
LSTM with TF-IDF	Epochs= 100, batch_size= 52, optimizer = ‘adam’
LSTM with GloVe	Epochs= 200, batch_size= 33, optimizer = ‘adam’
LSTM with W2V	Epochs= 200, batch_size= 43, optimizer = ‘adam’
ANN with TF-IDF	Epochs=250, batch_size=43, optimizer = ‘adam’

B. Results of Data Validation

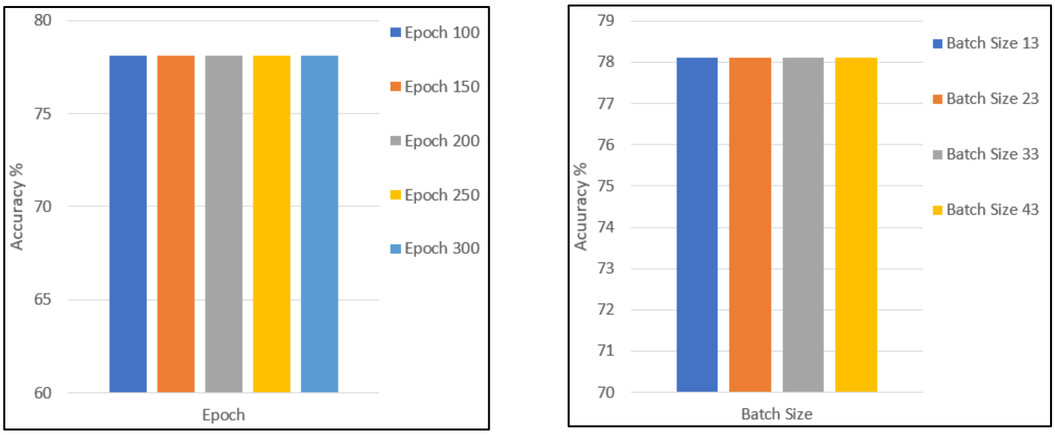
Priority values were predicted using LSTM three models which we implemented using TF-IDF, GloVe and Word2Vec and ANN with TF-IDF under validation dataset of 5000 bug reports. Then we check the algorithm outcomes by combining different feature extraction techniques results, we found majority priority value for each bug description. By considering that majority value we got the ensemble model priority value as shown in the Table 12 examples.

Table 13. Generating majority value for ensemble model

Bug Report	GloVe Priority	TF IDF Priority	Word2Vec Priority	Majority Value (LSTM Ensemble model)
1	P1	P1	P1	P1
2	P2	P1	P1	P1
3	P3	P3	P3	P3
4	P4	P3	P3	P3
5	P5	P5	P4	P5

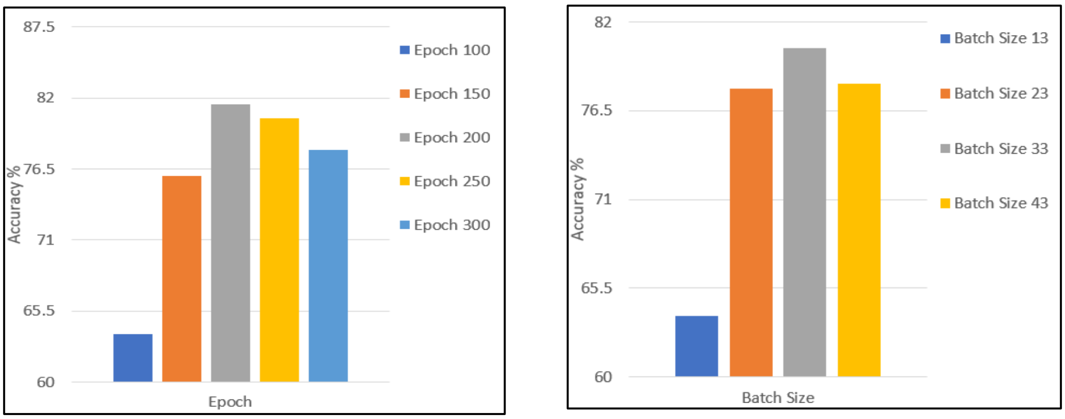
Then considering actual priority values and predicted priority values under five models including ANN model, LSTM GloVe, LSTM Word2Vec, LSTM TF-IDF, and ensemble model we done the evaluations to find a better model

.



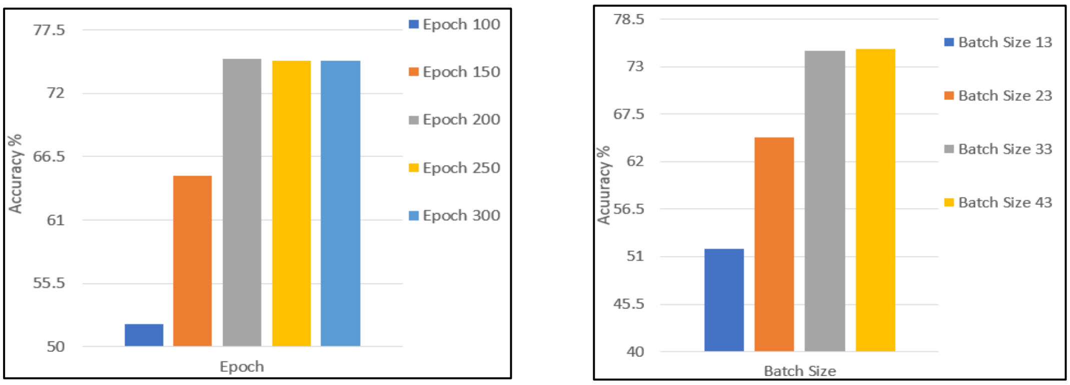
(a) (b)

Figure 5: (a) Epochs accuracy and (b) Batch sizes accuracy of LSTM with TF-IDF



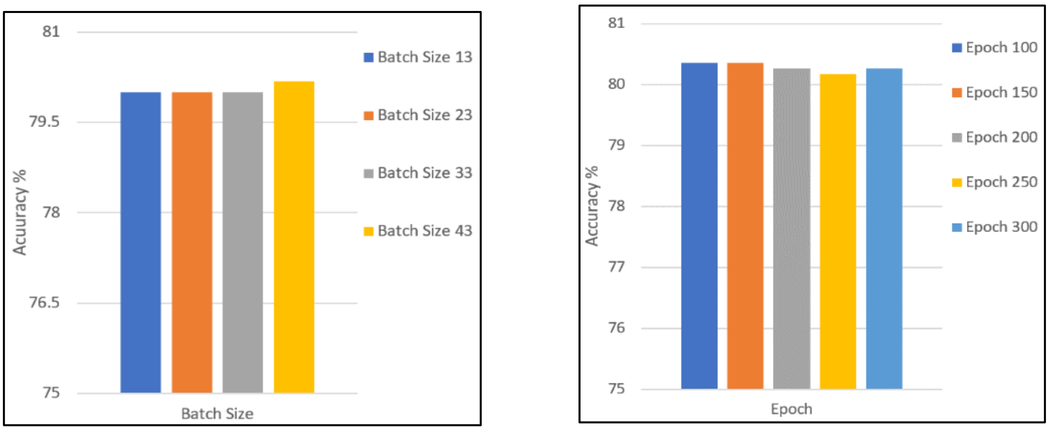
a) (b)

Figure 6: (a) Epochs accuracy (b) and Batch sizes accuracy of LSTM with GloVe



(a) (b)

Figure 7: (a) Epochs accuracy (b) and Batch sizes accuracy of LSTM with Word2Vec



(a) (b)

Figure 8: (a) Epochs accuracy (b) and Batch sizes accuracy of ANN with TF-IDF

When considering actual priority and predicted priority values ensemble model achieving more accuracy than other three individual models under ANN and LSTM classifier. Following Table 13 shows the summary of predicted values and its accuracy. Here total predictions are 5000.

Table 14. Summary of prediction and its accuracy with the ensemble model using validation data

Classifier	True Predictions	Wrong Predictions	Accuracy
ANN TF-IDF Model	4014	986	80%
LSTM GloVe Model	4479	521	89%
LSTM TF-IDF Model	4447	553	88%
LSTM W2V Model	4242	758	84%
LSTM Ensemble Model	4600	400	92%

Furthermore, following Recall, Precision, and F- measure were calculated under five prediction models based on validation data shown as Table 14.

Table 15. Precision, recall, f-measure of ANN, LSTM & Ensemble models using validation data

Models	Precision	Recall	F- measure
ANN TF-IDF Model	79%	81%	88%
LSTM TF-IDF Model	89%	95%	91%
LSTM GloVe Model	89%	96%	93%
LSTM Word2Vec Model	89%	88%	78%
LSTM Ensemble Model	94%	98%	95%

When comparing all the five models, ensemble model achieved the highest accuracy as well as best performance among all other evaluation metrics. It means ensemble model can gain more reliable and accurate predictions than other four models.

C. Results of statistical test (Student t-test)

It's important to use statistical hypothesis test to select the final model. Statistical significance tests are designed to address and compare the performance of machine learning models. In this study we used student t-test for find a better model. In the case of comparing the performance of models, we have to select two models to perform the paired Student's t-test. Among the above five models, based on the highest accuracy, LSTM GloVe model and Ensemble model were compared. In this student t-test which can compare the means of two independent samples to see if they are significantly different from each other. We conducted a two-sample t-tests to compare the means of these two sets of accuracies as below table.

Table 16. Results of t-test

Measurement	Value
t-statistic	6.5027
p-value	0.0001

The t-statistic is significantly high at 6.5027 while p-value is extremely low at 0.0001 Since the p-value is much less than 0.05, we reject the null hypothesis that there is no difference between the accuracies of two models. This indicates that the observed difference in accuracies between the LSTM GloVe model and the Ensemble model is statistically significant.

Therefore, based on this statistical evidence, we concluded that the proposed models perform statistically significant. Based on the accuracies, the ensemble model selected as the better model among these two algorithms.

IV. CONCLUSION

The process of manually assigning a bug priority value to a bug report takes time. There is a chance that a developer will reassign a wrong value, and this can affect several important software development processes. The main objective of this research is to incorporate three feature extraction approaches to create a model for automatically predicting the priority of bugs using the ANN and LSTM deep learning algorithm as a solution to the aforementioned problem.

We collect approximately 20,500 bug reports from Bugzilla; bug tracking system. Following preprocessing, created three models using the LSTM classifier and three unique feature vectors including GloVe, TF- IDF, and Word2Vec. After comparing the three LSTM models' outputs, the majority value was determined for creating an ensemble model. In parallelly, ANN model was built under TF-IDF feature vector. After validating those five models on 5000 bug reports and comparing outcomes, it was found that the ensemble model generated the most accurate results than other four models. In the prediction procedure, the ANN model's accuracy was 80.28%, LSTM GloVe model's accuracy was 89.58%, the LSTM TF-IDF model's accuracy was 88.94%, the LSTM W2V model's accuracy was 84.84%, and the accuracy of the ensemble model was 92%. For the purpose of evaluating the models, accuracy, recall, precision, and f-measure were used. Ensemble model achieving the highest values in all evaluation matrixes and shows it was the best method to predict the bug priority value in bug fixing during the software development process. As well as based on this statistical evidence, the models are statistically significant with higher t-statistic value 6.5027 and p-value 0,0001.

