SML Assignment 2 Report

University of Melbourne

Group Members
Alois Wang (1297543)
Xun Zhao (1297646)
Haoyang Du (1346554)

1 Introduction

Text generation has become an increasingly popular task with the rise of natural language processing (NLP) techniques and advancements in deep learning. While text generation has a wide range of applications, including chatbots, language translation, and content creation, it also poses significant challenges in ensuring content authenticity and authoritativeness. This project aims to develop a machine learning model capable of detecting whether a given text sample is human- or machine-generated, leveraging domain-generalised strategies to perform well across two distinct datasets.

2 Final Approach

2.1 Model Architecture

To enhance generalization across domains, we implemented an *Optimal Domain Adversarial Neural Network* (DANN) with a conservative architecture. The main idea behind DANN is to enjoin the network's hidden layer to learn a representation that is predictive of the source example labels, but uninformative about the domain of the input (source or target) [Ganin et al., 2016].
Our implementation comprises three components: a feature extractor, a label classifier, and a domain classifier. The feature extractor consists of a single hidden layer with ReLU activation and dropout (rate 0.1) for regularization, followed by linear layers for label and domain classification. We adopted a moderate hidden dimension size of 160 and set the Gradient Reversal Layer (GRL)'s penalty coefficient $\alpha = 0.03$ to softly discourage domain-specific representations, thereby promoting robustness to softly discourage domain-specific representations, promoting robustness to distribution shifts between domain1 and domain2.

2.2 Feature Engineering

For feature engineering, instead of using Bag-of-Words representations, we applied *TF-IDF* vectorization to convert the preprocessed text data into numerical feature vectors suitable for machine learning. We configured the `TfidfVectorizer` with conservative parameters to reduce overfitting, limiting the vocabulary size to a maximum of 3,000 features.
After fitting the vectorizer on the combined augmented training data from both domains, we transformed the training and validation sets accordingly. To further reduce dimensionality and retain the most informative features, we applied chi-squared feature selection using `SelectKBest`, selecting the top 2,000 features. This strategy balances expressiveness and generalization, ensuring robustness across domains while minimizing noise and computational complexity.

2.3 Data Augmentation

To address data imbalance and support robust model evaluation, we first split each domain's dataset into training and validation sets using stratified sampling, ensuring that class distributions were preserved.
Given the significant label imbalance in domain2, where human-generated samples are underrepresented, we applied a simple text augmentation strategy to artificially increase data diversity. This method involves randomly shuffling a small portion (20%) of tokens in longer texts to generate slightly altered versions while preserving the original label.
We adjusted the augmentation ratio per domain—40% for the smaller, balanced domain1, and 10% for the larger, imbalanced domain2—to avoid further skewing the class distributions. The resulting augmented training data was then used to enrich the dataset and support better generalization during model training.

2.4 Training Procedure

The model training process incorporates several key strategies:

- **Optimizer**: Adam with learning rate $\eta = 0.0012$ and weight decay $\lambda = 10^{-5}$
- **Training Epochs**: 20, with early stopping (patience = 3)
- **Domain Adversarial Strength**: $\alpha = 0.03$
- **Evaluation Weighting**: 60% Domain1, 40% Domain2

The overall training pipeline begins by applying TF-IDF vectorization to preprocess the input data. We then combine the datasets from both domains and split them into training and validation sets using stratified sampling to preserve label balance. After addressing class imbalance, we train the DANN model on the training set and evaluate it on the validation set.
Once the architecture and hyperparameters are finalized, we retrain the model on the entire training dataset to maximize generalization before generating test predictions.

3 Alternative Approaches

We evaluated several classical models to understand their limitations under domain shift and label imbalance, which motivated our final model choice.

Logistic Regression A linear model with 17K parameters, achieving 90.3% accuracy. However, it lacked the capacity to capture non-linear patterns or adapt across domains, making it fragile under domain shift [Tibshirani and Manning, 2014].

Random Forest Trained with 300 trees, it handled imbalance well (90.2% in Domain2), but dropped to 87.5% in Domain1 due to absence of domain-invariant learning [Chen et al., 2004].

SVM with RBF kernel SVM showed decent performance (88–89.5%) and compactness, but still failed under distribution shift, as it lacks flexibility to learn transferable representations [Singla and Shukla, 2020].

