Research Project– Software Requirements

Enhancing ML Prediction Model In Requirement Engineering

Sulbha Malviya
Masters in computer science

Mehenaz Afrin
Phd in computer science

CS 573 Advanced Software Engineering

Instructor – Dr. Nassir Eisty

Introduction

Enhancing Machine Learning Prediction Models for Requirement Engineering:

- This research extends existing studies that have identified biases in machine learning models used in requirement engineering.

- Our work focuses on addressing the unexplored areas and limitations of previous frameworks, particularly in the context of accurately identifying sensitive features from user stories(short description of software features from user's perspective)

- By leveraging advanced word embedding techniques and deep learning models, we aim to enhance the accuracy, and contextual understanding in these models, overcoming the biases observed in earlier approaches.

Problem Statement

- Machine learning models,may overlook important contextual clues or focus on irrelevant patterns, especially in complex textual data like user stories.

- This can result in models that are less accurate or make biased decisions, especially when the training data is imbalanced or fails to represent all user stories equally.

Problem Importance

- **Limited research work:** Only a few studies have assessed the potential biases of machine learning models during requirements specification and analysis in software development.

- **Current Frameworks Use Shallow ML Models:** Existing frameworks rely on simpler machine learning algorithms to classify tasks and application domains within user stories. Shallow models generally consist of a single or few layers, unlike deep learning models that may have dozens or even hundreds of layers.

- **Avoidance of Advanced AI Solutions:** Due to concerns over computational complexity and interpretability, advanced techniques like deep learning have not been fully explored, even though they may offer deeper insights.

Proposed Solution

Current Framework:

- Utilizes a synthetic dataset for experimentation.
- Employs shallow machine learning algorithms to classify application domains and tasks.
- The framework has shown promising performance, but limited in exploring advanced models and techniques.

Proposed Framework

Advanced Word Embedding Techniques:

- Implement several Bert version Albert, Distil Bert, Roberta for enhancing the text preprocessing and the performance Level.

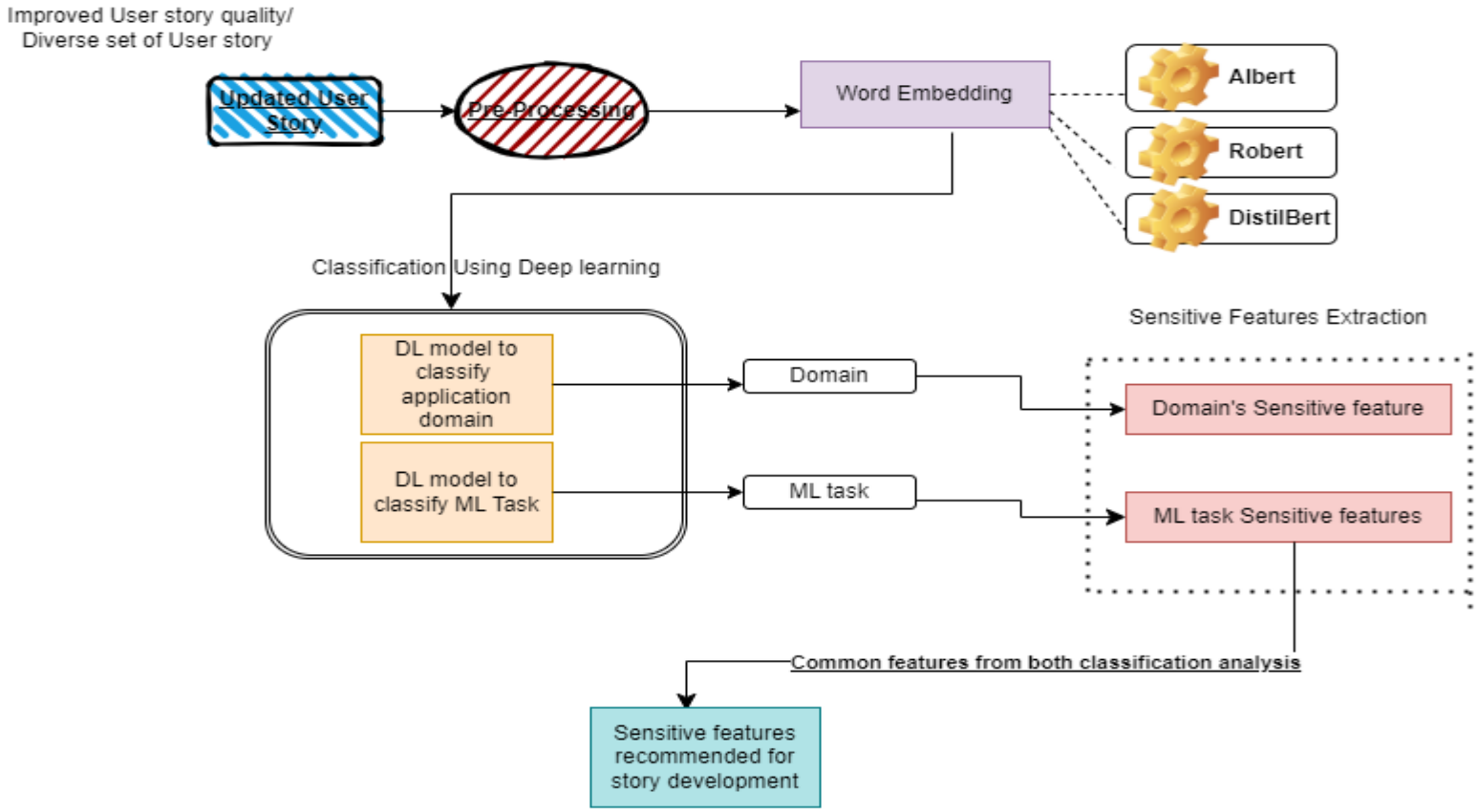
Advanced Classification Approaches:

- Explore deep learning architectures (e.g., Long Short-Term Memory, Recurrent Neural Network, Bidirectional Long Short-Term Memory) for effective classification of sensitive features, capturing complex patterns in the data.

Proposed Solution Cont.

Advanced Word Embedding Techniques:

Leverage advanced variants of BERT (e.g., RoBERTa, DistilBERT) to generate contextually rich embeddings, improving the model's understanding of nuanced language and enhancing classification accuracy.



Dataset approach

- We have used the existing dataset of ferrara et al which contains 12,401 synthetic USs related to 34 different application domains.
- For cleaning the dataset, we have dealt with the white spaces, punctuations, stop words.
- We have preprocessed the data with TFIDF Vectorizer, Glove, Word to Vec.

Synthetic User Stories

- This is the synthetic user stories dataset currently employed for our framework.

A	B	
Domain	**Machine Learning Task**	User Story
Biology	abstractive summarization	A group of researchers is using abstractive summarization to identify key trends and insights in large sets of biological data, enabling more efficient analysis and interpretation.
Plant Science	abstractive summarization	As a plant scientist, I want to use abstractive summarization to extract key findings from multiple research papers on plant genetics, so that I can better understand the latest
Biology	action model learning	As a molecular biologist, I want to use action model learning to predict the structure of complex biomolecules and design new drugs with higher specificity and efficacy, to deve
Plant Science	action model learning	As a plant scientist, I want to use action model learning to predict the growth of various plant species under different environmental conditions, so that I can optimize their grow
Biology	activation function	As a bioinformatics researcher, I want to use active learning settings to optimize the selection of training data for machine learning models that predict protein-protein interactio
Plant Science	activation function	As a plant scientist, I want to use machine learning activation functions to predict plant growth and yield based on environmental conditions, in order to optimize crop productio
Biology	active learning setting	As a genomics researcher, I want to use the AdaBoost algorithm to identify genetic variants that are associated with complex diseases like diabetes and heart disease, to imp
Plant Science	active learning setting	As a researcher, I want to use active learning to label a large dataset of plant images, so that I can train a model to identify different plant species with high accuracy while min
Biology	adaboost	A researcher in bioinformatics is using an Adaline model to classify different genetic sequences based on their underlying structures. By training the machine learning algorithr
Plant Science	adaboost	As a plant scientist, I want to use AdaBoost to classify different types of plant species based on their physical features so that I can better understand their characteristics and
Biology	adaptive resonance theory	As a cognitive neuroscientist, I want to use adaptive resonance theory to study how the brain processes and integrates information from multiple sensory modalities, to better u
Plant Science	adaptive resonance theory	As a plant biologist, I want to use adaptive resonance theory to analyze large sets of gene expression data and identify genes that are co-regulated under different environment

Research Questions:

RQ1: Do Bert versions enhance the performance level in classifying ML specific application domain from user stories?

RQ2: Do Bert versions enhance the performance level in classifying ML specific tasks from user stories?

RQ3: Can the application of Deep learning approaches higher the performance level in classifying ML specific application domain from user stories?

RQ4: Can the application of Deep learning approaches higher the performance level in classifying ML specific tasks from user stories?