These research findings will assist programmers, software developers and project managers in fixing bugs in software systems more quickly than before. As well as new researchers can gain knowledge regarding automate the bug priority prediction. In the future studies, we intend to gather data from sources other than Bugzilla, such as JIRA or a GitHub repository. Additionally, we try to apply other deep learning algorithms to improve the accuracy. And also, we are planning to improve the statistical test using all the proposed models in the future.

REFERENCES

[1] K. Moran, "Enhancing android application bug reporting," *10th Joint Meeting Foundation,* no. Aug, 2015, pp. 1045 - 1047, 2015.

[2] J. Xuan, H. Jiang, Z. Ren and W. Zou, "Developer Prioritization in Bug Repositories," in *34th International Conference on Software Engineering (ICSE)*, 2012.

[3] "Debian-wiki portal," 08 04 2022. [Online]. Available: https://wiki.debian.org/BTS. [Accessed 25 June 2023].

[4] "Jira Issue Tracker," Atlassian, [Online]. Available: https://www.atlassian.com/software/jira/. [Accessed January 2023].

[5] "Bugzilla Issue Tracker," Mozilla, [Online]. Available: https://www.bugzilla.org/. [Accessed January 2023].

[6] X. Xia, D. Lo, X. Wang and B. Zhou, "Accurate developer recommendation for bug resolution," in *20th Working Conference Reversen Eng. (WCRE)*, Oct., 2013.

[7] H. Bani-Salameh, M. Sallam and B. Al shboul, "A Deep-Learning-Based Bug Priority Prediction Using RNN-LSTM Neural Networks," *e-Information Software Engineering Journal*, vol. 15, no. 1, pp. 29-45, 2021.

[8] "Bugzilla - Reporting a Bug," www.Bugzilla.org, June 2023. [Online]. Available: https://www.bugzilla.org/contributing/reporting_bugs.html. [Accessed 14 June 2023].

[9] R. Harris, "Singlemindconsulting," 14 August 2020. [Online]. Available: https://www.singlemindconsulting.com/blog/prioritize-bug-fixes-vs-product-features/. [Accessed 25 June 2023].

[10] H. L. a. I. I. Q. Umer, "CNN-based automatic prioritization of bug reports," *IEEE transactions on reliability*, Vols. 69, no. 4, pp. 1341-1354, 2020.

[11] "Browserstack - Bug Severity Vs Priority," www.browserstack.com, 2023. [Online]. Available: https://www.browserstack.com/guide/bug-severity-vs-priority\. [Accessed 14 June 2023].

[12] R. Rathnayake, B. Kumara and E. Ekanayake, "CNN-Based Priority Prediction of Bug Reports," in *International Conference on Decision Aid Science and Application (DASA)*, 2021.

[13] Q. Umer, H. Liu and Y. Sultan, "Emotion Based Automated Priority Prediction for Bug Reports," *IEEE Access*, vol. 6, no. July 2, 2018, pp. 35743-35752, 2018.

[14] Q. Umer, H. Liu and . I. Illahi, "CNN-based automatic prioritization of bug reports," *IEEE trans. reliab*, vol. no. 4, no. 69, pp. 1341-1354, 2020.

[15] W. Y. Ramay, Q. Umer, C. Zhu, X. U. C. Yin and . I. Inam, "Deep Neural Network-Based Severity Prediction of Bug Reports," *IEEE Access*, vol. 7, no. 2019, pp. 46846 - 46857, 2019.

[16] P. Latha and M. Marlakunta, "Predicting the Severity of Bug Reports using Classification Algorithms," researchgate.net, Bangalore, India, 2021.

[17] J. KIM and G. YANG, "Bug Severity Prediction Algorithm Using Topic-Based Feature Selection and CNN- LSTM Algorithms," *IEEE Access*, vol. 10, no. 14, pp. 94643-94651, 2022.

[18] A. B. Farid, E. M. Fathy, A. S. Eldin and L. A. Abd-Elmegid, "Software defect prediction using hybrid model (CBIL) of convolutional neural network (CNN) and bidirectional long short-term memory (Bi-LSTM)," *PeerJ Computer Science*, p. 19, 2021.

[19] M. W. Thant and N. T. Aung, "Software Defect Prediction using Hybrid Approach," in *University of Information Technology, Yangon*, Myanmar.

[20] T. Chakraborty and A. K. Chakraborty, "Hellinger Net: A Hybrid Imbalance Learning Model to Improve Software Defect Prediction," Cornell University Library, 2020 September 12.

[21] G. Y. J. KIM, "Bug Severity Prediction Algorithm Using Topic-Based Feature Selection and CNN- LSTM Algorithms," *IEEE Access*, vol. 19, no. 14 September 2022, pp. 94643 - 94651, September 2022.

[22] R. R. Panda and N. K. Nagwani, "Software bug severity and priority prediction using SMOTE and Intuitionistic fuzzy similarity measure," *Applied Soft Computing*, vol. 150, 2024.

[23] A. Yadav and S. S. Rathore, "A Hierachical Attention Networks based Model for Bug Report Prioritization," in *17th Innovations in Software Engineering Conference (ISEC-2024)*, Bangalore, India, 2024.

[24] Z. Li, G. Cai, Q. Yu, P. Liang, R. Mo and H. Liu, "Bug priority change: An empirical study on Apache projects," *Journal of Systems and Software*, vol. 212, 2024.

[25] "Java Point - Data Preprocessing in Machine Learning," [Online]. Available: https://www.javatpoint.com/data-preprocessing-machine-learning. [Accessed 25 June 203].

[26] "Gartner," 2023. [Online]. Available: https://www.gartner.com/en/information-technology/glossary/tokenization#:~:text=Tokenization%20refers%20to%20a%20process,requires%20strong%20protections%20around%20it.. [Accessed May 2023].

[27] "ScienceDirect - Pattern Recognition," [Online]. Available: https://www.sciencedirect.com/journal/pattern-recognition. [Accessed 25 June 2023].

[28] A. Simha, "Capital One - Understanding TF-IDF for machine learning," 7 October 2021. [Online]. Available: https://www.capitalone.com/tech/machine-learning/understanding-tf-idf/. [Accessed 25 June 2023].

[29] "Stanford University," [Online]. Available: https://nlp.stanford.edu/projects/glove/. [Accessed 25 June 2023].

[30] shrisikotaiah, "Geeksforgeek," [Online]. Available: https://www.geeksforgeeks.org/word-embeddings-in-nlp/#:~:text=Word%20Embedding%20or%20Word%20Vector,can%20represent%2050%20unique%20features..

[31] "IBM - RNN," [Online]. Available: https://www.ibm.com/topics/recurrent-neural-networks. [Accessed 25 June 2023].

[32] "Scholar- Google," [Online]. Available: https://scholar.google.com/scholar?q=neuron-based+biological+systems&hl=en&as_sdt=0&as_vis=1&oi=scholart. [Accessed 23 June 2023].

[33] P. Srivatsavya, "LSTM - Implementation, Advantages and Disadvantages," 5 October 2023. [Online]. Available: https://medium.com/@prudhviraju.srivatsavaya/lstm-implementation-advantages-and-diadvantages-914a96fa0acb#:~:text=Handling%20Long%20Sequences%3A%20LSTMs%20are,NLP)%20and%20time%20series%20analysis.. [Accessed 30 June 2024].

[34] M. S. a. B. A. s. H. Bani-Salameh, *e-Information Software Engineering,* vol. 15, p. no. 1, 2021.

[35] J. Kim and G. Yang, "Bug Severity Prediction Algorithm Using Topic-Based Feature Selection And CNN-LSTM Algorithm," *IEEE Access,* vol. 10, pp. 94643-94651, 2022.

[36] "java Point - Data Preprocessing in Machine Learning," [Online]. Available: https://www.javatpoint.com/data-preprocessing-machine-learning. [Accessed 25 June 2023].

Acknowledgment: There are no any special parties to acknowledge.

Financial interest: The authors declare they have no relevant financial or non-financial interests to disclose.

Conflict of interest: The authors have no conflicts of interest to declare.

AUTHOR BIOGRAPHIES

D. N. A. Dissanayake was born in June 1998. She graduated from Sabaragamuwa University of Sri Lanka, with a second upper class bachelor’s degree in Information and Communication Technology. Her recent research interests include machine learning, artificial intelligence, the development of application based on data mining.

R.A.H.M. Rupasingha received her BSc in 2013 from Sabaragamuwa University in Sri Lanka. She obtained her MSc and PhD in 2016 and 2019, respectively, from the School of Computer Science and Engineering, the University of Aizu, Japan. Currently, she is a senior lecturer in Sabaragamuwa University in Sri Lanka. Her research interests include machine learning, ontology learning, data mining and recommendation

B. T. G. S. Kumara received the bachelor’s degree in 2006 from Sabaragamuwa University of Sri Lanka. He received the master’s degree in 2010 from University of Peradeniya, Sri Lanka and he received the PhD from School of Computer Science and Engineering, University of Aizu, Japan in 2015. Currently, he is a professor in Sabaragamuwa University in Sri Lanka. His research interests include semantic web, data mining, machine learning, web service discovery and composition.

Monocular 3D Reconstruction in Poorly Visible Environments

ITA Ilesinghe[1], NLNT Lekamge[1], and GDNN Samarutilake[1#]
[1]Department of Computer Science & Engineering, University of Moratuwa, Sri Lanka:
[#]nivinya.19@cse.mrt.ac.lk

ABSTRACT 3D reconstruction of real physical environments can be a challenging task, often requiring depth cameras such as LIDAR or RGB-D to capture the necessary depth information. However, this method is resource-intensive and expensive. To counter this problem, monocular 3D reconstruction has emerged as a research area of interest, leveraging deep learning techniques to reconstruct 3D environments using only sequences of RGB images, thus reducing the need for specialized hardware. Existing research has primarily focused on environments with good lighting conditions, leaving a gap in research for environments with poor visibility. In response, we propose a solution that addresses this limitation by enhancing the visibility of images taken in poorly visible environments. These enhanced images are then used for 3D reconstruction, resulting in the extraction of more features and producing a 3D mesh with improved visibility. Our solution employs a Generative Adversarial Network (GAN) to enhance the images, providing a complete pipeline from inputting images with poor visibility to generating an output mesh file for 3D reconstruction. Through visualization of these mesh files, we observe that our solution improves the lighting conditions of the environment, resulting in a more detailed and readable 3D reconstruction.

INDEX TERMS monocular 3D reconstruction, domain adaptation, GAN, poor visibility conditions

I. INTRODUCTION

Three-dimensional reconstruction aims to recover the geometric structure of a scene or an object by leveraging the visual cues that can be observed on the entity such as perspective, shading, and texture. Along with the appropriate numerical processes, 3D reconstruction algorithms estimate the spatial layout of objects and their relative positions from these visual cues. These reconstructed 3D models act as a bridge between the physical and digital worlds; thus, they are applicable in fields such as autonomous navigation, robotics, augmented reality and virtual reality.

Recently, 3D reconstruction mechanisms have rapidly progressed with the increasing availability of visual data, improved algorithms, and the availability of powerful computational resources. Among the various approaches to 3D reconstruction, *monocular* 3D reconstruction stands out as an area of intense research interest.

