ENHANCING PREDICTIVE ACCURACY AND RESOURCE UTILIZATION IN DISEASE PREDICTION : A COMPARATIVE STUDY OF SMALL VS LARGE LANGUAGE MODELS

MINI PROJECT REPORT

Submitted by

Tamanna	**210701281**
Sudhir S	**210701269**

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

ANNA UNIVERSITY:: CHENNAI 600 025

APRIL 2024

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

BONAFIDE CERTIFICATE

Certified that this Report titled “**Enhancing Predictive Accuracy and Resource Utilization in Disease Prediction: A Comparative Study of Small Versus Large Language Models**” is the bonafide work of **“Tamanna (210701281) Sudhir S (210701269)”** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Rahul Chiranjeevi. V
Assistant Professor,
Department of Computer Science and Engineering,
Rajalakshmi Engineering College,
Chennai – 602015

Submitted to Mini Project Viva-Voce Examination held on ___________________

Internal Examiner **External Examiner**

ABSTRACT

This project involves a comparative analysis of small language and large language models where it evaluates the models' capabilities in deciphering complex relationships between diseases and their symptoms using the distilgpt2(SLM) and gpt2(LLM) models. We utilized a large dataset where both small and large language models were used to study diseases and their symptoms. The models were evaluated based on their computational efficiency, predictive performance and the resources required for the operation. The methodology includes loading datasets, tokenization and model setup, monitoring training and validation losses, hyperparameter tuning to optimize the models and eventually generating text. The study showed SLM's proficiency in producing context-aware responses whereas LLM's shows its strength in generating refined and comprehensive text. The outcome showed the ability of large language models to possess higher predictive accuracy and they demanded significantly higher computational resources as compared to small language models, which may not be advised to use in resource-constrained environment. In contrast, small language models have efficient resource usage despite their low accuracy make their application more pertinent where computational resources are limited. This study paves the way for understanding the blueprint for health informatics practitioner highlighting the importance of domain-specific training in enhancing predictive accuracy and resource utilization of language models. It also underscores the requisite for a new approach to deployment of language models in healthcare settings.

ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavour to put forth this report. Our sincere thanks to our Chairman **Mr. S.MEGANATHAN, B.E, F.I.E.**, our Vice Chairman **Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S.,** and our respected Chairperson **Dr. (Mrs.) THANGAM MEGANATHAN**, **Ph.D.,** for providing us with the requisite infrastructure and sincere endeavoring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N. MURUGESAN, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Dr. P. KUMAR, Ph.D.**, Professor and Head of the Department of Computer Science and Engineering for his guidance and encouragement throughout the project work. We convey our sincere and deepest gratitude to our internal guide, **Rahul Chiranjeevi V** Professor, Department of Computer Science and Engineering. Rajalakshmi Engineering College for his valuable guidance throughout the course of the project.

Tamanna-210701281

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

SLM	Small Language Models
LLM	Large Language Model
BERT	Bidirectional Encoder Representations from Transformers
LSTM	Long Short Term Memory
GPT-2	Generative Pre-trained Transformer 2
CoT	Chain of thought

CHAPTER 1
INTRODUCTION

1.1 GENERAL

The area of investigation of small and large language models in disease prediction is a rapidly growing field of study that has significant potential for the future of healthcare. The SLMs with their few parameters and smaller datasets are characterized by their speed and efficiency in processing, thus, they are suitable for the applications where computational resources are limited making them suitable for real-time applications. The other way around, large language models (LLMs) are characterized by their huge number of parameters and the intensive training on the large datasets, thus, they can perform the complex language tasks with high accuracy and are capable of understanding complex symptom descriptions and providing more accurate disease associations.

1.2 OBJECTIVE

The aim of the study is to:

- To find out the linkage between diseases and symptoms by finding out how efficient their response is and its predictive performance.

- The resources needed by language models to carry out the functions needs to be specified.

- In order to improve the predictive performance of the model, it is necessary to understand the magnitude of domain-specific training.

- To understand the practicality of these models in real world environments, it is mandatory to deploy and put them to use.

1.3 EXISTING SYSTEM

The existing system involves using GPT-2 which is a large language model for prediction of disease based on the symptoms and vice versa. Through bespoke fine-tuning they have been trained on domain specific data to find the diseases based on the input symptoms [7]. They were trained on massive datasets to capture patterns and apply the medical knowledge. A well-known algorithm in this area is the Medical Concept Normalization-Bidirectional Encoder Representations from Transformers (MCN-BERT), which is a very accurate tool for disease prediction from symptom descriptions[12].They tackled this challenge by harnessing the power of language models like BERT that can learn by finding out intricate connections between the symptoms and the diseases .This model can enhance their disease predictive accuracy by getting trained on large medical data thereby improving domain-specific knowledge. Bidirectional LSTM layers were studied to investigate the influence of their use on the comprehension of the contextual relations between symptoms and diseases. The technique of hyperparameter optimization using Hyperopt was employed to increase the model's performance and thus, the model will be able to generalize well to the new data. In the existing system MCN-BERT model was use that followed the below steps:

- **Data collection and preprocessing** - This step involved collecting input datasets which consisted of symptoms mapped to their corresponding disease labels. The data was preprocessed by using a medical tokenizer where symptoms were diligently tokenized to enhance the contextual understanding of medical terms.

- **BERT model and tokenizer initialization** - The model was fed with pre-trained model weights which contain detailed linguistic knowledge that enhance the model's ability to grasp intricate medical terms. BERT models were architectured in such a way to include multiple transformer encoder layers and a specialized tokenizer was initialized. This tokenizer seamlessly converted the symptom description to tokens that were in turn integrated into the model as input.

- **Model Training** - The course of MCN-BERT training includes a thorough analysis of all stages to guarantee the correct learning process. Firstly, batches of tokenized symptom description and corresponding disease labels are organized as in the equation below

  BatchData=PrepareBatches(TokenizedSymptoms, DiseaseLabels)

  where BatchData and PrepareBatch are the prepared training batches, and function for PrepareBatches orchestrates this preparation. The following is the Model Forward Pass in which we propagate the batches through the BERT model using its forward pass mechanism.

- **Model Evaluation** - Well-known evaluation metrics, including accuracy, precision, recall, and F1-score, were used to assess the models' classification performance.

  Accuracy = TP + TN / TP + FP + TN + FN

  Precision = TP / TP + FP

  Recall = TP / TP + FN

  $F1 - \text{score} = 2 * (10)$ (Precision ×
  Recall) / (Precision + Recall)

1.4 PROPOSED SYSTEM

The proposed work aims to perform comparative research on small language models (SLMs) and large language models (LLMs) in a setting of the disease prediction. This work is aimed at making an achievement in the existing knowledge[14] about disease prediction with a comparative examination of small language models (SLMs) and large language models (LLMs).The aim is to assess and compare the forecasting ability, workload, and resource deployment of both types of models. The research will further utilize these SLMs such as DistilGPT-2 and LLMs which includes the GPT-2 or BERT which have been shown to perform impressively in natural language processing tasks in diverse ways. These models will be adjusted to the data set of symptoms and illnesses, their performance will be evaluated based on the level that they can correctly predict diseases from the symptom descriptions. Similarly, the study will evaluate both models for inference time in addition to the predictive accuracy parameter. This approach will help to determine a comparative study of resource usage by both SLMs and LLM. Besides, the study will use the comparison of the loss function of the both models to measure their predicting accuracy level. The proposed work will, therefore, prove to be a vital asset in the field of disease prediction as it will offer an exhaustive appraisal of SLMs and LLMs. The findings from this experiment might serve as a guide for later studies in the field and help create better and comprehensive disease prediction models. This work is an innovative and well separated concept that not only considers the precision of the models but also the computational efficiency and the resource utilization. This makes it a very important contribution to the field of disease prediction.

CHAPTER 2
LITERATURE SURVEY

[1]Victor Sanh, Lysandre Debut, Julien Chaumond, Thomas Wolf in this paper attempted to create a small language model that is equally efficient without compromising its performance. It mentions the issues they usually face while training large language models when the resources and data available are limited . Their approach of using knowledge distillation during the pre-training phase is notable and shows promising results. Their work provides a motivation to develop more efficient small language models with high accuracy and better performance.

[2]In this paper, the authors are proving the efficiency of BERT in different NLP cases, such as question answering and language inference tasks.The BERT model was put to test and it was observed how it outweighed the performance of other models. BERT’s capability to perform complex language processing tasks with utmost efficiency eventually leads to the understanding that it can similarly perform disease prediction with high accuracy by interpreting the intricate details of connection between symptoms and diseases.

This study[3] by Timo Schick and Hinrich Schütze analyzed the performance small language models compared to large language models.They showed how small models trained with few parameters as compared to large models can perform equally in predicting the result. By converting textual inputs into cloze questions and leveraging gradient-based optimization, these “greener” models achieve impressive natural language understanding.

This research by Leonardo Ranaldi and André Freitas in [4] tried to develop a method to bridge the gap in reasoning skills between small and large language models.With the help of Instruction-tuning-CoT method they provide SLMs with ability to perform multi step controlled reasoning when elicited with the CoT mechanism.

This paper [5] by Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, and others show how scaling up the models improve their performance sometimes even rivaling prior state-of-the-art fine-tuning approaches.They trained GPT-3 with 175 billion parameters and tested their performance without any fine tuning.They demonstrated how GPT-3 produced strong results on translation,question-answering and other reasoning based challenges.

[7]The paper "Large language models in health care: vulnerabilities, risks, and challenges" discusses the LLMs opportunities in healthcare. It briefly touches upon the problems of using machine learning systems in biomedical and clinical fields being applied in clinical environments, and those challenges which need to be solved in order to widely adopt their usage in this field.

This paper introduces us to TinyLlama which is a compact language model pre-trained on approximately 1 trillion tokens for around 3 epochs[8] It significantly outperforms existing open-source language models with comparable dimensions.

In [9] Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, Xinrong Zhang, Zheng Leng Thai, Kaihuo Zhang, Chongyi Wang, Yuan Yao, Chenyang Zhao, Jie Zhou, Jie Cai, Zhongwu Zhai, Ning Ding, Chao Jia, Guoyang Zeng, Dahai Li, Zhiyuan Liu, and Maosong Sun explores the potential of Small Language Models as resource-efficient alternatives to Large Language Models.While focusing on SLMs, our approach exhibits scalability in both model and data dimensions for future LLM research.

In [10], Awobade, Oduwole, and Kolawole investigate the effect of compression techniques on small language models. Practical strategies such as pruning, knowledge distillation, and quantization have been applied on AfriBERTa, a low-resource and small-data BERT system for Koine Greek. These techniques were performed to find out the which one had more effect in improving the model’s efficiency

The research gap in this study is the absence of the comparative analysis of small and large language models (SLMs and LLMs) majorly in disease prediction in terms of their predictive accuracy and resource utilization. Even though there are studies on the individual capabilities of SLMs and LLMs,this research aims to bridge the gap by giving a direct comparison within the stringent constraints and precise requirements of healthcare informatics.

This paper [11] showed NLPs effectiveness varies across diseases.They conducted a search across major databases including PubMed, Embase, Web of Science, and Scopus, up to December 2023, using keywords related to NLP, LLM, and infectious diseases.

In [12] the research explores the use of LLMs and deep learning techniques for predicting disease from symptoms.It analyzed two Medical Concept Normalization—Bidirectional Encoder Representations from Transformers (MCN-BERT) models and a Bidirectional Long Short-Term Memory (BiLSTM) model and demonstrated how accurately they predict the output.

The research in [14] by Minki Kang, Seanie Lee, Jinheon Baek, Kenji Kawaguchi, and Sung Ju Hwang devised a new approach to performing knowledge intensive tasks in small language models. By leveraging the use of external-knowledge resources ,they utilize knowledge-augmented reasoning distillation to improve the performance of SLMs.

This paper [16] focuses on developing open-source language models in the medical application area. It includes text collection from PMC articles, language model’s calibration, and implementing specific optimization strategies.

This paper by Wang et al. introduced a novel framework called Step-by-step knowledge distillation framework for recommendation (SLIM)[17] .In this research paper they tried to elevate the reasoning capabilities of large language models to work in a resource efficient manner by enabling sequential recommenders. They distill knowledge from large models and feed it into small models based on user behavior sequences.

In this paper [18] addresses the concerns associated with developing large language models with up to trillion parameters. To address these limitations, they present MiniCPM which consists of: 1. 2B and 2. 4B non-embedding parameter variants. It is worth to note that in each of these categories these Small Language Models (SLMs) themselves perform very well and are on a par with 7B-13B LLMs.

In [19] they introduced LLaVa-Phi which is an multi modal assistant that harnessed the power of small language models Phi-2.It showed that even small language models even those with 2.7B parameters can excellently engage in complex dialogues that combine text and visual elements trained on quality datasets.

This research paper [20] analyzes how well pruning, knowledge distillation, and quantization work on small language models that are low on resources. This study provides that compression techniques eventually improve SLMs efficiency.

CHAPTER 3
SYSTEM DESIGN

3.1 DEVELOPMENT ENVIRONMENT

3.1.1 HARDWARE SPECIFICATIONS

This project uses minimal hardware but in order to run the project efficiently without any lack of user experience, the following specifications are recommended:

Table 3.1.1 Hardware Specifications

PROCESSOR	Intel Core i5
RAM	4GB or above (DDR4 RAM)
GPU	Intel Integrated Graphics
HARD DISK	6GB
PROCESSOR FREQUENCY	1.5 GHz or above

3.1.2 SOFTWARE SPECIFICATIONS

The software specifications in order to execute the project has been listed down in the below table. The requirements in terms of the software that needs to be pre- installed and the languages needed to develop the project has been listed out below.

Table 3.1.2 Software Specifications

FRONT END	HTML, CSS, Bootstrap, JavaScript
BACK END	Python, Django
FRAMEWORKS	Pytorch, Tensor Flow
SOFTWARES USED	Visual Studio, Jupyter Notebook

3.2 SYSTEM DESIGN

3.2.1 ARCHITECTURE DIAGRAM

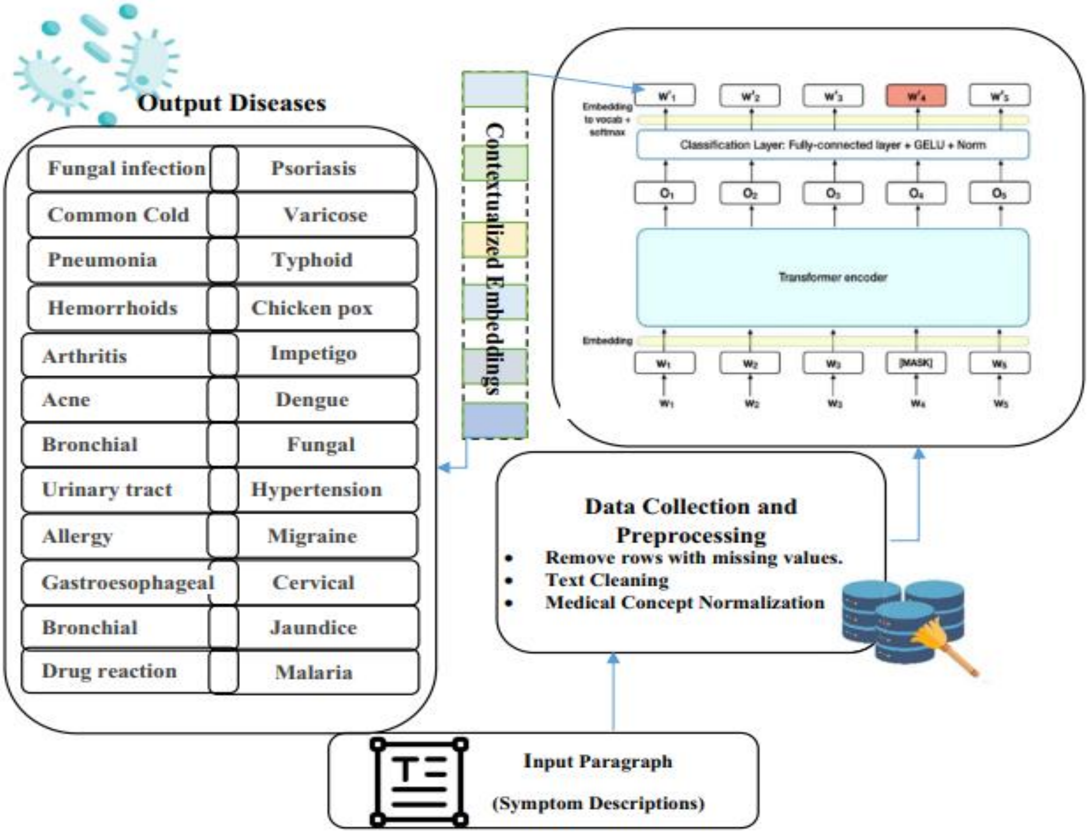


Fig 3.2.1 Architecture Diagram

APPROACH: UNDERSTANDING SLMs AND LLMs

Small Language Models (SLMs):

- **Architecture**: Language Models with a reduced context are called Small Language Models (SLMs). SLMs describe a neural network with fewer layers. They often utilize less numbers of layers and parameters, thus ensuring faster training and inference time.
- **Training:** SLM that are built by only using smaller datasets are less computationally expensive. This training method is optimal for situations, where there is no data available in high volume to work with.
- **Efficiency:** The lean construction or architecture of the SLMs is what leads to fast operational times and maximum efficiency[9].They can process language tasks rapidly, making them ideal for real-time applications.
- **Application:** Due to their resource efficiency, SLMs are being effectively implemented in situations where computational resources are a limitation or a real-time analysis is necessary.

Large Language Models (LLMs):

- **Architecture:** Large Language Models (LLMs) are based on a neural network of a great complexity that involves a deep learning approach. They are characterized by a higher number of trainable parameters that give them the ability to express complex patterns in data.
- **Training:** LLMs are trained with the aid of massive data that has an enormous variety. This procedure is an essential but energy-prodigal part of the model that enables it to carry out complicated language tasks.
- **Predictive Performance:** LLMs are well-known for their high accuracy and generating output that is not only detailed but also of good quality. Such language mastery helps to establish disease correlations with greater precision.
- **Resource Requirements**: The elaborate features of LLMs bear the tradeoff of more computing resources. They need more powerful hardware and consume more energy, thus in the places of resource scarcity, their performance can be low

CHAPTER 4
PROJECT DESCRIPTION

4.1 MODULE DESCRIPTION

4.1.1 DATA PRE-PROCESSING:

Like the present one, the data is prepared for processing by standardizing the symptoms in the form of a single string separated by commas.

4.1.2DATASET PREPARATION:

The processed data is then converted in a form that can be used for model training. This is tokenization of data input and the creation of the data loaders for training.

4.1.3MODEL INITIALIZATION:

The first thing that is done with evolved pre-trained LMs and evolving SLMs is the initialization. For SLMs, for instance, models like DistilGPT-2 can be toyed with, and for LLMs, GPT-2 or BERT can be used. The models are initialized with pre-weighted parameters and relocating these weights to the required device (CPU or GPU).

4.1.4MODEL TRAINING:

The layers are trained by using a regular training loop. During every epoch, the parameters of the model are adapted to minimize the loss value. Perturbation delivered through the forward method of the model provides the loss value to the output in the loss attribute.

4.1.5 MODEL EVALUATION:

After training, model accuracy is tested on the validation set considered. The way of the performance of the network is tested as the loss function is used which shows the difference between the predicted values by model and actual data. The metrics achieved in the present model (precision, accuracy, recall, and the F1-score) can be calculated in order to conduct a thorough assessment.

4.1.6 INFERENCE TIME MEASUREMENT:

Both such models' inference time is measured. It meant, feeding an input text into both the models and measuring the time, how fast these models generated the output.

4.1.7 INFERENCE TIME COMPARISON:

Two models are compared in terms of the inference times and a corresponding bar chart is used to illustrate this.

4.1.8 VALIDATION LOSS COMPARISON:

Bar graphs are employed to show the validation losses of the two models under testing. Moreover, the losses are compared.

CHAPTER 5

IMPLEMENTATION AND RESULTS

5.1 IMPLEMENTATION

The results highlight the trade-offs between SLMs and LLMs. While LLMs may provide better performance on the training data, SLMs can often generalize better to unseen data and provide faster inference times. This makes SLMs a viable option for applications where computational resources are limited or a fast response is required.
In this comparison:

Training Loss: The SLM has a higher training loss difference than the LLM. It could be that the SLMs simplifying of the architecture and the few parameters involved does not enable as close the fitting of the training data as can be done by the LLM.

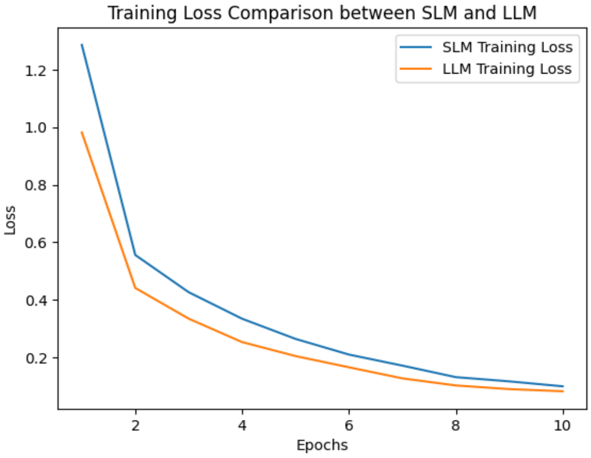


Fig.5.1.1 Training Loss Comparison

Validation Loss: In comparison, the validation loss of the SLM is less than that of the LLM. This demonstrates that the SLM has good generalizability to the data that the neural network has not encountered despite its higher training loss. This could be in connection with the discriminatory power of SLMs, where by avoiding data overfitting, they evade the overfitting problem of larger models such as the LLMs.

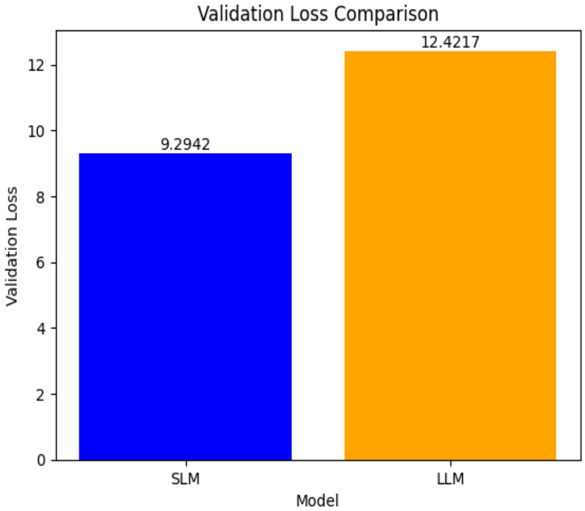


Fig.5.1.2 Validation Loss Comparison

Inference Time: The SLM speeds up the inference process relative to the LLM. This was to be anticipated as the SLMs are usually more computationally efficient owing to their small size/dimensionality and lesser number of parameters.

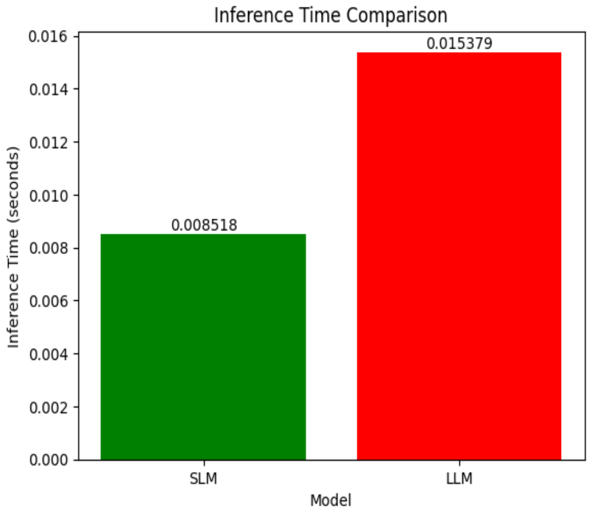


Fig.5.1.3 Inference Time Comparison

5.2 RESULTS

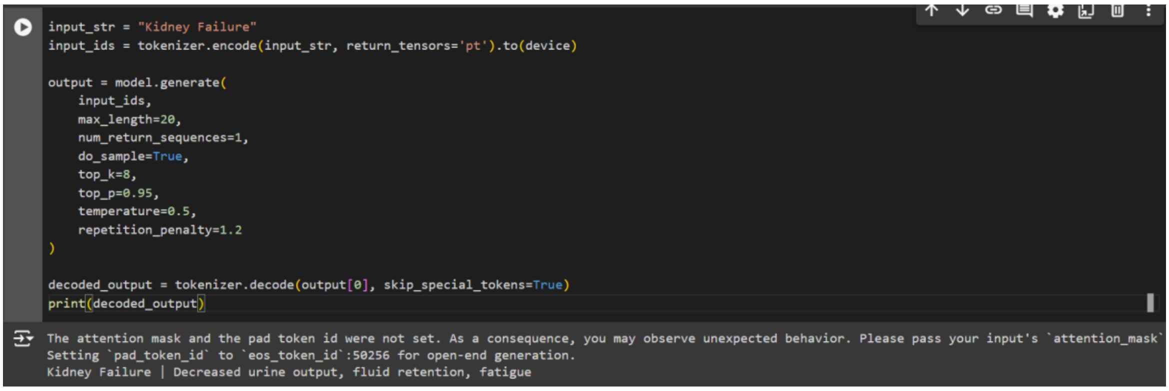
The comparison of the models based on the above three values gave the following results:

Model	Training Loss	Validation Loss	Inference Time
SLM	Higher	9.2942	0.008518 secs
LLM	Lower	12.4217	0.015379 secs

Table .5.2.1 Model Comparison

5.3 OUTPUT SCREENSHOTS

The below screenshot shows the prediction of diseases from symptoms and vice versa.

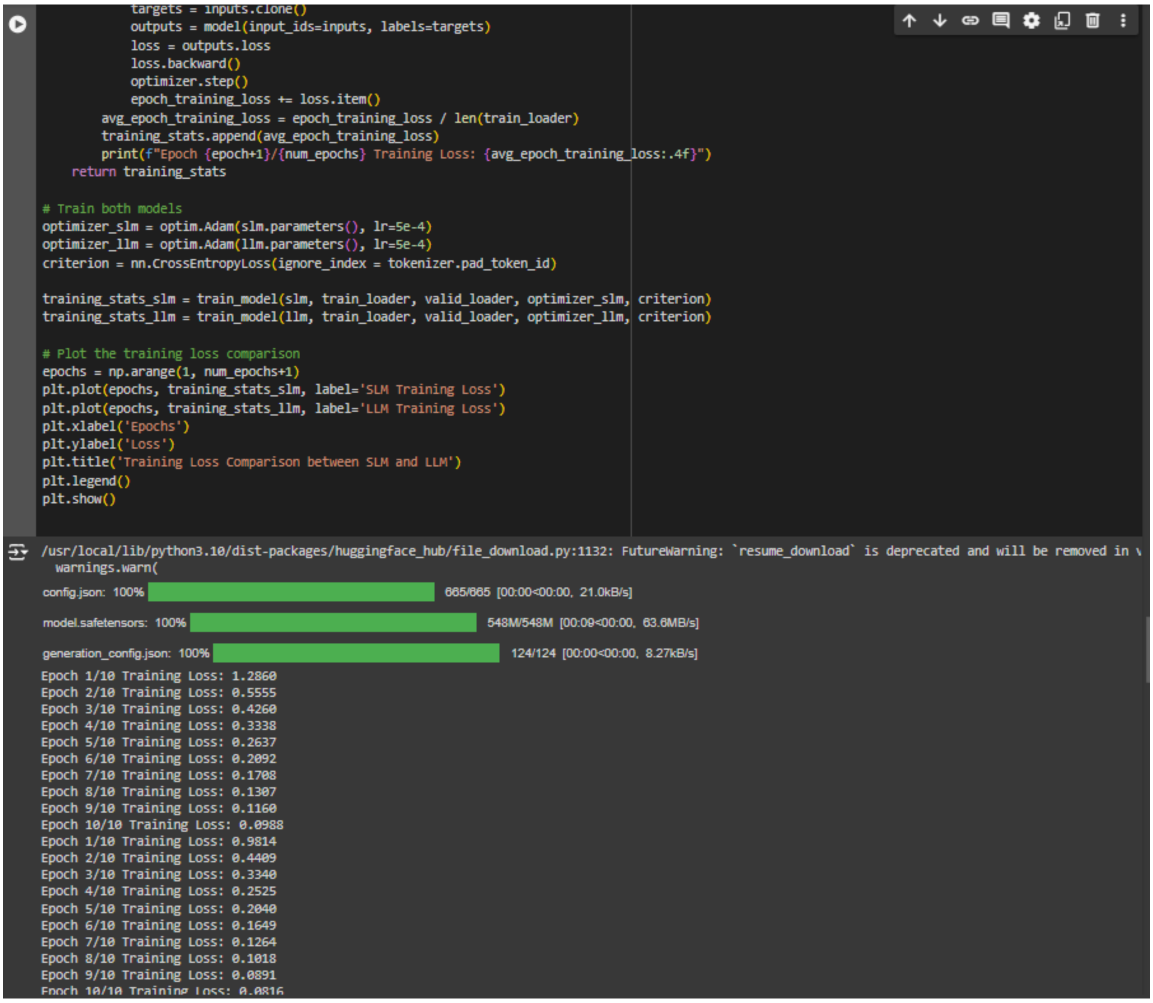


```
input_str = "Kidney Failure"
input_ids = tokenizer.encode(input_str, return_tensors='pt').to(device)

output = model.generate(
    input_ids,
    max_length=20,
    num_return_sequences=1,
    do_sample=True,
    top_k=8,
    top_p=0.95,
    temperature=0.5,
    repetition_penalty=1.2
)

decoded_output = tokenizer.decode(output[0], skip_special_tokens=True)
print(decoded_output)
```

```
The attention mask and the pad token id were not set. As a consequence, you may observe unexpected behavior. Please pass your input's `attention_mask`
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
Kidney Failure | Decreased urine output, fluid retention, fatigue
```

Fig.5.3.1 Prediction of symptom and disease

The below screenshot shows the comparison of performance of SLMs and LLMs based on certain parameters.



```
            targets = inputs.clone()
            outputs = model(input_ids=inputs, labels=targets)
            loss = outputs.loss
            loss.backward()
            optimizer.step()
            epoch_training_loss += loss.item()
        avg_epoch_training_loss = epoch_training_loss / len(train_loader)
        training_stats.append(avg_epoch_training_loss)
        print(f"Epoch {epoch+1}/{num_epochs} Training Loss: {avg_epoch_training_loss:.4f}")
    return training_stats

# Train both models
optimizer_slm = optim.Adam(slm.parameters(), lr=5e-4)
optimizer_llm = optim.Adam(llm.parameters(), lr=5e-4)
criterion = nn.CrossEntropyLoss(ignore_index = tokenizer.pad_token_id)

training_stats_slm = train_model(slm, train_loader, valid_loader, optimizer_slm, criterion)
training_stats_llm = train_model(llm, train_loader, valid_loader, optimizer_llm, criterion)

# Plot the training loss comparison
epochs = np.arange(1, num_epochs+1)
plt.plot(epochs, training_stats_slm, label='SLM Training Loss')
plt.plot(epochs, training_stats_llm, label='LLM Training Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.title('Training Loss Comparison between SLM and LLM')
plt.legend()
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/file_download.py:1132: FutureWarning: `resume_download` is deprecated and will be removed in v
  warnings.warn(
config.json: 100% 665/665 [00:00<00:00, 21.0kB/s]
model.safetensors: 100% 548M/548M [00:09<00:00, 63.6MB/s]
generation_config.json: 100% 124/124 [00:00<00:00, 8.27kB/s]
Epoch 1/10 Training Loss: 1.2860
Epoch 2/10 Training Loss: 0.5555
Epoch 3/10 Training Loss: 0.4260
Epoch 4/10 Training Loss: 0.3338
Epoch 5/10 Training Loss: 0.2637
Epoch 6/10 Training Loss: 0.2092
Epoch 7/10 Training Loss: 0.1708
Epoch 8/10 Training Loss: 0.1307
Epoch 9/10 Training Loss: 0.1160
Epoch 10/10 Training Loss: 0.0988
Epoch 1/10 Training Loss: 0.9814
Epoch 2/10 Training Loss: 0.4409
Epoch 3/10 Training Loss: 0.3340
Epoch 4/10 Training Loss: 0.2525
Epoch 5/10 Training Loss: 0.2040
Epoch 6/10 Training Loss: 0.1649
Epoch 7/10 Training Loss: 0.1264
Epoch 8/10 Training Loss: 0.1018
Epoch 9/10 Training Loss: 0.0891
Epoch 10/10 Training Loss: 0.0816
```

Fig.5.3.2 Comparison of model’s performance

CHAPTER 6
CONCLUSION AND FUTURE ENHANCEMENTS

6.1 CONCLUSION

Therefore, as the study shows both SLMs and LLMs are effective at disease prediction but there are clear allusions for improvements that guide future researchers to study behaviors of these language models. This result shows both the options between the SLMs and LLMs and also gamut's within each mode of learning. Although LLM may be unable to improve the notion, the SLM can manage to generalize well on new data and provide fast computing speed. Therefore, SLMs provide a reliable approach in situations where the computation resources are limited or response time is fast. Nonetheless, the preference between SLMs and LLMs depends on the operational needs that range from computational resources availability to need for real time responses and predictive accuracy. Contributing to the area of forecasting disease is an important outcome of the study by elaborating the difference between the two approaches: SLMs and LLMs, furthering research, and informing the building of systems that are better and more accurate for predicting disease.

The present study explores multiple lines of research in the near future. Another area of possible research is to identify approaches that would aid in increasing the efficiency of SLMs This could be done through approaches for model compression, quantization or hardware optimization as an example. A possibility can be to research how SLMs can be deployed in other healthcare applications, such as patient triage as well as getting medical information back. In addition, considerations of ethical and privacy issues should be included in future perspectives on SLMs for health care. It could mean looking into techniques of safeguarding personal data, obtaining written consent, and preventing the models from misusing. Finally, using a trained SLM in real life situations like a hospital or a health application and measuring their actual performance compared to others could be another exciting path for future research. This would offer innovative information about the realization and reliability of using SLMs in disease forecasting.

6.2 FUTURE ENHANCEMENTS

The present study explores multiple lines of research in the near future. Another area of possible research is to identify approaches that would aid in increasing the efficiency of SLMs This could be done through approaches for model compression, quantization or hardware optimization as an example. A possibility can be to research how SLMs can be deployed in other healthcare applications, such as patient triage as well as getting medical information back. In addition, considerations of ethical and privacy issues should be included in future perspectives on SLMs for health care. It could mean looking into techniques of safeguarding personal data, obtaining written consent, and preventing the

models from misusing. Finally, using a trained SLM in real life situations like a hospital or a health application and measuring their actual performance compared to others could be another exciting path for future research. This would offer innovative information about the realization and reliability of using SLMs in disease forecasting.

REFERENCES

[1] Sanh, V. et al. (2020) Distilbert, a distilled version of Bert: Smaller, faster, cheaper and lighter, arXiv.org.

[2] Devlin, J. et al. (2019) Bert: Pre-training of deep bidirectional Transformers for language understanding, arXiv.org.

[3] Schick, T. and Schütze, H. (2021) It’s not just size that matters: Small language models are also few-shot learners, arXiv.org.

[4] Aligning Large and Small Language Models via Chain-of-Thought Reasoning (Ranaldi & Freitas, EACL 2024)

[5] Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, et al. Language models are few-shot learners. Adv Neural Inf Process Syst. 2020; 33: 1877–901.

[6] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. Adv Neural Inf Process Syst. 2017; 30: 1–11.

[7] Yang R, Tan TF, Lu W, Thirunavukarasu AJ, Ting DSW, Liu N. Large language models in health care: development, applications, and challenges. Health Care Sci. 2023; 2: 255–263.

[8] Zhang, P. et al. (2024) TinyLlama: An open-source small language model, arXiv.org.

[9] Hu, S. et al. (2024) MiniCPM: Unveiling the potential of small language models with Scalable Training Strategies, arXiv.org.

[10] Awobade, B., Oduwole, M. and Kolawole, S. (2024) What happens when small is made smaller? exploring the impact of compression on small data pretrained language models, arXiv.org.

[11] “Utilizing Natural Language Processing and Large Language Models in the Diagnosis and Prediction of Infectious Diseases: A Systematic Review.” 2024. American Journal of Infection Control, April.

[12] Hassan, E., Abd El-Hafeez, T. & Shams, M.Y. Optimizing classification of diseases through language model analysis of symptoms. Sci Rep 14, 1507 (2024).

[13] StoryBuddiesPlay. 2024. Microsoft Phi 3 LLM: Powerful Small Language Model Demystified. StoryBuddiesPlay.

[14] Kang, Minki, et al. "Knowledge-augmented reasoning distillation for small language models in knowledge-intensive tasks." Advances in Neural Information Processing Systems 36 (2024).

[15] Armengol-Estapé, Jordi, et al. "SLaDe: A Portable Small Language Model Decompiler for Optimized Assembly." 2024 IEEE/ACM International Symposium on Code Generation and Optimization (CGO). IEEE, 2024.

[16] Wu, Chaoyi, et al. "PMC-LLaMA: toward building open-source language models for medicine." Journal of the American Medical Informatics Association (2024): ocae045.

[17] Wang, Yuling, et al. "Can Small Language Models be Good Reasoners for Sequential Recommendation?." arXiv preprint arXiv:2403.04260 (2024).

[18] Hu, Shengding, et al. "Minicpm: Unveiling the potential of small language models with scalable training strategies." arXiv preprint arXiv:2404.06395 (2024).

[19] Zhu, Yichen, et al. "LLaVA-$\phi $: Efficient Multi-Modal Assistant with Small Language Model." arXiv preprint arXiv:2401.02330 (2024).

[20] Awobade, Busayo, Mardiyyah Oduwole, and Steven Kolawole. "What Happens When Small Is Made Smaller? Exploring the Impact of Compression on Small Data Pretrained Language Models." arXiv preprint arXiv:2404.04759 (2024).

[21] Magister, Lucie Charlotte, et al. "Teaching small language models to reason." arXiv preprint arXiv:2212.08410 (2022).

[22] Fu, Yao, et al. "Specializing smaller language models towards multi-step reasoning." International Conference on Machine Learning. PMLR, 2023.

PLAGIARISM REPORT (PROJECT REPORT)

PAPER WITH PLAGIARISM REPORT

lack of evidence or lack of witnesses. Moreover it can also be due to the negligence of the law enforcement officers or tendency of them to zero in on someone who is not the suspect known as tunnel vision. In India, close to 21 lakh cases are unsolved.

In order to solve these crimes, we have to make use of the technological advancements in the field of Machine Learning and Deep Learning. By exploiting the power of Machine Learning algorithms, the law enforcement agencies can potentially unlock now leads and re-examine old evidences and retouch upon the earlier overlooked information. The exponential increase in the digital data has made internet the house of information and using ML techniques, patterns and unnoticed relations can be established.

With the passage of time, the crime rate worldwide is only going to increase which calls for new measures and aid from the recent technological advancements. This paper explores the power of Machine Learning and its impact on the field of law enforcement. By analyzing disconnected data points which are often overlooked by humans, the Machine Learning algorithms suggests already solved case files which might help the case to take a new direction. The ensemble model of classical clustering algorithms like K Means, DBSCAN, Hierarchical clustering algorithms have been used to increase the accuracy of the case files suggested.

K Means is a clustering algorithm is an unsupervised Machine Learning algorithm that is used to cluster similar data points according to a specific parameter. This algorithm takes the value of 'k' which is a pre-determined value which denotes the number of clusters. K Means is a centroid model which uses clustering of data points using the centroid value. The variance of the data points which are relatively closer to the centroid is calculated and based on the variance, the points clustered together. This process is repeated until the best possible group of clusters are formed and the number of outliers are decreased.

DBSCAN which stands for Density Based Spatial Clustering of Application with Noise, is another type of unsupervised clustering algorithm which is an improvement to the K Means clustering algorithm. In contrast to the centroid model used in K Means, DBSCAN uses density based clustering which determines whether the data points belong to a cluster or not using the density of the region. The DBSCAN algorithm uses two parameters minPts and eps.minPts is the parameter which is the minimum number of points required to form a cluster. eps is the parameter which is the distance measure that is used to locate the points in the neighborhood of any point.

Hierarchical Clustering is a type of clustering algorithm which considers each datapoint as a separate cluster and later combines them into larger clusters. There are two types of clustering namely Agglomerative Clustering and Divisive Clustering. Agglomerative Clustering is a type of clustering algorithm which follows a bottom up approach where initially, each data point is considered as a single entity and eventually, bigger clusters are formed by combining the clusters which are close to each other. Divisive clustering algorithm follows the top down approach where initially the data points are considered to be as a large cluster and as the distance between the data points decrease, the clusters are sub divided into individual components.

Ensemble learning is a Machine Learning strategy which uses the output of multiple models in order to generate a more accurate result. This is done by using a voting classifier which uses one of the two voting techniques namely hard voting and soft voting to choose the best model. Hard voting is a voting technique where the output class is based on the highest majority of votes whereas Soft voting technique determines the output class by taking the average probabilities of classes. While the above-mentioned clustering algorithms have their own advantages and benefits, the lack some key features which can be rectified by using the ensemble learning model. While K Means uses the centroid model for clustering, it is sensitive to initial conditions as the number of clusters needs to be predefined which may not be ideal in cases involving large number of datasets having large number of features to cluster with. While DBSCAN which uses density-based clustering, can form clusters or arbitrary size and does not need the number of cluster before-hand, it poses its own challenges that needs to be rectified. It is sensitive to the choice of Eps and MinPts and the cost of computation is high when the number of data points is large. Meanwhile Hierarchical clustering provides rich information and insight to the dataset but the inability to handle scalable datasets and time complexity makes the algorithm computationally expensive. An ensemble model of the above-mentioned algorithms will be robust to different data distributions, with enhanced cluster separation and outlier detection.

Using an ensemble model in this project improves pattern recognition as cold cases generally involve complex and multi-dimensional datasets which may be challenging to detect by using a single clustering algorithm. Understanding these patterns and relationships is crucial for the law enforcement officers in order to prioritize their investigation. By incorporating hierarchical clustering into the ensemble model, prioritization of the cases can be done as similar cases files will be clustered together. This may help them to lead the investigation in a certain direction.

II. LITERATURE SURVEY

This paper [1] discusses the importance and benefits of using ensemble learning methods in crime prediction and the advantages it holds over the conventional machine learning models when used as a single classifier. Due to the dynamic nature of crimes, it is difficult to find the right configuration for the datasets to feed as an input to the ensemble model. The paper proposes a model called a assemble-stacking based crime prediction method (SBCPM) which applies SVM model to achieve configurations that are specific to the domain. This model achieves 99% classification accuracy on training data.

One of the main types of crime, organized crimes have been studied in this paper [2] by identifying organized crimes through social media analysis. By analysing language cues in social media as indicators, this paper classifies crime type and location. The system generates Organized crime concepts to alert analysts of potential criminal activity by grouping information sources. Analysts can also investigate organized crime concepts through a prototype software system featuring social media scanning and map-based visualization. The system is illustrated using human trafficking and modern slavery.

This paper [3]describes a Crime Monitoring System (CMS) designed to detect crimes in real-time using camera surveillance. It aims to overcome human limitations like slow reactions by combining CCTV cameras with deep learning techniques. The system operates in three stages: detecting weapons, violence, and recognizing faces. It achieves high accuracy rates in each stage: over 80% for weapons, 95% for violence, and 97% for faces. Real-world testing shows the system effectively detects crime and alerts authorities promptly, improving overall security and safety measures.

The study conducted in the area of cross domain learning on crime prediction [4] gives an insight about the data insufficiency problem faced in small cities. As the number of researches on crime prediction are on the rise, the urban data has become more accessible to the researchers. But this paper discusses on how to overcome data insufficiency in small cities with respect to Canada. By using ensemble learning as it generalizes new data and the classified data is compared against the baseline models.

This review paper [5] gives a take on Artificial Intelligence being used in the field of crime prediction. By intensively analysing various criteria, the models are evaluated. Intensive research is carried out after reviewing 120 research papers that were published between 2008 and 2021. The research has concluded that crimes and spatial are the most applied categories in analysing crimes. The various ML models used across the 120 research papers were noted and supervised learning models were found to be the major contributors with 31% while a combination of supervised and unsupervised learning models contributed with 22% and unsupervised learning models alone contributed with 10%.

This study [6] draws attention towards the importance of crime prediction and forecasting to enhance urban safety which are considered as hotspots of crime. As an improvement to the existing studies which lack accuracy on learning models, this study uses various machine learning algorithms like SVM, XGBoost, KNN and ARIMA model to better fit the crime data. The findings suggest a moderate increase in Chicago's overall crime rate while Los Angeles experiences a decline. The study concludes that these predictive models can aid law enforcement in directing patrols and developing effective strategies to combat crime.

Aimed at aiding the Police Department with proper crime forecasting, this study [7] concentrates on machine learning algorithms for crime forecasting. By using Folium for data visualization, the year wise trends of crimes were discovered. The machine learning algorithms used were Random Forest, K-Nearest-Neighbours, AdaBoost and Neural network out of which Neural Network provided promising results when tested with Chicago Police Department's records with an accuracy of 90.77%.

This review paper [8] focuses on the growing interest among researchers in using machine learning and deep learning techniques to predict crime by analysing over 150 articles in the process. It examines the diverse algorithms employed and datasets utilized for crime prediction, exploring emerging trends and factors influencing criminal behaviour. By providing a overview of the research in crime prediction, it serves as a valuable resource for both academics and law enforcement agencies.

This paper [9] compares machine learning algorithms for crime prediction using historical data of public property crime from a coastal city in China between 2015 and 2018. It finds that LSTM model outperforms other algorithms like KNN, SVM, Random Forest and that incorporating environmental factors improves prediction accuracy compared to the model which only uses historical crime alone. Thus the paper concludes that crime prediction techniques should use both environmental factors and historical crime to maximize the accuracy.

This study [10] specifically focuses on classification of Crime category in the United States of America by using the data collected from socio-economic data from the US Census and crime data from FBI UCR. By using supervised classification algorithms like Naïve Bayes and Decision Tree, the study has concluded that the Decision Tree algorithm outperforms Naïve Bayes by having an accuracy of 83.9519% whereas the latter has an accuracy of 70.8124% in predicting crime of different states of the country.

This paper [11] deals with a unique type of crime called economic crime, which generally takes a lot of time for the law enforcement officers to solve. This paper develops an algorithm that detects fictitious enterprise using a classification algorithm called the Support Vector Machine. This model proved to be efficient as it resulted with a 99.7% accuracy in the testing data which consisted of the economic activities of 1100 companies in Ukraine out of which 355 were defined as fictious.

Various data mining algorithms and ensemble learning techniques applied in crime data analysis and prediction have been discussed in this paper [12]. It deals with the significance of crime forecasting to reduce criminal activities based on historical data. With the increasing rate of crime cases, accurate crime prediction becomes crucial. Data mining methods helps in finding out patterns. The study aims to analyse and discuss the effectiveness of different methods applied in crime prediction, which can be used as a foundation in solving further crimes.

Data Mining is one of the best practices that can be used to find out patterns and relationships within the dataset. This paper deals with [13] analysing each data mining technique extensively by finding out the pros and cons of each technique. This technique is mainly used in Crime Detection as the patterns which generally go unnoticed can be detected can be found out by applying Data Mining techniques. This survey is intended to serve as a state-of-the-art crime detection guide.

Criminology is a field which identifies crime characteristics. This study [14] uses data mining techniques to identify crime characteristics by using Decision Tree(J48). Decision Tree algorithm is considered to be most efficient among all the machine learning algorithms as experimental results have proved that Decision Tree(J48) holds an accuracy of 94.25% which can be considered as a safe score to be relied on.

As crime rates continue to rise, it poses a severe challenge to the law enforcement officers. To address this issue, this paper [15] proposes extracting data from crime records and applying data mining techniques, including classification and regression algorithms, to predict future crime trends. The law enforcement agencies can allocate their resources effectively by using this system since the model is trained on historical data and using this, future trends can be found out. The system also suggests visualizing predicted outcomes using clustering algorithms like K-means, providing a user-friendly interface for understanding and interpreting the data.

This study [16] looks at how people connect in social networks and predicts behaviour using fuzzy systems. It uses colors to show different levels of possible criminal behaviour based on factors like background and habits. By studying these connections, it helps spot unusual behaviour and adjusts the network to keep things safe, using fuzzy logic methods.

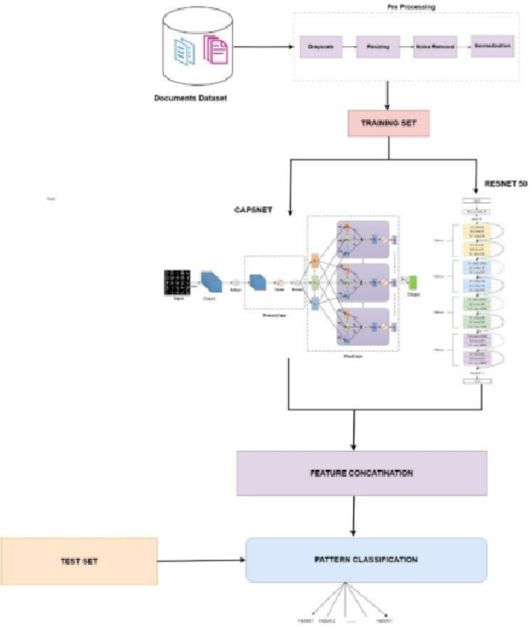
This research [17] explores using data mining and machine learning to predict violent crime patterns. It compares crime data from a dataset with actual statistics from Mississippi. Three algorithms are tested: Linear Regression, Additive Regression, and Decision Stump. The study finds that Linear Regression performs the best. It emphasizes the importance of using data mining in law enforcement for tasks like identifying crime hotspots and understanding trends. Despite challenges, the study highlights the value of these methods in improving public safety.

The changing nature of crime is making traditional approaches to crime ineffective. This paper [18] discusses on a growing pattern which is the combined use of computer vision and machine learning technology to deal with these limitations. It makes use of machine learning algorithms that use historical crime data to predict crimes in the future. In public places, computer vision algorithms are used simultaneously for anomaly detection and real-time surveillance. The capacity of these integrated techniques to improve law enforcement operations and reduce the negative effects of criminal activity on society is examined.

This paper [19] reveals the importance of predicting crimes based on various factors like weather, geographic location, literacy rate of the location and so on. These features creates a base for these crimes. Moreover, the paper suggests some us to use Hotspots based on geographic location to predict and stop it. The paper proposes a K-Means for clustering the data and creating the hotspot. This paper also proves the vitality of these kind of applications in police department. Plotting of 2D Hotspot in location of crimes are based on historic crime data.

This research [20] assumes that traditional algorithms use concepts which are stationary and expect them to be stationary. Using such algorithms in real world forecasting where the concepts and scenarios would change could cause a real problem as the machine is not future proof and it is susceptible to errors. Further, the paper deals with the predicting vulnerable victims of crimes occurred in large cities. It mentions that a significant number of types of victims are changed based on police countering.

III. PROPOSED MODEL



A. PRE-PROCESSING:

Digitalizing Tamil and Hindi text from FIR (First Information Report) comes with a challenge of ignoring background noises as the paper's age can be decades. Before feeding the image into the models it has to be cleaned and uniformly resized. To reduce the number of dimensions of the data the image is grey scaled using binarization for separation of foreground and background which further helps in removal of image noise and text classification. The background is changed to black and the text and noise is converted to white. The colors values of 0 to 255 is normalized to 0 to 1. The uniform images of the dataset is being resized into 64x64 size in order to reduce the training and working time of the system implemented along with that the zoom, height shift and other shifts are applied to make the character centered. The morphological operation of images are applied into the dataset in order to convert the text and background more clear and less vulnerable to errors. These helps in identifying shapes of different letters and strokes and erosion operations are done to diminish the sizes of boundaries. The data is then processed using Vott in order to convert it into JSON to feed it into CAPSNET and RESNET 50 the image is resized into 9x9 after Vott. Each image is given its own JSON file and then their coordinates are stored accordingly. The JSON can be further used in formation of classes for CAPSNET.

B. Training Set

In the domain of computer vision, there are two important techniques which greatly help in the application for the problem statement discussed here: CASPNET for handwritten regional text recognition and ResNet50 for image classification in crime scenes. CASPNET, an advanced convolutional neural network, specializes in deciphering handwritten text in various languages and styles, aiding in document digitization and analysis. Meanwhile, ResNet50, a deep learning model, excels in recognizing objects and patterns within crime scene images, aiding in forensic investigations. By deploying CASPNET, researchers can gather valuable information from handwritten documents, enhancing archival and investigative processes. On the other hand, ResNet50 plays a crucial role in identifying crucial elements within crime scenes, aiding law enforcement agencies in identifying suspects and reconstructing events accurately. Together, these technologies offer great tools for enhancing efficiency and accuracy in document analysis and crime scene investigation, contributing to the advancement of forensic science and law enforcement practices.

CAPSNET:

Image Annotation and Segmentation:

Data from various case files that were documented in the regional language where the crime was reported are collected together. The languages can be in English, Tamil or Hindi.

Initially, images are segmented individual characters. This is carried out using the Vott tool, an open-source software designed for image annotation. Vott provides the annotation process where users can define bounding boxes around objects of interest, in this case, characters within the case files. These boxes bounded by the users provide a visual representation of the segmentation process, outlining the boundaries of each character on the image. After processing the input image, Vott generates a JSON file for each annotated image, containing detailed coordinates of the bounding boxes, including various characteristics like width, height, and the coordinates of the upper-left corner. This enables precise identification and separation of characters for further analysis. Vott is capable of saving segmented character images, further helping the segmentation process. Following segmentation, all isolated characters were standardized to a size of 9 by 9 pixels to prepare them for input into the developed model. This step proved crucial in preprocessing case files written in regional language, enabling segmentation into lines, words, and individual characters for subsequent analysis.

CapsNet Model:

CapsNet is a type of neural network that uses a group of neurons to represent different parts of an object, like an object's characteristic or a specific part of it. Two convolutional layers are combined with a fully connected layer called RecCaps . The first layer converts the character image into blocks of activity. The second layer acts like a primary capsule and turns single output neurons into vectors with 8 dimensions. Then, RecCaps is used to capture the spatial relationships between all the local features from the primary capsule, and fed all these features into a higher dimensional capsule with 16 dimensions. The network's first layer works like a typical convolution layer in a CNN, using the ReLU activation function. In further layers, a special "squashing" activation function to shrink short vectors to zero and longer ones to a number close to 1 is used. This helps represent the probability of certain features being present. To run the network, segmented characters is fed from the case files into it. The first layer, extracts lower-level features from the characters. Then, the PrimaryCaps layer applies convolutional operations to get a 3-D matrix. This matrix has dimensions of 18 * 16 * 128, which is then split into 16 capsules, each with dimensions 18 * 16 * 8. We use a dynamic routing algorithm to connect primary capsules with advanced capsules, helping the model understand how different features relate to each other. This improves the model's ability to recognize and interpret handwritten characters. The algorithm calculates coupling coefficients for each iteration, which are used to compute input vectors for parent capsules and output vectors for capsules in the next layer. Finally, the result is gathered by the agreement between all capsules

Implementation:

The developed CapsNet model, created using the Keras Python library is implemented on computers having minimum specifications RAM and graphic support. The CapsNet model has a nested structure with 800 hidden units, and tested against 399 different classes. Adam optimizer is used for model training, which is a popular model in deep learning. The model used here is a hybrid one, which includes convolution layers, dense layers, and hidden units. ReLU and sigmoid activation functions is used, along with a loss function called binary cross-entropy.

RESNET50:

ResNet50 is a deep residual network designed to address the problem of network degradation. It introduces cross-layer connections to construct residual blocks, which learn the difference between input and output. This approach helps protect information integrity and simplifies the learning process. ResNet's structure accelerates network training by enabling faster loss reduction. The ResNet50 model consists of an initial independent convolutional layer, followed by pooling and four distinct convolutional residual modules. Each residual block comprises multiple convolutional layers and cross-layer connections, concluding with a pooling layer. The pool5 and fc1 features are extracted from the ResNet50 network for experimentation.

The Faster R-CNN object detection algorithm is utilized for semantic annotation of images, followed by content-based image retrieval. This method involves using ResNet50 for feature extraction, where the ResNet50 model is enhanced with a multi-scale pooling technique at the ROI-Pooling layer to improve feature representation. Specifically, the ROI-Pooling layer, responsible for mapping candidate regions back to the original image, undergoes modification by replacing single-scale pooling with multi-scale pooling. This alteration involves employing multiple pooling panes for maximum pooling at various scales (8x8, 4x4, 2x2, and 1x1) on the feature maps generated by the last convolutional layer, resulting in 85-dimensional feature vectors. This enhancement aims to overcome limitations in feature representation encountered with single-scale pooling. The improved multi-scale features are then utilized for object classification, contributing to enhanced accuracy in the classification process. Additionally, the semantic information obtained from Faster R-CNN object detection is integrated with the content-based image retrieval process, leading to improved retrieval rates and accuracy. This integrated approach involves utilizing the semantic information derived from object detection for image filtering, thereby reducing the search space for subsequent content-based retrieval based on depth features.

C. *Pattern Classification*

After digitalizing the case files, some features are extracted from the case files based on various parameters. Some of the many possible features are listed below:

Case Identifier: Each case will be represented by a unique identifier. Though this may not be a contributing factor towards the feature concatenation, this feature will contribute towards removing duplicate files from the dataset as combining case files from various jurisdiction can result in the dataset having duplicate case files and by using this feature, duplicate files will not be considered for further process.

Address: This field contains the location details of the crime containing the house number, street name, city, and country where crime has reported. Further, each of the above-mentioned attribute is separated for ease of classification for the model and the attributes present as text are converted into an integer for standardization .

Date of Crime: This feature contains the date and time of the crime. This date is in the format of Date, Month, Year followed by the time when the crime took place. Later each of the individual aspects are separated and concatenated along with the place where the crime took place to uniquely identify the crime.

Binary Classification: Certain features contain only binary classification of true or false. Features like whether the crime is Domestic Crime, whether an arrest was made or not. These classifications contribute to the overall classification to the type and severity of the crime.

Evidence Description: This features contains the details of evidences that are collected from the crime spot during the investigation process like DNA, Fingerprints, Weapon description.

Case Status: Case status contains the current status of the crime (ie) whether it is Active, Closed, Under Process or if there are developments since the initial report.

Based on the various features extracted, patterns are identified and classified together.

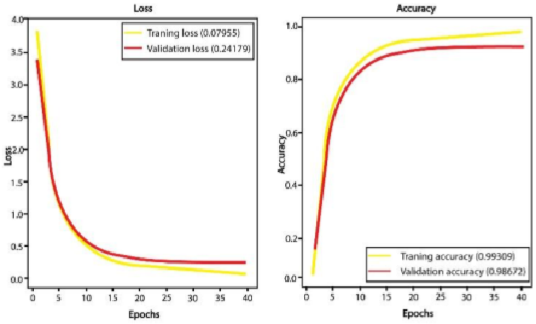
The test set, which is an active case file, is tested against the pattern classified cases files. When a new case file is given to the model, it is compared with the historical crime files and pattern matching is done with those files. Based on the output from the pattern classification algorithms, the historical case files which are closely related to the test case file are grouped together and are given as suggestions.

IV. RESULT

A. *STATISTICAL ANALYSIS:*

The analysis shows that strokes, edges, and curves are identified in caspnet to identify the specific regional language and also preprocessing removed all the unnecessary noises. The analysis on text recognition in Tamil gives around 90.12 +or - 2.24% and Hindi gives around 95.32 + or - 4.34%. The resnet50 used to identify crime scenes gives an sensitivity analytical score of 88.43 + or - 2.23% .This is a huge score in crime scene image classification. ResNet-50 exhibited the highest accuracy of 95.39% in digitizing handwritten text, surpassing alternative methods such as HWT, CSO, and BBO. ResNet-50's effectiveness is attributed to its utilization of residual modules and convolution layers, enabling superior feature extraction and training efficiency.

Characters	precision	recall	f1-score	support
ka	0.95	0.96	0.98	300
cha	0.95	0.97	0.97	300
ta	0.96	0.95	0.95	300
pa	0.97	0.95	0.93	300
ya	0.93	0.94	0.94	300
ra	0.94	0.91	0.91	300
va	0.91	0.93	0.95	300
na	0.95	0.97	0.97	300
gna	0.98	0.95	0.96	300
zha	0.97	0.98	0.95	300
Accuracy			0.951	3000
Macro Average	0.95	0.95	0.95	3000
Weighted Average	0.95	0.95	0.95	3000



(a) Loss curve (b) Accuracy curve

The graph is plotted for Accuracy and Loss. The graph is plotted against the number of Epochs. The Loss Curve results in having a training loss of 0.07955 and a validation loss of 0.24179. The accuracy curve increases exponentially having a training accuracy of 0.951 and validation accuracy of 0.98

V.CONCLUSION

In conclusion, this paper has concentrated two important areas of research: handwritten regional language text classification using CapsNet and crime scene image classification using ResNet50 which provides a foundation for prediction of crime using historical crime data. In the domain of handwritten regional language text classification, the implementation of CapsNet proved to be promising. By taking advantage of the unique architecture of CapsNet, the model showed excellent performance in accurately categorizing handwritten text of various regional languages. This achievement helps greatly in digitalizing old case files which can help to solve cases which remain unsolved in that period. On the other hand, in the domain of crime scene image classification, the utilization of ResNet50 showcased remarkable capabilities in accurately identifying and classifying objects and scenes within crime scene images. This can be of great help in forensic investigation. The robustness and efficiency of ResNet50 make it a valuable tool for law enforcement agencies and forensic experts in analyzing and interpreting visual evidence, ultimately aiding in criminal investigations and ensuring justice.

Moreover, this research emphasizes the importance of using state-of-the-art deep learning techniques in addressing complex real-world problems. However, it's important to acknowledge the limitations and areas for future exploration. While CapsNet and ResNet50 show promising results, further research is needed to enhance their performance, scalability, and applicability across different datasets and scenarios. Additionally, the ethical implications of deploying such technology in sensitive domains like law enforcement warrant careful consideration as there might be a potential data leak issue and if done, it might have a serious impact.

In summary, this study lays a solid foundation for future researches aimed at advancing the fields of handwritten text classification and crime scene image analysis. This lays a solid foundation on which crime prediction is done. By incorporating regional language handwritten recognition, the scope and application of the project is increased multifold.

References

[1] S. S. Kshatri, D. Singh, B. Narain, S. Bhatia, M. T. Quasim, and G. R. Sinha, "An Empirical Analysis of Machine Learning Algorithms for Crime Prediction Using Stacked Generalization: An Ensemble Approach," *IEEE Access*, vol. 9, pp. 67488–67500, 2021, doi: 10.1109/ACCESS.2021.3075140.

[2] S. Andrews, B. Brewster, and T. Day, "Organised crime and social media: a system for detecting, corroborating and visualising weak signals of organised crime online," *Secur Inform*, vol. 7, no. 1, Dec. 2018, doi: 10.1186/s13388-018-0032-8.

[3] M. M. Mukto *et al.*, "Design of a real-time crime monitoring system using deep learning techniques," *Intelligent Systems with Applications*, vol. 21, Mar. 2024, doi: 10.1016/j.iswa.2023.200311.

[4] F. K. Bappee, A. Soares, L. M. Petry, and S. Matwin, "Examining the impact of cross-domain learning on crime prediction," *J Big Data*, vol. 8, no. 1, Dec. 2021, doi: 10.1186/s40537-021-00489-9.

[5] F. Dakalbab, M. Abu Talib, O. Abu Waraga, A. Bou Nassif, S. Abbas, and Q. Nasir, "Artificial intelligence & crime prediction: A systematic literature review," *Social Sciences and Humanities Open*, vol. 6, no. 1. Elsevier Ltd, Jan. 01, 2022. doi: 10.1016/j.ssaho.2022.100342.

[6] W. Safat, S. Asghar, and S. A. Gillani, "Empirical Analysis for Crime Prediction and Forecasting Using Machine Learning and Deep Learning Techniques," *IEEE Access*, vol. 9, pp. 70080–70094, 2021, doi: 10.1109/ACCESS.2021.3078117.

[7] A. Tamir, E. Watson, B. Willett, Q. Hasan, and J.-S. Yuan, "Crime Prediction and Forecasting using Machine Learning Algorithms," 2021. [Online]. Available: https://www.researchgate.net/publication/355872171

[8] V. Mandalapu, L. Elluri, P. Vyas, and N. Roy, "Crime Prediction Using Machine Learning and Deep Learning: A Systematic Review and Future Directions," *IEEE Access*, vol. 11, pp. 60153–60170, 2023, doi: 10.1109/ACCESS.2023.3286344.

[9] X. Zhang, L. Liu, L. Xiao, and J. Ji, "Comparison of machine learning algorithms for predicting crime hotspots," *IEEE Access*, vol. 8, pp. 181302–181310, 2020, doi: 10.1109/ACCESS.2020.3028420.

[10] R. Iqbal *et al.*, "An Experimental Study of Classification Algorithms for Crime Prediction." [Online]. Available: www.indjst.org

[11] A. Krysovatyy, H. Lipyanina-Goncharenko, S. Sachenko, and O. Desyatnyuk, "Economic Crime Detection Using Support Vector Machine Classification."

[12] A. Almaw and K. Kadam, "Survey Paper on Crime Prediction using Ensemble Approach." [Online]. Available: http://www.ijpam.eu

[13] S. Qayyum and H. Shareef Dar, "A Survey of Data Mining Techniques for Crime Detection," 2018.

[14] E. Ahishakiye, D. Taremwa, E. O. Omulo, and I. Niyonzima, "Crime Prediction Using Decision Tree (J48) Classification Algorithm," 2017. [Online]. Available: www.ijcit.com188

[15] V. Pande Student, C. Engg, V. Samant Student, B. E. Computer Engg, and S. Nair Asst Professor, "Crime Detection using Data Mining." [Online]. Available: http://www.ijert.org

[16] S. Gupta and S. Kumar, "Crime Detection and Prevention using Social Network Analysis," 2015.

[17] L. McClendon and N. Meghanathan, "Using Machine Learning Algorithms to Analyze Crime Data," *Machine Learning and Applications: An International Journal*, vol. 2, no. 1, pp. 1–12, Mar. 2015, doi: 10.5121/mlaij.2015.2101.

[18] N. Shah, N. Bhagat, and M. Shah, "Crime forecasting: a machine learning and computer vision approach to crime prediction and prevention," *Visual Computing for Industry, Biomedicine, and Art*, vol. 4, no. 1. Springer, Dec. 01, 2021. doi: 10.1186/s42492-021-00075-z.

[19] G. Hajela, M. Chawla, and A. Rasool, "A Clustering Based Hotspot Identification Approach for Crime Prediction," in *Procedia Computer Science*, Elsevier B.V., 2020, pp. 1462–1470. doi: 10.1016/j.procs.2020.03.357.

[20] A. J. De Souza, A. P. Borges, H. M. Gomes, J. P. Barddal, and F. Enembreck, "Applying ensemble-based online learning techniques on crime forecasting," in *ICEIS 2015 - 17th International Conference on Enterprise Information Systems, Proceedings*, SciTePress, 2015, pp. 17–24. doi: 10.5220/0005335700170024.

kags

ORIGINALITY REPORT

26%	5%	8%	19%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to University of Illinois at Urbana-Champaign Student Paper	18%
2	Aditi Moudgil, Saravjeet Singh, Vinay Gautam, Shalli Rani, Syed Hassan Ahmed. "Handwritten Devanagari Manuscript Characters Recognition using CapsNet", International Journal of Cognitive Computing in Engineering, 2023 Publication	2%
3	Dongyuan Li, Xiaojun Bai. "Criminal Investigation Image Retrieval Based on Deep Learning", 2020 International Conference on Computer Network, Electronic and Automation (ICCNEA), 2020 Publication	1%
4	www.medrxiv.org Internet Source	1%
5	ijcsmc.com Internet Source	<1%
	sci-hub.se	

6	Internet Source	<1%
7	Fatima Dakalbab, Manar Abu Talib, Omnia Abu Waraga, Ali Bou Nassif, Sohail Abbas, Qassim Nasir. "Artificial intelligence & crime prediction: A systematic literature review", Social Sciences & Humanities Open, 2022 Publication	<1%
8	link.springer.com Internet Source	<1%
9	www.researchgate.net Internet Source	<1%
10	Submitted to Capitol College Student Paper	<1%
11	Guofu Zhai, Zhigang Sun, Guotao Wang, Pengfei Li, Qi Liang, Min Zhang. "Instance-based transfer learning method for locating loose particles inside aerospace equipment", Measurement, 2023 Publication	<1%
12	www.aiirjournal.com Internet Source	<1%
13	repository.tudelft.nl Internet Source	<1%
14	5wwwww.easychair.org Internet Source	<1%

15	file.techscience.com Internet Source	<1%
16	www.cse.griet.ac.in Internet Source	<1%
17	"Intelligent Systems and Applications", Springer Science and Business Media LLC, 2019 Publication	<1%
18	Asit Kumar Das, Priyanka Das. "Graph based ensemble classification for crime report prediction", Applied Soft Computing, 2022 Publication	<1%
19	Oghenevovwero Zion Apene, Nachamada Vachaku Blamah, Gilbert Imuetinyan Osaze Aimufua. "Advancements in Crime Prevention and Detection: From Traditional Approaches to Artificial Intelligence Solutions", European Journal of Applied Science, Engineering and Technology, 2024 Publication	<1%
20	doaj.org Internet Source	<1%
21	polynoe.lib.uniwa.gr Internet Source	<1%
22	scitepress.org Internet Source	<1%

23	www.jcdronline.org Internet Source	<1%
24	Junxiang Yin. "Crime Prediction Methods Based on Machine Learning: A Survey", Computers, Materials & Continua, 2023 Publication	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		