Performance analysis of Word Embeddings for Cyberbullying Detection

Subbaraju Pericherla¹, E Ilavarasan²

¹ Research Scholar, Dept. of CSE, Pondicherry Engineering college, Pondicherry, India-605014.
² Professor, Dept. of CSE, Pondicherry Engineering college, Pondicherry, India-605014.
*Email: raju.pericherla74@gmail.com

Abstract. Cyber bullying activities are increasing day by day with the increase of Social Media Platforms such as Facebook, Twitter, Instagram etc. Bullies take the advantage of these large online connected platforms due to which it became as a big challenging task in Natural Language Processing (NLP). In this paper, we compare the performance of various word embedding methods from basic word embedding methods to recent advanced language models such as RoBERTa, XLNET, ALBERT, etc. for cyberbullying detection. We used LightGBM and Logistic regression classifiers for the classification of bullying and non-bullying tweets. Among all the models, RoBERTa is outperformed as compared to state-of-the-art models.

1. Introduction

The usage of social media is much more increasing from day to day with increase of digital devices like smartphones, tabs, and electronic gadgets. The most common social media platforms like Facebook, Twitter, Instagram, WhatsApp etc. are majorly used to share the messages and express their opinions. According to Statista® website information, the number of social media users increased from 10 million to 260 million. These social media platforms give more freedom to users to express their ideas and feelings in different forms like text message, sharing image, audio, and graphics. Some of the people using these social media networks for anti-social activities like online trolling, cyberbullying etc.

In recent years, cyberbullying is a big challenging task in social media era. Cyberbullying is a kind of online hate speech by intentionally posting hate kind of text messages, images, audios and videos[1]. The effect of cyberbullying is more dangerous than traditional bullying because, it reaches very large number of people in short time. The major victims of cyberbullying activities are school children and adolescents. The victims are psychologically affected, and sometimes lead to suicide attempts[2]. To save the bullying victims from these activities, some researchers developed machine learning algorithms that can predict cyberbullying text messages which are having humiliating content. In natural language processing tasks, the machine learning algorithms depends on word embeddings. Each text message must convert into numeric form by different word embedding techniques.
In this paper, we study the performance of word embedding techniques on cyberbullying detection on two classifier algorithms called Logistic regression and LightGBM. The major contributions of the paper as follows

➢ We conducted experiments on most advanced word embedding techniques in natural language processing research area.
➢ Compared the performance of word embeddings in three different sampling modes.
➢ Demonstrated the mis-classification rate of word embedding techniques.

2. Related work
In this section, we discussed about previous works on cyberbullying detection using various word embedding techniques.

M. Karan and J. Snajder [3] proposed a pre-emptive detection system for toxic language. They used support vector machine as classifier and tf-idf used for word embeddings. Xu et al.[4] used bag of words word embeddings and studied different roles of participants in cyberbullying activities such as victims, bullies, accuser. They achieved f1-score of 0.77 using support vector machine classifier. Nahar et al.[5] implemented graph model to estimates victim scores on Twitter dataset. They used tf-idf to extract features from the text. Pang et al.[6] employed three machine learning algorithms SVM, Navie Bayes, Maximum Entropy for sentiment classification on IMDb dataset to classify the movie review is positive or negative. They used Bag of words word embeddings technique. Zhao and Mao[7] proposed a method called Semantic-enhanced Marginalized Stacked Denoising Autoencoder (smSDA) for cyberbullying detection. This smSDA capable of reducing noise and able to identify best features from Bag of Words representation in effective and efficient way. Yen et.al [8] employed a supervised learning approach to identify online harassment in web 2.0. They combine different features like contextual features, sentiment features and content features of documents and trained on support vector machine classifier. The proposed technique outperforms the traditional tf-idf. Vikas et al.[9] included two additional features called pronoun capturing and skip-grams to the features extracted by tf-idf and n-gram. This method increases the accuracy of the detection system up to 86% using logistic regression classifier. Kasture [10] considered Linguistic Inquiry and Word count(LIWC) for feature extraction. They train with different machine leaning classifiers and found that random forest is the best among the twitter dataset on recall of 96.3% and precision of 98.4%. Ghasem et al[11], chosen the top most 500 informative words as the feature vector and train on support vector machine(SVM) and achieved f1-score of 95%. Jiale Wu et.al[12], PROPOSED ITFT (improved tf-idf based fastText) model for bullying detection by considering keywords, position weight to traditional tf-idf word embedding method. The proposed model able to filter the noise and improved the accuracy. B Sri Nandini et al.[13] proposed a model using Fuzzy logic and genetic algorithms to classify the bullying actions such as Terrorism, Racism, Harassment. They used to genetic algorithm for optimization of features and fuzzy rule set to retrieve the data for classification task. Charkrabarty N[14] used Bag of Words and tf-idf word embedding techniques for toxic comments classification from Wikipedia dataset. Salminen et al.[15] proposed online hate content detection approach for various social media networks. The authors collected 197,566 comments from YouTube, Reddit, Wikipedia and Twitter.

