Natural language processing

One Week FDP on AI and Deep Learning

Dr. Lov Kumar

Department of Computer Science and Information Systems,
Birla Institute of Technology & Science, Pilani - Hyderabad
Shameerpet, Hyderabad, Telangana 500078

21-11-2019

BITS Pilani
Hyderabad Campus

Overview

Introduction

- Natural Language Processing is the technology used to aid computers to understand the human's natural language
- Natural Language Processing, usually shortened as NLP, is a branch of artificial intelligence that deals with the interaction between computers and humans using the natural language.
- The ultimate objective of NLP is to read, understand, and make sense of the human languages in a manner that is valuable.
- Most NLP techniques rely on machine learning to derive meaning from human languages.

Introduction

- **Language Translation:** Translation of a sentence from one language to another.
- **Sentiment Analysis:** To determine, from a text corpus, whether the sentiment towards any topic or product etc. is positive, negative, or neutral.
- **Spam Filtering:** Detect unsolicited and unwanted email/messages.
- **Software defect severity level** : Assign an appropriate severity level to the defects present in the defect reports.

Defect Severity Level Prediction

- Software defect severity level prediction models aim to assign an appropriate severity level to the defects present in the defect reports.
- These prediction models help to improve the quality of software with the effective allocation of testing resources.

Project Name

- Software defect severity level prediction
- Sentiment Analysis for Software Engineering.
- Fake News
- Fake fake job posting prediction
- Twitter-airline-sentiment
- COVID 19 Data analysis

Technical Challenges

- Word Embedding
- High-Dimensional Data
- Imbalanced Data

Data Preprocessing

- **Tokenization** - convert sentences to words
- **Removing stop words** ? frequent words such as "the", "is", etc. that do not have specific semantic
- **NLTK** - The Natural Language ToolKit is one of the best-known and most-used NLP libraries, useful for all sorts of tasks from t tokenization, stemming, tagging, parsing, and beyond

```
import nltk
from nltk.tokenize import word_tokenize
tokens = word_tokenize("BITS Hyderabad Campus")
print(tokens)
```

Data Preprocessing

innovate achieve lead

stopwords

```
from nltk.corpus import stopwords
stop_words = set(stopwords.words('english'))
tokens = [w for w in tokens if not w in stop_words]
print(tokens)
```

BITS Pilani
Hyderabad Campus

Feature Extraction

- Bag of Words (BOW)
- Term Frequency
- Inverse Document Frequency (**TF-IDF**)
- term Frequency (TF) = (Number of times term t appears in a document)/(Number of terms in the document)
- Inverse Document Frequency (**IDF**) = log(N/n), where, N is the number of documents and n is the number of documents a term t has appeared in.
- **TF-IDF** value of a term as = TF * IDF

Feature Extraction

CountVectorizer

```
from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer(ngram_range=(1,2))
x = vectorizer.fit_transform(data.DescTex)
x=x.toarray()
print(x)
print(vectorizer.get_feature_names())
```

TFIDF

```
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf = TfidfVectorizer()
 x = tfidf.fit_transform(data.DescTex)
df_tfidf = x.toarray()
```

Word embedding

- CBOW
- Skip-gram
- Word2Vec
- Glove
- BERT
- Fasttext

Experimental setup

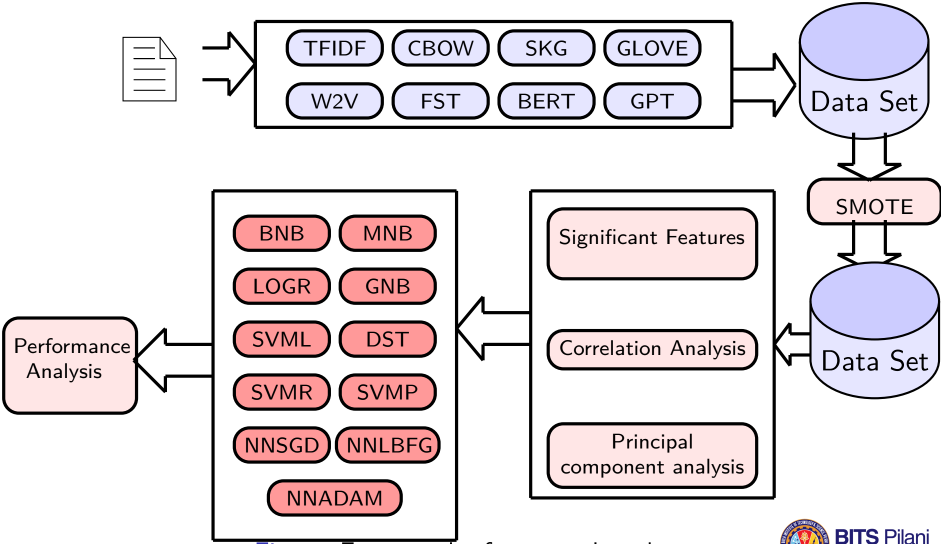


Figure: Framework of proposed work

Feature Selection Techniques

Why we need FS:

- to improve performance (in terms of speed, predictive power, simplicity of the model).
- To visualize the data for model selection.
- To reduce dimensionality and remove noise.