Methodology

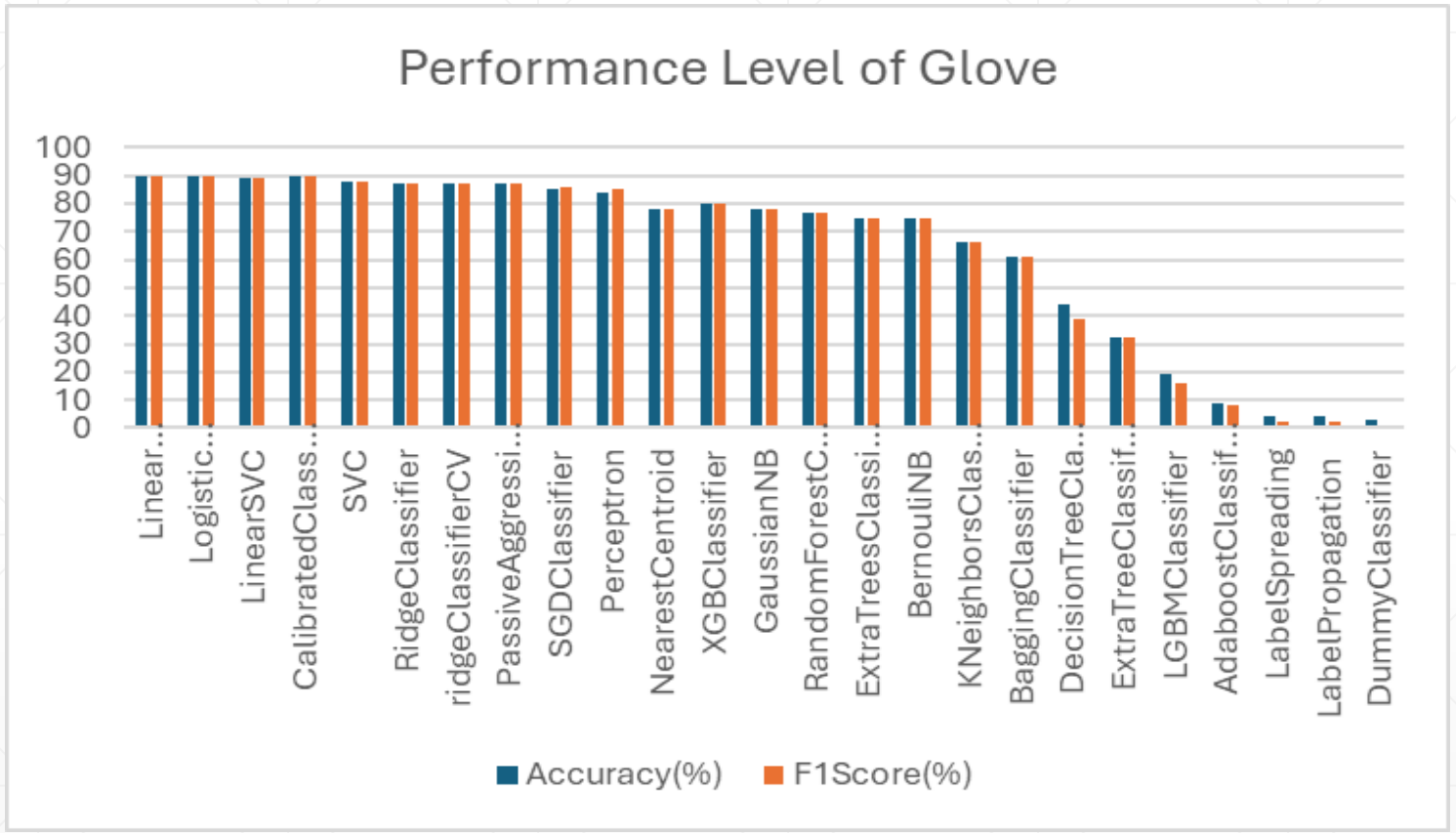
If the application domain or ML tasks are misclassified, this framework may suggest sensitive features that do not align with the specific context of a given use case.

Application Domain Classification: In this part, we have exploited multiple word embedding techniques Bert, Albert,DistilBert,Roberta,Glove. Here we are trying to classify the most likely application domain of the user stories among the 34 domains available in the ontology. This domain detection is a multiclass classification problem. This framework support 25 different machine learning algorithms along with different Bert versions.

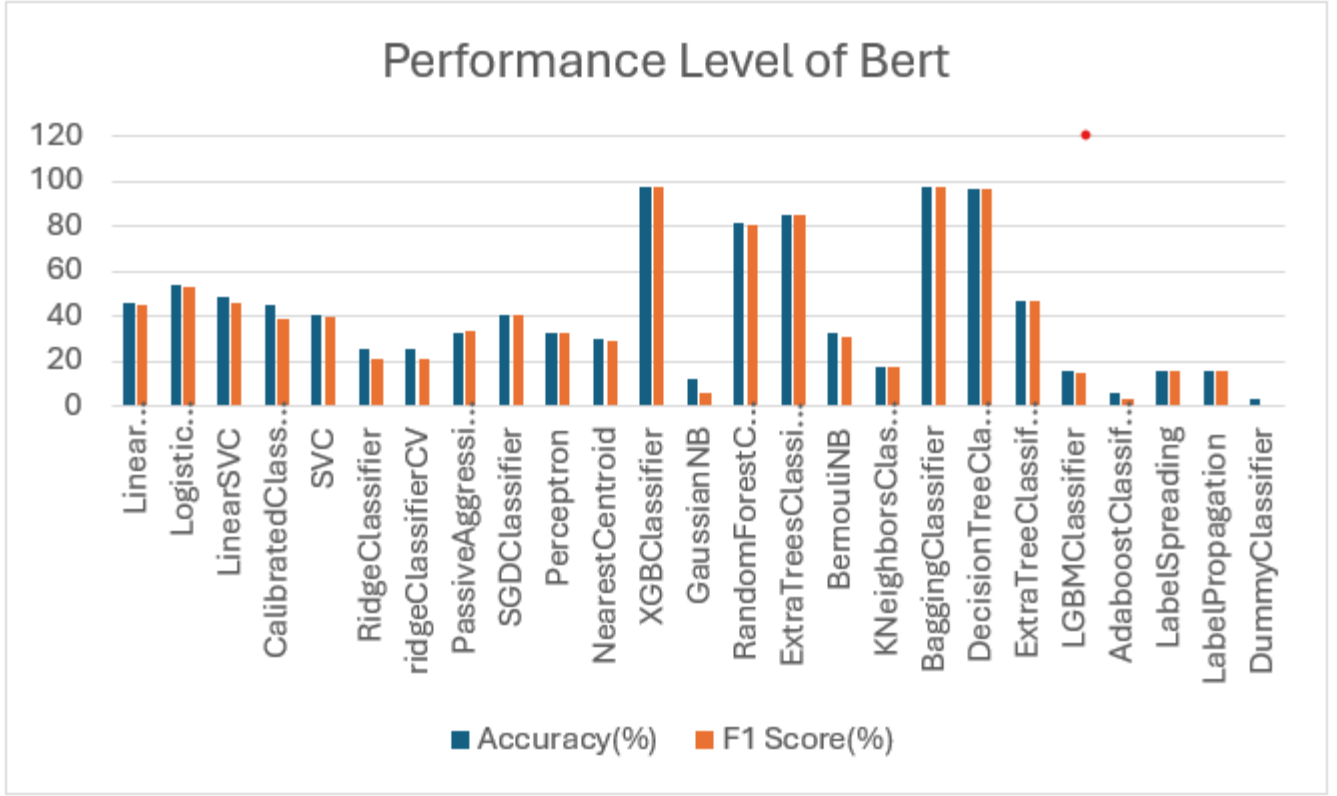
Application Domain Classification with ML classifier

Approach 1.1:

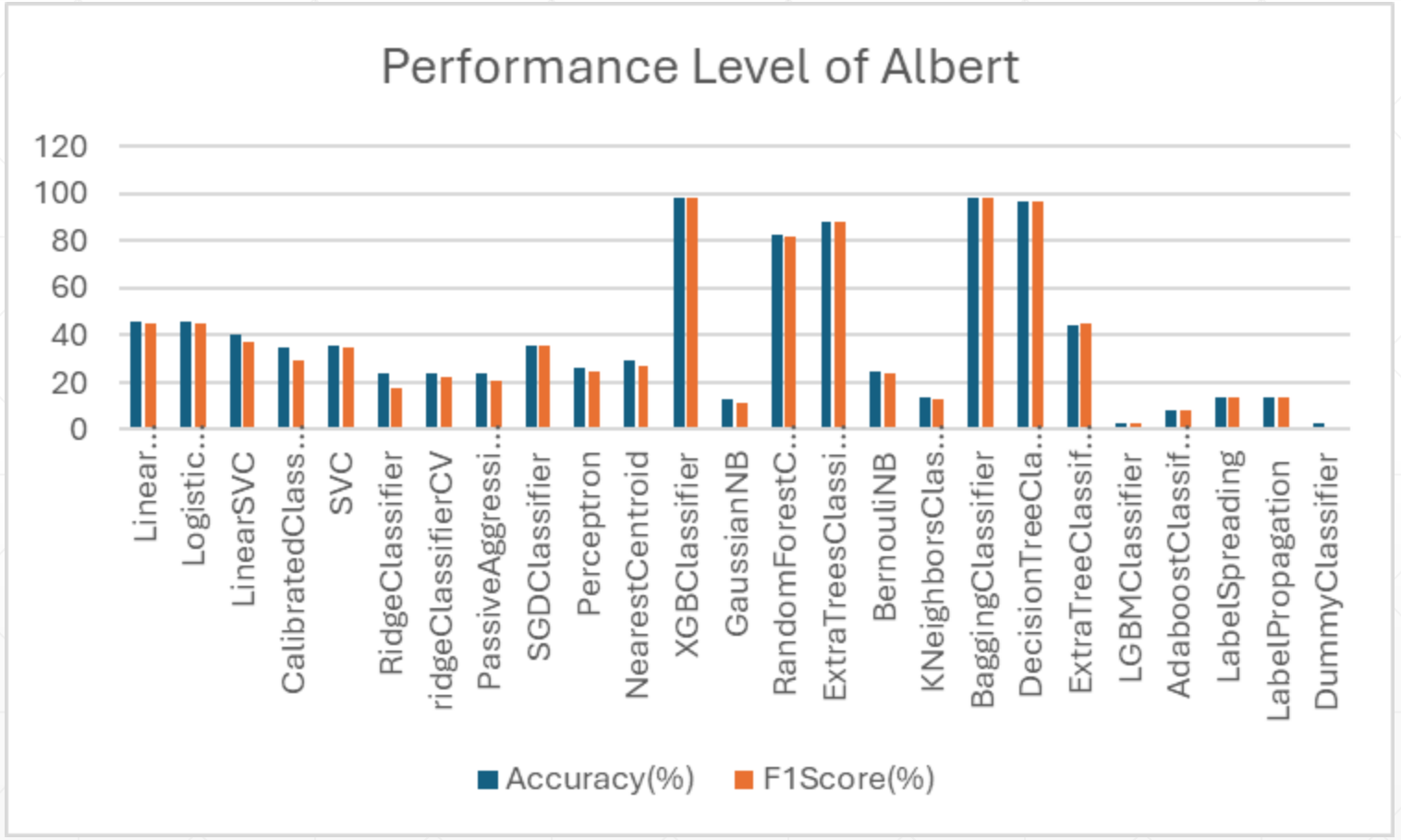
Initially we set the test size 0.5 and then measure the performance level of 25 ML classifiers with different word embedding techniques.