3. Materials and Methods
In this section we discuss about data collection and proposed methodology for cyberbullying detection and evaluation metrics. Fig 1 shows the key components in the proposed methodology. We collected the dataset from the source Waseem and Hovy[16]. The Waseem and Hovy collected tweets from popular social networking site called Twitter over a period of two months with the help of Twitter API. They retrieved 136,052 tweets totally and assigned the labelling for 13,762 tweets. The 9,533 tweets neither sexist nor racist labelled as 0, and 4,229 tweets labelled as 1 which contains either sexism or racism. We assumed labelled ‘0’ tweets are non-bullying comments and labelled ‘1’ tweets are bullying comments. Table 1 shows the example labelling of bullying and non-bullying tweets.
3.1 Data collection:

Table 1. Labelling of tweets. Label 0 represents non bullying and label 1 is for bullying.

| Example Tweet                                                                 | Label |
|-----------------------------------------------------------------------------|-------|
| user if we're lucky. <hashtag>mkr                                           | 0     |
| i think i just threw up in my mouth a little <hashtag>mkr                   | 0     |
| trapped in a pole-dancing venue with that hotch potch and sausage impersonating food <hashtag>mkr | 0     |
| rt user violence censorship & intolerance of dissent. it's why we call feminism a hate movement. User | 1     |
| self described "radical feminist" makes death threats against <hashtag>gamergate woc for not doing/thinking as she was told. | 1     |
| rt user it's absolutely crystal clear now that with the amount of sjw feminists on campuses that guys must tape every sexual en emoticon | 1     |

Table 2. Example of Noisy Data.

| Example Tweets                                                                 | Noisy Data        |
|-------------------------------------------------------------------------------|-------------------|
| â€œAnd so weâ€™ll have whittled 18 down to 14 in just a few short monthsâ€   | Numbers           |
| #MKR that was useless from nikki                                               | Hash Tags         |
| #MKR2015#MKRHhttp://t.co/mD9TyU1MOn                                            | URL links         |
| Twitter makes me laugh....Better than the series                               | Emojis            |
| 😊 😞 4 rounds of instant restaurants????? Craaaazy                             | Spelling corrections |
The input raw data is first pre-processed to avoid inconsistencies and then different feature extraction techniques are used to extract the robust features. Then the classification methods are used to compare the effectiveness of each feature in terms of evaluation metrics.

3.2 Data pre-processing
The pre-processing is very important step for any natural language processing (NLP) tasks. The data collected from various social media consists of noise data such as stop words, URL, unwanted numbers. Table 2 shows the example of noise in social media tweets. To remove noise data from the tweet, we applied various natural language pre-processing steps such as tokenization, lemmatization [17] to get the original root words and stemming [18] to reduce inflections words to root form.

3.3 Feature Extraction
For any natural language processing tasks, feature extraction methods play important role. Machine learning classifier accepts only numerical representation as input data to get trained. So every text information needs to convert into numeral form by applying various feature extraction techniques. In this proposed methodology, we used different feature extraction methods such as Bag of Words (BoW), TF-IDF, word2vec, glove, FastText and various language models such as ALBERT [19], ELECTRA [20], GPT-2[21], XL-NET[22] and RoBERTa[23].

3.3.1 Feature Extraction Techniques
In this section, we give outline of the feature extraction techniques that are used for comparison in our work.

ALBERT:
ALBERT is a language model stands for A LITE BERT. It is self-supervised learning language model of language representations. ALBERT incorporates two-parameter reduction techniques which are factorized embedding parameterization and cross-layer parameter sharing in order for lifting the major obstacles in scaling pre-trained models in NLP. In this model, number of parameters are reduced 89% compared to BERT [24] model and it achieves respectable performance on benchmark datasets. The major advantage of ALBERT is parameter efficiency, because it can fit much larger batches into memory for quick inference.