Traditionally, 3D reconstruction methods have relied on depth data captured by sensors such as LIDAR or RGB-D cameras, or stereo vision or multi-view geometry to infer depth information. However, these approaches often require specialized hardware (such as stereo cameras) and precise camera calibration, limiting their practicality.

In monocular 3D reconstruction, however, the aim is to extract the structural information of a scene from single view 2D images. The challenge lies in extracting depth information from a single viewpoint, where the loss of stereo cues makes the task inherently ill-posed. Various techniques have been explored to address this challenge throughout the past few years, and methods such as structure from motion (SfM) and visual odometry (VO) have yielded acceptable results. The latest trend that has emerged is the utilization of deep learning based methods for the task of 3D reconstruction. Convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have been employed to predict depth maps directly from single images.

Most of the explored deep learning based mechanisms for 3D reconstruction have been evaluated on datasets which present well-illuminated, high resolution daytime images of scenes [1, 2], thus the visual cues are easily perceivable even by the human eye. However, in practical scenarios the visibility of scenes could be hindered by poor illumination or non-uniform lighting by multiple light sources. For example, an outdoor scene might be poorly visible due to rainy, foggy weather conditions.

A night-time outdoor scenario will consist of non- uniform lighting or will have low visibility in general. Due to these conditions, the 3D reconstruction models have difficulty in understanding the scene and rendering a proper 3D model. Therefore, these environments are referred to as complex environments [3]. Since the existing models have been developed with the assumption of consistent illumination and static scenarios, they have not been evaluated against such complex environments.

To address the above limitations, in this work we are proposing a 3D reconstruction module which can reconstruct a 3D model of environments in challenging conditions such as:

- Low lighting / visibility (e.g: Night-time)
- Multiple light sources demonstrating inconsistent lighting conditions (e.g: a night-time image of an urban road with vehicle lights / streetlights),

from a sequence of monocular images. In our proposed method, a monocular sequence of images which depict a complex environment will be converted into a more visible image through a domain adaptation network. Our focus is on night-time outdoor images and poorly lit indoor images; thus, the conversion will enhance the visibility of these images. The enhanced images will be fed into a separate 3D reconstruction model which will produce the required 3D data. The domain adaptation network is based on a Generative Adversarial Network (GAN) called AU-GAN [4] which converts the domain from night-time to daytime and the 3D reconstruction model is based on the state-of-the-art 3D reconstruction model SimpleRecon [5]. The overall network has been trained on both indoor and outdoor data with poor visibility conditions, enabling higher performance even in environments such as the above-mentioned complex ones.

Through this research, we have explored the following research objectives:

- **RO1:** Developing a framework (A basic structure for a system) that can reconstruct a 3D scene from an image sequence of a poorly visible environment.
- **RO2:** Enhance the applicability of the 3D reconstruction framework for a wider range of applications.

II. LITERATURE REVIEW

A. Domain Adaptation with GANs

Generative Adversarial Networks (GANs), particularly Deep Convolutional GANs (DCGANs), have significantly advanced image generation in artificial intelligence. DCGANs utilize deep convolutional neural networks to capture intricate features and spatial relationships, enhancing image realism. They have demonstrated their capability in learning hierarchical representations from object parts to full scenes, using techniques like batch normalization to stabilize training and mitigate issues like mode collapse. These networks, trained on large-scale image datasets such as Imagenet-1k, are adept at generating high-quality visual samples. A critical application of GANs is image-to-image translation, transforming an input image into a corresponding output while preserving essential visual characteristics. This task, essential for image enhancement, style transfer, synthesis, and editing, can be approached in supervised, unsupervised, or semi-supervised ways. Supervised methods, like pix2pix [6], require paired examples, whereas unsupervised methods, such as CycleGAN [7] and UNIT [8], aim to learn the mapping without explicit supervision, simplifying data collection. UNIT (Unsupervised Image-to-image Translation) introduces a shared-latent space assumption, suggesting that images from different domains can be mapped to a common latent representation. This approach uses a combination of GANs and variational autoencoders (VAEs) to model each image domain, incorporating weight-sharing and adversarial training to enforce the shared-latent space. The UNIT framework also addresses domain adaptation, achieving high accuracy on benchmark datasets. By integrating the cycle-consistency constraint, UNIT ensures a robust mapping between domains, facilitating realistic image translations in various challenging tasks, such as street scene transformations, synthetic to real image translation, and facial attribute modifications. The framework demonstrates proficiency in handling diverse and complex image translation tasks, producing visually realistic results even in scenarios with substantial domain differences.

Domain translation between challenging conditions like night-time and standard daytime poses significant challenges in unsupervised or weakly-supervised learning due to the impracticality of obtaining precisely aligned ground-truth image pairs, especially in dynamic driving scenes with numerous moving objects. Visual variations across different weather conditions, such as vehicles and streetlamps, along with global texture differences like raindrops and regional changes such as reflections on wet roads, further complicate the problem. Despite these variations, a commonality in semantic and geometrical aspects exists between adverse and normal domains. The primary objective of a general night-to-day domain adaptation model is to disentangle invariant and variant features without relying on supervision or task-specific knowledge.

Optimal task-agnostic image translation should preserve image content at all scale levels, from overall scene layout to intricate object details, while dynamically adapting to varying illumination and weather conditions. CycleGAN-based models such as [7] demonstrate effectiveness in altering global conditions but often fail to preserve local feature details. ForkGAN [9] addresses this limitation by coupling two encoding spaces of CycleGAN to retain invariant information in both domains. ForkGAN enforces domain agnosticism by ensuring that encoded features do not reveal their domain of origin, introducing a 'Fork' branch to assess the sufficiency of encoded information for reconstructing original image data in both domains.

ForkGAN introduces a fork-shaped architecture for image translation using unpaired data, featuring one encoder and two decoders. For example, in night-to-day translation, a night-time image is encoded to extract a domain-invariant representation, which is then processed by two decoders: one reconstructs the original night-time image, and the other generates a plausible daytime image. Adversarial training and a perceptual loss ensure content representation consistency between the original and translated images. ForkGAN's architecture enhances image recognition tasks in both domains by ensuring retention of essential information.

Experiments using the Alderley and BDD100K [10] datasets demonstrate ForkGAN's efficacy. The Alderley dataset, designed for the SeqSLAM algorithm [11], includes images captured along the same route under different conditions, while the BDD100K dataset contains annotated high-resolution images from diverse cities and environmental conditions. ForkGAN achieves superior or comparable results to methods like UNIT [8], CycleGAN [7], MUNIT [12], and StarGAN [13] in localization, semantic segmentation, and object detection tasks.

Conventional symmetric architectures like those in CycleGAN-based approaches struggle with adverse domain translation due to significant domain gaps. Rainy night images, with artifacts, blur, and reflections, necessitate an asymmetric approach. AUGAN [4] proposes an asymmetric architecture with a feature transfer network between the encoder and decoder, enhancing encoded features from adverse domain images.

An asymmetric feature matching loss aids in disentangling domain-invariant from domain-specific features. AUGAN also introduces an uncertainty-aware cycle-consistency loss to mitigate artifacts in adverse domains, penalizing regions based on a confidence map.

AUGAN’s asymmetric framework excels in adverse weather translation tasks on the Alderley and BDD100K datasets. It consistently produces superior visual results, outperforming models like CycleGAN [7], TodayGAN [13], and ForkGAN [9], especially in dark or blurry areas.

AUGAN's robust performance is attributed to its innovative approach to feature enhancement and disentanglement, ensuring well-preserved objects and high-quality transformations across various challenging conditions.

B. *3D Reconstruction Methods*

When considering 3D reconstruction techniques, including stereo reconstruction, multi-view stereo (MVS), volumetric reconstruction, structure from motion (SfM), and deep learning-based methods have been extensively studied. Recently, the application of sparse truncated signed distance function (TSDF) for 3D reconstruction has shown enhanced performance and accuracy.

NeuralRecon [14] is a neural network that processes a sequence of images from a moving camera and their corresponding camera poses to generate a 3D representation of the scene as a TSDF volume. It reconstructs and fuses sparse TSDF volumes incrementally using sparse 3D convolutions and gated recurrent units (GRUs). Unlike methods that estimate single-view depth maps and fuse them later, NeuralRecon directly reconstructs local surfaces for each video fragment, ensuring global consistency and eliminating redundant computations. This results in dense, accurate, and coherent 3D scene geometry while maintaining real-time performance. NeuralRecon captures both local smoothness and global shape priors, achieving real-time performance at 33 key frames per second, significantly faster than previous methods like Atlas [15].

TransformerFusion [16] employs a transformer-based approach for 3D scene reconstruction by fusing monocular RGB video frames into a volumetric feature grid. The transformer architecture allows the network to attend to the most relevant image frames for each 3D location, enhancing surface reconstruction accuracy. The coarse-to-fine formulation of transformer-based feature fusion improves both reconstruction performance and runtime. FineRecon [17] addresses the challenge of coarse and detail-lacking 3D reconstructions with a depth-aware, end-to-end network. By using posed RGB images and a depth-prediction network to guide back-projection, FineRecon achieves significant improvements across various depth and 3D reconstruction metrics, outperforming other state-of-the-art methods. However, its computational efficiency is lower compared to NeuralRecon, and the requirement for camera poses adds complexity to its usage.

The SimpleRecon approach [5] presents a novel method for 3D indoor scene reconstruction by enhancing multi-view depth prediction quality instead of direct 3D volumetric reconstruction.

This method integrates keyframe and geometric metadata into the 4D cost volume, allowing for informed depth plane scoring. It employs a 2D Convolutional Neural Network (CNN) that leverages strong image priors and geometric losses, enabling real-time, low- memory reconstruction. SimpleRecon's results demonstrate a considerable lead over current state-of-the-art methods for depth estimation, showing close or better performance on standard datasets like ScanNet [1] and 7-Scenes.

III. METHODOLOGY

A. *Method Overview*

The proposed method consists of two main components: a generative adversarial network (GAN) that can transform images that are taken in the presence of challenging lighting conditions such as poor visibility or artificial lighting at night (referred to as "night images") into images with clear contrast and color range (referred to as "day images"), and a neural network that can reconstruct 3D surfaces from monocular day image sequences.