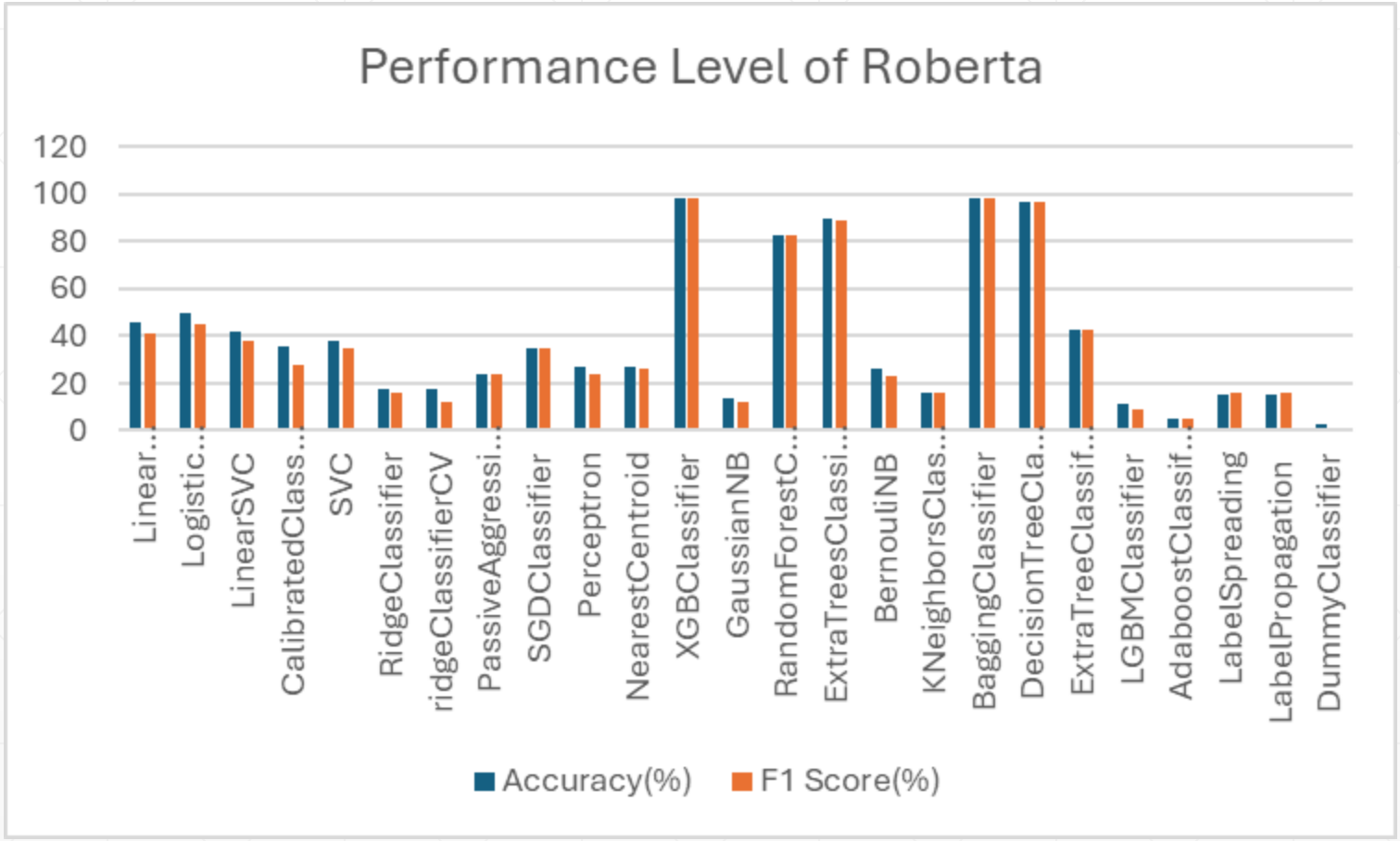
Application Domain Classification with ML classifier



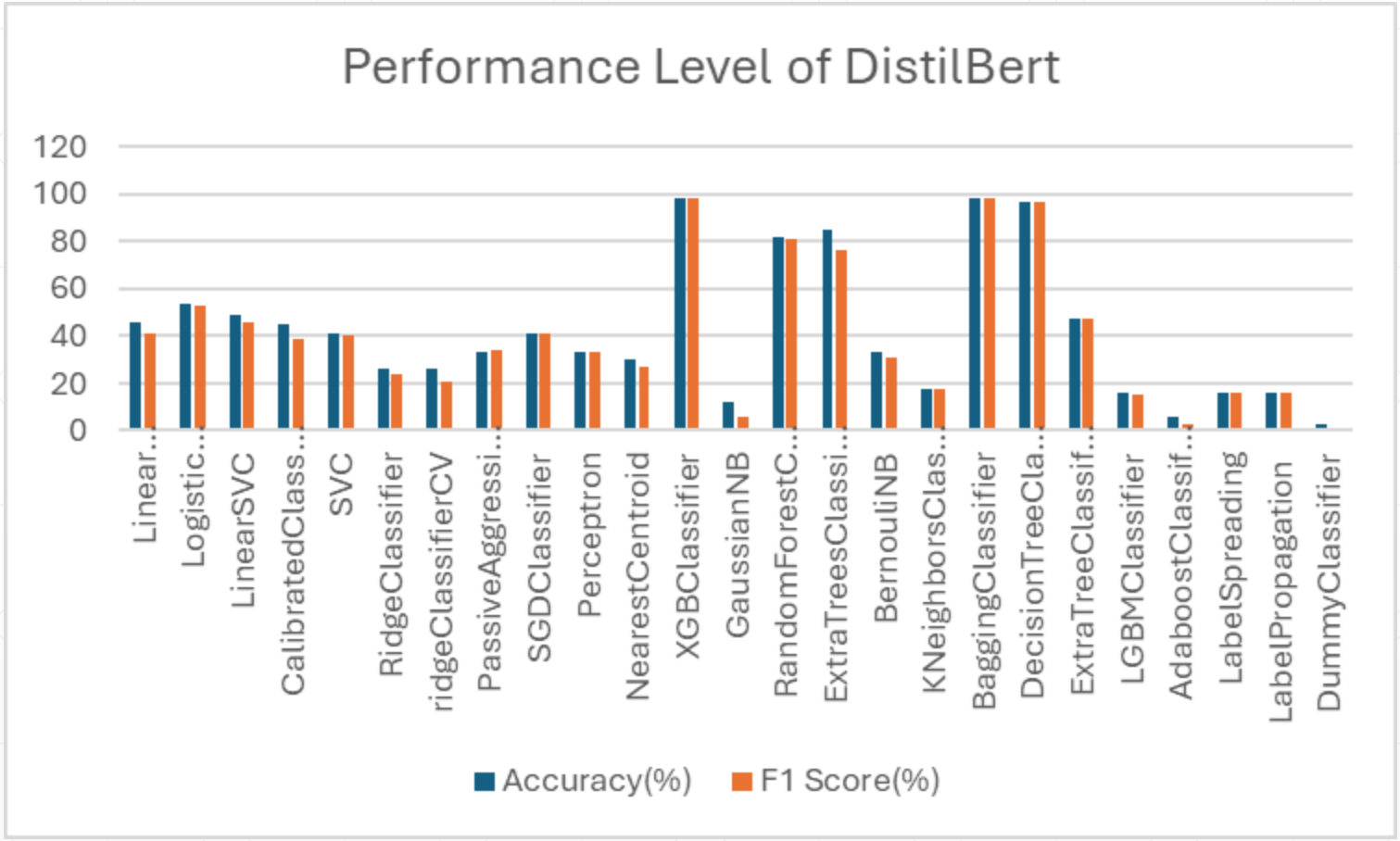
Application Domain Classification with ML classifier



Application Domain Classification with ML classifier



Application Domain Classification with ML classifier



Application Domain Classification with ML classifier

Approach 1.2: Later we have employed 10 fold cross validation technique with that 25 classifiers as well as the Bert versions for the reliability of the result.

Distil Bert		
Classifier	Accuracy	f1 score
XGBClassifier	99	99
BaggingClassifier	98.6	98.2
DecisionTreeClassifier	98.3	98.3

Albert		
Classifier	Accuracy	f1 score
XGBClassifier	99	99
BaggingClassifier	98.5	98.5
DecisionTreeClassifier	98.1	98.1

Roberta		
Classifier	Accuracy	f1 score
XGBClassifier	98.6	98.6
BaggingClassifier	98.3	98.3
DecisionTreeClassifier	98.1	98.1