ELECTRA
It is novel pre-training method that uses a technique is called Replaced Token Detection (RTD). Fig 2 shows the Replaced Token Detection system in ELECTRA. The model returns the efficient performance as same as masked language model such as BERT. It overcomes the drawback of masking approach in BERT models.

GPT-2
GPT-2 is a pre-trained language model which is useful for various natural language processing tasks such as text generation, question and answer system and language translation. It is a transformed-based model trained on a dataset eight million web documents. It uses huge vocabulary size in training. It can predict the next occurring word in a sentence. The GPT-2 architecture based on Transformers concept which provides mechanism of encoder-decoders to identify input and output dependencies. It was evaluated through zero-shot learning.
XL-NET
It is a general auto-regressive pretraining method that enables learning bidirectional contexts by maximizing the expected likelihood over all permutations of the factorization order. It integrates two methods called autoregressive and auto-encoding. In a given text sequence \( k = (k_1, \ldots, k_T) \), autoregressive models use probability of \( p(k) = \prod_{t=1}^{T} p(k_t | k_{<t}) \) for forward prediction of the word and use probability of \( p(k) = \prod_{t=T}^{1} p(k_t | k_{>t}) \) backward prediction of the word. XL-Net outperforms BERT on 20 tasks, including question answering, natural language inference, sentiment analysis, and document ranking.

RoBERTa
RoBERTa is an enhanced version of BERT model by improved training methodology. Dynamic masking pattern is introduced whereas BERT follows static masking pattern. It removes Next Sentence Prediction (NSP) task from BERT’s pre-training. It uses 160GB of text for pre-training with various data such as BookCorpus, English Wikipedia, OpenWebText whereas BERT trained against BookCorpus only. This model outperforms both XL-NET and BERT on GLUE benchmark results.

3.4 Classifiers
Classification is supervised learning approach in which the model predicts discrete output variable based on input variables. Machine learning classifier plays vital role for after generating word embeddings. These classifiers are used in many applications such as speech recognition, handwriting recognition, face detection, document classification etc. A bunch of machine learning classifiers such as Support Vector Machine (SVM), Naive Bayes (NB), Random Forest (RF), K-nearest neighbor (kNN) used for these tasks. We used simple baseline linear classifier Logistic Regression (LR)[25] and non-linear classifier called LightGBM[26] in the experiments for classification task.

3.4.1 Logistic Regression
It is a supervised machine learning algorithm and intended for binary classification tasks. It uses logistic function called sigmoid function which is in the range of 0 and 1. Eq.1 is sigmoid equation for logistic regression. The sigmoid function generates probability scores based on input features. For example, the probability score is \( \geq 0.5 \), it belongs to bullying tweet; otherwise, the prediction will be for the other class (non-bullying).

\[
\text{Sigmoid}(z) = \frac{1}{1 + \exp(-z)}
\] 

3.4.2 Light GBM
LightGBM (Light Gradient Boosting Machine) improved many demerits of Decision Tree and Gradient Boosting Decision Tree algorithms and hence is very useful in bringing great insights from Cyberbullying Data. The LightGBM classifier follows leaf-wise tree growth which is big advantage for this classifier. Earlier boosting algorithms unable to handle large datasets while LightGBM able to handle a huge amount of data with many advantages such as faster training speed, good accuracy, parallel learning, and less memory usage. Fig 3 shows the leaf-wise growth in LightGBM classifier. Each node represent a decision of classify the tasks.

![Figure 3: Leaf-wise growth of LightGBM classifier.](image-url)
3.5 Evaluation Metrics
We considered precision, recall and f1-score as evaluation metrics.

**Precision:** The precision can be defined as the ratio of correctly predicted as bullying tweets to total number of predictions as bullying tweets. Eq.2 shows precision formula.

\[
P_r = \frac{C}{C+B}
\]

Where C indicates correctly predicted as bullying tweets, B indicates non bullying tweets predicted as bullying tweets.