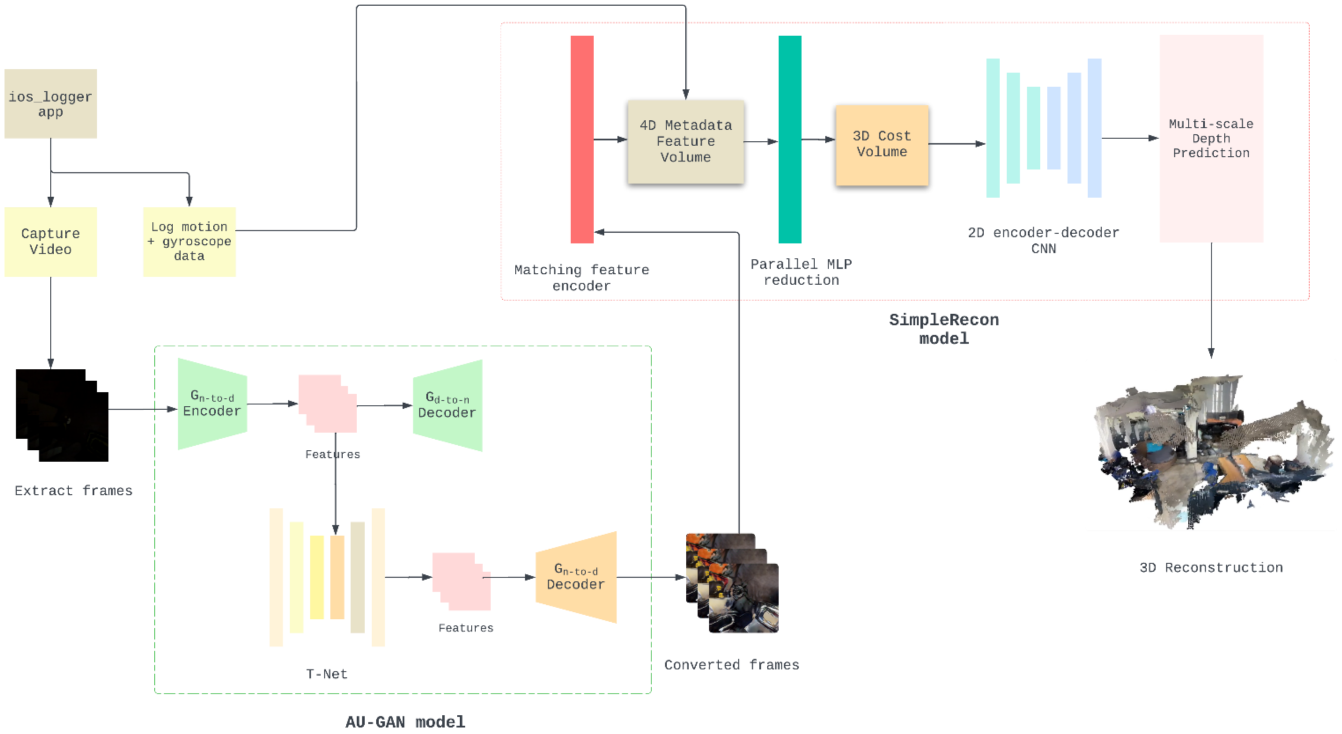


Figure 1 Architecture of Our Implementation

The first component is a GAN that can learn to map night images to day images in an unsupervised manner. The GAN consists of two networks: a generator and a discriminator. The generator tries to produce realistic day images from night images, while the discriminator tries to distinguish between real and fake day images. The GAN is trained on a large dataset of unpaired night and day images, both indoor and outdoor, collected from various sources.

As there is no established dataset for this task, a dataset created comprising of natural night-time/poorly visible environments was needed. The GAN is expected to capture the poor lighting conditions and color variations in night scenes and to generate natural-looking day images that preserve the scene geometry and semantics.

The second component is a 3D reconstruction model comprising of a neural network that can reconstruct 3D surfaces from a sequence of monocular images. The model takes as input a sequence of images captured by a moving camera and outputs a 3D representation of the scene in the form of mesh reconstruction.

The final step is to combine the two components to achieve 3D reconstruction from monocular video of complex environments. The idea is to first apply the GAN to convert the night video into a day video, and then feed the day video to the neural network to obtain the 3D reconstruction. The advantage of this approach is that it leverages the existing methods for reconstructing day scenes, which are more mature and robust than the methods for reconstructing night scenes and avoids the challenges of dealing with complex lighting and shadows in night scenes.

B. *Model Selection*

For the base models of the 2 main components mentioned above, we tested several existing models and selected the following:

- Domain adaptation network - AU-GAN [4]
- 3D reconstruction model - SimpleRecon [5]

For the domain adaptation network, we tested ForkGAN [9], CycleGAN [7], AU-GAN [4], and UNIT [7]. Out of them, we have selected AU-GAN as the base model for the domain adaptation network.

Unlike symmetric approaches like ForkGAN [9], which struggle with the pronounced domain gap between standard and adverse weather conditions, AUGAN introduces a novel framework adept at handling rainy night images replete with artifacts, blur, and reflections. By incorporating a feature transfer network exclusively within the generator responsible for adverse domain translation, AUGAN enhances and disentangles features crucial for domain translation without compromising on local object details. Moreover, its incorporation of an uncertainty-aware cycle consistency loss, inspired by uncertainty modeling, ensures better preservation of details in dark or blurry regions, addressing a common shortfall of other models like CycleGAN and ForkGAN. Through comprehensive qualitative and quantitative evaluations, AUGAN consistently outperforms its counterparts, showcasing superior visual quality and robustness in adverse weather translation tasks across diverse outdoor datasets, thereby solidifying its status as the leading choice for night-to-day conversion endeavors.

For the base of the 3D reconstruction model, the other frameworks tested were NeuralRecon [14], TransformerFusion

[16], and FineRecon [17]. The SimpleRecon model is a comparatively newer 3D reconstruction model which provided excellent results in a small amount of time. It outperformed the other 3 models in reliability and efficiency, although there is a small tradeoff on accuracy, as FineRecon has provided better accuracy. However, FineRecon model takes up a considerably larger amount of time and computational resources than SimpleRecon to produce the final result. Therefore, after carefully considering our requirements and cost-effectiveness, SimpleRecon was selected as the base model for the 3D reconstruction component of our pipeline.

After selecting the base models, we made several modifications to the domain adaptation model to produce a convincing daytime image, when a night-time image is input. One significant improvement involved architectural adjustments of the base AUGAN model, such as the adoption of demodulated convolutions and the use of upsample-plus-convolution operations instead of transposed convolutions, which were instrumental in stabilizing training and mitigating artifacts commonly associated with GANs, such as droplet and checkerboard artifacts. Furthermore, augmenting the training data, particularly with the BDD100K dataset [10], mitigated biases towards specific environmental features like trees and sky prevalent in the original dataset. This augmentation strategy aimed to enhance the model's ability to generalize across diverse environments and scenarios, addressing the second research objective we were aiming for.

Through architectural refinements and data augmentation strategies, the model's robustness and applicability across various environmental conditions were improved, paving the way for more effective image-to-image translation tasks in both research and practical applications.

C. *Pipeline*

Figure 1 denotes the final architecture of our implementation. This method relies on the input from an iOS application called ios_logger which can capture a video whilst logging the motion information corresponding to the video frames. This application was introduced and utilized in NeuralRecon [14] for monocular video recording purposes. The captured video frames are extracted and fed into the domain adaptation model component, which would generate the frames with enhanced visibility. These generated images are next fed into the 3D reconstruction model component for the 3D reconstruction, along with the motion data captured earlier. This component outputs the 3D mesh file for the input image sequence.

D. *Datasets*

The research project necessitates datasets suitable for training and evaluating both a domain adaptation network and a 3D reconstruction model. Selection criteria were established, including the requirement for RGB-D images of real-world indoor/outdoor scenes with complex conditions, availability of camera pose data or related information, presence of sequential images with corresponding labels and semantics, and preference for minimal dynamic components.

Despite thorough searching, no single dataset met all the criteria. However, three datasets were selected:

- BDD100K [10]: Originally designed for autonomous driving algorithm evaluation, this dataset comprises over 100,000 videos with high-resolution images captured under various weather conditions and in night-time conditions. While it lacks some complex conditions, it offers diverse scenarios for certain tasks like object detection and semantic segmentation.

- ScanNet [1]: Primarily used for indoor scene understanding, ScanNet provides RGB-D video data of indoor environments with annotations like 3D camera poses and semantic segmentations. Although it lacks complex conditions, its static environments align with the requirement for minimal dynamic components.

- Custom Dataset: To address the absence of datasets for poorly lit indoor environments, a small custom dataset was created using an iPhone 15 Pro and the ios_logger app. This dataset captures indoor locations of a university during night-time, focusing on areas like hostels, study spaces, and parking lots. Some daytime captures were also included for comparison with night-time reconstructions, providing insights into differences between lighting conditions.

Each dataset offers unique strengths and limitations, fulfilling specific criteria outlined for the research project.

E. *Implementation Details*

The training of the GAN and the implementation of the pipeline was done in 2 separate GCP VMs. Each GCP VM had a NVIDIA V100 GPU. As for the training time, it took around 10 hours to train 1 epoch of the GAN model. The resolution of the images was downsized to 256 x 256 for training. Each epoch had more than 100,000 iterations.

IV. RESULTS & DISCUSSION

A. *Domain Adaptation*

The Fréchet Inception Distance (FID), or FID score, introduced by Heusel et al. [18] improves the Inception Score. FID utilizes the Inception v3 model, specifically the final pooling layer before image classification, to capture important image features. By calculating activations for both real and generated images in this layer, FID forms multivariate Gaussian distributions. The Fréchet distance (Wasserstein-2 distance) measures the divergence between these distributions. A lower FID score indicates that generated images more closely match the statistical properties of real images, signifying higher fidelity.

Table 1. FID-scores comparison

Model	**Train Dataset**	**Evaluation Dataset**	**FID-Score**
Original AU-GAN	bdd100k	bdd100k	45.6886
Original AU-GAN	bdd100k	Custom indoor dataset	369.5709
pt-AUGAN	bdd100k	bdd100k	120.9566

pt-AUGAN	bdd100k-augmented	bdd100k	113.7694
pt-AUGAN	bdd100k-augmented	Custom indoor dataset	180.9832

The provided Table 1 shows that the original AU-GAN trained on the BDD100K dataset achieved an FID score of 45.6886 when evaluated on both the BDD100K dataset itself and a custom indoor dataset.

In contrast, the pt-AUGAN which is the improved model, consistently showed higher FID scores than the original AU-GAN. Specifically, when tested on the BDD100K dataset, the pt-AUGAN achieved an FID score of 120.9566, reflecting a decline in performance. However, when the pt-AUGAN was evaluated on an augmented version of the BDD100K dataset ("bdd100k-augmented"), it performed slightly better, with an FID score of 113.7694.

In summary, the pt-AUGAN exhibits mixed performance compared to the original AU-GAN. While there are some indications of improvement in certain scenarios, it also shows notable performance degradation in others. Further evaluation is necessary, especially regarding its performance on a wider range of datasets, and that currently remains as a future work.

B. 3D Reconstruction

We randomly picked 16 scans from the Scannet dataset [1] as the test set for evaluating our pipeline. Each of these scans represents an indoor environment image sequence, containing around 600-3000 images (the number of images vary). Table 2 provides results obtained and the averaged values are provided in Table 3.

The method of evaluation was to compare the 3D reconstruction from the night-time image sequence itself (without any domain adaptation), and the 3D reconstruction generated from our improved pipeline with domain adaptation. These 2 types of 3D reconstructions are referred to as ‘night-time mesh’ and ‘daytime mesh’ respectively. First, the night-time mesh was evaluated against the ground truth mesh, and the precision, recall, f-score metrics were obtained for it. Next, the same metrics were obtained for the daytime mesh, comparing it against the same ground truth mesh. This evaluation method provides a relative understanding of how well the 3D reconstruction could be done on a night-time environment (an environment in poorly visible conditions) as it is, and how much it could be improved by employing our solution instead.