Comparison in Domain Classification

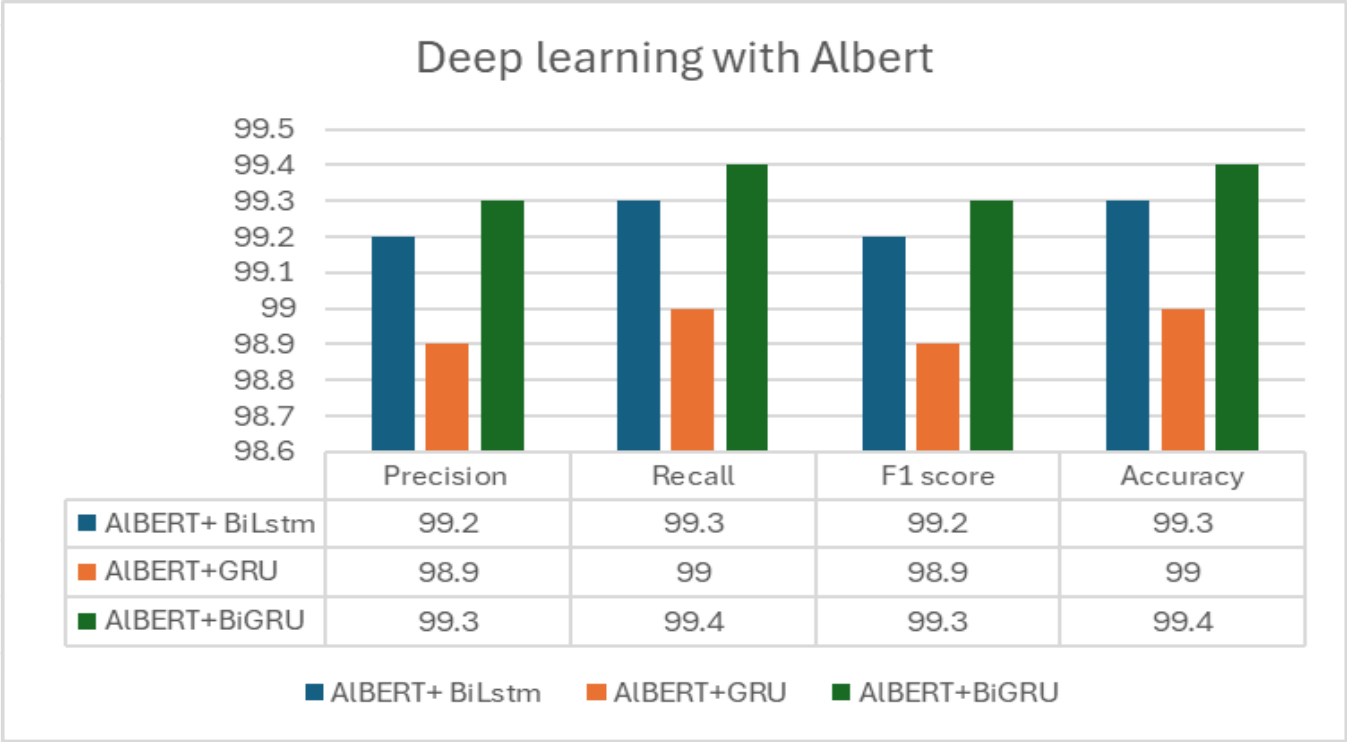
Our RQ1 achieved here. we get the better result in the domain classification task which will eventually lead better sensitive feature extraction.

Column1	Column2	Column3	Column4	Column5	Column6	Column7
Their framework				Our framework		
Classifier	Accuracy	F1 Score		Classifier	Accuracy	F1 Score
XGBClassifier	98	98		XGBClassifier	99	99
BaggingClassifier	98	98		BaggingClassifier	98.6	98.2
DecisionTreeClassifier	98	98		DecisionTreeClassifier	98.3	98.3

Deep Learning Classifier in Domain Classification

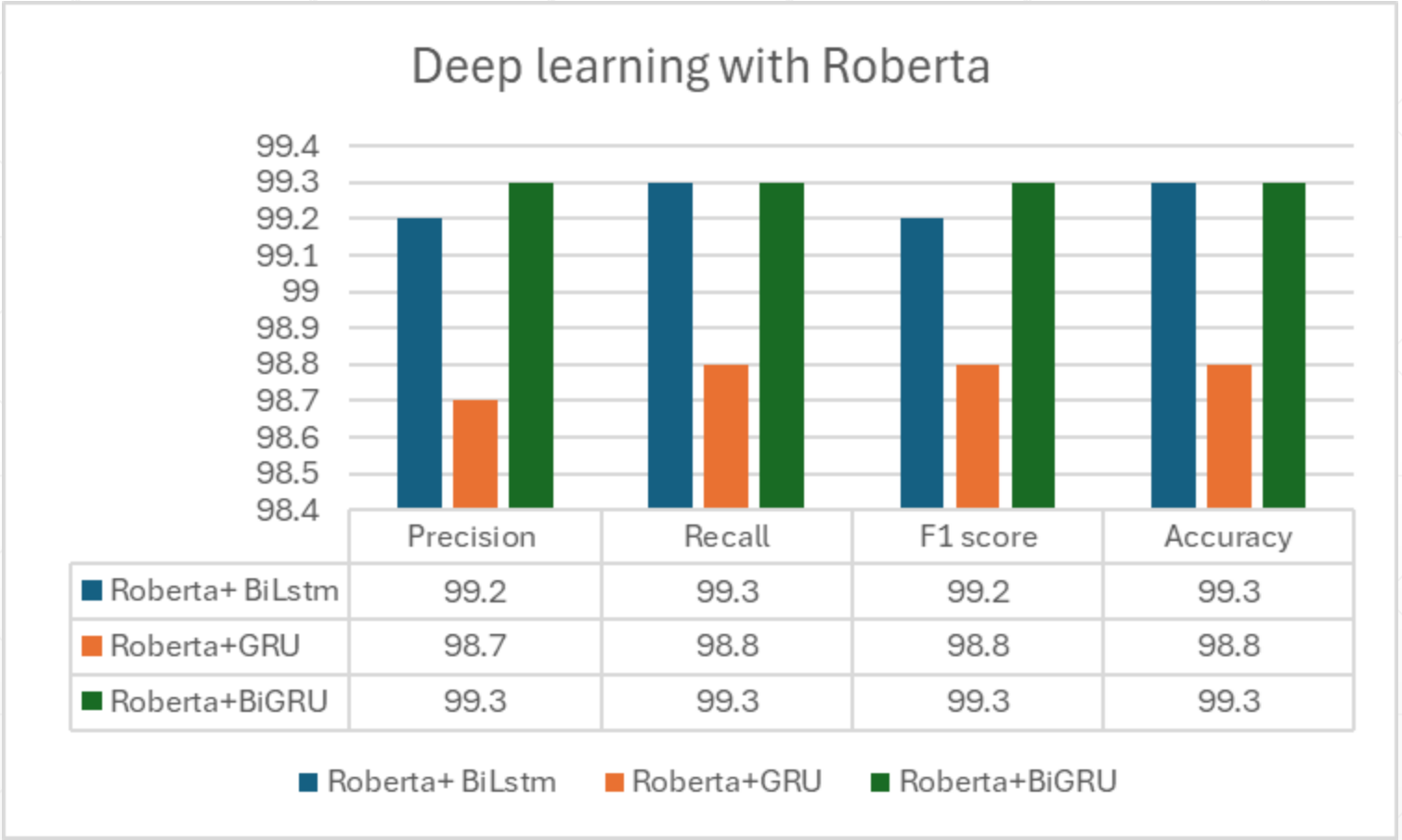
Approach 2: To enhance the domain detection part, we have applied three Deep learning algorithms(variants of RNN model) as in BiLSTM, BiGRU, GRU with three advanced Bert versions.

Here we are using 4 evaluation metrics unlike previous framework.



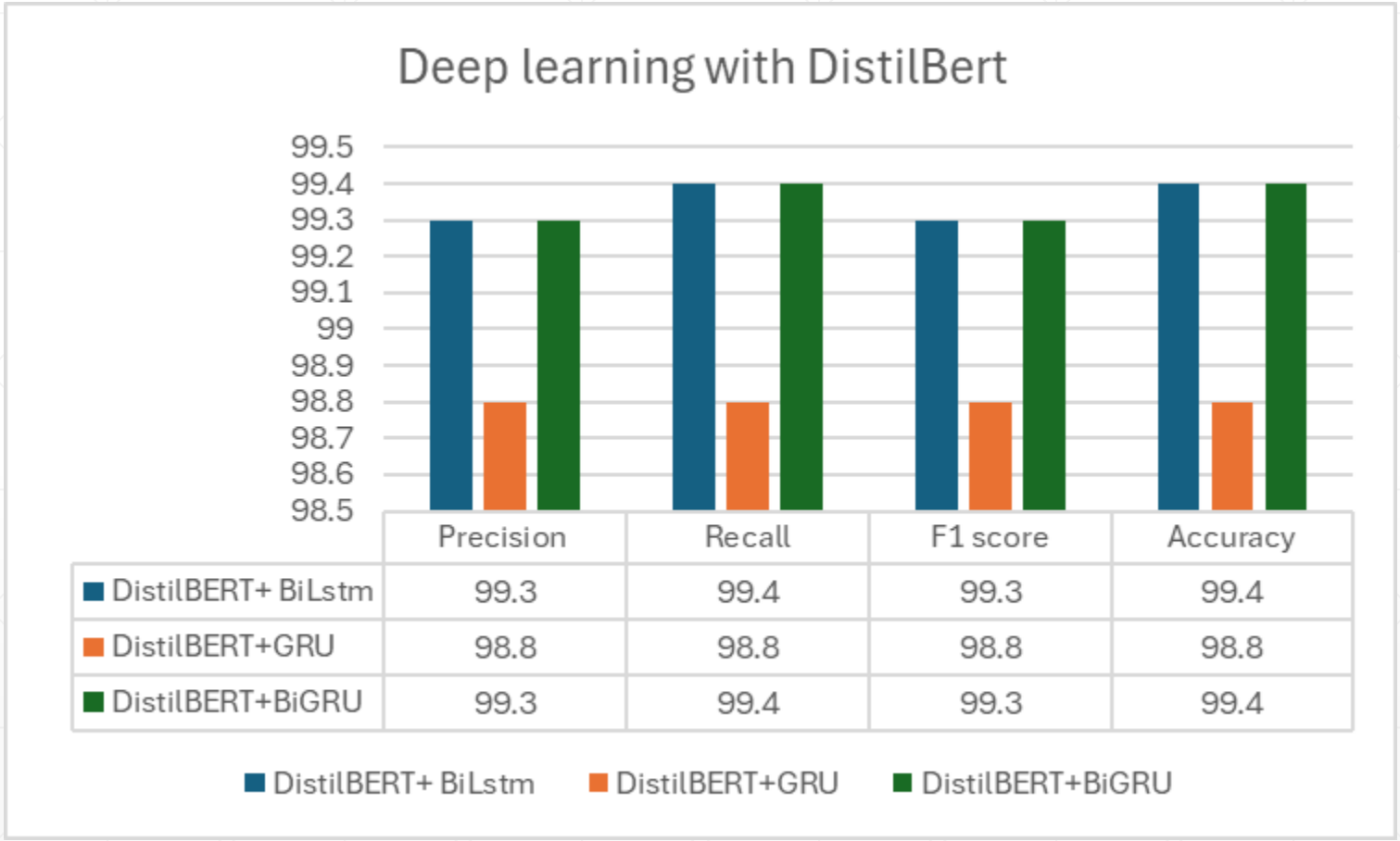
	Precision	Recall	F1 score	Accuracy
AlBERT+ BiLstm	99.2	99.3	99.2	99.3
AlBERT+GRU	98.9	99	98.9	99
AlBERT+BiGRU	99.3	99.4	99.3	99.4

Deep Learning Classifier in Domain Classification



	Precision	Recall	F1 score	Accuracy
Roberta+ BiLstm	99.2	99.3	99.2	99.3
Roberta+GRU	98.7	98.8	98.8	98.8
Roberta+BiGRU	99.3	99.3	99.3	99.3

Deep Learning Classifier in Domain Classification



	Precision	Recall	F1 score	Accuracy
DistilBERT+ BiLstm	99.3	99.4	99.3	99.4
DistilBERT+GRU	98.8	98.8	98.8	98.8
DistilBERT+BiGRU	99.3	99.4	99.3	99.4

Comparison in Domain Classification

We have satisfied our RQ3 by getting enhanced result in identifying domain classification task. We compare our best combination classifier with their best combination.