**Recall:** The recall can be defined as the ratio of correctly predicted as bullying tweets to total number of actual bullying tweets. Equation-3 shows recall formula. Here, NB indicates non bullying tweets.

\[
R_c = \frac{C}{C+NB}
\]

**F1-score:** f1-score is the weighted average of recall and precision. Equation-4 gives f1-score formula.

\[
f1\text{-score} = \frac{2 \times P_r \times R_c}{P_r + R_c}
\]

4. Results and Discussion
The proposed approach is implemented in Python programming language with help of several packages called Pandas, Numpy, matplotlib, Scikit-Learn, LightGBM, Tensorflow in Linux operating system. We ran our codes on Intel i7 8th Gen 12core CPU processor and Nvidia Max-Q 1070 8GB RAM.

We conducted several experiments to check the performance of various Word Embedding methods for cyberbullying detection data. We performed 5-fold validation of the dataset. We consider macro average of precision, recall, f1-scores of class-0 and class-1. Fig 4 shows the f1-scores of Waseem dataset with No sampling mode. We observed that most of the language models scored high f1-scores as compared to traditional word embedding such as BoW and T-IDF and semantic word embeddings such as word2vec, glove, FastText. Over sampling and under sampling are two techniques used for imbalanced datasets to avoid class bias problems. We conducted experiments in Over sampling and Under sampling. Table 3 shows precision, recall and f1-score of Waseem dataset with over sampling and under sampling modes. We noticed that the performance of language models is better than traditional word embedding techniques in terms of precision, recall and F1-score.

![Figure 4. f1 score on Waseem dataset with No sampling.](image-url)
### Table 3. Precision, Recall and f1-scores of under sampling and over sampling of various word embeddings.

| Word Embeddings + classifier | Under Sampling |   |   | Over Sampling |   |   |
|-----------------------------|----------------|---|---|---------------|---|---|
|                             | Precision      | Recall | F1-score | Precision      | Recall | F1-score |
| BoW+ LR                     | 71.2           | 72.2   | 71.4     | 71.2           | 72.4   | 71.4     |
| BoW+ LightGBM               | 72.0           | 74.4   | 72.8     | 73.8           | 74.2   | 74.0     |
| TFIDF+LR                    | 70.0           | 71.8   | 70.8     | 70.2           | 71.8   | 70.8     |
| TFIDF+LIGHTGBM              | 70.6           | 73.0   | 71.0     | 73.4           | 73.4   | 73.4     |
| WORD2VEC+LR                 | 75.2           | 77.8   | 75.8     | 75.6           | 78.0   | 76.6     |
| WORD2VEC+lightGBM           | 77.6           | 80.0   | 78.6     | 81.0           | 78.4   | 79.4     |
| GLOVE+ LR                   | 71.8           | 74.2   | 72.6     | 72.2           | 74.2   | 72.6     |
| GLOVE+ light GBM            | 73.8           | 76.8   | 74.8     | 78.6           | 76.0   | 77.0     |
| FastText + LR               | 75.6           | 77.2   | 76.2     | 76.0           | 78.2   | 76.2     |
| FastText + light GBM        | 78.0           | 80.4   | 78.6     | 80.2           | 79.2   | 80.2     |
| ALBERT+LR                   | 77.0           | 79.4   | 77.4     | 76.4           | 79.0   | 77.2     |
| ALBERT+ light GBM           | 78.0           | 80.6   | 78.6     | 81.4           | 78.2   | 79.8     |
| ELECTRA+LR                  | 71.8           | 74.6   | 72.0     | 71.8           | 74.6   | 72.6     |
| ELECTRA+ light GBM          | 73.2           | 75.6   | 73.8     | 77.4           | 73.8   | 75.2     |
| GPT-2 + LR                  | 74.4           | 77.0   | 75.2     | 74.2           | 76.8   | 75.0     |
| GPT-2 + light GBM           | 78.2           | 80.6   | 79.2     | 83.2           | 79.4   | 81.2     |
| XL NET + LR                 | 73.6           | 76.4   | 74.6     | 74.4           | 76.6   | 75.4     |
| XL NET + light GBM          | 75.2           | 77.6   | 76.0     | 80.4           | 75.4   | 77.4     |
| RoBERTa + LR                | 77.2           | 79.2   | 78.2     | 77.8           | 80.0   | 78.8     |
| RoBERTa + light GBM         | 78.0           | 80.8   | 79.0     | 82.8           | 79.4   | 80.6     |