Table 2. 3D reconstruction evaluation

Scan scene no.	**Night-time mesh (w/o domain adaptation)**			**Daytime mesh (with domain adaptation)**		
	precision	**recall**	**f-score**	**precision**	**recall**	**f-score**
0025	0.219	0.239	0.229	0.284	0.313	0.298
0046	0.236	0.257	0.246	0.221	0.237	0.229
0068	0.296	0.334	0 .313	0.366	0.333	0.349
0167	0.220	0.251	0.235	0.237	0.272	0.253
0257	0.173	0.183	0.178	0.248	0.276	0.261
0303	0.318	0.317	0.318	0.246	0.233	0.239
0325	0.339	0.346	0.343	0.288	0.306	0.296
0428	0.176	0.190	0.182	0.162	0.157	0.159
0642	0.243	0.261	0.252	0.298	0.311	0.304
0715	0.165	0.176	0.171	0.149	0.143	0.146
0725	0.231	0.239	0.235	0.214	0.212	0.213
0737	0.215	0.213	0.214	0.300	0.301	0.301
0746	0.299	0.311	0.305	0.273	0.278	0.275
0761	0.201	0.220	0.210	0.249	0.252	0.251
0780	0.183	0.180	0.182	0.225	0.227	0.226
0795	0.159	0.148	0.153	0.177	0.194	0.185

Table 3 shows the averaged values of the above evaluation results.

Table 3. Summary of 3D reconstruction evaluation

Night-time mesh			**Day time mesh**		
precision	recall	f-score	precision	recall	f-score
0.229	0.241	0.235	0.246	0.252	0.249

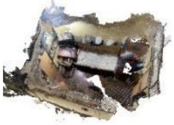
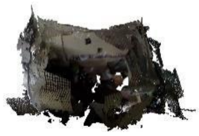
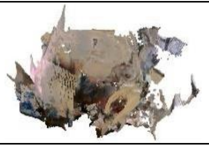
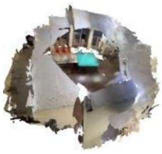
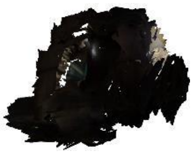
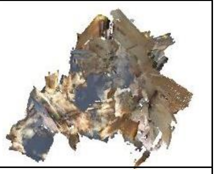
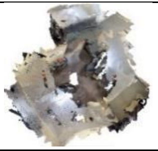
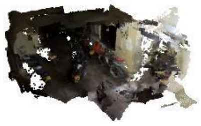
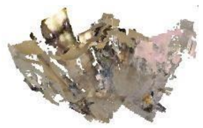
From these results it can be concluded that there is relatively little accuracy improvement in the reconstructed meshes of our pipeline. There are various factors that affect these results as we have identified:

- The density of the ground truth mesh and the density of the predicted mesh are vastly different. The ground truth mesh is an extremely dense reconstruction, whereas our method produces a comparatively sparse mesh. This could be the main reason affecting the low measurements of accuracy when it comes to the surface distance metrics.

- The FID score of the GAN model is high due to the high resolution of the images and the unavailability of the two domains (night and day) of the same indoor environments. Although we retrained the model with new indoor environment images from the custom dataset, the amount of training data and time seems to be insufficient for the modified AU-GAN model to produce a convincing result.

- Due to the lack of ground truth data, we converted an existing day-time image dataset into night-time images. The resulting night-time images, in some cases, were not passable as convincing captures of a night-time environment. The poor quality of these night-time images may have resulted in a poor-quality output of the predicted mesh. Table 4 shows visual results comparison.

Table 4. Visualization of 3D reconstructions

Original	Night-time reconstruction	Reconstruction on our method
		
		
		

V. FUTURE WORK

One of the main improvements that can be made to our method is the fine tuning of the GAN component to be inclusive of night-time indoor environments. Although we attempted this improvement, the scale of our custom dataset was not sufficient for the model to produce a satisfactory result. This leads to the requirement of a dataset which includes images of night-time indoor environments and corresponding ground truth data. Also, using region-based spatial attention methods with the GAN will reduce the bias introduced by the dataset.

To further elaborate, our suggestion would be to develop a dataset which has an equal distribution of image sequences of night-time and daytime environments, both indoor and outdoor. The same location must be captured in both high-visibility and low-visibility conditions using the exact same setup of cameras, along with the ground truth data. The dataset can be created by compiling a large number of such scans. If this dataset can be created, it would benefit the further training and evaluation tasks of our pipeline.

The generated day-time images have artefacts in them, which is an additional noise that hinders the accuracy of the final output. These can be reduced by optimizing the GAN further with more experimentation.

VI. CONCLUSION

Monocular 3D reconstruction seeks to overcome the limitations posed by the need for specialized hardware and precise calibration, by extracting structural information from single 2D images, a task that is inherently ill-posed due to the loss of stereo cues. Recent advances in deep learning have significantly improved the performance of monocular depth estimation and 3D reconstruction. Despite these advancements, most existing deep learning-based methods have been trained and evaluated on datasets that assume well-illuminated, such as daytime environments with consistent lighting conditions. To address these limitations, our research proposes a novel 3D reconstruction framework capable of handling complex environments characterized by challenging lighting conditions. Our approach leverages a sequence of monocular images and utilizes a domain adaptation network to enhance image visibility before feeding them into a 3D reconstruction model. This method shows slight improvements against 3D reconstructions done on the captured night-time environment itself. However, our solution can be further improved with the availability of a night-time environment dataset which includes ground truth data.

Our approach addresses the need for a more generalized and adaptable 3D reconstruction model. By training our system on diverse datasets that include both indoor and outdoor scenes with varying lighting conditions, we enhance its ability to generalize across different environments. This versatility is crucial for cost-effective applications in autonomous navigation, robotics, augmented reality, and virtual reality, where the ability to accurately reconstruct 3D environments in real-time under various conditions is essential.

REFERENCES

[1] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser and M. Nießner, "ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes," in *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[2] A. Geiger, P. Lenz, C. Stiller and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, p. 1231–1237, 2013.

[3] C. Zhao, Y. Tang and Q. Sun, "Unsupervised monocular depth estimation in highly complex environments," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 6, p. 1237–1246, 2022.

[4] J.-g. Kwak, Y. Jin, Y. Li, D. Yoon, D. Kim and H. Ko, "Adverse weather image translation with asymmetric and uncertainty-aware GAN," *arXiv preprint arXiv:2112.04283*, 2021.

[5] M. Sayed, J. Gibson, J. Watson, V. Prisacariu, M. Firman and C. Godard, "SimpleRecon: 3D Reconstruction Without 3D Convolutions," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[6] P. Isola, J.-Y. Zhu, T. Zhou and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[7] J.-Y. Zhu, T. Park, P. Isola and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017.

[8] M.-Y. Liu, T. Breuel and J. Kautz, "Unsupervised image-to-image translation networks," *Advances in neural information processing systems*, vol. 30, 2017.

[9] Z. Zheng, Y. Wu, X. Han and J. Shi, "ForkGAN: Seeing into the Rainy Night," in *The IEEE European Conference on Computer Vision (ECCV)*, 2020.

[10] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan and T. Darrell, *BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning*, 2020.

[11] M. J. Milford and G. F. Wyeth, "SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights," in *2012 IEEE international conference on robotics and automation*, 2012.

[12] X. Huang, M.-Y. Liu, S. Belongie and J. Kautz, *Multimodal Unsupervised Image-to-Image Translation*, 2018.

[13] A. Anoosheh, T. Sattler, R. Timofte, M. Pollefeys and L. V. Gool, *Night-to-Day Image Translation for Retrieval-based Localization*, 2019.

[14] J. Sun, Y. Xie, L. Chen, X. Zhou and H. Bao, "NeuralRecon: Real-Time Coherent 3D Reconstruction from Monocular Video," *CVPR*, 2021.

[15] Z. Murez, T. van As, J. Bartolozzi, A. Sinha, V. Badrinarayanan and A. Rabinovich, "Atlas: End-to-End 3D Scene Reconstruction from Posed Images," in *Computer Vision – ECCV 2020*, Cham, 2020.

[16] A. Bozic, P. Palafox, J. Thies, A. Dai and M. Nießner, "Transformerfusion: Monocular rgb scene reconstruction using transformers," *Advances in Neural Information Processing Systems,* vol. 34, p. 1403–1414, 2021.

[17] N. Stier, A. Ranjan, A. Colburn, Y. Yan, L. Yang, F. Ma and B. Angles, "FineRecon: Depth-aware Feed-forward Network for Detailed 3D Reconstruction," in *ICCV*, 2023.

[18] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems,* vol. 30, 2017.

[19] T. Karras, S. Laine and T. Aila, *A Style-Based Generator Architecture for Generative Adversarial Networks,* 2019.

ACKNOWLEDGMENT

We would like to extend our gratitude to the Computer Science and Engineering Department at the University of Moratuwa for providing us with the opportunity to undertake this project and for their resources and facilities that made this research possible. We are deeply thankful to all those who directly or indirectly contributed to this project. Their support and encouragement were essential and greatly appreciated.

Enhancing Human-Computer Interaction on Educational Websites: Color Preferences for 7-8 Years Old

DMS Sathsara [#1], **DVDS Abeysinghe** [1], **and KGK Abeywardhane** [1]

[1]Department of Computer Science, Faculty of Computing, General Sir John Kotelawala Defence University, Ratmalana, 10390, Sri Lanka.

[#]sathsara.dms.@kdu.ac.l

ABSTRACT For any application development first impression will always be a matter to attract users. User Interfaces (UI) will be the first handshake of an application with the user. This concern brings more impact when creating applications for children for educational purposes. In that case, having a vibrant and playful UI will be supporting to spark joy in every click of an application. The study aims to evaluate the impact of the effective selection of colors in educational website UI designs. Initially, the study conducted a comparative analysis of color preferences among 7-8-year-old students, aiming to identify the most preferred colors. A systematic color selection process was employed by gathering data from both primary and secondary data sources, resulting in 220 data from primary and a sample population of 323 data from secondary. Based on the findings, the most preferred colors among 7-8-year-old students were identified as red, yellow, green, blue, and purple. Then, the obtained color preferences were used in designing an UI for an educational website. The newly created design was then compared with three existing websites to evaluate the attractiveness of UI as an educational website. Finally, the study has concluded that a successful color selection in the UI design will enhance the UI, as it was proven by identifying the newly created UI design with the most preferred colors of the 7-8 years old as the most liked with 52.9% of positive feedback during the post surveys conducted to validate the aim of the study.

INDEX TERMS Applications for children, Color preferences, Human Computer Interaction, UX/UI designing

I. INTRODUCTION

"*It's not enough that we build products that function, that are understandable and usable, we also need to build products that bring joy and excitement, pleasure and fun, and, yes, beauty to people's lives*." - **Don Norman**

User interface (UI) is where human-computer interactions (HCI) take place and it includes all user-interactive elements, such as buttons, pages, screens, icons, and more. A positive user experience can only be achieved with a well-designed user interface. To give a better user experience color combinations on an interface can be influential, especially when creating interfaces for young children. To design effective user interfaces that satisfy children's wants and preferences, it is essential to comprehend children's color preferences and their potential influence on user engagement and interaction. This study intends to investigate how youngsters between the ages of 7 and 8 feel about color preferences and User Experience (UX) for UI designs.

The concept of e-learning is a dominant point of current centuries. Student Content Interaction (SCI) is a major part of the HCI. It helps to keep the student interaction for the e learning platforms in the educational domain. User satisfaction is the final output which expect from any product. This is belonging to the Web UI concept as well [1].