Their framework			Ourframework		
Classifier	F1 score	Accuracy	Classifier	F1 score	Accuracy
BERT+ XGBClassifier	98	98	DistilBERT+ BiLstm	99.3	99.4
BERT+ BaggingClassifier	98	98	AlBERT+GRU	98.9	99
BERT+DecisionTreeClassifier	98	98	DistilBERT+BiGRU	99.3	99.4

Integration of BERT Variants with ML Models

Approach 1:

ML Task Classification: We exploited different version of Bert i.e. Albert, Robert and Distill bert along with combination with various ML algorithm such as Decision tree, Random Forest Classifier, LogisticRegression etc.

This ML classification task is Multilable classification for that we are using PowerLabelSet, BinaryRelevance, ClassifierChain in combination with Robert,Albert and DistillBert to see if these new variants is surpassing the current framework capabilities and to answer RQ2.

We have performed 45 experiments with different combination of multilable classification algorithm with variants of bert and ML models and extracted F1 score and hamming loss.

Integration of BERT Variants with Deep Learning Models

Approach 2: BERT Variants with Deep Learning Models

Variants Used:
ALBERT, RoBERTa, DistilBERT.

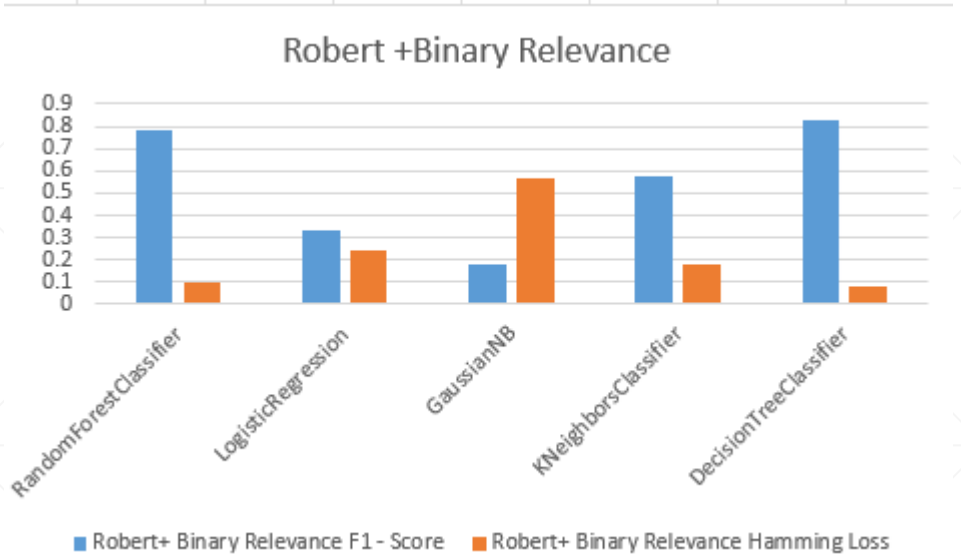
Deep Learning Models:
BiLSTM, BiGRU, GRU.

Key Result:

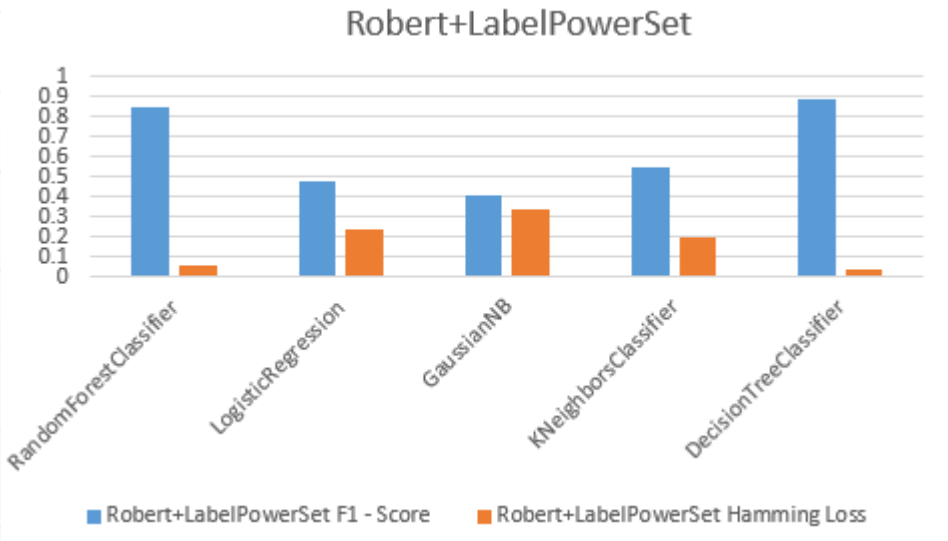
- Successfully classified ML tasks as a multilabel classification problem.
- Outperformed ML algorithms with significant F1 score and hamming loss improvements.
- Conducted 9 experiments that demonstrated enhanced efficiency of our proposed framework.

Results so far(Approach 1)

Robert+ Binary Relevance		
ML Model	F1 - Score	Hamming Loss
RandomForestClassifier	0.779	0.095
LogisticRegression	0.33	0.239
GaussianNB	0.18	0.57
KNeighborsClassifier	0.58	0.18
DecisionTreeClassifier	0.83	0.08

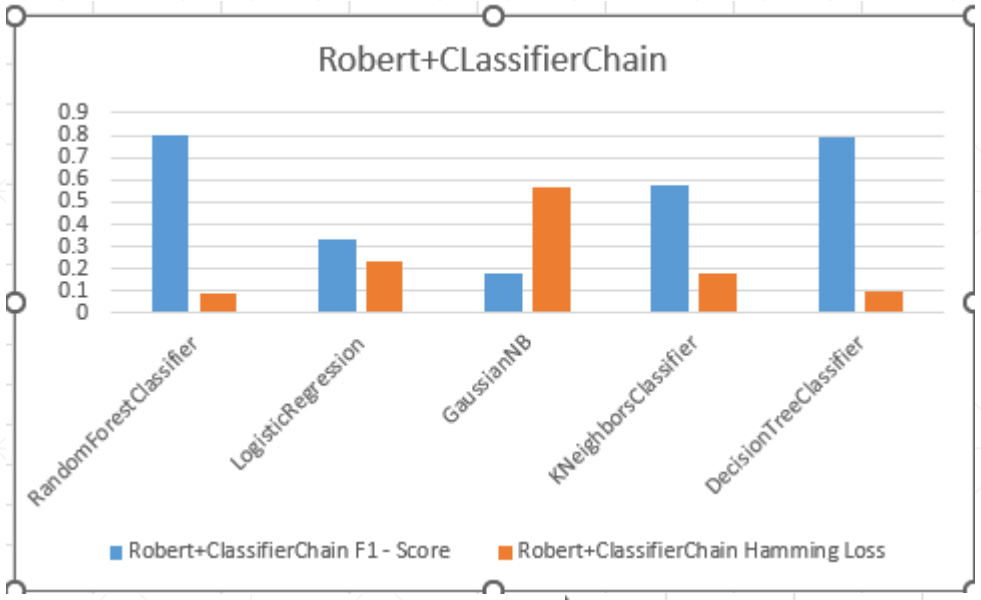


Robert+LabelPowerSet		
ML Model	F1 - Score	Hamming Loss
RandomForestClassifier	0.85	0.06
LogisticRegression	0.48	0.24
GaussianNB	0.41	0.34
KNeighborsClassifier	0.55	0.2
DecisionTreeClassifier	0.89	0.04

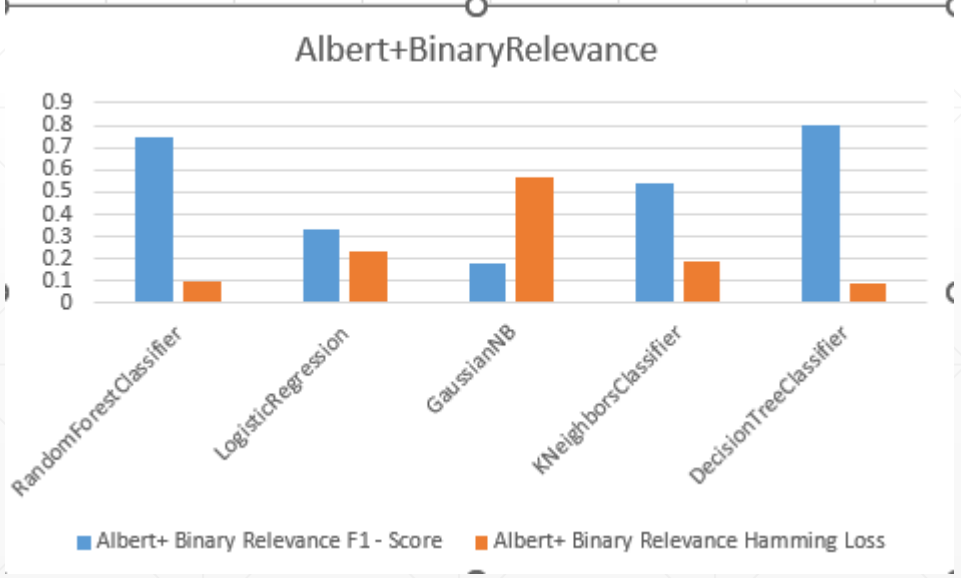


Results so far(Approach 1)

Robert+ClassifierChain		
ML Model	F1 - Score	Hamming Loss
RandomForestClassifier	0.8	0.09
LogisticRegression	0.33	0.23
GaussianNB	0.18	0.57
KNeighborsClassifier	0.58	0.18
DecisionTreeClassifier	0.79	0.1

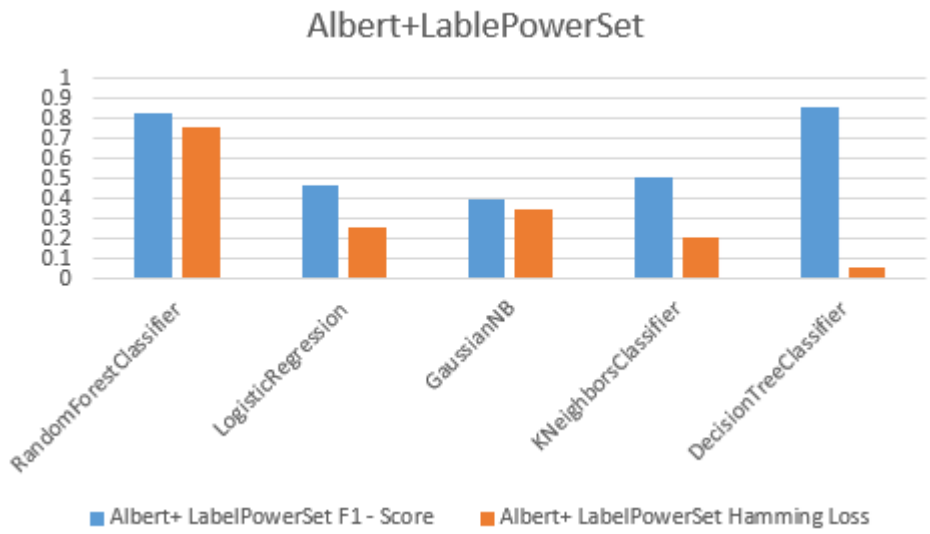


Albert+ Binary Relevance		
ML Model	F1 - Score	Hamming Loss
RandomForestClassifier	0.75	0.1
LogisticRegression	0.33	0.23
GaussianNB	0.18	0.57
KNeighborsClassifier	0.54	0.19
DecisionTreeClassifier	0.8	0.09

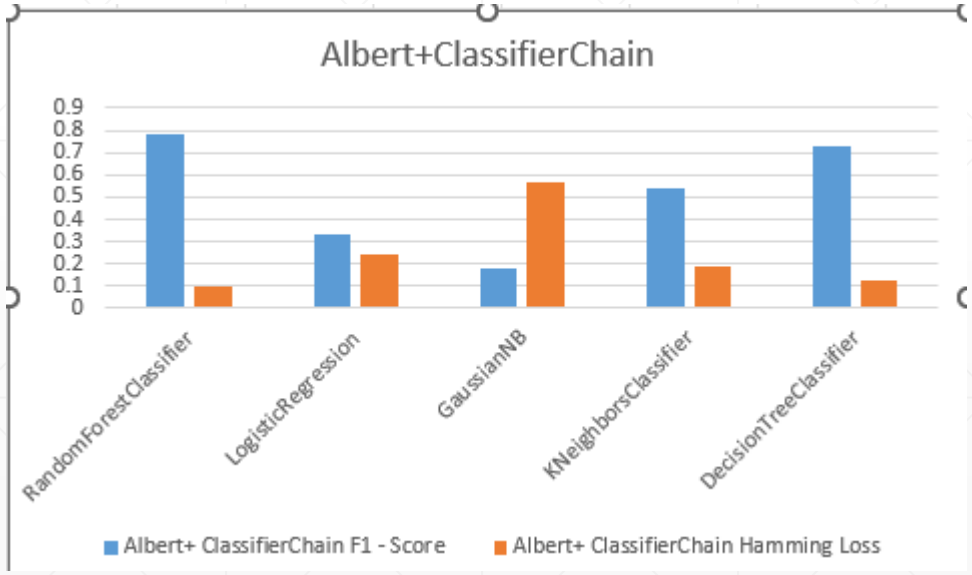


Results so far(Approach 1)

Albert+ LabelPowerSet		
ML Model	F1 - Score	Hamming Loss
RandomForestClassifier	0.83	0.76
LogisticRegression	0.47	0.26
GaussianNB	0.4	0.35
KNeighborsClassifier	0.51	0.21
DecisionTreeClassifier	0.86	0.06

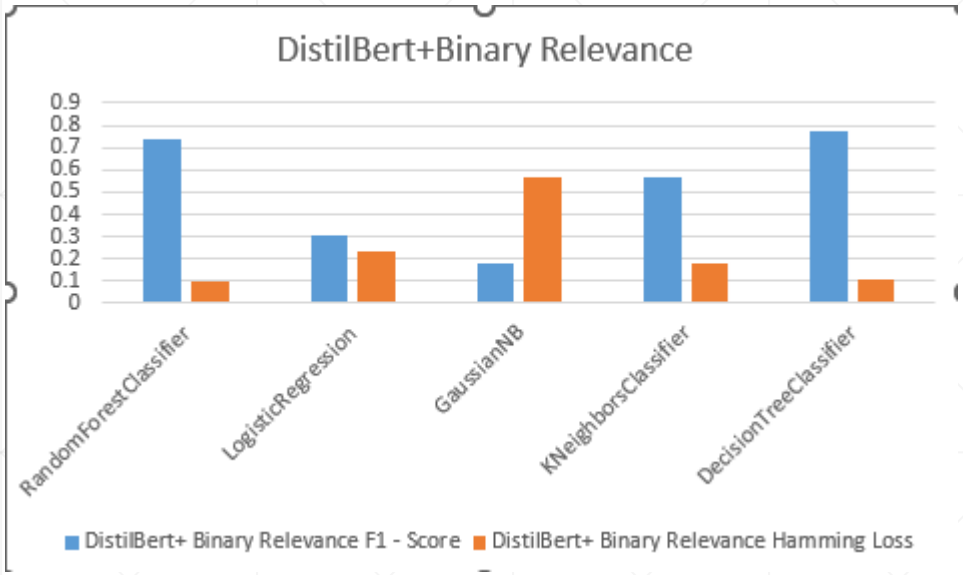


Albert+ ClassifierChain		
ML Model	F1 - Score	Hamming Loss
RandomForestClassifier	0.78	0.1
LogisticRegression	0.33	0.239
GaussianNB	0.18	0.57
KNeighborsClassifier	0.54	0.19
DecisionTreeClassifier	0.73	0.13

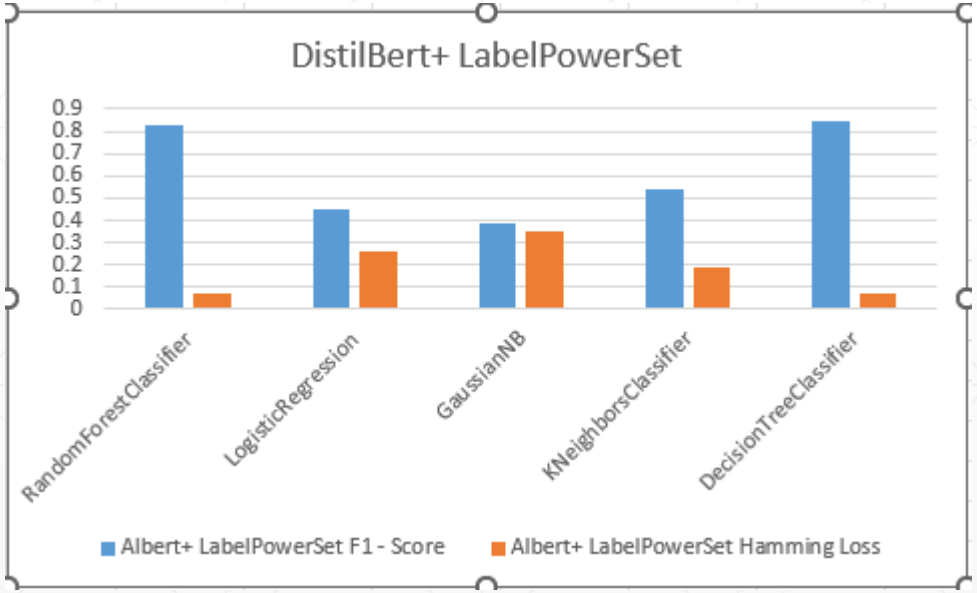


Results so far(Approach 1)

DistilBert+ Binary Relevance		
ML Model	F1 - Score	Hamming Loss
RandomForestClassifier	0.74	0.1
LogisticRegression	0.31	0.23
GaussianNB	0.18	0.57
KNeighborsClassifier	0.57	0.18
DecisionTreeClassifier	0.77	0.11

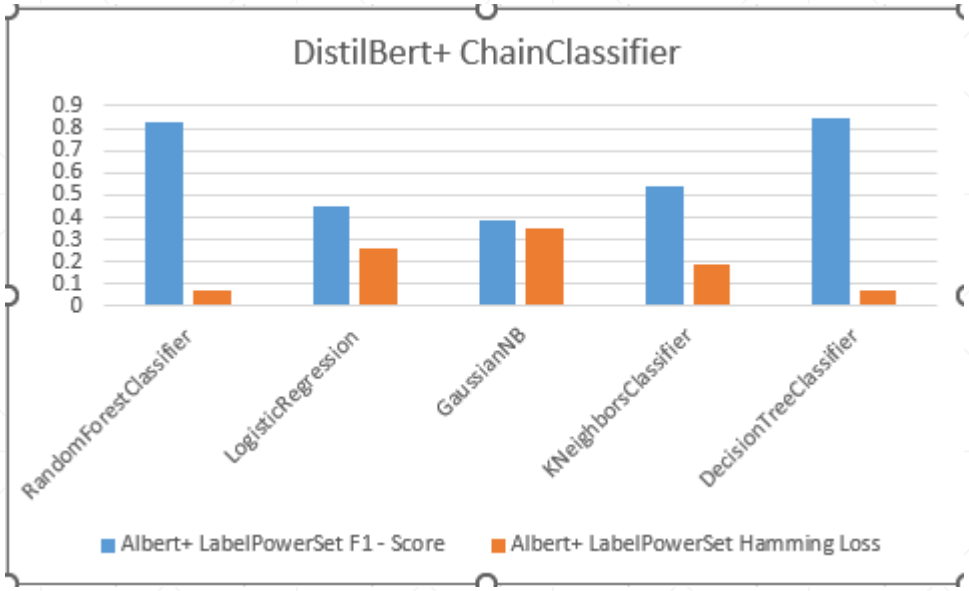


DistilBert+ LabelPowerSet		
ML Model	F1 - Score	Hamming Loss
RandomForestClassifier	0.83	0.07
LogisticRegression	0.45	0.26
GaussianNB	0.39	0.35
KNeighborsClassifier	0.54	0.19
DecisionTreeClassifier	0.85	0.07



Results so far(Approach 1)

DistilBert+ ClassifierChain		
ML Model	F1 - Score	Hamming Loss
RandomForestClassifier	0.77	0.1
LogisticRegression	0.31	0.23
GaussianNB	0.18	0.57
KNeighborsClassifier	0.57	0.18
DecisionTreeClassifier	0.69	0.06



Comparison in ML Task Classification

Our RQ2 achieved here. we get the better result in the ML classification task which will eventually lead better sensitive feature extraction.

Combination of Robert + LabelPowerSet+ Decision tree has given us enhanced F1 score and Hamming loss.

Current Framework

BERT		
Technique + Model	F1-Score	Hamming Loss
LP + DT	0.86	0.07
LP + RF	0.84	0.08
BR + DT	0.78	0.11

Our Framework

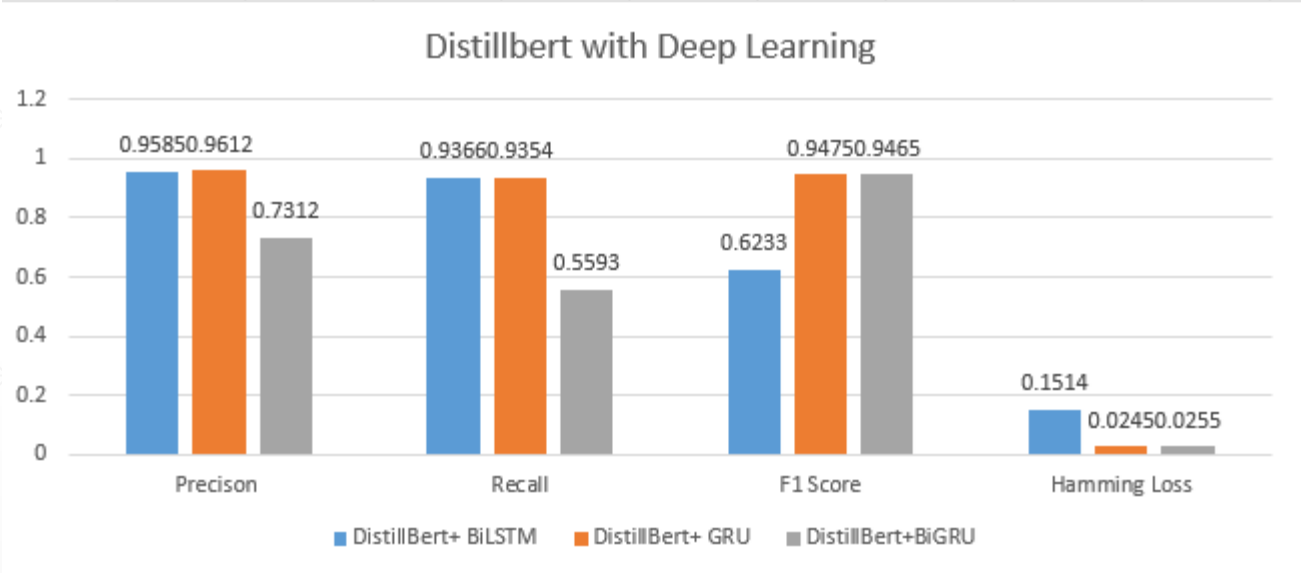
Robert+LabelPowerSet		
ML Model	F1 - Score	Hamming Loss
RandomForestClassifier	0.85	0.06
LogisticRegression	0.48	0.24
GaussianNB	0.41	0.34
KNeighborsClassifier	0.55	0.2
DecisionTreeClassifier	0.89	0.04

Deep Learning Classifier in ML Task Classification

Approach 2:

Distillbert with Deep Learning

Model Combination	Precison	Recall	F1 Score	Hamming Loss
DistillBert+ BiLSTM	0.9585	0.9366	0.6233	0.1514
DistillBert+ GRU	0.9612	0.9354	0.9475	0.0245
DistillBert+BiGRU	0.7312	0.5593	0.9465	0.0255

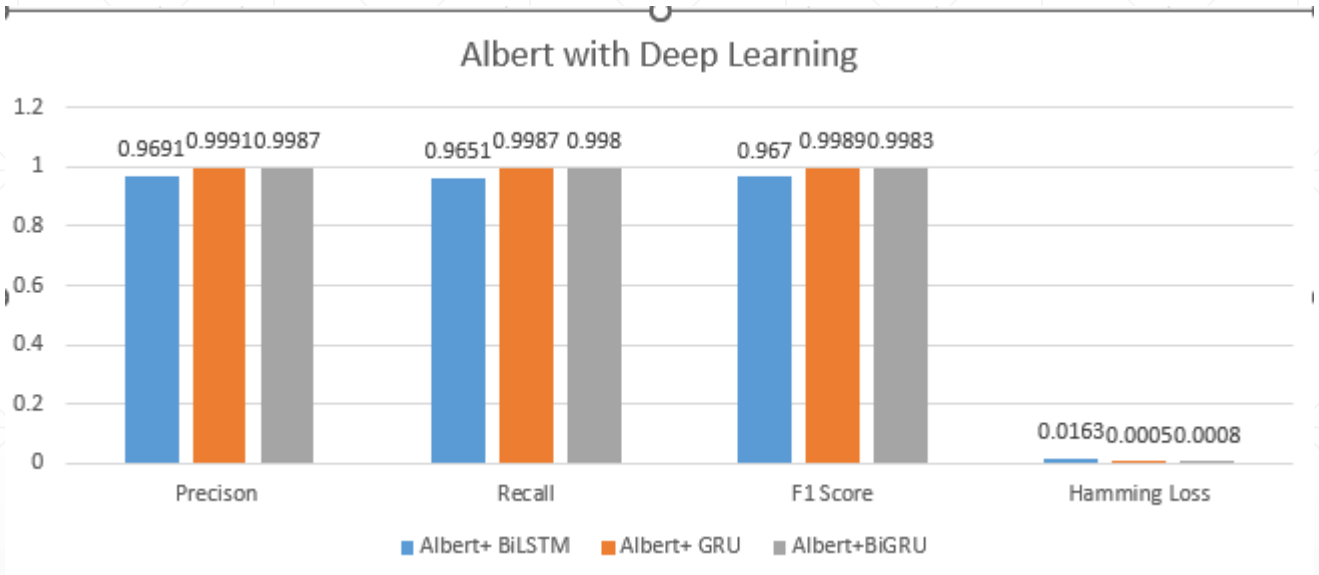


Deep Learning Classifier in ML Task Classification

Approach 2:

Albert with Deep Learning

Model Combination	Precison	Recall	F1 Score	Hamming Loss
Albert+ BiLSTM	0.9691	0.9651	0.967	0.0163
Albert+ GRU	0.9991	0.9987	0.9989	0.0005
Albert+BiGRU	0.9987	0.998	0.9983	0.0008

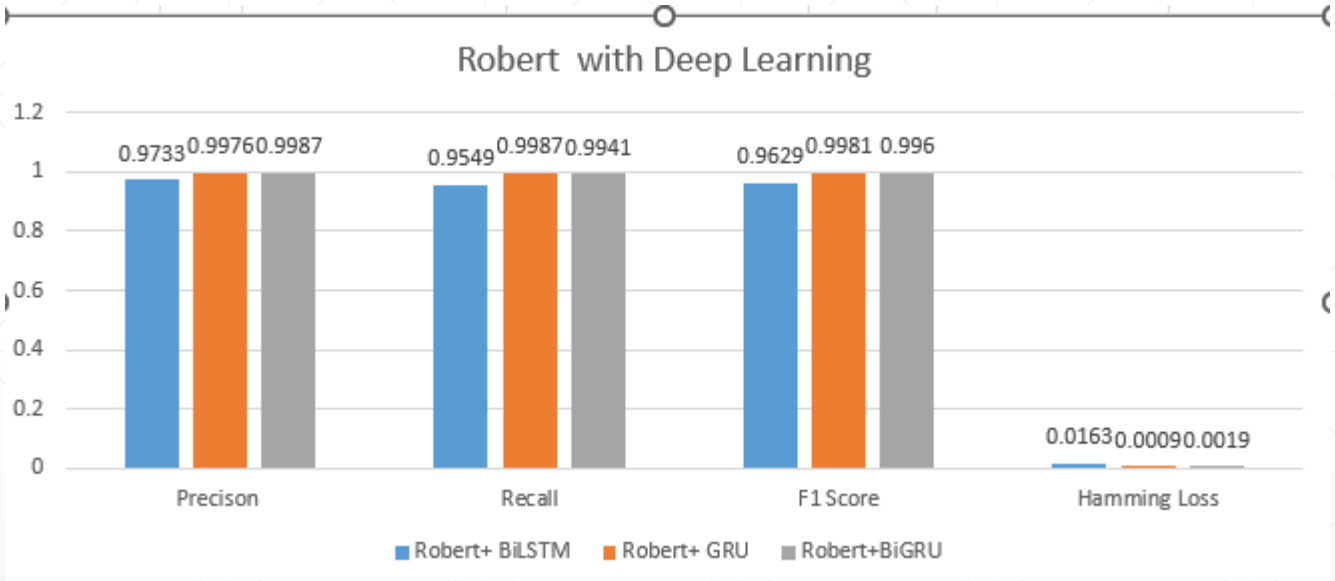


Deep Learning Classifier in ML Task Classification

Approach 2:

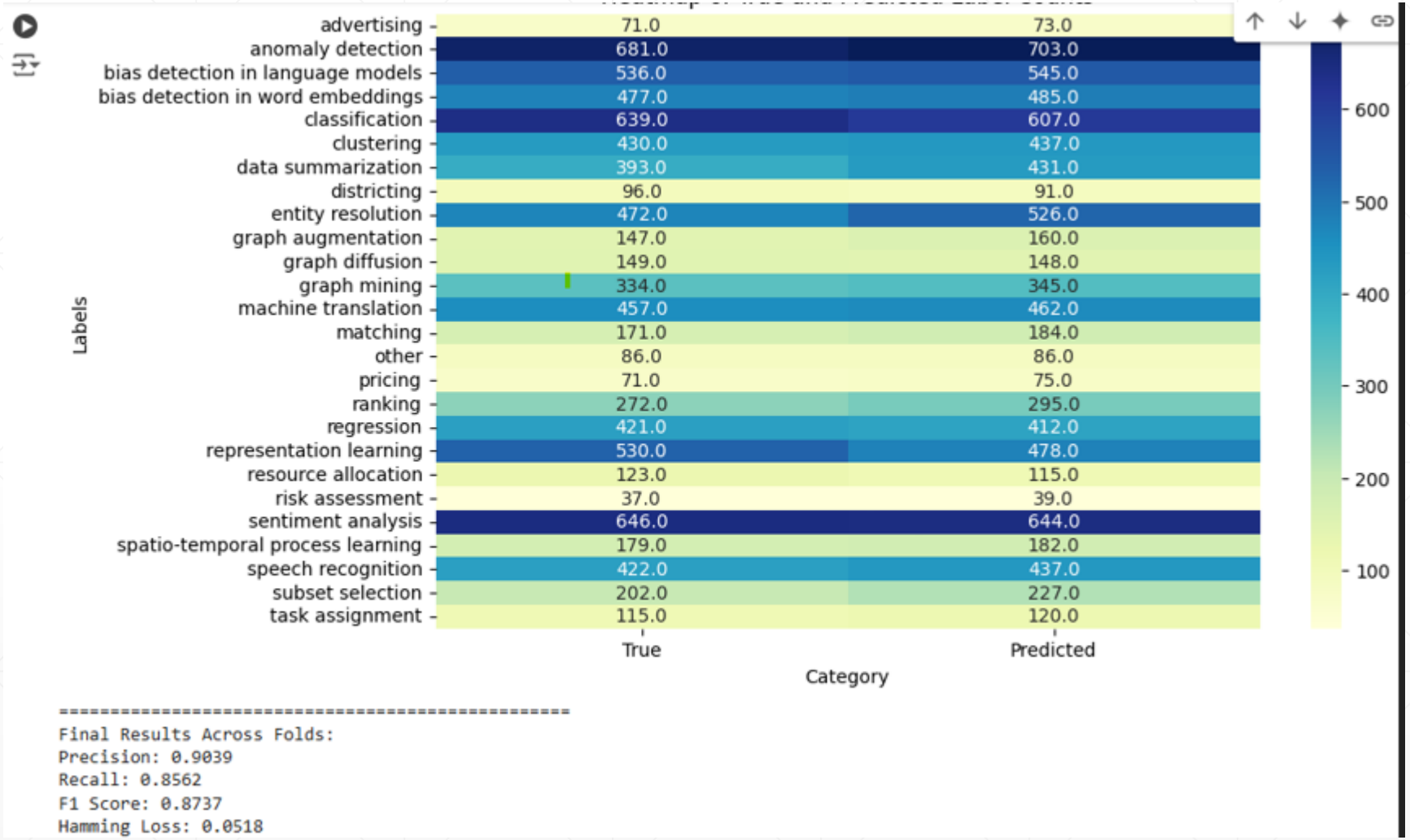
Robert with Deep Learning

Model Combination	Precison	Recall	F1 Score	Hamming Loss
Robert+ BiLSTM	0.9733	0.9549	0.9629	0.0163
Robert+ GRU	0.9976	0.9987	0.9981	0.0009
Robert+BiGRU	0.9987	0.9941	0.996	0.0019



HeatMap of misclassified labels

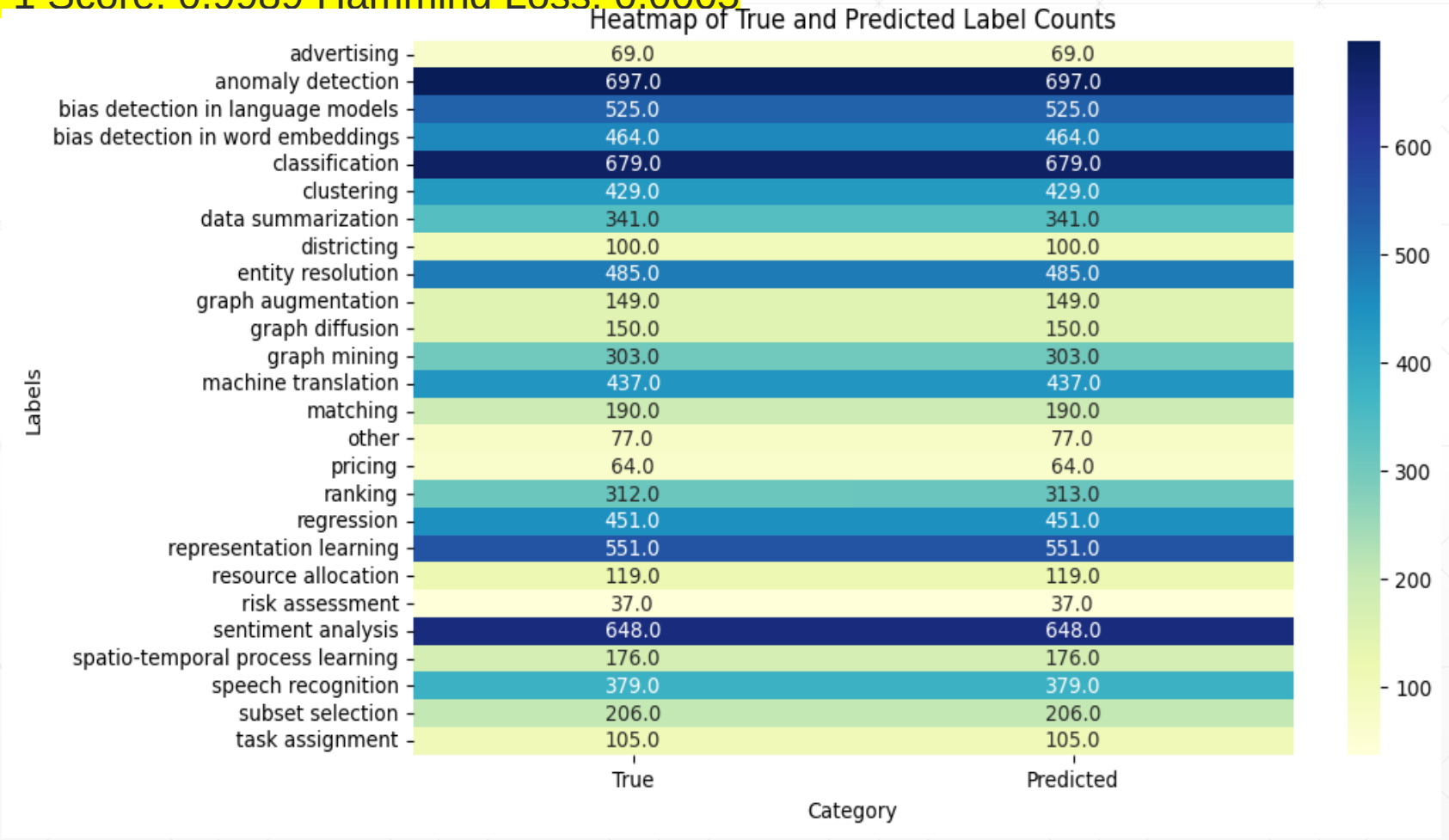
Distillbert+BiLSTM




```
==================================================
Final Results Across Folds:
Precision: 0.9039
Recall: 0.8562
F1 Score: 0.8737
Hamming Loss: 0.0518
```

HeatMap for 26 Multi-Label ML classification

Albert+GRU F1 Score: 0.9989 Hamming Loss: 0.0005



Comparison in ML Task Classification

We have satisfied our RQ4 by getting enhanced result in identifying MT classification task. We compare our best combination classifier with their best combination.

Current Framework

Glove		
Technique + Model	F1-Score	Hamming Loss
LP + LSVC	0.90	0.05
LP + GNB	0.72	0.15
BR + KNN	0.67	0.15

Our Framework

Model Combination	F1 Score	Hamming Loss
Albert+ GRU	0.9989	0.0005
Albert+BiGRU	0.9983	0.0008
Robert+ GRU	0.9981	0.0009
Robert+BiGRU	0.996	0.0019

Resolved Challenges

- ➔ Deep learning models often have many hyperparameters (e.g., learning rate, optimizer) that has been carefully tuned for optimal performance.

- ➔ Employing Deep learning along with 10 fold cross validation technique was computationally expensive, spent significant time to find results.

Conclusion

- In this work, we enhanced the classification of machine learning (ML) tasks and application domains from user stories by integrating advanced word embedding techniques and deep learning models.
- Building upon insights from existing studies, we utilized variants of BERT (Albert, RoBERTa, and DistilBERT) with deep learning architectures such as BiLSTM, GRU, and BiGRU.
- Among the combinations tested, DistilBERT+BiGRU achieved the best performance for domain classification with 99.4% accuracy and 99.3% F1 score, while Albert+GRU excelled in ML task classification with a 0.9989 F1 score and 0.0005 Hamming loss.
- Our findings demonstrate that deep learning surpasses traditional methods, providing more accurate and nuanced insights for early-stage ML task and domain detection.

Future Work

Enhancing Dataset Diversity

- ➔ Utilize authentic and diverse user stories instead of synthetic datasets.
- ➔ Incorporate real-world data from various domains to improve model robustness.

Addressing Dataset Limitations

- ➔ Current synthetic dataset, while effective, lacks variety.
- ➔ Authentic user stories will help in validating and refining the framework for real-world scenarios.

THANK YOU

Q & A