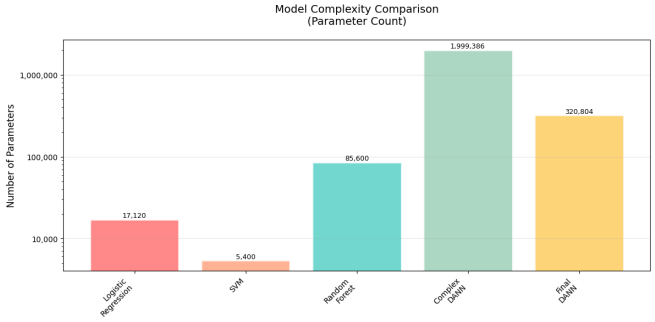
Complex DANN This deeper model overfit despite high validation accuracy, dropping to 80.7% on test data. It highlighted the need for a moderately expressive, stable architecture—leading to our final DANN design [Ganin et al., 2016].

These comparisons highlight the need for a moderately expressive, domain-invariant model. Hence, our final DANN with soft gradient reversal, dropout regularisation, and reduced hidden size balances generalisation and robustness across both domains.

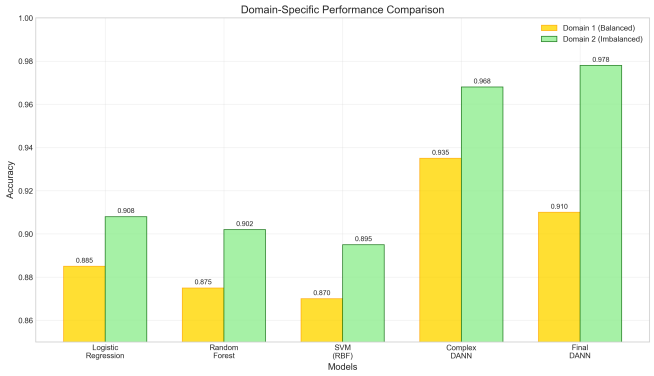
4 Domain Analysis

During our analysis, we observed a significant discrepancy between domain1 and domain2. Models trained exclusively on domain2 data performed poorly when applied to domain1, indicating that the underlying data distributions differ substantially between the two domains.
This observation further motivated the adoption of DANN as our final model. DANN leverages a *Gradient Reversal Layer* (GRL), which plays a critical role in learning domain-invariant features. The key idea behind GRL is to encourage the feature extractor to learn representations that are discriminative for the label prediction task, but invariant with respect to the domain.
By reversing the gradient from the domain classifier during backpropagation, the network is penalized for encoding domain-specific features, thus promoting better generalization across domains.

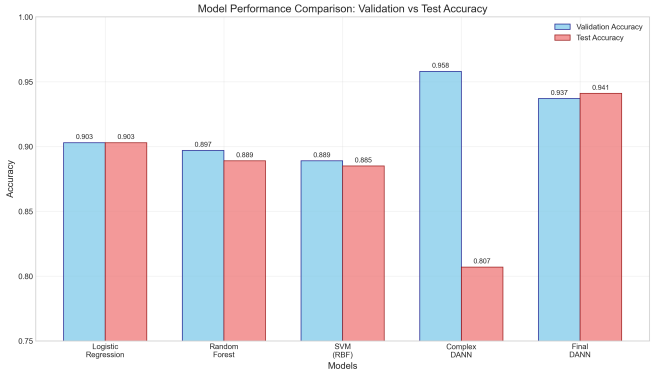
5 Results



(a) Model parameter counts



(b) Accuracy across domains



(c) Validation vs. test accuracy

Model	Acc.	Prec.	Recall	F1
LogReg	90.3	0.89	0.91	0.90
SVM	88.5	0.88	0.89	0.88
RF	88.9	0.87	0.90	0.88
CDANN	80.7	0.82	0.79	0.80
FinalD	**94.1**	**0.93**	**0.95**	**0.94**

(d) Performance metrics summary

Figure 1: Overview of model complexity, accuracy, generalisation, and performance metrics.

6 Conclusion

In conclusion, our final model demonstrated strong performance in handling the domain shift between two distinct datasets. Its ability to learn domain-invariant representations using the *Gradient Reversal Layer* (GRL) significantly improved generalization across domains, particularly when compared to traditional models. In addition, our feature engineering strategies—including TF-IDF vectorization and chi-squared feature selection—contributed to stable and interpretable performance. However, there remains room for improvement. Future work could explore more advanced augmentation techniques, domain-specific fine-tuning, or transformer-based architectures to further enhance both performance and robustness.

References

Tibshirani, J., and Manning, C.D. Robust Logistic Regression using Shift Parameters. In *Proceedings of ACL 2014 (Short Papers)*, pages 124–129.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, et al. Domain-Adversarial Training of Neural Networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016.

Chao Chen, Andy Liaw, and Leo Breiman. Using Random Forest to Learn Imbalanced Data. Technical Report, Department of Statistics, UC Berkeley, 2004.

Manisha Singla and K. K. Shukla. Robust statistics-based support vector machine and its variants: A survey. *Neural Computing and Applications*, 32:11173–11194, 2020.