The importance of color associations and their impact on children's emotional and cognitive responses have been addressed in earlier research [2]. The study has stated for instance that, yellow has been proven to be both highly favoured by 7-year-olds and to have a calming impact on kids with respiratory disorders like asthma. Additionally, fourth graders have been identified color yellow with honesty, demonstrating the complex nature of color preferences in many settings. Red color, on the other hand, has been associated with improved motor skills as well as elevated blood pressure and respiration rate. Girls respond to color red more favourably than boys do, evoking sentiments of exhilaration and passion. Children with sensory impairments may benefit especially from blue color, which is a popular color among 7 to 11-years-old and well known to have a soothing effect on the respiratory and cardiac systems.

Children prefer hues like red, yellow, green, blue, purple, violet, and orange, according to UI evaluations and research [3-5]. Age-related differences in color preferences have been observed, with color orderings for various age groups. For instance, blue is the color that 7-year-olds enjoy most, followed by red, yellow, violet, orange, and green colors [4].

Cool hues like purple, blue, and green have been proven to affect attentiveness similarly, but red has a detrimental impact on children's attention [5]. Hence, the influence of colors on student attentiveness has been investigated through numerous studies as explained above. Moreover, gender distinctions have been observed to possess a relationship with color preferences. In the study [6], the above factor has been proved as it has identified girls rating for colors such as pink and purple whereas boys do the opposite.

By understanding these color preferences and their psychological associations, UI designers can create interfaces that are visually appealing, engaging, and tailored to children's needs.

This research paper contribute to the existing body of knowledge by investigating the specific color preferences and their potential impact on children's UI experiences. The study has aimed to evaluate the impact of the effective selection of colors in educational website UI designs rather than designing UIs without considering the color preferences. The authors evaluate the color preferences to keep the interaction of 7- to 8-year-old student for educational websites. Sample population and the educational websites are selected from the Sri Lankan context.

Most students prefer using online materials for learning. Among these, websites are a popular medium used by 7- to 8-year-old students for their educational activities. Therefore, creating websites with colors that appeal to children can further enhance their use of educational sites.

II. LITERATURE REVIEW

The relationship between UX/UI and color in creating rich interfaces has been evaluated in many aspects through existing works. Further, this study has been conducted as a part of the previously conducted work by one of the authors. In that study [7], the author has conducted an overall study about several factors such as color, fonts, user behaviour, user interactions, and gestures affecting the UX/UI designing under the field of HCI. In this section, the previously conducted study of the author and a few existing works will be highlighted.

In [7], the authors have investigated the user interactions for educational websites for the age group of 7-8 years old. The study has investigated about five factors namely as color, fonts, user behaviour, user interactions and gestures. Three existing local websites were considered, and the factors were evaluated. As the results, the study has concluded when designing interfaces for children to keep them engaged and to enhance the UX, designers should properly identify the preferences for each factor for the targeted audience. Hence, the study has emphasized the room for further studies to investigate about these factors one by one.

The study conducted by C. Llinares et. Al [8], has focused on the usage of colors, and warm and cold hues in the virtual classroom environments and the impact of that for education. The investigation has been conducted in two main aspects through psychological and neurophysiological methods. Psychological testing has been conducted with attention and memory tasks where neurophysiological methods have been monitored through heart rate variability and electroencephalogram. Hence collected factors have been analysed to relate to the cognitive functions. Through the study, authors have found that cold hue colors not only improve attention and memory function but also raise arousal and facilitate the formulation of design principles. Furthermore, correlations between the psychological and neurophysiological measures were discovered, which represents a remarkable advancement in the field of neuroarchitecture. Moreover, the study has emphasized the use of the impact of colors properly in architectural designs for educational environments will be a crucial factor in learning settings.

In [9], the study has discussed about the term ‘color symbolism’. It has shown the impact the colors in thinking and decision making. Further, the study has shown that color preferences often reflect gender significance. The study shows that some colors were categorized as girls’ colors and some as boys’ colors which clearly defines the gender significance. Moreover, the study has shown that color impacts for the society, preferences, performances, environment, and emotions. This clearly shows that having suitable colors in the learning environments are highly impactful.

The study [10], has discussed about color as one of the essential elements of design as it has a direct impact on children's psychology and behaviour. The authors have selected a color range as orange, yellow, green, blue, purple, and pink, and have investigated about the preferences of grade 1 and 2 students. As results, study has observed blue and orange colors as the most favourable impacts on children. With blue color creativity and artistic skills were found to be improved while orange and yellow colors were identified to support logical rational thinking associated with mathematics. Moreover, purple color has been identified as a balanced color in both logical and creative aspects with the research. Another significant finding of the study was that blue color had an impact on attendance. Hence, the study overall shows the impact of the colors in the learning spaces including environment, materials, and resources.

In [11], the study has discussed the role of colors in establishing a visual hierarchy within a design. The study has cleared stated that the colors are not just only an aesthetic element, and it also can be used as a strategic tool in communication and emotional resonance. Further, the study has emphasized that users can easily learn and navigate through a clear visual language which is supported by a well-defined color scheme. Warm hues like red and yellow, according to the study, can arouse feelings of urgency and energy, which makes them appropriate for aspects that call for user action. Additionally, colder colors like blues and greens can have a calming impact. This makes them useful for backgrounds or spaces where people are supposed to unwind and take in information. Authors have also stated that utilizing color psychology can be instrumental in designing compelling and successful user interfaces that evoke feelings and actions from users is a valuable technique. In concluding remarks, the study has shown that utilizing color to highlight the most significant and pertinent aspects of the user interface is important, and a visual hierarchy and contrast should be established. Also, the colors can be used to evoke strong feelings in users and to build an emotional connection with the design. Further, making use of color to appeal to the target audience and establish cultural and personal significance and using it to make users happy is also important in UX/UI designs.

The study [12], has directly discussed colors in UIs. The study has shown that the main factor to be considered when selecting colors to the designs should be to make user-friendly interactive interfaces. That is because color is a powerful tool that a designer can use and influence the user with easy visuals. The study also stated that colors will support to distinguish and separate objects. Hence, the study has concluded that color plays a vital role in UX/UI design and it a highly challenging resource to utilize.

The article [13], has discussed color psychology. The author has stated that UI/UX designers may use color psychology as a potent tool to craft memorable and compelling digital experiences. Through a proper understanding about the colors and the emotional effects the UIs can be designed to arise the favourable feeling and encourage user interactions. The article has discussed about colors and user engagement and color associations with UX. In user engagement, it has been identified that color is a direct impact, and the vision can be attracted easily with good and pleasing colors. In terms of user experience the behaviours of the users and satisfaction can be impacted with colors. Overall, the article has clearly shown the importance of colors, and using it to promote psychological aspects in a UI design is tactful.

Hence from the above highlighted works, it has clearly identified that colors play a vital role in human life impacting the psychological aspects of humans. In designing UI to provide a rich UX colors can be used as a powerful tool. With the psychological impact of colors, it is efficient to use it, especially in educational aspects. Overall, the review has emphasized that the effective use of colons in educational application UI design will play a pivotal role.

III. METHODOLOGY AND EXPERIMENTAL SETUP

This study's primary goal is to investigate how color affects educational websites, with a particular emphasis on kids between the ages of 7 and 8. This age group is perfect for the research because they consider a lot about the looks and comfort of their learning environment. These kids also possess improved memory and the mental capacity to classify items according to their size, shape, and color. Understanding color's impact is crucial for creating captivating and educationally useful websites that may improve learning and promote cognitive development in this delicate age group since color plays a significant part in a child's visual environment [7], [14-15].

The overall methodology has been done following several sub-processes including data gathering, data analysis, UI designing, validation, and the result analysis as shown in Figure 1 below.

As a beginning of the methodology, data gathering was done in two methods as primary data gathering (pre-survey) and the literature review. It was observed that depending only on primary data will not be sufficient to make the decisions. Hence, the study also conducted a sound literature review referring to the existing work regarding the color preferences of the targeted student groups.

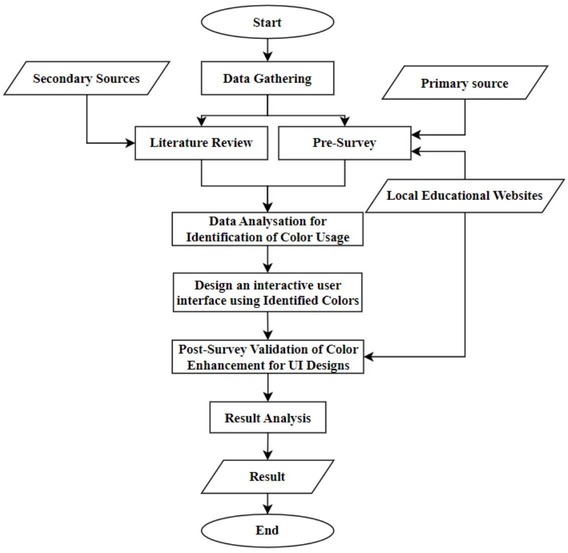
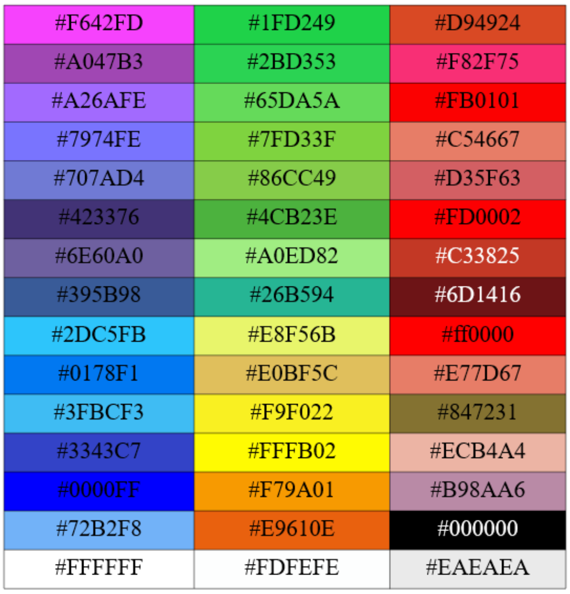

Start
Secondary Sources
Data Gathering
Primary source
Literature Review
Pre-Survey
Local Educational Websites
Data Analysation for Identification of Color Usage
Design an interactive user interface using Identified Colors
Post-Survey Validation of Color Enhancement for UI Designs
Result Analysis
Result
End

Figure 1. Methodology of the Study
Source: Authors

A. Data gathering and preprocessing.

1) Primary data gathering (pre-Survey): In the study, a pre-survey has been conducted to gather the data to identify the impact of colors on user experience and preference changes. Structured interviews were conducted as questionnaires with the selected participants of the age group 7-8 years old in identifying their color preferences. The questionnaire was used to learn in-depth about how children of the age group interact with instructional websites and to get their opinions on color components in existing websites. To ensure uniformity and comparability, a common operating protocol was adhered to during each data collection session. During the interviews, audio-visual recordings were used to guarantee reliable data acquisition. The comments, gestures, and nonverbal clues made by the participants were reviewed and analyzed later with the recordings. 45 existing colors were identified from the secondary data sources as shown in Figure 2 below. Evaluating the identified colors the questionaries were generated with the support of the selected 3 websites.