Feature Selection is a process that chooses an optimal subset of features according to a certain criterion.

Feature selection methods

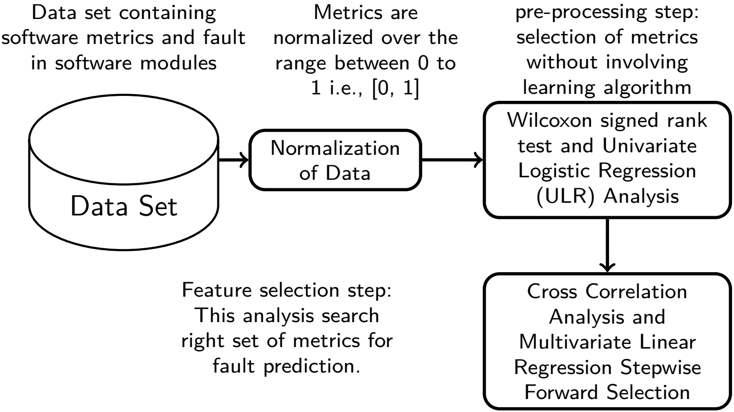
Feature ranking techniques:

In Feature ranking technique, some decisive factors have been considered to rank each individual feature and then some features are selected that are suitable for a given project.

Feature subset selection techniques:

In feature subset selection, subset of features are searched which collectively have good predictive capability.

PROPOSED Features VALIDATION METHOD



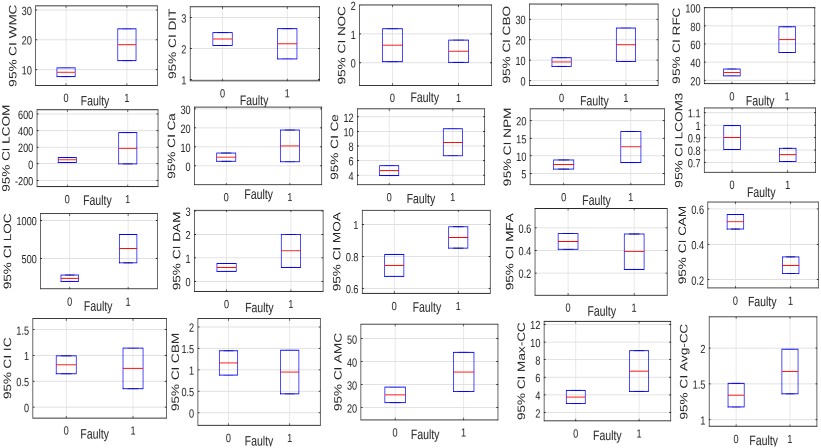
Confidence Intervals

- A Confidence Interval is a range of values we are fairly sure our true value lies in.
- Calculating the Confidence Interval
- Step1: find the number of observations n, calculate their mean X, and standard deviation s.
- Step2: Decide what Confidence Interval we want: 95% or 99% are common choices. Then find the "Z" value (1.96 for 95% and 2.576 for 99%) for that Confidence Interval here:
- Step3: use that Z in this formula for the Confidence Interval

$$\bar{X} \pm Z * s/\sqrt{(n)} \quad (1)$$

BITS Pilani
Hyderabad Campus

Error box plots



BITS Pilani
Hyderabad Campus

Python code

Python code

```
from scipy.stats import mannwhitneyu
w,p=mannwhitneyu(f0,f1)
```

Python code

```
from matplotlib import pyplot
pyplot.boxplot(x,labels=['Not-faulty','Faulty'])
pyplot.grid(True)
pyplot.xlabel('Metrics')
pyplot.ylabel('95%CI')
fna='C:/Users/lov/Documents/dsv/'+str(i)+".png"
pyplot.savefig(fna) pyplot.close()
```

95% confidence interval

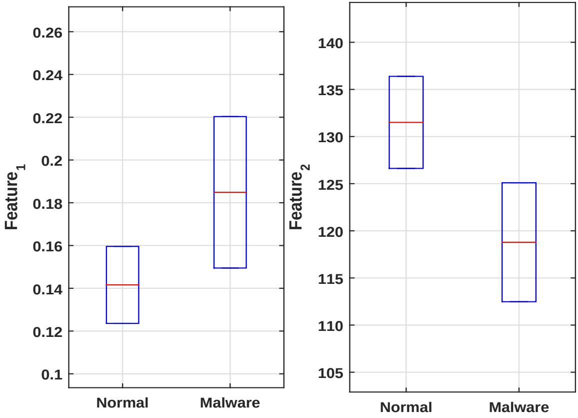


Figure: 95% confidence interval of two features

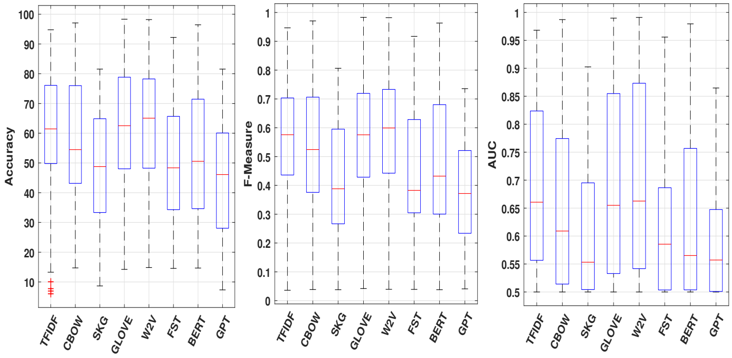
Performance: Accuracy

	OD										
	MNB	BNB	GNB	LOGR	DST	SVML	SVMP	SVMR	NNLBFG	NNSGD	NNADAM
TFIDF	78.11	75.56	22.78	78.22	62.44	78.22	77.78	78.00	72.67	78.11	73.89
CBOW	78.11	78.11	73.44	78.11	60.22	78.11	70.89	78.11	72.78	78.11	78.00
SKG	78.11	78.00	60.11	78.00	60.22	78.11	74.56	78.11	74.56	78.11	77.67
GLOVE	78.11	78.00	51.00	77.67	59.56	78.00	71.11	78.00	70.22	78.11	73.89
W2V	78.11	77.67	55.11	78.11	60.11	77.89	70.44	78.22	68.56	78.11	75.33
FST	78.11	77.56	24.22	78.00	58.56	78.11	75.11	78.11	72.89	78.11	78.11
BERT	74.22	76.89	17.22	78.11	59.67	78.22	72.11	78.22	78.11	78.11	78.11
GPT	54.67	73.89	7.33	78.11	72.33	78.11	78.00	78.11	78.11	78.11	78.11
	SMOTE										
TFIDF	60.18	62.60	59.50	64.35	79.53	67.47	86.95	88.85	85.78	83.32	91.90
CBOW	42.94	16.18	58.21	54.38	76.17	48.53	91.92	95.92	90.46	92.06	95.71
SKG	24.55	16.21	25.92	44.62	74.39	42.32	75.95	68.59	72.14	58.24	74.71
GLOVE	45.89	16.43	56.77	76.52	78.60	78.29	95.79	98.28	92.09	95.60	95.65
W2V	46.52	16.22	59.43	79.95	77.56	80.43	95.07	98.18	91.88	95.93	94.08
FST	27.45	15.97	34.14	37.29	77.35	33.71	86.66	82.33	67.96	84.94	87.25
BERT	24.13	15.97	31.22	77.99	78.90	81.07	94.72	84.63	15.97	15.97	15.97
GPT	22.49	19.05	27.93	44.52	68.06	46.57	61.16	50.58	15.97	15.97	15.97

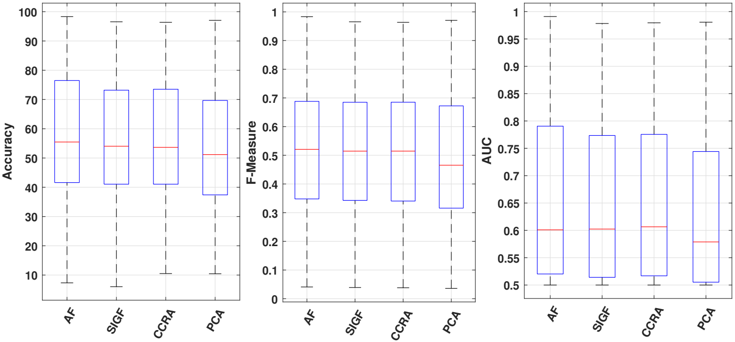
Performance: AUC

	MNB	BNB	GNB	LOGR	DST	SVML	SVMP	SVMR	NNLBFG	NNSGD	NNADAM
	OD										
TFIDF	0.51	0.51	0.59	0.51	0.52	0.51	0.51	0.50	0.53	0.50	0.54
CBOW	0.50	0.50	0.53	0.50	0.55	0.50	0.56	0.50	0.52	0.50	0.50
SKG	0.50	0.50	0.58	0.51	0.57	0.50	0.56	0.50	0.54	0.50	0.51
GLOVE	0.50	0.51	0.64	0.51	0.56	0.50	0.54	0.50	0.53	0.50	0.55
W2V	0.50	0.50	0.63	0.53	0.54	0.51	0.57	0.50	0.57	0.50	0.54
FST	0.50	0.51	0.54	0.51	0.55	0.50	0.52	0.50	0.51	0.50	0.50
BERT	0.53	0.52	0.57	0.51	0.54	0.50	0.53	0.50	0.50	0.50	0.50
GPT	0.57	0.52	0.54	0.50	0.52	0.50	0.50	0.50	0.50	0.50	0.50
	SMOTE										
TFIDF	0.80	0.82	0.80	0.80	0.87	0.82	0.90	0.93	0.92	0.91	0.95
CBOW	0.67	0.51	0.76	0.76	0.85	0.72	0.95	0.98	0.94	0.96	0.98
SKG	0.56	0.50	0.55	0.71	0.84	0.70	0.87	0.83	0.87	0.78	0.88
GLOVE	0.72	0.50	0.78	0.88	0.87	0.89	0.97	0.99	0.96	0.97	0.97
W2V	0.73	0.51	0.79	0.90	0.86	0.91	0.97	0.99	0.96	0.97	0.96
FST	0.57	0.50	0.60	0.64	0.87	0.65	0.93	0.90	0.82	0.92	0.93
BERT	0.56	0.50	0.58	0.88	0.87	0.90	0.97	0.91	0.50	0.50	0.50
GPT	0.55	0.54	0.59	0.70	0.81	0.73	0.78	0.72	0.50	0.50	0.50

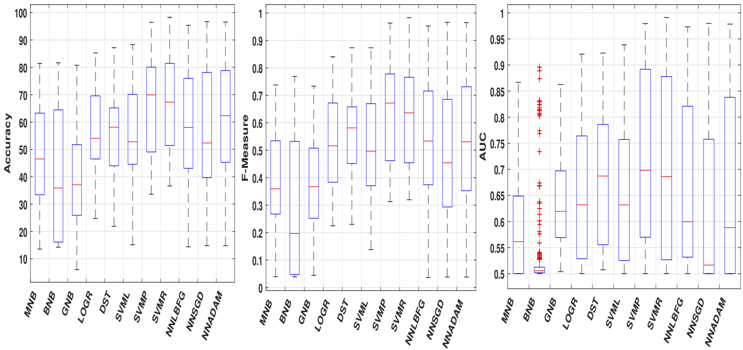
Comparison: Performance of Different Word Embedding



Comparison: Performance of Different S Features



Comparison: Performance of Different Classification Techniques



Conclusion

The high value of AUC confirms that the developed models using word embedding on balanced data have the ability to predict severity levels of the defects present based on defect descriptions.

The models developed by considered word vector computed using GLOVE and w2v have a better predictive ability as compared to other models.

The defected severity levels prediction models developed using different word embedding methods are significantly different.

Conclusion

The predictive ability of the models developed using significant uncorrelated features has a better ability to predict severity level as compared to all features.

The models developed using SVM with polynomial kernel achieve significantly better performance as compared to other techniques.

Selected References

[1] Malay Kumar, Jasraj Meena, Rahul Singh, and Manu Vardhan.
Data outsourcing: A threat to confidentiality, integrity, and availability.
In *2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*, pages 1496–1501. IEEE, 2015.

[2] Andrea Saracino, Daniele Sgandurra, Gianluca Dini, and Fabio Martinelli.
Madam: Effective and efficient behavior-based android malware detection and prevention.
IEEE Transactions on Dependable and Secure Computing, 15(1):83–97, 2018.

[3] Niall McLaughlin, Jesus Martinez del Rincon, BooJoong Kang, Suleiman Yerima, Paul Miller, Sakir Sezer, Yeganeh Safaei, Erik Trickel, Ziming Zhao, Adam Doupe, et al.
Deep android malware detection.
In *Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy*, pages 301–308. ACM, 2017.

[4] Ambra Demontis, Marco Melis, Battista Biggio, Davide Maiorca, Daniel Arp, Konrad Rieck, Igino Corona, Giorgio Giacinto, and Fabio Roli.
Yes, machine learning can be more secure! a case study on android malware detection.
IEEE Transactions on Dependable and Secure Computing, 2017.

Any Question Please ?

Thank You!

BITS Pilani
Hyderabad Campus