**Error Analysis:**
Mis-classification rate (MCR) is the ratio of false positive (FP) and false negatives (FN) to the total number of records. Eq.5 shows mis-classification rate. In this experiment we applied five fold validation. The total number of tweets in each fold is 2753. Table 4 shows the total number of misclassifications i.e both false positives and false negatives in each fold in terms of percentage. The least number of misclassifications found in language models such as RoBERTa, OpenAI, GPT-2. We noticed that LightGBM classifies better than logistic regression classifier in all the word embedding methods. Fig 5 show the misclassification rate of word embedding methods in terms of percentage. The basic word embeddings rate is 21% which is reduced to 16% with language models.

$$MCR = \frac{FP+FN}{Total \ number \ of \ Records}$$

(5)
### Table 4. Mis-classification rate in each fold in percentage.

| Word Embedding Techniques | Fold-1(%) | Fold-2(%) | Fold-3(%) | Fold-4(%) | Fold-5(%) |
|---------------------------|-----------|-----------|-----------|-----------|-----------|
| BoW+ LR                   | 22.23     | 21.76     | 22.96     | 22.96     | 22.12     |
| BoW+ LightGBM             | 20.81     | 20.41     | 22.3      | 21.32     | 19.69     |
| TF-IDF+LR                 | 23.03     | 23.1      | 21.79     | 22.34     | 21.5      |
| TF-IDF+LIGHTGBM           | 21.32     | 21.72     | 20.96     | 22.12     | 21.25     |
| WORD2VEC+LR               | 18.85     | 19.22     | 20.09     | 18.63     | 20.16     |
| WORD2VEC+lightGBM         | 15.66     | 15.76     | 16.64     | 16.6      | 16.78     |
| GLOVE+ LR                 | 23.14     | 21.61     | 22.12     | 22.63     | 20.09     |
| GLOVE+ light GBM          | 19.40     | 19.87     | 18.38     | 19.29     | 18.71     |
| FastText + LR             | 19.83     | 20.27     | 18.82     | 19.22     | 19.25     |
| FastText + light GBM      | 16.16     | 15.98     | 15.04     | 16.6      | 16.31     |
| ALBERT+LR                 | 18.31     | 17.18     | 16.06     | 16.89     | 18.96     |
| ALBERT+ light GBM         | 16.78     | 16.27     | 15.51     | 16.02     | 17.14     |
| ELECTRA+LR                | 20.74     | 20.01     | 20.52     | 20.6      | 21.21     |
| ELECTRA+ light GBM        | 18.82     | 19.03     | 20.12     | 18.63     | 19.4      |
| GPT-2 + LR                | 20.27     | 17.14     | 18.63     | 19.07     | 19.8      |
| GPT-2 + light GBM         | 16.13     | 14.89     | 16.24     | 15.66     | 16.02     |
| XL NET + LR               | 19.03     | 18.92     | 19.72     | 19.25     | 20.12     |
| XL NET + light GBM        | 17.22     | 17.58     | 18.09     | 17.62     | 17.87     |
| RoBERTa + LR              | 17.14     | 16.67     | 16.82     | 17.4      | 17.47     |
| RoBERTa + light GBM       | 16.27     | 15.76     | 15.51     | 16.6      | 15.29     |

**Figure 5.:** Mis-classification rate of Word Embeddings.
5. Conclusion and Future Works

With the increase of social media platforms, the bullying activities are increasing significantly. This paper proposed a methodology for cyberbullying detection by experimenting various word embeddings techniques with two classifiers. We conducted experiments in three types of sampling modes on the Twitter dataset. 1) no sampling 2) under sampling 3) oversampling. The f1-score of language models in sampling mode almost close to 80 percentage whereas basic word embeddings methods are around 75%. The mis-classification rate of basic word embeddings is 21%, whereas mis-classification rate of language models reduced to 16%. We observed that the drawbacks of basic word embedding techniques can overcome with these language models for many downstream tasks in Natural language processing. As part of the future work for cyberbullying detection system, language models have to be considered to build the robust features.

6. References

[1] Cohen A and Raphael 2018 Social responsibility on the internet: Addressing the challenge of cyberbullying. Aggression and Violent Behavior. DOI: 10.1016/j.avb.2018.01.001
[2] Nikolaou and Dimitrios 2017 Does cyberbullying impact youth suicidal behaviors? Journal of health economics 56 30–46
[3] Karan M and Snajder J 2019 Preemptive toxic language detection in Wikipedia comments using thread-level context in Proceedings of the Third Workshop on Abusive Language Online, Florence, Italy: Association for Computational Linguistics 129–134
[4] Xu, X Zhu J M and Bellmore A 2012 Fast learning for sentiment analysis on bullying proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining - WISDOM 12 1–6
[5] Nahar V Li X and Pang C 2014 An Effective Approach for Cyberbullying Detection Communications in Information Science and Management Engineering 3 5 238-247
[6] Pang B Lee L and Vaithyanathan S 2002 sentiment classification using machine learning techniques In: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing 10 79–86
[7] Zhao R and Mao K 2016 Cyberbullying detection based on semantic-enhanced marginalized denoising auto-encoder IEEE Transactions on Affective Computing 8 328–339
[8] Yin D Xue Z Hong L Davison B D Kontostathis A and Edwards L 2009 Detection of harassment on web 2.0 Proceedings of the Content Analysis in the WEB 2 1–7
[9] Chavan V S and Shylaja S S 2015 Machine learning approach for detection of cyber-aggressive comments by peers on social media network International Conference on In Advances in computing communications and informatics (ICACCI) 2354–2358
[10] Kasture A S 2015 A predictive model to detect online cyberbullying Auckland University of Technology
[11] Ghasem Z Frommholz I and Maple C 2015 Machine learning solutions for controlling cyberbullying and cyberstalking J Inf Secur Res 6 55-64
[12] Jiale Wu Wen M Lu R Li B and Jinguo Li 2020 Toward efficient and effective bullying detection in online social network Peer-to-Peer networking and Applications Springer 13 1567–1576
[13] Sri Nandhini B and Sheeba J I 2015 Online Social Network Bullying Detection Using Intelligence Techniques International Conference on Advanced Computing Technologies and Applications (ICACTA-2015), Elsevier 45 485 – 492
[14] Chakrabarty N 2019 A Machine Learning Approach to Comment Toxicity Classification Computational Intelligence in Pattern Recognition Advances in Intelligent Systems and Computing Springer Singapore 999 183-193
[13] Salminen J Hopf M Chowdhury S A Jung S G Almerekhi H and Jansen B J 2020 Developing an online hate classifier for multiple social media platforms Human-centric Computing and Information Sciences 10 15-20

[14] Skorkovská L 2012 Application of Lemmatization and Summarization Methods in Topic Identification Module for Large Scale Language Modeling Data Filtering 7499 Springer

[15] Waseem Z and Hovy D 2016 Hateful symbols or hateful people? Predictive features for hate Speech detection on twitter Proceedings of the NAACL Student Research Workshop Association for Computational Linguistics 88-93

[16] Pande B P and Dhami H S 2011 Application of natural language processing tools in stemming International Journal of Computer Applications 27(6) 14-19

[17] Lan Z Chen M Goodman S Gimpel K Sharma P and Soricut R 2020 ALBERT: A Lite BERT for Self-supervised Learning of Language representations ICLR

[18] Clark K Luong M Le QV and Manning C 2020 ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators ICLR

[19] Radford A Wu J Child R Luan D Amodei D and Sutskever I 2019 Language Models are Unsupervised Multitask Learners

[20] Yang Z Dai Z Yang Y Carbonell J Salakhutdinov R Le and Quoc V 2019 XLNet: Generalized Autoregressive Pretraining for Language Understanding NeurIPS

[21] Liu Y Ott M Goyal N Du J Joshi M Chen D Levy O Lewis M Zettlemoyer L and Stoyanov V 2019 RoBERTa: A Robustly Optimized BERT Pretraining Approach ICLR

[22] Devlin J Chang M W Lee K and Toutanova K 2019 BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding Association for Computational Linguistics DOI:10.18653/v1/N19-1423

[23] Park H 2013 An introduction to logistic regression: from basic concepts to interpretation with particular attention to nursing domain Journal of Korean Academy of Nursing

[24] Guolin K Meng Q Finely T Wang T Chen W Ma W Ye Q and Liu T 2017 LightGBM: A Highly Efficient Gradient Boosting Decision Tree Advances in Neural Information Processing Systems 3149–3157