#F642FD	#1FD249	#D94924
#A047B3	#2BD353	#F82F75
#A26AFE	#65DA5A	#FB0101
#7974FE	#7FD33F	#C54667
#707AD4	#86CC49	#D35F63
#423376	#4CB23E	#FD0002
#6E60A0	#A0ED82	#C33825
#395B98	#26B594	#6D1416
#2DC5FB	#E8F56B	#ff0000
#0178F1	#E0BF5C	#E77D67
#3FBCF3	#F9F022	#847231
#3343C7	#FFFB02	#ECB4A4
#0000FF	#F79A01	#B98AA6
#72B2F8	#E9610E	#000000
#FFFFFF	#FDFEFE	#EAEAEA

Figure 2. Identified colors from the secondary data gathering.
Source: Authors

Data collecting was facilitated by a data gathering scale, as Grade 3 students had weak writing abilities. This Likert scale, as shown in Figure 3, was included in the questionnaire to gather the responses from students. Students responded by identifying their choices on a five-point rating scale: 5 for excellent, 4 for good, 3 for average, 2 for terrible, and 1 for extremely dissatisfied. A color-coded method was implemented to improve clarity even further. A green picture was allocated to a question that received a higher preference rating from students (5 out of 5). A standardized evaluation of the student's preferences and perceptions was made possible by the constant application of this approach to all their replies. Further, for an additional clarity the websites' interfaces were shown with a few selected similar activities through a largely visible screen devices.

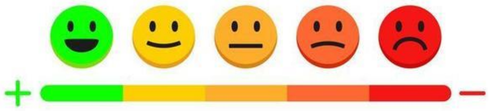


Figure 3. Likert scale [16]

Overall, the interview was conducted by focusing on three local educational websites. The websites were selected based on their different usage levels according to a previously conducted work by one of the authors of the study [7]. "Website 1" from the most used category, "Website 2" from the moderately used category, and "Website 3" from the least used category. Altogether 220 responses were gathered through the pre-surveys to identify the most preferred colors through the websites.

2) Secondary data gathering (Literature review): Among the various studies reviewed, a study has selected the three most suitable research papers. These papers were focused on the color preferences for educational interface backgrounds, colors related to their emotions, and color combinations to keep students attentions [4-6]. The studies were referred to collect a sample population for the preferred colors. By considering each paper, 6 sample groups were identified together creating a sample population of 323 for this study for the selected age categories from both private and public schools from both genders.

Table 1. Selected Sample populations from selected research papers
Source: Authors

Sample group	Sample population	Study
1	62	[4]
2	62	[4]
3	34	[5]
4	44	[5]
5	20	[6]
6	20	[6]

Group 01 and Group 02 consisted of 62 students each, representing the 7-year-old and 8-year-old categories, respectively. The students were asked to express their preferences for single colors, including violet, green, blue, yellow, orange, and red. The researcher recorded the students' 1 to 6-order of color preferences.

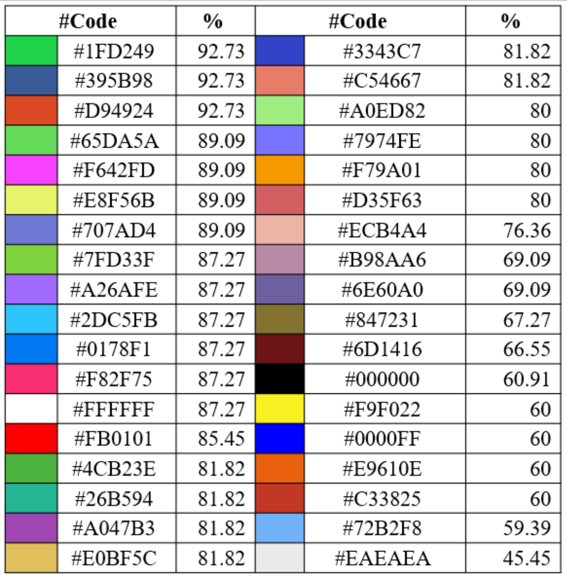
Group 03 and Group 04 comprised 34 and 44 students, respectively, from state and private schools in the 8-year-old category. These students were also asked to express their preferences for yellow, red, purple, blue, and green colors. The researcher recorded the average color preference percentage for each student.

Group 05 and Group 06 consisted of 20 boys and 20 girls, respectively, in the 7-8-year-old age range. The students were asked to indicate their happy preferences for red, orange, yellow, green, blue, purple, pink, white, brown, and black colors. The researcher recorded the average color preference percentage for each student.

B. Data analyzation

The core objective of this analysis was to identify the most suitable colors for the target age categories. Hence, the JASP tool was used to analyze the primary data as well as the secondary data. Data about 11 colors from the secondary data gathering and data about 36 colors from the primary surveys were analyzed.

1) *Primary data analysis*: Data gathered from the survey was analyzed using the JASP tool to identify the highest preferences of the students. The process of analysis was able to produce more than 80% valuable feedback for 24 colors. *The Figures 4 to 6 given below shows the analysed preferences by the primary data.*



#Code		%	#Code		%
🟩	#1FD249	92.73	🟦	#3343C7	81.82
🟦	#395B98	92.73	🟥	#C54667	81.82
🟥	#D94924	92.73	🟩	#A0ED82	80
🟩	#65DA5A	89.09	🟪	#7974FE	80
🟪	#F642FD	89.09	🟧	#F79A01	80
🟨	#E8F56B	89.09	🟥	#D35F63	80
🟦	#707AD4	89.09	🟧	#ECB4A4	76.36
🟩	#7FD33F	87.27	🟪	#B98AA6	69.09
🟪	#A26AFE	87.27	🟪	#6E60A0	69.09
🟦	#2DC5FB	87.27	🟫	#847231	67.27
🟦	#0178F1	87.27	🟫	#6D1416	66.55
🟥	#F82F75	87.27	⬛	#000000	60.91
⬜	#FFFFFF	87.27	🟨	#F9F022	60
🟥	#FB0101	85.45	🟦	#0000FF	60
🟩	#4CB23E	81.82	🟧	#E9610E	60
🟩	#26B594	81.82	🟥	#C33825	60
🟪	#A047B3	81.82	🟦	#72B2F8	59.39
🟨	#E0BF5C	81.82	⬜	#EAEAEA	45.45

Figure 4. Analysed feedback of primary data – Overall
Source: Authors

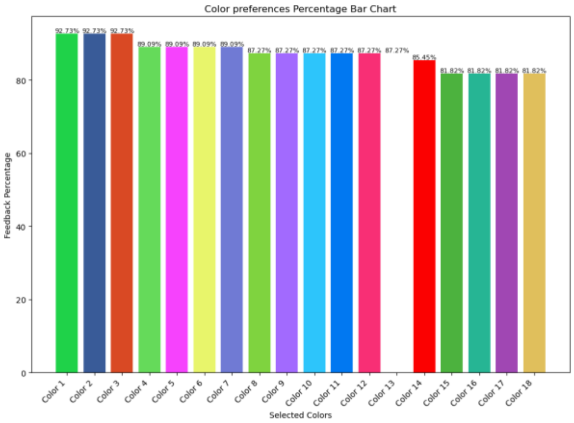


Figure 5. Analysed feedback of primary data – part 1
Source: Authors

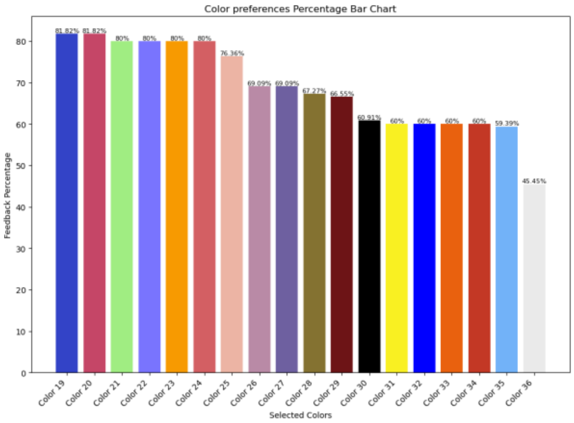


Figure 6. Analysed feedback of primary data – part 2
Source: Authors

2) *Secondary data analyzation:* The collected data from the sample groups were analyzed by grouping them based on similarities in available colors. Sample Group 1 and Group 2 were combined, and the average percentage for each 1st, 2nd, 3rd, 4th, 5th, and 6th color preferences was calculated. Sample Group 3, Group 4, Group 5, and Group 6 were also combined to calculate the average percentage of color preferences. Study [4] evaluated the color preference for the age 7- and 8-years students using 6 colors (Violet, Green, Blue, Yellow, Orange, Red). Both age categories have the highest color preference for both blue and red colors than others [4]. Table 2 given below shows the obtained results from [4].

Table 2: Color ranks for age 7- and 8-years students [4].

Color	Age	Violet	Green	Blue	Yellow	Orange	Red
Rank	7	5	3	1	6	4	2
	8	4	6	1	3	5	2

In [5], they have used 5 colors (Purple, Blue, Green, Yellow, Red) to test the relation between attention and colors for the students' educational processes. According to the result of this study purple, green and blue colors can be used to keep a highest students' attention for their studies than red and yellow [5].

According to the study [6], six feelings, three complicated emotions: proud, envious, and nervous and three fundamental emotions: happy, sad, and love were concentrated. Females provided more effective responses regarding both happy and negative emotions, and gender had a major effect on the appropriateness and quality of the emotional responses. Nevertheless, there were no appreciable variations in the replies for happy and negative feelings, nor were there any appreciable interactions between gender and categorization. The Munsell method of color notation was examined in the study; there were no appreciable variations in the orange, yellow, purple, or pink ratings between boys and girls. Blue was ranked as joyful, whereas black, white, red, green, and brown were more frequently rated as unpleasant hues. There were no discernible gender differences for orange, yellow, black, white, blue, or green [6].

According to the above analysis eleven colors, namely as: Violet, Green, Blue, Yellow, Orange, Red, Purple, Pink, Black, White, Brown were identified and among them seven colors, Blue, Red, Purple, Green, Orange, Pink could be highlighted as the most suitable colors for age 7- and 8-years students.

C. User Interface Design

After conducting a sound data gathering and analysation, the study's aim was to evaluate the suitability of identified color preferences at a newly created UI for the desired scope in educational websites. Therefore, a new website interface was designed as a sample that can be used for any educational website based on the identified color preferences. Then along with other three websites which were identified from study [7] during the data gathering were compared with the newly developed website among the age groups of 7- and 8-years old students by conducting a post survey. Figma tool was used to draw the User interface design. *The Figure 7 given below shows the newly created UI design by the authors based on the identified highly preferred colors by the age group.*

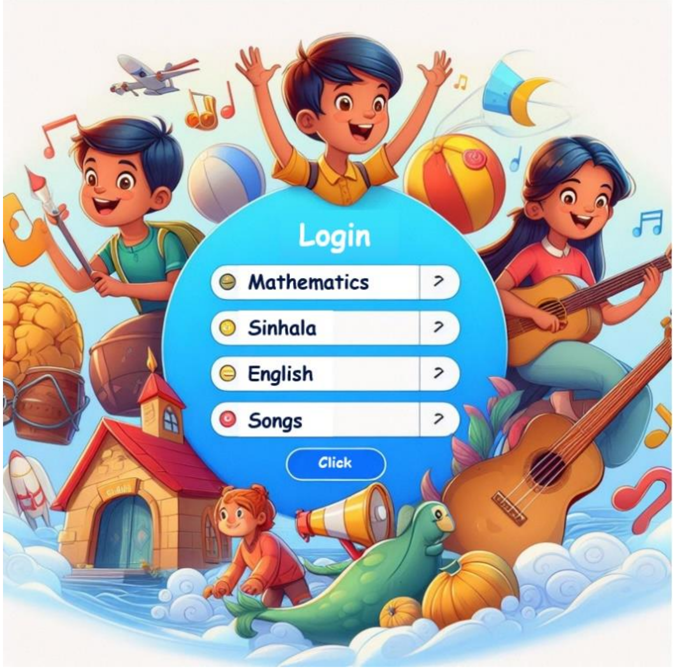


Figure 7: Interactive UI design created with the identified colors

Source: Authors

IV. TESTING AND VALIDATION

Due to privacy and ethics, the names and UIs of the other three websites have not been revealed in the study. Nevertheless, the study conducted an open-ended detailed interview as a post-survey to identify the impact of colors on a website for students. During the interviews, students were shown the four websites, three existing, 'Website 01', 'Website 02', and 'Website 03' along with the website UI designed by the authors with the identified colors. 34 detailed interviews were conducted, and records were taken to a Google form to present the results. Post surveys were supportive in validating the impact of colors in the enhancement of UI in the selected domain for educational websites. The results discussed in the section below will validate the findings.

V. RESULTS

During the post surveys, authors let students choose the most preferred website image considering the appealing nature for them and colors from the included images of four UI designs. The image of the newly created web interface design was considered as "Image 01", selected Website 1 as "Image 02", Website 2 as "Image 03", and Website 3 as "Image 04" in the post surveys. Table 3 and Figure 8 given below shows the results recorded to a google form from the detailed interviews conducted as the post survey.

Table 3: Results of the post-survey
Source: Authors

Website	Selection Criteria	Image Name
Newly created UI design	Used Identified Colors	Image 01
Website 01	Most-used category	Image 02
Website 02	Medium-used category	Image 03
Website 03	Least-used category	Image 04

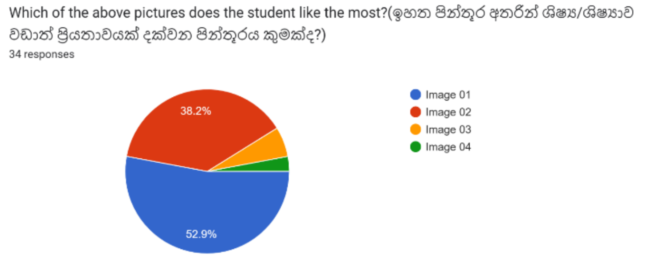


Figure 8: Post survey responses
Source: Authors

As shown in above Figure 8, 34 responses were gathered from the students of the respected age group 7-8 years old from private and public schools. According to the results, 52.9% of the sample preferred Image-01 which was created by the authors using identified colors. 38.2% of feedback was obtained by Image-02 of Website 1 which was from the most used category. 5.88% of feedback was taken from Image-03 which represents Website 2 and 2.94% of feedback for Image-04 which was Website 3. Overall, 31 responses from the 34 were taken by Image-01 and Image-02.

As per the results of the post-survey which was conducted to validate the impact of considering the color preferences in UI designs, it was identified the students preferred the website created by the authors with the colors identified as the highest preferred by age group as the most preferred educational website design among the considered four interfaces.

Hence, the results clearly have shown that even though there are existing websites which are highly used, students will be most likely to get attracted to an UI design which has successfully selected the colors according to the preferences of the user age, gender or any other relevant criteria. Hence, the results conclude that the colors can be considered as an important factor in UI design enhancement.

VI. DISCUSSION AND CONCLUSION

The study was conducted with a balanced experimental setup, where there is more room to succeed. This section hence discusses the limitations, drawbacks and successful facts that the study has flown through. During the data gathering process, it has been limited only for the local educational websites to balance the scope. Hence, the study has only reached to the Sri Lankan educational context and produced the results related to it. Moreover, another critical challenge that study has experienced was to gather the correct feedback from the age 7- and 8-years students. In that aspect, school teachers' and parents' support were taken to get most accurate preferences and to simplify the questions to the student audience. Also, the data-gathering process was limited to only 34 students due to the above-mentioned situations. The selected age category has no experience and much knowledge for scaling their preferences as a presentation or ratio. To avoid this matter, the Likert scale was used to gather the correct preferences. Secondary data sources were also reviewed to make the color selections accurate. Further, three factors were observed when identifying the color preferences based on the specific scenarios such as: gender, keeping attention, and initial feelings. Various lists of colors were identified through this study and both colors identified from primary and secondary data gathering were analyzed to interpret the most suitable colors. Equal size samples were taken from the population to avoid color preference ratio based on gender. Moreover, the privacy of the selected websites had to be preserved. The website names had to be processed anonymously due to ethical considerations. Hence, a study has addressed them as Website 1, Website 2, and Website 3.

In conclusion, the study has proved that there is an impact of color on educational websites. According to the overall process from requirement gathering to the result interpretation, a study has identified different colors from the existing educational websites, emotional preference, attention, and initial feelings. The study emphasized the direct impact of gender and emotional preferences on colors. Also, developers need to focus more on ways to keep the students' attention on the educational web pages. As the final output of this study, it can be highlighted that the created UI design got the highest preference (52.9%) from the students over the existing websites selected from all three categories, highest-used, moderately- and lowest-used. Color may affect the attraction for educational purposes. The study can hence conclude that the colors do impact the UI designs and proper selection of colors can be used as a greatly impacted element to enhance users' interaction with UI designs and create a rich user experience.

Concluding the work, the authors emphasize that as a part of the overall process of education with the educational websites, enhancing the knowledge of the future world holders can also be dependent on color selection that a designer chooses. Hence it is compulsory to keep suitable and preferred color combinations and consistencies when designing user interfaces to keep the interaction.

.

VII. FUTURE WORKS

The study has room to expand the work in many aspects in the future. Many other elements in a UI design can be evaluated to check the impact of them in enhancing the UI design. Further, factors such as fonts, user behaviour, user interactions, and gestures, etc can be evaluated in enhancing the user interfaces and user experiences.

REFERENCES

[1] Gunesekera, A. I., Bao, Y., & Kibelloh, M. (2019). The role of usability on e-learning user interactions and satisfaction: a literature review. *Journal of Systems and Information Technology, 21*(3), 368-394.

[2] K. Gaines, and Z. Curry, "The Effects of Color on Learning and Behavior", Journal of Family and Consumer Sciences Education, Vol. 29 Issue 1, 2011, pp 46-57.

[3] G. P. Park, "Correlations between color attributes and children's color preferences", Color Research & Application, Vol 39(5), 2014, pp.452-462.

[4] G.M. Michaels, "Color preference according to age", The American Journal of Psychology, Vol 35(1), 2012, pp.79-87.

[5] Duyan and R. Ünver, "A research on the effect of classroom wall colors on student's attention",. A| Z ITU Journal of the Faculty of Architecture, Vol 13(2),2016, pp.73-78.

[6] D.J. Pope, H. Butler and P. Qualter, “Emotional Understanding and Color-Emotion Associations in Children Aged 7-8 Years.”, Child Development Research, 2012.

[7] M. S. Sathsara, W. Gamage, and S. K. Jayathunga, “Investigating User Interaction in User Interface Designs of Educational Websites for 7 to 8 Years Old Children: A Comparative Study,” ir.kdu.ac.lk, Sep. 2023, Accessed: May 09, 2024. [Online].Available:http://ir.kdu.ac.lk/handle/345/7387

[8] C. Llinares, J. L. Higuera-Trujillo, and J. Serra, “Cold and warm colored classrooms. Effects on students’ attention and memory measured through psychological and neurophysiological responses,” Building and Environment, vol. 196, p. 107726, Jun. 2021,doi:https://doi.org/10.1016/j.buildenv.2021.107726.

[9] “Color Symbolism and Child Development,” obo. https://www.oxfordbibliographies.com/display/document/obo-9780199791231/obo-9780199791231-0270.xml

[10] “
The Effect Of Class Room Color On Learning With Reference To Primary Education; A Case Study In Sri Lanka,” ResearchGate. https://www.researchgate.net/publication/320831687_THE_EFFECT_OF_CLASS_ROOM_COLOR_ON_LEARNING_WITH_REFERENCE_TO_PRIMARY_EDUCATION_A_CASE_STUDY_IN_SRI_LANKA

[11] P
ratibodh - Journal Editor, P. Khandelwal, and N. Chaudhary, “The Psychology of Colors in UI/UX Design,” PRATIBODH, 2023. https://pratibodh.org/index.php/pratibodh/article/view/154

[12] C
. Daļa, P. Zinātnes, and Humanities, “Daugavpils Universitātes 61. Starptautiskās Zinātniskās Konferences Proceedings Of Rakstu Krājums The 61 Daugavpils Universitātes 61. Starptautiskās Zinātniskās Konferences Rakstu Krājums Proceedings Of The 61 St International Scientific Conference Of Daugavpils University,” 2019. Available: https://dukonference.lv/files/978-9984-14-901-1_61_konf_kraj_C_Hum%20zin.pdf#page=79

[13] “
The Role of Color Psychology in UI/UX Design,” www.linkedin.com. https://www.linkedin.com/pulse/role-color-psychology-uiux-design-keyur-vadhadia/

[14] R
aising Children Network, “6-8 years: child development,” Raising Children Network, Nov. 17, 2017. https://raisingchildren.net.au/school-age/development/development-tracker/6-8-years

[15] H
. Mahnke, Color, environment, and human response : an interdisciplinary understanding of color and its use as a beneficial element in the design of the architectural environment. New York ; Chichester: Wiley, 1996.

[16] *F*
reepik.com, 2024. https://img.freepik.com/premium-ector/satisfaction-rating-vector-level-concept_824631-733.jpg?w=1380 (accessed Jul. 15, 2024).

ABBREVIATIONS AND SPECIFIC SYMBOLS

UI - User Interface

ACKNOWLEDGMENT

This work would not have been a success without the immense support of academic staff of General Sir John Kotelawala Defence University. We would express our gratitude to them and for all the individuals who has supported in completing this work as a success.

AUTHOR BIOGRAPHIES

DMS Sathsara, the first author of the research work, is a graduate from the University of Sri Jayewardenepura, Sri Lanka, with a Bachelor of Information Communication Technology Honours Degree specialized in Software Technology. Currently she serves as an instructor at General Sir John Kotelawala Defence University, showcasing her expertise in the field.

DVDS Abeysinghe, the second author of the research work, is a graduate from the Eastern University of Sri Lanka, with a Bachelor of Science Honours in Specialization of Computer Science Degree and has completed her master's at University of Moratuwa in Cloud computing specialization. Currently she serves as a Lecturer (probationary) at General Sir John Kotelawala Defence University, showcasing her expertise in the field.

KGK Abeywardhane, the third author of the research work, is a graduate from the Vavniya Campus, University of Jaffna, Sri Lanka with a Bachelor of Information Communication Technology Honours Degree and has completed her MSc in IT from SLIIT, Sri Lanka. Currently she serves as an instructor at General Sir John Kotelawala Defence University, showcasing her expertise in the field.

Journal website: