Enhancing Network Security: Pinpointing Similarities and Differences of Machine Learning Models for Firewall Intrusion Detection

Asfiya Shireen Shaikh Mukhtar
✉ asfiyashireen768@gmail.com

R. N. Jugele
✉ rn_jugele@yahoo.com

Department of Computer Science
Shivaji Science College, Nagpur
RTMNU
Nagpur, Maharashtra

ABSTRACT

As cyber threats become increasingly sophisticated, the need for a robust Firewall intrusion detection system to protect network infrastructure becomes paramount. This study represents an in-depth Knowledge to improve Firewall security through a comparable assessment of various machine learning algorithms for firewall intrusion detection. Researcher analyzed classic machine learning ML models such as support vector machines (SVMs), random forests and k-nearest neighbors (k-NNs) Bayesian learning, Linear Regression, Decision tree, K-mean etc. Traditional machine learning models perform well in scenarios with well-defined functions and low computational requirements. Our results contribute to the continuing debate on hardening network defenses and help practitioners and researchers select models tailored to the specific needs of their network security environments. This study provides a comprehensive overview of the firewall intrusion detection landscape and emphasize to evolving ML algorithm's function in improving interconnections security and show different strengths and weaknesses within each model category also Insights gained from this comparative analysis gave the way for informed decision-making and advancements in the development of robust firewall intrusion detection systems.

KEYWORDS: *Firewall anomaly detection, Machine learning models.*

INTRODUCTION

Networks are having a greater influence on contemporary society, rendering information security a crucial field of study information privacy technologies primarily incorporate anti-malware software, firewalls, and Security tool for IDS. These innovations defend your network against threats from within as well as outside. with each other., firewall may be a comparatively anomaly impedances region framework that serves a basic parcel in guaranteeing internet safety by checking the success of program and technology performing on the organize. The goal of this propose work is to group together and analyze ML model applied on firewall anomaly detection system and study their metrics,demerits, and characteristics. Incorporating machine learning algorithms such as Logistic Regression,K-Neighbors Classifier, Gaussian NB, Linear SVC, and Random Forest Classifier expands the scope of anomaly detection capabilities. Each model has unique strengths that improve the system's ability to detect anomalous patterns or drops in network traffic

COMMON MACHINE LEARNING ALGORITHMS

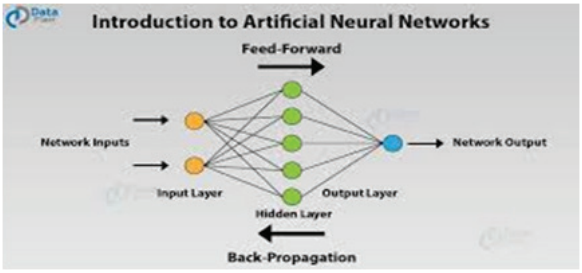


Fig 1 : Working of ANN

Artificial Neural Networks (ANN)

The ideology of ANN is to imitate the function of the brain of an individual. ANN incorporate a data input

level, an undetectable layer, and an outcome level. In Ann, neighboring level units are integrally correlated. ANN has multiple units and can potentially estimate every function. As a result, it excels at adapting to non-linear functions.

Support vector machine (SVM)

SVMs are useful for both classification and regression problems. By using this technique, the decision boundary is defined as a hyper plane. A decision plane is needed when a collection of objects from different classes needs to be divided. Trigonometric functions called kernels are required to divide objects that belong to different classes if they cannot be separated linearly. Correct object classification using examples from the training data set is the aim of SVM [3] [4].

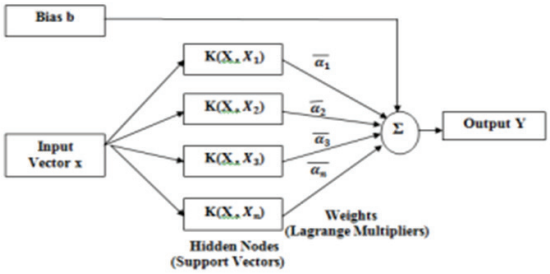


Fig 2. Schematic diagram of SVM model

K-nearest neighbor (KNN)

One technique for categorization [7] is the K Neighbor (KNN). This Approach uses organized repository to classify sample values into various categories efficiently. KNN called non-parametric as it does not assume anything about the distribution of the underlying data. It is an easy-to-implement, uncomplicated strategy. The model is inexpensive to build. It is a very flexible multi-modal categorization method. Records have several class labels assigned to them. Compared to the Bayes error rate, the error rate is twice as high. The process of classifying unidentified records is quite costly.

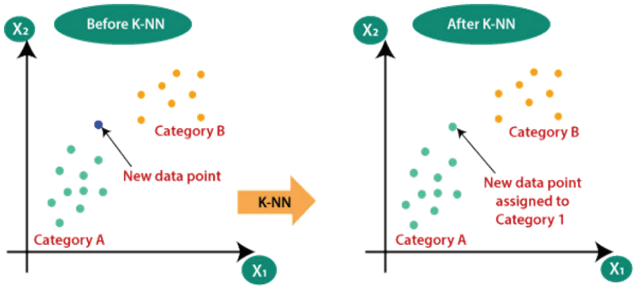


Fig 3. Working of KNN

Naïve Bayes

Bayesian learning identifies an initial probability range and then upgrades it to derive a subsequent distribution. The earlier observations will be updated if fresh ones become available later. A bazillion of data record may be control by incomplete Naïve Bayes networks. This learning strategy may be used to stop over fitting of the data. Identifying victims of disasters and diagnosing illnesses are two examples of applications for Bayesian learning.

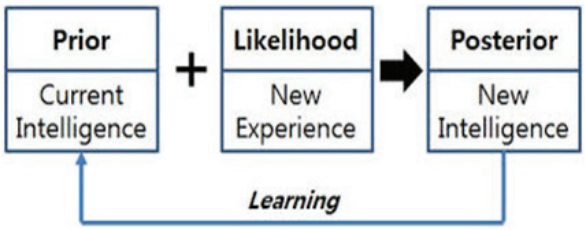


Fig 4 : Naïve Bayes Process

E. Logistic Regression

Logistic Regression an identification technique with a linear regression transformation. It anticipates binary outcomes. A sigmoid arc is constructed, having values from 0 to 1. LR implements the sigmoid algorithm [14]. It shipped with OVR and multiclass attributes for multiclass classification. Over fitting can occur in high-dimensional datasets, but prevent by regularization approaches.

The logistic regression equation can be expressed as:

$$P(Y = 1) = 1/(1 + e\text{^}(-(b\_(()) + b\_1X\_1 + b\_2X\_2 + \cdots + b\_nX\_n)))$$

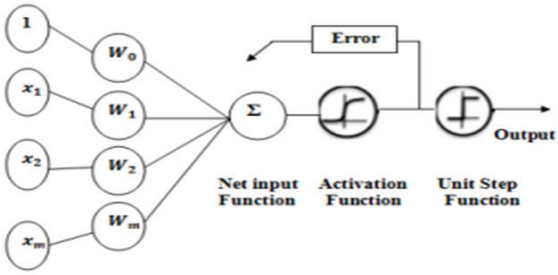


Fig 5: Logistic Regression Model

Decision Tree

Decision tree is a type of ML method with nodes and edges that split input into nodes. It can be used for classification and regression problems.

ID3 and C4.5 algorithms are commonly used for Decision tree construction.

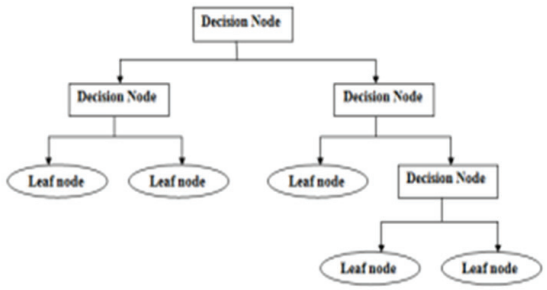


Fig 6: Decision Tree

K-means

K-means isfor clustering algorithm. K stands for the clustering method. An uncontrolled recurrent approach that splits a raw data set into clusters of K's and recognizes classification based on proximity. The method's main idea is center-based, with every statistic sample split to the closest center. proposed an innovative method of anomaly recognition, incorporating KNN and Naive Bayes. The K-means technique laid out an intrusion detection approach employing KNN and LR classification.

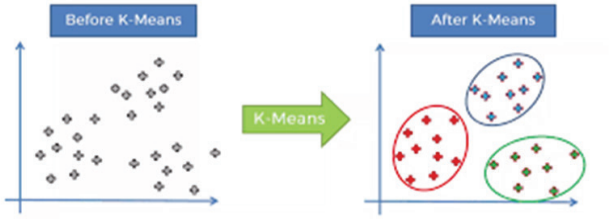


Fig 7 : K-mean clustering

CHARACTERISTICS OF DIFFERENT MACHINE LEARNING MODEL

The following table shows the characteristics of different Machine learning Model.

Table 1: Matching of different Machine Learning Algorithm

Characteristics	Machine Learning Algorithm						
	ANN	SVM	KNN	Naïve Bayes	Logistic Regression	Decision Tree	K-mean
Supervised Learning	✓	✓	☐	✓	✓	✓	☐
Unsupervised Learning	☐	☐	✓	☐	☐	☐	✓
Classification	✓	✓	✓	✓	☐	✓	✓
Regression	✓	✓	✓	✓	☐	☐	☐
Scalability	✓	✓	✓	✓	✓	✓	✓
Bias-Variance Trade-off	✓	✓	✓	✓	☐	☐	☐
Generalization	✓	✓	✓	☐	☐	☐	☐
Adaptability	✓	✓	✓	✓	✓	✓	✓
Non-linearity	✓	✓	✓	✓	✓	✓	✓
Fault Tolerance	✓	✓	✓	✓	✓	✓	✓
Simple and Intuitive	✓	✓	✓	✓	✓	☐	✓
Non-Parametric	☐	☐	✓	☐	☐	✓	✓
Memory Efficient	☐	✓	☐	☐	☐	☐	☐
Versatile	✓	✓	✓	✓	✓	✓	✓
Robust to Overfitting	☐	✓	☐	☐	✓	☐	☐
Sensitive to Outliers	☐	✓	✓	✓	✓	☐	☐
Feature Scaling/Selection	✓	✓	✓	☐	✓	☐	☐
Hyperparameter Tuning	✓	✓	✓	☐	☐	☐	☐
Probabilistic Model	✓	☐	✓	✓	☐	☐	☐
Interpretability	✓	✓	✓	✓	✓	✓	✓
Robustness	✓	✓	✓	✓	☐	✓	✓
Sequential Data Processing	✓	☐	✓	☐	☐	☐	☐
Activation Functions	☐	☐	☐	☐	☐	☐	☐
Feed Forward Network	✓	☐	☐	☐	☐	☐	☐
Encoder-Decoder Architecture	☐	☐	☐	☐	☐	☐	☐
Dimensionality Reduction	☐	✓	☐	☐	☐	☐	☐
Regularization	✓	✓	✓	☐	☐	☐	☐

PROS AND CONS OF VARIOUS MACHINE LEARNING MODEL

The following Table Summarize the Pros and cons of various Machine learning models

Table 2: Pros and cons of various Machine Learning Model

Algorithm	Functions	Advantages	Disadvantages
ANN	Pattern Recognition, Feature Extraction, Anomaly Detection, Adaptability and Learning, Reducing False Positives,Behavioural Analysis.	Capable of handling nonlinear data, Excellent fitting capability.	Overfitting tendency, tendency to get trapped in a local optimum, model training
SVM	Classification, Non-Linearity Handling, Margin Maximization, Dimensionality Reduction.	Acquire valuable knowledge from a limited number of trains, robust generation capacity	low performance on many categorization tasks or large data sets; kernel function-sensitive
KNN	Instance-Based Learning, No Assumptions about Data Distribution,Handling Multimodal Data,Localized Decision Boundaries	Utilize large amounts of data, appropriate for non-specific evidence Develop rapidly,Sensitive to noise.	Unreliable impacts in the minority category, extended periods of testing.
Naïve Bayes	Probabilistic Modelling, Efficient Training and Prediction,Assumption of Feature Independence,Adaptability to Streaming, DataIncremental Learning.	Soundproof,Skilled grade learning.	Does not work well for data related to attributes.
LR	Interpretability,Scalability,Regularization,Probabilistic Modelling.	Simple,quick to learn, Scalable,works automatically.	Does not work well with non-linear data; Apt toover fitting
Decision tree	Rule Extraction, Detecting Interaction Effects, Rule-based Representation,Handling Mixed Data Types.	Intelligently Choose Features,Powerful perception.	Disregard the correlation between the data points and proceed with every single class's classification outcome.
K-means	Network Segmentation, Traffic Profiling, Dynamic Threshold Setting, Adaptive Monitoring.	Straightforward, can be preparedquickly, Solidversatility,Able to accommodate vast amounts of data	When there is no convex knowledge, perform inadequately. Sensitive when using parameter K, dependent on activation.

CONCLUSIONS

The paper proposes a Firewall intrusion detection taxonomy. It showcases the several machine learning techniques in detail and analyzed the comparison of machine learning models based on own stronger fitting and generalization abilities. Researcher analyzed machine learning models offer valuable contributions to firewall intrusion detection. Nonetheless, the interpretability of machine learning models remains essential for understanding detection decisions and ensuring transparency in security operations. We also study the various machine learning algorithm and distribute the various deep learning and machine learning algorithm according to their characteristics. Future advancements may lie in refining hybrid approaches, addressing Interpretability challenges in deep neural networks algorithms and improving firewall IDS resistance to emerging threats.

REFERENCE

1. Zeyuan Fu, "Computer Network Intrusion Anomaly Detection with Recurrent Neural Network", Mobile Information Systems, vol. 2022, Article ID 6576023, 11 pages, 2022. https://doi.org/10.1155/2022/6576023
2. El-Nagar, Ahmad &Zaki, Ahmad &Soliman, Fouad& el Bardini, Mohammad. (2022). Hybrid deep learning diagonal recurrent neural network controller for nonlinear systems. Neural Computing and Applications. 34. 1-20. 10.1007/s00521-022-07673-9.
3. Khan, Muhammad Ashfaq. 2021. "HCRNNIDS: Hybrid Convolutional Recurrent Neural Network-Based Network Intrusion Detection System" Processes 9, no. 5: 834. https://doi.org/10.3390/pr9050834
4. ErmalElbasani, Jeong-Dong Kim, "LLAD: Life-Log Anomaly Detection Based on Recurrent Neural Network LSTM", Journal of Healthcare Engineering, vol.

2021, Article ID 8829403, 7 pages, 2021. https://doi.org/10.1155/2021/8829403

5. M. Rathika, P. Sivakumar, K. Ramash Kumar, IlhanGarip, "Cooperative Communications Based on Deep Learning Using a Recurrent Neural Network in Wireless

6. Communication Networks", Mathematical Problems in Engineering, vol. 2022, Article ID 1864290, 12 pages, 2022. https://doi.org/10.1155/2022/1864290

7. Jin Gao, Jiaquan Liu, SihuaGuo, Qi Zhang, Xinyang Wang, "A Data Mining Method Using Deep Learning for Anomaly Detection in Cloud Computing Environment", Mathematical Problems in Engineering, vol. 2020, Article ID 6343705, 11 pages, 2020. https://doi.org/10.1155/2020/6343705

8. Junjie Cen, Yongbo Li, "Deep Learning-Based Anomaly Traffic Detection Method in Cloud Computing Environment", Wireless Communications and Mobile Computing, vol. 2022, Article ID 6155925, 8 pages, 2022. https://doi.org/10.1155/2022/6155925

9. SajadEiny, HasanSaygin, HemrahHivehch, YahyaDorostkarNavaei, "Local and Deep Features Based Convolutional Neural Network Frameworks for Brain MRI Anomaly Detection", Complexity, vol. 2022, Article ID 3081748, 11 pages, 2022.

10. Chao Wang, Bailing Wang, Hongri Liu, HaikuoQu, "Anomaly Detection for Industrial Control System Based on Autoencoder Neural Network", Wireless Communications and Mobile Computing, vol. 2020, Article ID 8897926, 10 pages,2020.https://doi.org/10.1155/2020/8897926

11. LerinaAversano, Mario Luca Bernardi, Marta Cimitile, Riccardo Pecori, Luca Veltri, "Effective Anomaly Detection Using Deep Learning in IoT Systems", Wireless Communications and Mobile Computing, vol. 2021, Article ID 9054336, 14 pages, 2021. https://doi.org/10.1155/2021/9054336

12. R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," arXiv preprint arXiv:1901.03407, 2019.

13. LirimAshiku, CihanDagli, Network Intrusion Detection System using Deep Learning, Procedia Computer Science, Volume 185, 2021, Pages 239-247, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2021.05.025.

14. JOUR Wanjau, Stephen Wambugu, Geoffrey Oirere, Aaron 2022/06/01 16 Network Intrusion Detection Systems:

15. A Systematic Literature Review of Hybrid Deep Learning Approaches 10 10.35940/ijese. F2530.0610722 International Journal of Emerging Science and EngineeringYao, H., Li, C., & Sun, P. (2020).

Perspective on Big Data Hadoop Tools and Technologies

[1]Shital P. Adkine, [2]Dr. Manish T. Wanjari, [3]Dr.Keshao D. Kalaskar

[1]Dept.of Computer Science,Sardar Patel Mahavidyalaya, Chandrapur, Maharashtra-India
[2]Dept. of computer science, Shivaji Science College, Nagpur, Maharashtra-India
[3]Dept. of computer science, Dr, Ambedkar College, Chandrapur, Maharashtra-India

ABSTRACT

The word 'big data' itself tells everything is big like huge volume of data which can be in a structured, semi-structured, unstructured form. As it is Big so generated in a very large amount making it difficult to process using traditional techniques. To process these generated data traditional big data management system is incompetent to manage the large amount of data with different structures, thus Hadoop the framework which is designed to process the large data sets and provides high performance and fault tolerance from a single server to thousands of machines with different. In this paper we describe a study of big data, Hadoop along with a comparison of various tools and technologies used in big data management.

Keywords-component, Privacy, Unstructured Big Data, Big Data Classification, Big Data Tools

1. Introduction

Big data the term itself indicating its meaning a massive pool of data. Now a day's valuable asset means data. The usage of big data spread due to commercialization and digitization in each area. Mainly big data lies on 6 pillars of v's, volume variety, velocity, veracity, value and variability.

1.1. Big data is characterized with the help of Six Vs

1. Volume: Volume describes as a large quantity of data produced by every day in sets, tables and files by any organization like healthcare, education institutes & commercial business in terms of petabytes & zettabytes.
2. Variety: It refers to types of data which deals with may be structured or unstructured, semi-structured.
3. Velocity: It refers to the rate of growth; the speed of big data is generated Telecommunication produces 35TB of data on per day.
4. Veracity: Veracity defines availability and accountability biases, noise and anomaly in data.
5. Value: Having endless amounts of data but it can be move into value.
6. Variability: Variability deals with the data whose meaning is constantly changing

1.2. Sources of Big Data:

Big data are coming from several different sources. The three main primary sources of big data which includes the organization in

1. Social networks.
2. Traditional business system.
3. Internet of Things (IoT).

The data from these sources can be structured, semi-structured, or unstructured, or any combination of these varieties. Social Networks includes the data, human-sourced information from LinkedIn whatsup Twitter and Facebook,

Instagram, Flickr, Pinterest, etc. Traditional Business Systems deals with customersservices likeCommercial transactions, E-commerce like Alibaba, Amazon, Flipkart generates huge amount of logs from which users buying trends, Banking records, Credit cards, healthcare records and Internet of Things include Sensors, traffic, weather, mobile phone location, etc. Security, surveillance videos, and images Satellite images, Data from computer systems (logs, weblogs, etc.) The connectivity of large number of heterogeneous devices produces huge data [3], which includes features such as heterogeneity, variety, unstructured feature, noise, and high redundancy.

1.3 Behavioral types of big data

Different types of big data based on content format are as follows:

Structured Data

The data stored in relational databases table in the format of row and column. Structured data include numbers, text, and dates; in terms of database, it is called *strings*. Data have fixed structures and these structures used for organizations to creating a perfect model. Data model permission to store, process and operate on data. Analysis and storing of structured data is very easy. Because of high cost, limited storage space and techniques used for processing, causes RDBMS the only path to store and process the data effectively. Programming language called Structured Query Language (SQL) is used for managing this type of data.

Unstructured Data

Without any specific structure and due to this could not be stored in a row and column format is unstructured data. The data is contradictory to that of structured data. It cannot be stored in a databank. Volume of this data is growing extremely fast which is very tough to manage and analyze it completely. To analyze the unstructured data advanced technology knowledge is needed.

Semi-structured Data

Data which is in the form of structured data but it does not fit the data model is semi-structured data. It cannot be stored in the form of data table, but it canbe stored in some particular types of files which hold some specific marker or tags. These markers are distinguished by some specific rule and the data is enforced to be stored with a ranking. This form of data increased rapidly after the introduction of the World Wide Web where various form of data need medium for interchanging the information like XML and JSON.

1.4 Margins of Existing Systems

The existing systems have major restrictions preventing their use in applications. The limitations are:

1. Lack of integrity
2. Lack of availability and continuity of service
3. Lack of accuracy
4. Existing systems provide vertical scalability.
5. Inconsistency in data format
6. Risk of mismanagement

Big Data deals with modern tools and techniques, and to process this huge data set the previously work traditional data management system is not work properly to handle this enormous amount of data. Traditional relational databases are obsolete and cannot store and process the data generated from recent business applications [4]. Traditional

computational frameworks, system architectures and processing systems are designed to handle structured data [5]. One solution to this is Hadoop which work to solve the problems in existing big data management system, which is design to process effectively by providing scalability fault tolerance h and high performance

Table 1:Comparison of traditional and big data [1]

	Traditional data	Big data	Pros of big data
Types of data	Structured data	Structured, Unstructured and semi-structured	develop variety
Volume	Small amount of data. Range- Gigabyte - terabytes	Large amount of data. Range-<petabytes.	Cost reduces and help business intelligence
Data schema	Fixed schema	Dynamic schema	Preserves the information in data.
Data Relationship	Relationship with data is explored easily	Difficulty in relationship betweendata items.	-
Scaling	More than one server for computing	Single server for computing	Cost effective
Accuracy	Less accurate results	High accurate results	Confident results and reliable

Why Hadoop?

The key features and ability to process enormous amount of data with effective storage, computation and analysis has been a great impulseto take a look into the structural design of the industry leading big data processing framework byApache, Hadoop. Earlier days due to the less advanced technology to deals with unstructured data is not handling by several industries. Hadoop is a solution for big data, change the way and decision-making process be used for unstructured data. Hadoop provides a reliable and scalable platform which is used to solve problems caused by massive amount of heterogeneous data. Hadoop technology accepted because of the features like flexibility, scalability, performance, and cost effective. The Hadoop consists of Hadoop kernel, MapReduce, Hadoop distributed file system (HDFS) and Apache hive etc. MapReduce is a programming model which is used to processing large datasets and analyzing it in a cost-effective manner based on divide and conquers method. The divide and conquer method are implemented by two steps such as Map step and Reduce Step. Hadoop data analytics environment deals with data storage, data processing, data access, data management, privacy data protection.

Hadoop distributed file system (HDFS) for the analysis of massive type of data sets using theMapReduce programming model. Hadoop work on three master points scalability, computation capacity, and storage. Hadoopstores file system metadata known as block**s.** It contains the name node and data nodes. HDFS work on master-slave architecture. An HDFS cluster contains a single name node, a master server that manages the file system namespace and directories in the form of hierarchy structure. Data node divides in totwo files, the first one contains data itself and the second deals with block's generation stamp. Hadoop Distributed File System is designed to tackle fault tolerant and effective-cost hardware. HDFSHadoop stands for "YET ANOTHER RESOURCE NEGOTIATOR". It provides different processing techniques, like batchprocessing, interactive processing, stream processing graph processingetc. Hadoop common contains libraries and directories.

Table 2 – List of latest tools available to handle big data [2]

Features	Hadoop	MapReduce	HBase	Hive	Spark	Pig
Data flow	Hadoop is a chain of stages.	MapReduce is bas on distributed programming moc that was designed for processing of huge volumes of datasets in paralle such that it is independently wo without bothering sub work.	HBase is run on top of HDFS and it stores data in the key / value form.	Data flow in Hive behaves at the query execution level right from the UI. Meta store sends metadata info back to the compiler.	Spark represents a data flow in a form of a direct acyclic graph (DAG).	The Pig is used to analyze larger sets of data that presents them as data flows
Data processing	Hadoop uses batch processing system.	MapReduce is a tool which is suitable for parallel processing of huge data.	HBase is used to store data into a column-oriented database format.	Hive is used to summarizatio n of data, query, and analysis.	Spark is micro-batch processing and system	Pig is a tool used for analyzing of huge data sets.
Streaming engine	Hadoop deals with large data sets as input, processes it and produces the output.	MapReducestreamin g is a type of native batch processing engine.	The batch load is optimized to run on the Spark execution engine	Hive streaming provides for Software based enterprise content delivery that is done behind the firewall for efficiency and security.	Spark streams data in to micro-batches.	Pig provides a parallel architecture-oriented streaming engine that can update Hadoop data over small portions.
Scalability	Hadoop provides scalable, flexible data storage and analysis.	MapReduce provides scalability means that single server to thousands of different machines	HBase provides extreme scalability, reliability, and flexibility for data.	Hive is much familiar, fast, scalable and extensible.	Spark provides linear scalability in the distributed environmen t	Pig provides high level scalability
Latency	Hadoop gives higher latency than both Spark and Fink.	MapReduce gives low latency.	HBase is fast and used for low latency data access. It stores data in - memory table	Hive has high Latency as compared to HBase.	Spark gives low latency than Hadoop	Pig is streaming writes, just like Map Reduce. Low latency queries are not supported in

			known as MemStore.			Pig;thus it is not suitable for OLAP and OLTP.
Scheduler	Hadoop provides two types of schedulers. fair scheduler and Capacity scheduler in Hadoop. The scheduler in Hadoop becomes the pluggable component.	MapReduce provides the Fair Scheduler, which provides a way to share large clusters.	HBase Scheduler uses a polling to change state at controlintervals . if required based on configuration it can trigger jobs.	Hive schedules table every hour by use of Oozie schedule.	Spark deals With its own flow scheduler due to in-memory computation.	Oozie is the tool for workflow scheduler in Hadoop for Apache Pig – Secondly, writes a brief Pig script for each data file to extract the required data fields.
Cost	A mid-range Intel server is recommended an enterprise-class Hadoop cluster.	Map reduce Cost is high but Hadoop cluster, a mid-range Intel server is recommended for it.	The cost of HBaseis depends on your usage pattern; S3 listing and file transfer might cost money.	Hive is also open source, and built on top of Hadoop for data querying.	Spark is very costly	Pig is lower in cost to write and maintain compared to MapReduce
Developme nt	Hadoop is developed by Apache Software Foundation.	MapReduce is developed by Google for a new style of large data processing	HBaseis an open-source project that was incubated by Apache Software Foundation.	Hive was initially developed by Facebook, but also some other companies develop and use it.	. Spark is developed in the University of California after some time it's codebase donated to Apache Software Foundation	. Pig is originally developed by Yahoo & Facebook

5. Conclusion

In this paper we concentratedon Big Data &Hadoop along with six V's and big data tools. Traditional big data management systemsare not up to the mark to handle massive data sets. Many issuesarise while handling the big data due to some sort of lack of accuracy, lack of integrity, lack of privacy, etc. Some major issues have to come with the traditional big data management system. To overcome these types of issues Hadoop is the solution for the processing of large data sets very effective manner. It is a future technology which provides excellent scalability as from a single server to thousands of machines according to our requirements. We have discussed a big data tool to handle

heterogeneous data. This review is conducted to give academics an appropriate guideline in determining the promising region regarding the Hadoop. Hadoop is indeed a technology to store and process the huge data sets. Major concern that is associated with big data is ensuring its security and integrity. Apache Spark is another tool used in analytics of big data.It is faster performance than Map Reduce. There are almost all tools covered which deals with big data.

References

[1]. *Chunarkar-Patil P, Bhosale A. Big data analytics. Open Access J Sci. 2018;2(5):326–335. DOI: 10.15406/oajs.2018.02.00095*

[2]. *Toshifa, Aniruddh Sanga, Shweta Mongia, International Conference on Advancements in Computing & Management (ICACM-2019)*

[3]. *P. R. B. B, P. Saluja, N. Sharma, A. Mittal, and S. V. Sharma, "Cloud Computing for Internet of Things & Sensing Based Applications," Sensing Technology (ICST), pp. 374–380, 2012.*

[4]. *M. Junghanns, M. Neumann, and E. Rahm, "Management and Analysis of Big Graph Data: Current Systems and Open Challenges," in Handbook of Big Data Technologies, 2017, pp. 457–505.*

[5]. *S. Akter and S. F. Wamba, "Big data analytics in E-commerce: a systematic review and agenda for future research," Electronic Markets, pp. 173–194, 2016.*

[6]. *S. Srivastava, "Appraising a Decade of Research in the Field of Big Data 'The Next Big Thing,'" Computing for Sustainable Global Development (INDIACom), no. 2014, pp. 2171–2175, 2016.*

[7]. *J. Quackenbush, "Microarray data normalization and transformation," Nature Genetics, vol. 32, no. December, pp. 496– 501, 2002.*

[8]. *J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," Communications of the ACM, vol. 51, no. 1, pp. 107-113, 2008.*

[9]. *R. Casado and M. Younas, "Emerging trends and technologies in big data processing," 2014.*

[10]. *S. Venkatraman, S. Kaspi, Kiran Fahd, and R. Venkatraman, "SQL Versus NoSQL Movement with Big Data Analytics," International Journal of Information Technology and Computer Science, vol. 8, no. 12, pp. 59–66, 2016.*

[11]. *D. J. Abadi, P. A. Boncz, and S. Haritopoulos, "Column-oriented Database Systems," Proceedings of the VLDB Endowment, vol. 2, no. 2, pp. 1664–1665, 2009.*

[12]. *F. Naumann, "Data Profiling Revisited," vol. 42, no. 4, 2013.*

[13]. *M. Seeger, "Key-Value stores: a practical overview," pp. 1–21, 2009.*

[14]. *B. Baesens, R. Bapna, J. R. Marsden, J. Vanthienen, and J. L. Zhao, "Transformational issues of big data and analytics in networked business," MIS quarterly, vol. 38, no. 2, pp. 629–631, 2014.*

[15]. *S. Amini and C. Prehofer, "Big Data Analytics Architecture for Real- Time Traffic Control," Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2017.*

[16]. *S. Yu, "Big Privacy: Challenges and Opportunities of Privacy Study in the Age of Big Data," IEEE Access, vol. 4, pp. 2751–2763, 2016.*

[17]. *B. Zhou and J. Pei, "The k-anonymity and l-diversity approaches for privacy preservation in social networks against neighborhood attacks," Knowledge and Information Systems, vol. 28, no. 1, pp. 47– 77, 2011.*

[18]. *A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkatasubramanian, "L-diversity: Privacy beyond k-anonymity," ACM Transactions on Knowledge Discovery from Data (TKDD), vol. 1, no. 1, 3–es, 2007.*

[19]. *N. Li, T. Li, and S. Venkatasubramanian, "T-closeness: Privacy beyond k-anonymity and l-diversity," in2007 IEEE 23rd International Conference on Data Engineering, IEEE, 2007, pp. 106–115.*

Lane Detection for Autonomous Vehicles using Deep Learning Methods

Manish T. Wanjari
Department of Computer Science
SSESA's Science College
Nagpur, Maharashtra
✉ mwebwanjari@gmail.com

Harshala B. Pethe
Department of Computer Science
Dr. Ambedkar College, Deekshabhoomi
Nagpur, Maharashtra
✉ harshapethe@gmail.com

ABSTRACT

Lane departure warning and adaptive cruise control in autonomous cars are two common uses for lane detection, which is the act of recognizing lane markers as approximated curves. In this research paper, we discussed a comprehensive survey of recent visual-based lane detection methods, especially those based on deep learning. We provide qualitative and quantitative comparisons with several existing methods. Additionally, we undertake a comparative evaluation of some pre-existing lane detection models on a small subset of the TuSimple dataset, offering insights into their performance under controlled conditions. We also discussed the advantages and limitations of the existing methods and suggested some possible directions for future research. In this paper we implement the methodologies of lane detection for autonomous vehicles and produce the results.

KEYWORDS: *Lane detection, ResNet, CNN, Python, Deep Learning, Pytorch, TuSimple, Lane segmentation, Classification, Transformer.*

INTRODUCTION

With the rapid evolution of high-precision optic sensors and electronic sensors, alongside the development of highly efficient computer vision and machine learning algorithms, real-time understanding of driving scenes has become increasingly attainable. Both academic and industrial research groups have invested substantial resources in advancing algorithms for driving scene understanding, focusing on applications for autonomous vehicles or advanced driver assistance systems (ADAS). Lane detection, amongst the several study areas for driving scene comprehension, holds particular significance as a fundamental aspect [1].

Since deep segmentation approaches inherently have more semantic representation capabilities than standard image processing methods, they are becoming popular for lane recognition, which requires higher-level semantic analysis of lanes [2].

Traditional lane detection techniques depend upon visual information, employing techniques like HSI color models, edge extraction algorithms, and tracking for post-processing. These methods also include machine learning algorithms including template matching and support vector machines. HSI color models segment lane lines based on hue, saturation, and intensity values, effectively handling various lighting conditions and road materials but requiring precise color thresholding and camera parameter calibration.

Edge extraction algorithms detect lane boundaries using filters like Sobel or Canny on grayscale images, reducing noise and enhancing lane contrast, but they are sensitive to edge parameters and the presence of shadows, cracks, or other road markings. Tracking for post-processing enhances stability and accuracy by utilizing temporal information from previous frames, effectively managing lane changes and occlusions but necessitating reliable initialization. Template matching matches lane lines using predefined templates, accommodating curved and dashed lanes but requiring numerous templates for diverse scenarios and incurring high computational costs. Support vector machines utilize a classifier to differentiate lanes from non-lane pixels, effective in complex road environments but requiring abundant labelled data for training and feature extraction to reduce input dimensionality.

In contrast, the advancement of deep learning, lane detection approaches relying upon deep neural networks have demonstrated superiority, treating the problem as semantic segmentation. These methods include specialized convolution operations, lightweight approaches for real-time applications, alternative formulations such as sequential prediction with clustering, and techniques such as LSTM networks or Fast-Draw to handle long lane structures and predict lane directions sequentially. Clustering approaches have also been proposed for accurate segmentation of lane markings, alongside 3D formulations to address non-flat ground issues in lane detection. An anchor-driven ordinal classification approach efficiently and accurately detects lanes for real-time applications [3].

METHODS AND MATERIAL

Data Preparation

The TuSimple dataset is a collection of 3626 video clips of 1 second duration each, captured by cameras mounted on a vehicle dashboard. Each video clip contains 20 frames, of which the last frame is annotated with the lane boundaries. The annotations are in JSON format, containing the x and y coordinates of the lane points, as well as the file path of the image. The dataset covers various scenarios, such as urban roads, highways, day and night, different weather conditions, and different lane types.

To prepare the data for our deep learning model, we performed the following steps:

1. We extracted a subset of the TuSimple dataset consisting of 100 video clips from the training set and 75 clips for the testing set all from the highway scenario.

2. We ensured that the data distribution was balanced across the sets, in terms of the number of lanes, the lane types, and the scenarios.

3. The Dataset consists of positive ground truth labels for the lanes for target output.

Table 1: Data Preparation for Study

Dataset	Train	Test	Lane	Environment
TuSimple	100	75	`<5	Highway

Tools and Libraries

We have used state of the art Python tools and libraries for the implementation of various CNN based Algorithms and Architectures. The following libraries and tools are used for the implementation, PyTorch offers dynamic neural network development using tensors and automatic differentiation. Numpy excels in numerical computation, enabling efficient operations on arrays and matrices. Pandas simplifies data analysis with high-level structures like Series and DataFrame, while Matplotlib customizes plots and graphs. An intuitive interface for applying machine learning algorithms is offered by Scikit-learn, including data pre-processing and evaluation utilities.

Architectures and Algorithms

We implemented various state of the art CNN based lane detection models having different architectures and backbone algorithms.

1) The CNN architecture

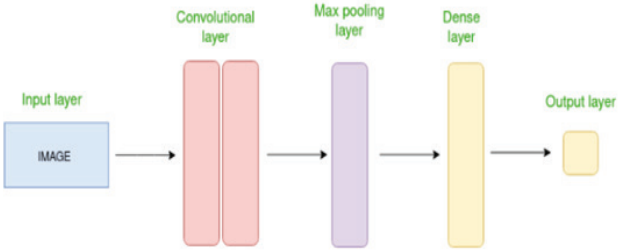


Fig. 1 Simple CNN Architecture

In this lane detection paper, the Convolutional Neural Network acts as the central component, orchestrating the functionality of all implemented models [5].

2) Structure Aware Deep Lane Detection Architecture

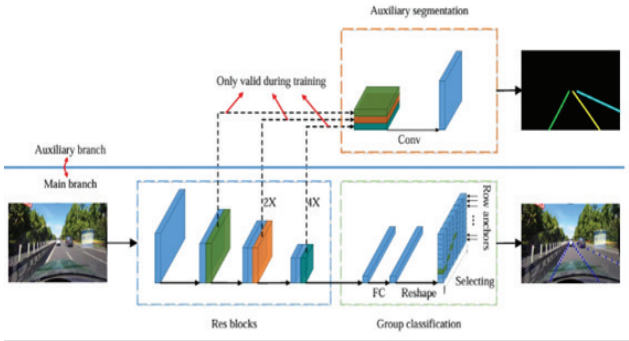


Fig. 2 SADL Architecture

This architecture efficiently processes images for lane detection. It uses a main branch with residual blocks to extract features and detect lanes through row anchor selection and group classification. It achieves remarkable processing speeds, capable of handling over 300 frames per second, making it highly suitable for real-time applications in various driving conditions.

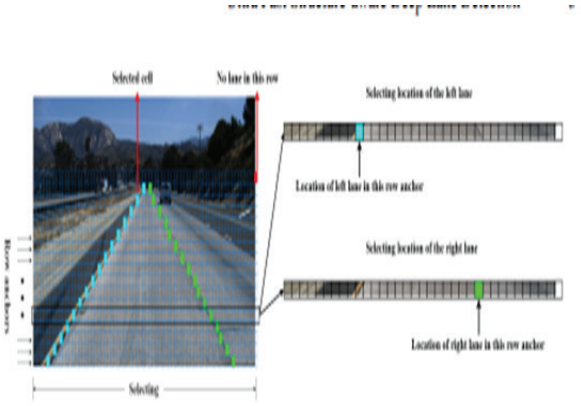


Fig. 3 Illustration of selecting on the left and right lane.

A detailed view of the selection of a row is displayed in the right portion of the above picture. The predetermined row positions are known as row anchors, and our formulation is characterized by horizontal selection on each row anchor. A background gridding cell is added to the image on the right to show that there are no lanes in this row [2].

Backbone Algorithm used: ResNet18

ResNet-18 is a 18-layer residual network that consists of five convolutional blocks, each with a different number of filters and layers. The final layer is a global average pooling layer, followed by a fully connected layer with 1000 output units for the ImageNet classification task [9].

3) Hybrid Anchor Driven Ordinal Classification Architecture

This architecture presents a novel approach to lane detection. It treats the task as an anchor-driven ordinal classification problem, utilizing global features for efficient and accurate detection.

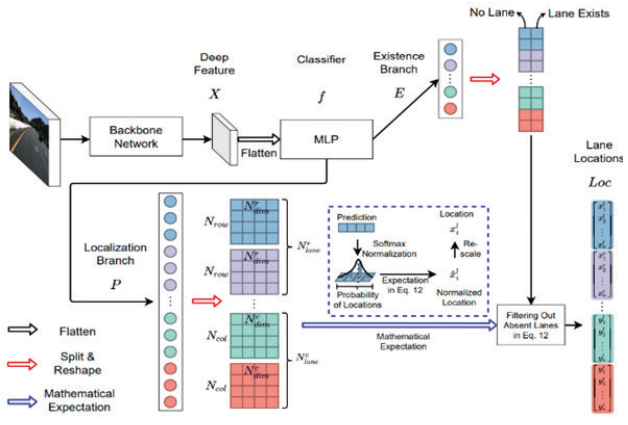


Fig. 4 HAOC architecture

The method handles difficult situations including strong occlusions and harsh lighting conditions by representing lanes with sparse coordinates on hybrid anchors, which combine row and column anchors. This technique is appropriate for real-time applications in autonomous driving systems since it drastically lowers processing costs and processes pictures quickly [7].

Backbone Algorithm: ResNet34, ResNet18

ResNet-34 is a 34-layer residual network that follows the same structure as ResNet-18, but with more layers in each convolutional block [9].

4) LSTR Architecture

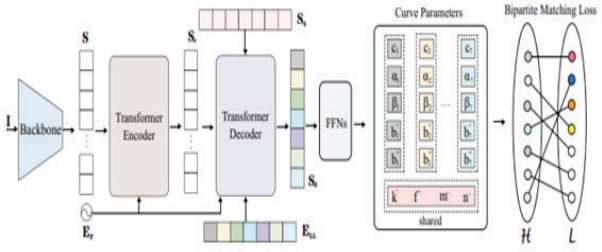


Fig. 5 LSTR Architecture

LSTR architecture applies the Transformer model's principles to the task of lane detection. It utilizes the self-attention mechanism to analyse sequences of road images, identifying and focusing on key features such as lane markings and road edges. The encoder processes the input data to extract spatial features and understand the context of the lanes within the scene. The decoder then uses this information to predict the position and trajectory of lanes across a series of images, allowing for accurate real-time lane detection. This architecture benefits from the Transformer's ability to handle long-range dependencies and variable input sizes, making it well-suited for the dynamic nature of road environments [4].

Transformer Algorithm

Transformers can be adapted for lane detection by leveraging their ability to handle sequential data. In lane detection, sequences can be the series of images or the pixels along the lanes in an image. The mechanism of self-attention permits the model to concentrate on pertinent segments of the input, such as lane markers, while ignoring irrelevant information. The encoder can process the input images to identify features that represent lanes, and the decoder can then predict the trajectory of lanes in a sequence of images. This approach can potentially improve

lane detection accuracy by considering the context and relationships between different parts of the road [10].

RESULTS AND DISCUSSION

Evaluation and metrics

The evaluation of the implemented models is done using the following metrics:

Precision: The precision of the model in identifying the positive class is measured by the ratio of properly predicted positive examples to the total number of anticipated positive instances. The precision formula is as follows:

Precision = TP/ (TP + FP)

Recall: It is the proportion of all real positive cases to all correctly projected positive instances. It gauges the model's memory for the positive class. Recall is calculated using the following formula:

Recall = TP/ (TP+FN)

F1 Score: The harmonic mean of recall and accuracy is what it is. When both measures are high, it balances them and returns a greater number. It comes in handy when there is an imbalance in the data or when recall and precision are equally crucial. For an F1 score, use this formula:

F1 = (2 x Precision x Recall)/ (Precision + Recall)

Accuracy: Accuracy is defined as the proportion of correctly predicted events to all occurrences. It evaluates the model's accuracy in classifying the data. For the lane detection models, the accuracy is calculated by iterating through each predicted lane and comparing it with the ground truth lane values

Accuracy = (Correctly predicted instances)/ (Total no. of instances)

Experiment and results

We compare the performance of three lane detection models: Ultra-Fast Structure Aware Deep Lane detection (UFLD-SADL), Ultra-fast lane detection Hybrid Anchor Ordinal Classification (UFLD-HAOC), and LSTR (end-to-end lane shape detection with transformer). The models were evaluated on the subset of TuSimple dataset of images containing road scenes with various lane markings.

The tables below summarizes the results of the evaluation:

Table 2: Optimizers used in Models

Model	Algorithm	Optimizer
SADL	ResNet	Adam
HAOC	ResNet_18	SGD
HAOC	ResNet_34	SGD
LSTR	Transformer	Adam

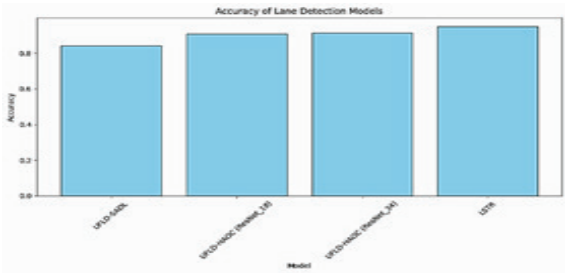
Table 3: Comparison of FP, FN, TP for Different Algorithms

Model	Algorithm	Head	FP	FN	TP
SADL	ResNet	CNN	0.46	0.40	0.54
HAOC	ResNet_18	CNN	0.18	0.16	0.81
HAOC	ResNet_34	CNN	0.17	0.15	0.81
LSTR	Transformer	CNN	0.05	0.06	0.94

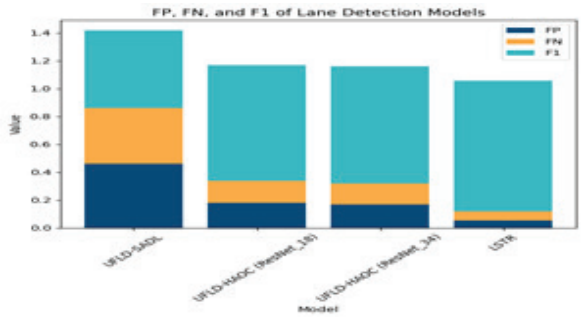
Table 4: Comparison of Precision, Recall, Accuracy and F1 Score Across Implemented Models

Model	Algorithm	Precision	Recall	Accuracy	F1
SADL	ResNet	0.54	0.58	84.15%	56%
HAOC	ResNet_18	0.81	0.83	90.91%	82%
HAOC	ResNet_34	0.83	0.84	91.23%	84%
LSTR	Transformer	0.94	0.93	95.54%	94%

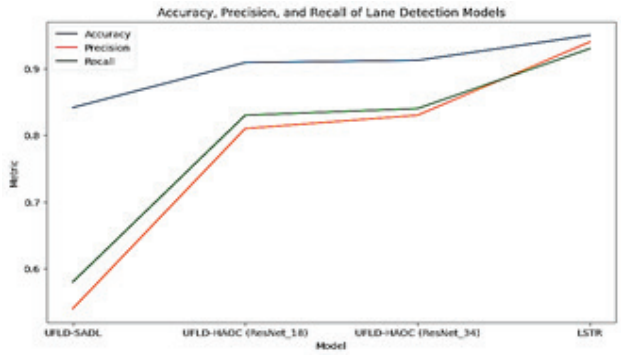
As shown in the table, LSTR achieved the best overall performance, with a F1 score of 94% and an accuracy of 95.54%. UFLD-HAOC with ResNet_34 backbone also performed well, with a F1 score of 84% and an accuracy of 91.23%. UFLD-SADL had the lowest performance among the three models, with a F1 score of 56% and an accuracy of 84.15%. These results suggest that LSTR is a promising model for lane detection, especially in applications where high accuracy is required. However, UFLD-HAOC with ResNet_34 backbone may be a good alternative if computational efficiency is a concern.



Graph 1: Comparison of Accuracy across the implemented lane detection models



Graph 2 Comparison of FP, FN and F1 Score across the implemented lane detection models



Graph 3 Comparison of Accuracy, Precision and Recall across the implemented lane detection models

Fig. 6 Lane detection result for UFLD-SADL

Fig. 7 Lane detection result for UFLD-HAOC (ResNet_18)

Fig. 8 Lane detection result for UFLD-HAOC (ResNet_34)

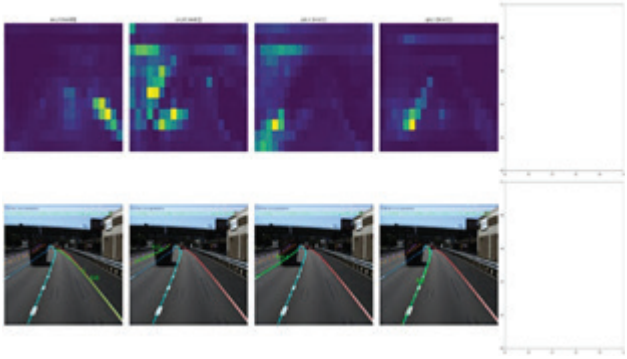


Fig. 9 Lane detection result for LSTR

CONCLUSION

In this research paper we studied, lane detection capabilities of three deep learning models: UFLD-SADL, UFLD-HAOC, and LSTR. While all models exhibited strengths, limitations emerged based on training data size and computational demands. LSTR show cased exceptional accuracy (F1 score: 94%, accuracy: 95.54%) but suffered from extensive training time due to its iterative approach. This highlights its potential, particularly for accuracy-critical scenarios, but also necessitates exploring computational optimizations for real-time applications.

UFLD-HAOC struck a compelling balance, achieving good accuracy (F1 score: 84%, accuracy: 91.23%) with significantly faster training compared to LSTR. This combination presents itself as a strong contender for practical applications with limited computational resources. UFLD-SADL (F1 score: 56%, accuracy: 84.15%) underperformed despite its promising structure-aware approach.

REFERENCES

1. Zou, Qin & Jiang, Hanwen & Dai, Qiyu & Yue, Yuanhao & Chen, Long & Wang, Qian, “Robust Lane Detection

From Continuous Driving Scenes Using Deep Neural Networks". IEEE Transactions on Vehicular Technology. PP. 1-1. 2019, DOI: 10.1109/TVT.2019.2949603.

2. Qin, Z., Wang, H., Li, X.,"Ultra Fast Structure-Aware Deep Lane Detection", In: Vedaldi, A., Bischof, H., Brox, T., Frahm, JM. (eds) Computer Vision. ECCV 2020. Lecture Notes in Computer Science, vol 12369. Springer, Cham, 2020, DOI: 10.1007/978-3-030-58586-0_17.
3. Wei Wang, Hui Lin, Junshu Wang, "CNN based lane detection with instance segmentation in edge-cloud computing", 2020, DOI: 10.1186/s13677-020-00172-z.
4. Ruijin Liu and Zejian Yuan and Tie Liu and Zhiliang Xiong, "End-to-end Lane Shape Prediction with Transformers", 2020, DOI: 10.48550/arXiv.2011.04233.
5. Pizzati, Fabio & García, Fernando. "Enhanced free space detection in multiple lanes based on single CNN with scene identification",2019, DOI: 10.1109/IVS.2019.8814181.
6. D. Neven, B. De Brabandere, S. Georgoulis, M. Proesmans, and L. Van Gool, "Towards End-to-End Lane Detection: an Instance Segmentation Approach", 2018, DOI: 10.48550/arXiv.1802.05591.
7. Zequn Qin and Pengyi Zhang and Xi Li, "Ultra Fast Deep Lane Detection with Hybrid Anchor Driven Ordinal Classification", 2022, DOI:10.48550/arXiv.2206.07389.
8. Zhengyang Feng, Shaohua Guo, Xin Tan, Ke Xu, Min Wang, Lizhuang Ma, "Rethinking Efficient Lane Detection via Curve Modeling", 2023, DOI:10.48550/arXiv.2203.02431
9. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, "Deep Residual Learning for Image Recognition", 2015, DOI:10.48550/arXiv.1512.03385.
10. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, "Attention Is All You Need", 2017, DOI:10.48550/arXiv.1706.03762.

Analysing Machine Learning Approaches for Fake News Detection

Harshala B. Pethe
Department of Computer Science
Dr. Ambedkar College, Deekshabhoomi
Nagpur, Maharashtra
✉ harshapethe@gmail.com

Manish T. Wanjari
Department of Computer Science
SSESA's Science College
Congress Nagar
Nagpur, Maharashtra
✉ mwebwanjari@gmail.com

ABSTRACT

Fake news articles on social media have a significant concern due to its potential to cause destruction to society and the nation. It can spread false information about a country, including fabricating government expenses, leading to various societal problems. The proliferation of fake news and scams coincided with the advent of the internet, aimed at misleading people, gaining followers, and even perpetuating psychological conflicts. The dissemination of news across multiple media platforms without adequate fact-checking has exacerbated the issue of fake news, making it a pervasive problem. The article will address a novel a technique for spotting fabricated information by leveraging the Scikit Learn library for data processing plus Natural Language Processing. The model utilizes TF-IDF vectorization for feature extraction, enabling the identification of key patterns in textual data and converting text into numerical data. In this study report, we shall apply machine learning techniques (for Recognizing False News), comprising Decision Tree, Logistic Regression, Random Forest with Support Vector Machine. The accuracy of all the algorithms is calculated and analysed.

KEYWORDS: *Fake News Detection (FND), Machine Learning (ML), TF-IDF vectorizer, Stop words, Stemming, tokenization, Logistic Regression (LR), Random forest, SVM, Decision tree.*

INTRODUCTION

Living in the digital age brings many benefits, but it also comes with its challenges, one of which is the prevalence of fake news. Fake news refers to false information intentionally created to deceive people, often with the aim of damaging someone's reputation or promoting a particular agenda [1]. Facebook, Twitter, and Instagram have become primary sources of news. However, the reliability and accuracy of news on social media can be questionable compared to traditional news sources such as TV or newspapers [2].

Fake news is deliberately fabricated to mislead, whereas rumours are uncertain stories that may or may not be true but are not intentionally created to deceive [3].

Fake news detection is crucial for several reasons. Firstly, it can have serious implications for public health. For instance, misinformation about vaccines can bring about reduced vaccination rates, resulting in the spread of preventable diseases.

Now, scientists are turning to machine learning technology to address this issue. Automated data analysis and decision-making are made possible by machine learning. By training machine-learning models with examples of fake and real news, researchers aim to develop automated systems capable of discriminating the authenticity of news [2].

Fake News Detection Architecture

Understanding the problem is the initial step, involving thorough analysis to grasp the key components and objectives of solving it. This includes breaking down the problem into smaller, manageable parts, identifying crucial information and understanding the interconnectedness of elements. This step lays the foundation for devising effective solutions by offering a thorough comprehension of the current issue.

Next step is dataset selection, where relevant data is chosen to address the problem. Since we are dealing with fake news detection, selecting a dataset containing

information about news articles is essential. The dataset should encompass various attributes relevant to news articles, facilitating the classification of news into different categories.

Following dataset selection is the pre-processing step, wherein the dataset undergoes filtration to eliminate any null values or irrelevant data. This ensures that the dataset used for analysis is clean and conducive to accurate modelling and prediction. Pre-processing enhances the quality of data and prepares it for further analysis and modelling stages.

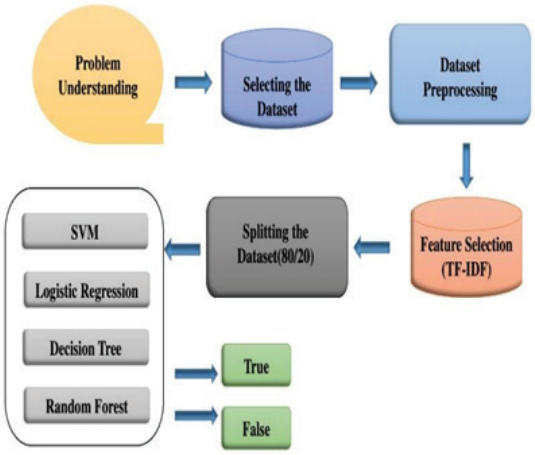


Fig. 1. Fake News Detection Architecture

METHODOLOGY USED FOR FAKE NEWS DETECTION

The steps below are part of the methodology applied for detecting fake news: data collection, pre-processing, feature extraction, model training, also evaluation. Central to our approach is the utilization of datasets sourced from Kaggle, which provide essential data for training as well as testing machine learning frameworks. We outline the details of these datasets, emphasizing their structure, size, and attributes utilized in our analysis.

Data Collection

The Fake News Dataset, comprises two files: fake.csv and true.csv, sourced from Kaggle. These files contain a total of 23481 rows and 21417 rows, respectively, with each row representing a news article. Both files consist of five columns, with the primary attribute of interest being the title. Additionally, each article is labelled with binary indicators, denoting whether it is classified as “fake” or “real” news.

Tools and Libraries

In this paper, we utilized Jupyter Notebook as our IDE for Python programming, providing an interactive environment for code execution and documentation. Python operated as our primary programming language due to its versatility and extensive libraries. Key libraries employed include Use Scikit-learn to put different machine learning algorithms into practice, Natural Language Toolkit for text processing and Natural Language Processing tasks, Pandas for efficient data manipulation and analysis, and Matplotlib for data visualization.

Algorithms used for Fake News Detection

1) Logistic Regression: Analyzing the connection between input characteristics (X) and a binary output (y), logistic regression is a crucial statistical model in machine learning, such as distinguishing real from fake news articles [4]. Unlike linear regression, logistic regression focuses on classification tasks by estimating the likelihood of the result with a sigmoid function, yielding values ranging from 0 to 1. [7].

2) SVM: Provide assistance for problems involving regression and classification, Vector Machine is a supervised learning technique, delineating data points in a multidimensional space based on specific features [10]. It excels in binary classification, such as distinguishing true from false articles, by establishing hyperplanes as decision boundaries. These hyperplanes effectively segregate data points, yielding high precision results ideal for semi-structured datasets and high-dimensional spaces [7].

3) Decision Tree: Decision tree algorithms, pivotal in supervised learning, classify data by iteratively splitting it based on specific parameters [10]. This tree-like structure comprises decision nodes representing attribute conditions and branches indicating decision rules. Leaf nodes store outcomes, forming classification rules. Decision trees excel in identifying important variables and relationships, aiding in feature generation and efficient data exploration [10]. Despite benefits such as interpretability, they may overfit and perform poorly on testing data, particularly with numerous sparse features [11].

4) Random Forest: The machine learning algorithm Random Forest is widely used, influences multiple decision trees to enhance decision-making through techniques like bagging [5]. By constructing numerous trees and using feature subsets, it ensures diverse predictions, leading to higher accuracy compared to single tree models. Random

Forest's simplicity and superior results make it a preferred choice for various tasks [6]. Techniques like bootstrapping and feature randomness yield uncorrelated trees, enhancing its robustness in classification tasks [7].

Data Preprocessing

Data Preprocessing involves organizing and refining data, crucial in contexts like social media where content is often unstructured. Techniques like text preprocessing in NLP streamline data for analysis, including stemming, tokenization, stopwords removal, and special character handling.

1) Tokenization and Stopwords: Tokenization breaks text into individual units like words or tokens, aiding additional investigation.

2) Encoding: Encoding converts categorical data into numerical representations for analysis, aiding in model training tasks [10].

3) Feature Extraction: Feature Extraction simplifies raw data for machine learning models, aiding comprehension and analysis. TF-IDF vectorizer calculates the relative importance of words in documents, crucial for tasks like search engine scoring and text summarization. TF computes word frequency in a document, whereas IDF reduces the importance of common terms. TF-IDF assigns values proportional to word occurrences in a document but offsets by corpus frequency, aiding in text similarity checks and sentence matching [8].

TF (t, d) = No. of times t occurs in d/ (Total word count of document 'd'

IDF (t, d) = Total number of documents/ (No.of documents with item t in it)

TF-IDF (t, d) = TF (t, d) * IDF(t)

Train-Test Split

Train-Test Split divides the dataset into training, testing subsets, essential for model evaluation. Model Training employs machine learning algorithms like SVM, Logistic Regression, Random Forest and Decision Tree. Classifier to train model, facilitating accurate prediction tasks. To ensure robust model training and evaluation, the Fake News dataset was meticulously divided into an 80-20 ratio, whereby 80% of the data was designated for training purposes, whereas the final 20% served for model validation and testing, thus optimizing generalization and performance assessment.

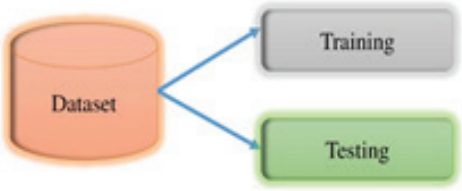


Fig. 2. Train-Test Split

Evaluation and Metrics

1) Confusion Matrix: The effectiveness of classification models is evaluated using a confusion matrix that demonstrates instances of True Positive (TP), True Negative (TN), False Positive (FP), along with False Negative (FN).

	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

Fig. 3. Confusion Matrix

2) Precision: Precision measures how well the model predicts favourable outcomes. It's especially helpful when the cost of false positives is considerable. It's the ratio of genuine positives to all expected positives.

Precision = TP/ (TP + FP)

3) Recall: Recall gauges how well the model can locate all pertinent examples, especially true positives. It's important when the cost of false negatives is considerable since it's the ratio of real positives to all actual positives.

Recall = TP/(TP+FN)

4) F1 Score: F1 score is defined as the harmonious average of recall with accuracy. If there is a discrepancy in the costs associated with false positives and false negatives or where there is an uneven distribution of classes, this metric can be used to evaluate models since it strikes a compromise between accuracy and recall.

$$\text{F1 Score} = (2 * \text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$$

5) Accuracy: Taking into account both true positives and true negatives, accuracy gauges how accurate the model's predictions are overall.

$$\text{Accuracy} = (TP+TN)/(TP+TN+FP+FN)$$

RESULTS

We evaluated the efficacy of four machine learning algorithms—Logistic Regression, Support Vector Machines, Decision Trees, along Random Forest Classifiers—in detecting false news. The "Fake News" dataset, which comprises of the two CSV files true.csv as well as fake.csv, was used to test the models. Numerous performance metrics, notably F1 score, recall, accuracy, as well as precision, were used to evaluate each machine learning algorithm after it had been both trained then tested on the respective sets of data. Additionally, the computational efficiency and scalability of each algorithm were also considered, contributing to a holistic comparison of their performance.

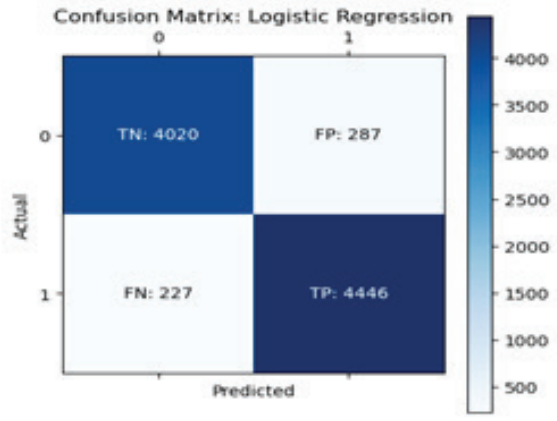


Fig 4. Confusion Matrix: Logistic Regression

For Logistic Regression, the confusion matrix indicates the following:

In 4020 cases, the model accurately classified instances as negative (TN). However, it misclassified 287 instances as positive when they were actually negative (FP). Additionally, in 227 instances, negative as anticipated by its design when they were actually positive (FN). On the positive side, the model correctly classified 4666 instances as positive (TP).

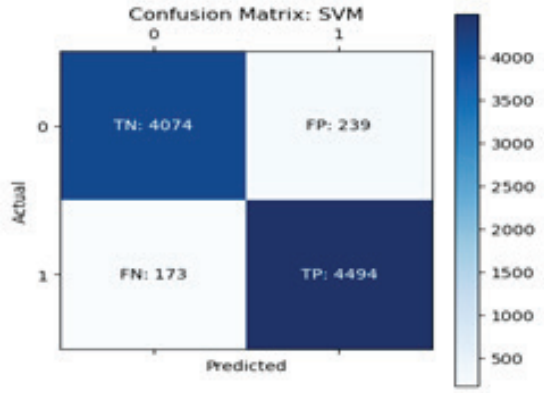


Fig. 5. Confusion Matrix: SVM

For SVM, the confusion matrix indicates the following:

In 4074 cases, the model accurately classified instances as negative (TN). However, it misclassified 239 instances as positive when they were actually negative (FP). Additionally, in 173 instances, negative as anticipated by its design when they were actually positive (FN). On the positive side, the model correctly classified 4494 instances as positive (TP).

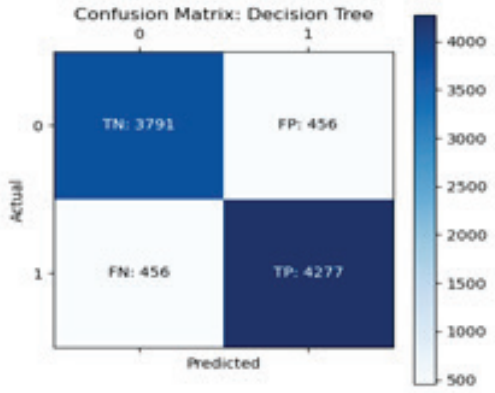


Fig. 6. Confusion Matrix: Decision Tree

For Decision Tree Classifier, the confusion matrix indicates the following:

In 3791 cases, the model accurately classified instances as negative (TN). However, it misclassified 456 instances as positive when they were actually negative (FP). Additionally, in 456 instances, negative as anticipated by its design when they were actually positive (FN). On the positive side, the model correctly classified 4277 instances as positive (TP).

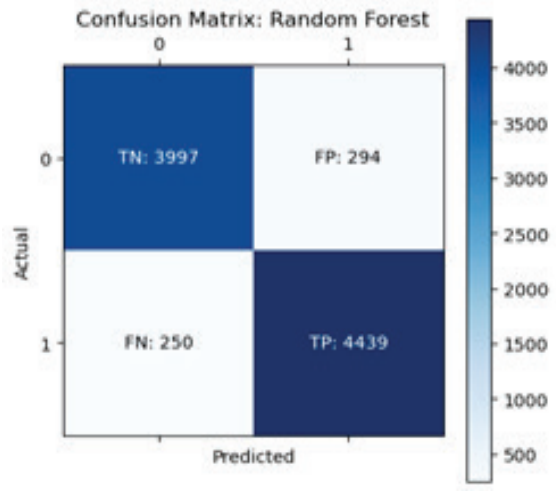


Fig. 7. Confusion Matrix: Random Forest

For Random Forest Classifier, the confusion matrix indicates the following:

In 4020 cases, the model accurately classified instances as negative (TN). However, it misclassified 287 instances as positive when they were actually negative (FP). Additionally, in 227 instances negative as anticipated by its design when they were actually positive (FN). On the positive side, the model correctly classified 4666 instances as positive (TP).

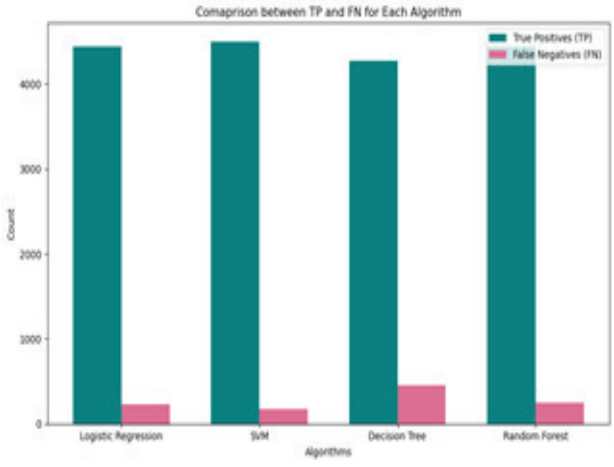


Fig. 8. Comparison between TP and FN

The comparison between True Positives (TP) and False Negatives (FN) provides important indications regarding how well the categorization strategy is working. While True Positives symbolize instances correctly identified as positive, False Negatives designate instances falsely categorized as negative when they are actually positive.

Table 1: Comparison Of Metrics

Algorithms	Precision	Recall	F1 Score	Accuracy
Logistic Regression	0.94	0.95	0.95	94.27%
SVM	0.95	0.96	0.96	95.41%
Decision Tree	0.90	0.90	0.90	89.98%
Random Forest	0.94	0.95	0.94	93.94%

The above table shows the comparison of key metrics used for all the algorithms applied in this task.

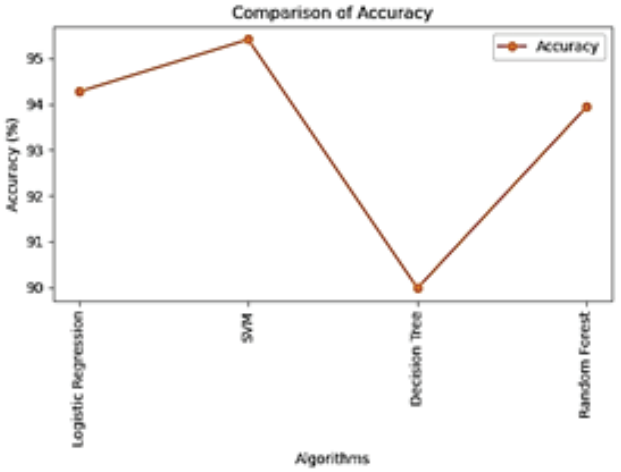


Fig. 9. Comparison of Accuracy

CONCLUSION

Through the utilization of diverse algorithms comprising Logistic Regression, Support Vector Machine, Decision Tree Classifier, along with Random Forest Classifier, we aimed to distinguish between authentic and fabricated news articles across various datasets. Our analysis showcases promising performance metrics, notably accuracy. Logistic Regression and Support Vector Machine emerged as frontrunners, achieving accuracy rates of 94.27% and 95.41%, respectively. While Decision Tree and Random Forest also exhibited commendable accuracy with rates of 89.98% and 93.94% respectively. Conclusively, our findings affirm that SVM stood out as the best-performing algorithm, boasting the maximum accuracy relative to all other methods employed.

REFERENCES

1. Pshko Rasul Mohammed Amin, “Fake News Detection Using Machine Learning”, Indonesian Journal of

Computer Science, Vol. 12, No. 4, 17 Ags 2023, ISSN 2549-7286.

2. Uma Sharma, Sidarth Saran and Shankar M. Patil, "Fake News Detection Using Machine Learning Algorithms", International Journal of Creative Research Thoughts (IJCRT), Volume 8, Issue 6 June 2020, ISSN: 2320-2882.

3. M. F. Mridha, Ashfia Jannat Keya, MD. Abdul Hamid, Muhammad Mostafa Monowar, and MD. Saifur Rahman, "A Comprehensive Review on Fake News Detection With Deep Learning", IEEE Access, VOLUME 9, 18, 2021.

4. Johnson Adeleke Adeyiga, Philip Gbounmi Toriola, Temitope Elizabeth Abioye, Adebisi Esther Oluwatosin , oluwasefunmi 'Tale Arogundade, "Fake News Detection Using a Logistic Regression Model and Natural Language Processing Techniques", Research Square, 14 Jul, 2023, DOI: 10.21203/rs.3.rs-3156168/v1.

5. Fopa Yuffon Amadou Olabi, Mohamadou Moctar, Mikayilou Namba, "Fake News Detection: A Machine Learning Approach using Automated-Text Analysis Technique" ResearchGate, 13 October 2021.

6. Pawar A B, Jawale M A, Kyatanavar D N, "Analyzing Fake News Based on Machine Learning Algorithms", 2020, doi:10.3233/APC200146.

7. Ajit Patil, "Fake News Detection Using Machine Learning Algorithms", Researchgate, VOLUME 8, ISSUE 12, 2021, August 2022, ISSN NO : 1869-9391.

8. Anjali Jain, Harsh Khatter, Amit Kumar Gupta, "A smart system for fake news detection using machine learning", Sep 2019, DOI:10.1109/ICICT46931.2019.8977659.

9. Vasu Agarwal, Parveen Sultana, Srijan malhotra, Amitrajit Sarkar, "Analysis of Classifiers for Fake News Detection", international conference on recent trends in advance computing 2019, icrtac 2019, 2019.

10. Z Khanam, B N Alwasel, H Sirafi, M Rashid, " Fake News Detection Using Machine Learning Approaches", ResearchGate, 2021, doi:10.1088/1757-899X/1099/1/012040.

11. Abdulaziz Albahr, Marwan Albahar, "An Empirical Comparison of Fake News Detection using different Machine Learning Algorithms", ResearchGate, Vol. 11, No. 9, 2020, 03 February 2022, DOI: 10.14569/IJACSA.2020.0110917.

Study of Classification Models using Ensemble based and Non Ensemble based Mining techniques using Astronomical Data

S. R. Gedam*[1], R. S. Gedam[2], R. A. Ingolikar[3]
[1]*Department of Computer Science, Shivaji Science College, Congress Nagar, Nagpur*
[2]*Department of Physics, Visvesvaraya National Institute of Technology, Nagpur*
[3]*Department of Computer Science, Saint Francis De' Sales College, Seminary Hills, Nagpur*
[1]*shilpagedam2020@gmail.com*, [2]*rupeshgedam411@gmail.com*,
[3]*ranjana.ingolikar@gmail.com*

Abstract: *Accurate classification of huge data is a matter of concern for Data miners. In this paper, study of various data mining models for classification of astronomical data is done. Ensemble based and non-ensemble based methods are used for classification. Summary of all classification results is presented. Comparative analysis of classification results of Ensemble based and non-ensemble based classifier is done. The result shows that classification accuracy of Ensemble based classifier is better than non-ensemble based classifier .*

Keywords: **Ensemble based Mining, Non-Ensemble based Mining, Random forest, Weighted Random forest, Filtered classifier**

1. INTRODUCTION

Classification is troublesome if the data is huge. Classification model can be prepared using ensemble based and non-ensemble based methods. Ensemble based method considers the results of various individual classifiers and then aggregates the outcome for better classification results. Non Ensemble based methods are based on single classifier. Astronomical data (related to celestial bodies) being massive in number is considered here for classification.

Similar work related to astronomical data is done by SergiiKhlamov et al. used collection of light technology software for processing astronomical information. They also described the benefits of Online Data Analysis System for solving data mining problem[1] .TheeranaiSangjan et al presented a data level approach to solve imbalance data problem. They used classifiers like K-Nearest Neighbor, decision tree and Support Vector Machine for Investigation on a data set of Light curve profile[2]. M. Klush proposed a novel hybrid neural network approach for fully automated spectral and luminosity classification of stars. Hybrid neural system used neural classifiers and a semantic networks for similarity based reasoning [3]. Shiyu Deny et al applied Manhatten Distance density algorithm to variety of spectral data and concluded that the average classification stable number of the Manhatten Distance Density algorithm is smaller and the computing time is shorter [4]. Liangping Tu et al used local mean based K- Nearest Neighbor method for automated classification of Galaxies and Quasars. Their experimental results showed that local mean based K- Nearest Neighbor performs best and better than KNN [5]. Zhenping YI et al tried to evaluate the effectiveness of random Forest on stellar spectra. Their results also showed that random forest gave better efficiency and less root mean square error as compared to Multilayer perceptron network [6]. Jiang Bin et al presented a novel technique for automatically classifying massive stellar spectra selected from SDSS. Their results indicated the classification accuracy upto 90% [7]. In an attempt to classify

*S. R. Gedam

Celestial body we are dealing with Astronomical data (classification of celestial object-Star).

2. DATA

The astronomical data is generated using the spectra available on Slogan Digital Sky Server (SDSS)-10 [8]. SDSS provides information about various celestial bodies in the sky. SDSS provides data of about ~10^9 objects in the sky. Parameters that are considered for classification of Star are right ascesion of star, declination of star, intensity of light from star, wavelength of light, radial velocity of star, redshift of an object, temperature and colour of star. Classification model identifies the class of the star (using training and test data). The star is classified as of class A, F, K, G and M. Table 1 shows the sample data of size 20 records.

Table 1. Sample size of 20 records

Sr no	RA	DEC	u	g	r	i	z	Redshift	Intensity	Wavelength	Colour	Radial velocity	Temperature	Class
1	53.63515	-5.42961	24.35	22.44	20.34	19.14	18.49	0.0002	7	7600	RED	60	3812.86	K
2	42.69537	1.15301	15.19	14.01	13.47	13.81	13.42	-0.0003	170	4000	VIOLET	-90	7244.43	F
3	356.6797	16.0897	17.6	16.7	16.42	16.32	16.26	-0.0006	120	4000	VIOLET	-180	7244.43	F
4	134.3652	42.7043	17.91	18.25	18.74	19.08	19.37	0.0007	50	3800	UV	210	7625.72	A
5	182.208	6.17178	18.91	18.93	19.3	19.52	19.62	0.0003	20	4100	VIOLET	90	7067.74	F
6	176.7317	1.15892	17.6	16.7	16.42	16.32	16.26	0.0002	80	3800	UV	60	7625.72	A
7	215.8062	0.42469	21.16	18.46	17.1	16.48	16.19	0.0005	48	7700	RED	150	3763.34	K
8	220.2851	1.17219	16.64	16.84	17.29	17.6	17.88	0.0001	160	3800	UV	30	7625.72	A
9	183.6272	1.08106	20.48	18.14	16.98	16.51	16.25	-0.0001	38	7600	RED	-30	3812.86	K
10	195.0071	-1.17447	15.83	14.69	14.3	14.16	14.12	-0.0002	500	4500	VIOLET	-60	6439.49	F
11	239.6784	1.19479	17.6	16.7	16.42	16.32	16.26	0.0002	40	5600	GREEN	60	5174.59	K
12	241.4534	1.10398	18.85	17.58	17.08	16.86	16.77	0.0005	32	3800	UV	150	7625.72	A
13	255.6053	64.7947	17.8	16.64	16.13	15.9	15.78	-0.0008	110	4700	BLUE	-240	6165.47	F
14	247.2433	1.13857	22.21	20.29	19.44	19.11	18.87	-0.0001	28	5600	GREEN	-30	5174.59	K
15	24.3607	1.24467	20.06	19.15	18.73	18.56	18.51	0.0007	17	3800	UV	210	7625.72	A
16	30.3438	1.15482	17.97	16.90	16.54	16.43	16.41	0	96	4500	VIOLET	0	6439.49	F
17	48.25589	1.09948	19.15	18.73	18.67	18.74	18.84	0	22	3900	UV	0	7430.18	F
18	220.2851	1.17219	16.64	16.84	17.29	17.6	17.88	0.0001	160	3800	UV	30	7625.72	A
19	114.4405	38.8352	21.53	20.29	20.27	20.29	20.21	0.0005	6	3800	UV	150	7625.72	A
20	46.63369	-6.64844	18.34	17.89	17.81	17.81	17.87	0.0001	48	3800	UV	30	7625.72	A

3. METHODS

Classification Models are prepared using Weka [9] using ensemble and non-ensemble based methods. Non-ensemble based methods considered are BayesNet, Naïve Bayes, Logistic, SMO, KStar, LWL, MultiClass Classifier, Filtered Classifier, InputMapped Classifier, Jrip and ZeroR.

BayesNet is a classification method which assumes all variables to be discrete and finite. BayesNet treats the attributes and class as a random variable. The random variable is defined by a probability density function. The probability that x object belongs to class C is calculated using probabitity density function P(C/x). This probability is determined using Bayes theorem [10].

Naïve Bayes algorithm is used for predictive modeling. It is collection of algorithms based on Bayes theorem. All the algorithms assume that every pair of feature being classified is independent of each other [11].

Logistic Regression classifier only supports binary classification problem. It has been adapted to support multiclass classification problem. It predicts a coefficient for each input value which is combined into a regression function and is converted using logistic function[12].

SMO stands for Sequential Minimal Optimization. It uses a specific algorithm used by Support Vector Machine. SMO works on numerical input values. It works by finding a line that separates the data into groups. SMO uses the instances in the training dataset that are closest to the line and separates the dataset into classes [13].

KStar is an instance based classifier. The class of a test instance is based upon the class of those training instances similar to it. This similarity is determined by some similarity function. Sometimes it may differ from other instance based learner by selecting different function such as entropy based distance function [14].

Locally Weighted Learning uses an instance based algorithm to assign instance weight which are then used for classification. A classification is obtained from Naïve Bayes model by taking the attribute value of the test data as input. It can be used for classification or regression [15]. The subset of data used to train each locally weighted naïve Bayes model are determined by a nearest neighbor algorithm [16].

Filtered Classifier filters the dataset or alter it in someway like deleting a particular attribute, removing misclassified instances etc. For classification it selects any arbitrary classifier but initially the data is passed through a filter [19].

InputMapped Classifier addresses the incompatibility between training data and test data. It does this by building a mapping between the model built using training data and incoming test instances. If some important attributes are not found in the incoming instance then this classifier puts some nominal attribute value which the classifier has not seen before and then the model developed is trained accordingly for proper classification [20].

Jrip is propositional rule learner which does repeated incremental purning for error reduction. This classifier is proposed by Willaim W. Jrip divides the data into classes and generates rules by including all attributes of the class and for each instance until all the classes have been covered [21].

ZeroR is the classification method which relies on the target and ignores all predictors. ZeroR classifier just predicts the majority category (class). Although there is no predictability power in ZeroR, it is useful for determining a baseline performance as a benchmark for other classification methods [22].

Ensemble based classification methods considered are Bagging, Multiclass classifier, Random Forest and Weighted Random Forest.

Bagging is also called Bootstrap Aggregation. Bagging is an ensemble method that combines the predictions from multiple machine learning algorithms to make more accurate prediction than an individual model. This procedure is used to reduce the variance for those algorithm that have high variance [17].

Multiclass Classifier is type of supervised machine learning. It uses Decision tree (data is visualized in the form of tree), Support vector machine (feature vector is high dimensional) and K nearest Neighbour (does not depend on structure of data) classifiers on the training data to predict the label for the test data [18].

Random forest is a recently proposed ensemble method [23] which uses many tree classifiers and aggregates their results. Random forest uses different bootstrap sample of data to construct each tree. Then a subset of predictors is chosen randomly, each node of the trees is split using the best among the subset instead of all predictors [24]. There are several ways to calculate output of random forest. The simplest is simple majority voting method for classification, while average output of trees is considered. **Weighted Random forest** is an updation of Random forest. Here forest is generated based on the weights assigned to trees. Weights are assigned in such a way that only useful trees are selected for model development [25].

4. Experimental Results

Classification Models are prepared for different sample sizes using ensemble and non-ensemble based methods. Sample Sizes are taken in an incremental manner. Initially all non-ensemble based methods are considered.

Table 2 and Table 3 shows the classification accuracy obtained using different non-ensemble based methods for Training data and Test data respectively.

Table 2. Classification Accuracy using non-ensemble based methods for training data

Sample Size / Methods	300	500	750	1000	1300	1500
BayesNet	100	100	100	100	99.904	99.917
Naïve Bayes	96.68	98.75	95.83	96.76	95.568	98.496
Logistic	100	100	100	100	100	100
SMO	89.21	94.25	96.17	96.63	95.279	95.489
KStar	100	100	100	100	99.904	99.666
LWL	95.07	88.25	87.17	88.15	82.081	94.152
Filtered Classifier	100	100	100	100	100	100
Input mapped Classifier	50.85	47.25	47.25	43.14	42.857	31.913
Jrip	100	99.75	99.75	99.88	99.904	99.917
ZeroR	50.21	47.25	47.25	43.14	34.584	31.913

Table 3. Classification Accuracy using non-ensemble based methods for test data

Sample Size / Methods	300	500	750	1000	1300	1500
BayesNet	98.3051	100	100	100	100	100
Naïve Bayes	88.1356	99	93.33	94.581	96.947	98.68
Logistic	100	96	97.33	95.567	98.473	98.68
SMO	86.4407	95	93.33	92.611	95.802	96.04
KStar	79.661	90	91.33	95.074	95.038	96.04
LWL	81.3559	88	87.33	66.997	82.06	96.37
Filtered Classifier	98.3051	100	100	100	100	100
Input mapped Classifier	50.8475	47	46	42.857	34.351	32.013
Jrip	100	99	99.33	100	100	100
ZeroR	50.8475	47	46	42.857	34.351	32.013

Table 4 shows the root mean square error generated by different Non ensemble based classifiers for different sample sizes.

Table 4.Root mean square error generated by different Non ensemble based classifiers

Non-Ensemble based Classifier	Sample Size					
	300	500	750	1000	1300	1500
Bayes Net	0.0925	0.0077	0.0016	0.015	0.0037	0.0015
Naïve Bayes	0.1802	0.0717	0.1421	0.1299	0.0972	0.0647
Logistic	0.0003	0.1216	0.0916	0.1332	0.0781	0.0705
SMO	0.3247	0.3183	0.3197	0.3197	0.3164	0.316
K Star	0.2768	0.1861	0.1643	0.1399	0.1239	0.126
LWL	0.2527	0.2183	0.2192	0.3033	0.2167	0.1756
Filtered Classifier	0.0823	0.005	0.002	0.008	0.002	0.002
Input mapped Classifier	0.3036	0.3696	0.3724	0.3758	0.3867	0.3887
Jrip	0.001	0.0633	0.0517	0.0012	0.001	0.0009
ZeroR	0.3633	0.3696	0.3724	0.3758	0.3867	0.3887

Figure 1 shows the Average Root Mean Square Error generated by different Non ensemble based classifiers during classification.

Now all ensemble based methods are considered. Table 5 and 6 the classification accuracy obtained using Random Forest and Weighted Random Forest for different

sample sizes using training data and test data respectively. Table 7 gives the root mean square error generated by each ensemble based method for different sample sizes during model development.

Table 5. Classification Accuracy using Ensemble based methods for training data

Ensemble based Classification Method	Sample Size					
	300	500	750	1000	1300	1500
Bagging	100	99.25	99.25	99.38	99.711	99.666
Multiclass Classifier	100	100	100	100	100	100
Random Forest	100	100	100	100	100	100
Weighted Random Forest	100	100	100	100	100	100

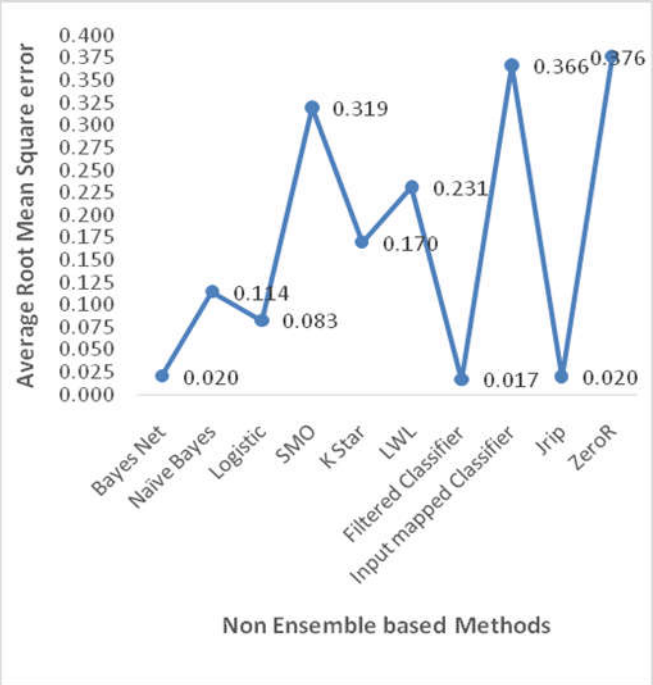


Figure 1.Average Root Mean Square Error generated by different Non ensemble based classifier

Table 6. Classification Accuracy using Ensemble based methods for test data

Ensemble based Classification Method	Sample Size					
	300	500	750	1000	1300	1500
Bagging	100	100	100	96.059	100	100
Multiclass Classifier	96.6102	99	97.33	98.03	99.618	99.34

Random Forest	100	100	100	100	100	100
Weighted Random Forest	100	100	100	100	100	100

Figure 2 given below shows the Average Root Mean Square Error generated by different Ensemble based classifiers during classification.

Table 7: Root mean square error generated during classification by ensemble methods

Ensemble based Classification Method	Sample Size					
	300	500	750	1000	1300	1500
Bagging	0.0326	0.02	0.509	0.1125	0.0034	0.0202
Multi Class Classifier	0.3533	0.3541	0.3543	0.3548	0.3531	0.3532
Random Forest	0.0197	0.0528	0.0403	0.0291	0.0205	0.0202
Weighted Random Forest	0.028	0.0246	0.0076	0.003	0	0.0031

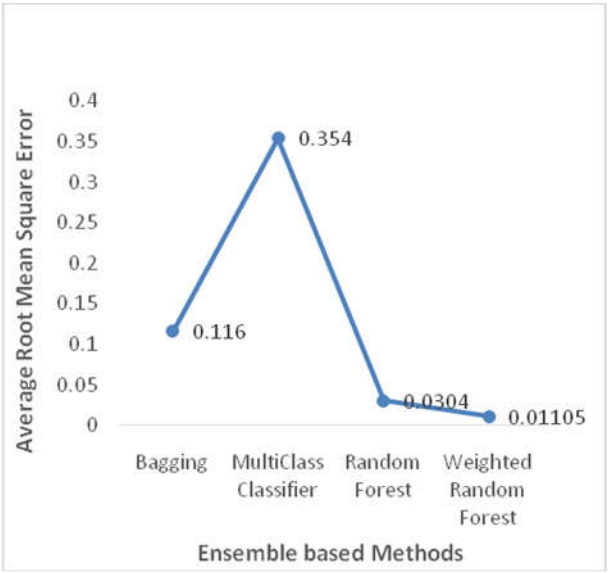


Figure 2. Average root mean Square error generated for different Ensemble based classifiers

From Figure 1, it is seen that Filtered classifier (Non Ensemble based Classifier) gives lowest average root mean square error (0.016883). From Figure 2, it can be concluded that Weighted Random Forest gives lower average root mean square error (0.0115) .

If both Filtered Classifier and Weighted Random forest are compared based on average root mean square error then it can be concluded that the performance of Weighted Random forest is better . So the performance of Ensemble based classifier is better as compared to non ensemble based classifiers.

5. Conclusion

In this paper study of different ensemble based and non ensemble based classifier is done. Classification Models are prepared using ensemble based and non ensemble based classifier . For comparative analysis, Root mean square error and average root mean square error generated during model development are considered. After comparative analysis, it can be concluded that the performance of Filtered classifier is best, if Non Ensemble based classification methods are considered and Weighted Random forst is best in case of ensemble based classification methods. Overall if Ensemble based and Non – Ensemble based methods are considered altogether than Ensemble based methods outshines in making better classification.

REFRENCES

[1] *SeriiKhlamovet al., "Colitec Software for Astronomical Data Sets Processing", Proceeding of IEEE second International Conference on Data Stream mining and processing(2018) Aug 21-25, 227-230.*

[2] *TheeranaiSangjan et al.."Classification of Astronomical objects Using Light Curve Profile."Proceeding of IEEE Eurasia Conference on IOT, Communication and Engineering(2019).*

[3] *M. Klush, "HNS- a hybrid neural system and its use for classification of stars", Proceeding of International Conference on Neural Network (1993).*

[4] *Shiyu Deny, Liangping Tu. "Classification of Celestial spectral based on improved desnsity clustering", Proceeding of 10th International Conference on Image and Signal Processing Bio Medical Engineering and Informatics. (2017).*

[5] *LiangpingTu, Huiming Wei and Liya Ai. "Galaxy and Quasar classification based on local mean- based K-Nearest Neighbor method", Proceeding of IEEE 5th International Conference on Electronics Information and Emerging Communication. (2015).*

[6] *Zhenping YI, Jingchang PAN. " Application of Random Forest to stellar spectral classification", Proceeding of 3rd International Congress on Image and Signal Processing , Volume 7. (2010).*

[7] *Jiang Bin, Wang Wenyu, Ma Chunyu, Wang Wei, Qu Meixia. "The Application of automative classification of massive SDSS spectra " , Proceeding of 2nd IEEE International Conference on Computer and Communication (2016) (1376-1380).*

[8] *D.G. York, et al., and SDSS Collaboration. The Sloan Digital Sky Survey: Technical Summary. AJ, 2000. 120:1579-1587.*

[9] *M.Hall, E. Frank, G. Homes, B. Pfahringer, P. Reutemann and I.H. Witten.2009. The weka data mining software: An update. SIGKDD Explorations, 11(1):10-18.*

[10] *Introduction to Bayes Net? Retrieved 10 20, 2018,from Tutorial on Bayes Network with Netica: http://www.norsys.com/Sec_A/tutA1.htm(1995).*

[11] *Naive Bayes. Retrieved 9 21, 2018, from Scikit learn developers: http://scikit-learn.org/stable/module/naive_bayes.html (2007).*

[12] *https://machinelearningmastery.com/use-classification-machine-learning-algorithms-weka/*

[13] *https://machinelearningmastery.com/use-classification-machine-learning-algorithms-weka/*

[14] *K-Star. Retrieved 11 10, 2017, from KStarpentahoData Mining-pentaho Wiki: http://www.wiki.pentaho.com/display/DATAMINING/KStar (2008, 12 5).*

[15] *https://wiki.pentaho.com/display/DATAMINING/LWL.*

[16] *Eibe Frank, Mark Hall, Bernhard Pfahringer. 2003. Locally Weighted Naive Bayes. In Proceeding of 19th Conference in Uncertainty in Artificial Intelligence(249-256), 2003.*

[17] *Bagging. Retrieved 10 23, 2018, from Bagging and random Forest Ensemblle Algorithms for Machine Learning: https://machinelearningmastery.com/bagging-and-randon-forest-ensemble-algoritms (2016, 4 22).*

[18] *Multiclass classifier. Retrieved 11 6, 2017, from Multiclass classification using scikit-learn-GeeksforGeeks: https:www.geeksforgeeks.org/multiclass-classification-using-scikit-learn/ , (2010).*

[19] *https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781787122536/4/ch04lvl1sec46/classifying-unseen-test-data-with-a-filtered-classifier*

[20] *https://nodepit.com/node/org.knime.ext.weka37.classifier.WekaClassifierNodeFactory%23InputMappedClassifier%20(3.7)*

[21] *https://nodepit.com/node/org.knime.ext.weka37.classifier.WekaClassfierNodeFactory%23JRip%20(3.7)*

[22] *https://www.saedsayad.com/zeror.htm*

[23] *Leo Breiman. "Random Forests, Machine Learning", (2001), 45,(5-32).*

[24] *Liaw, A, Wiener, M. "Classification and regression by randomForest, R News", 2:18-22(2002).*

[25] *S. Gedam, R. Ingolikar, "Decision Support System Using Weighted Random Forest For Astronomical Data", IOSR Journal of Computer Engineering , Volume 20, Issue 4, Ver. I (2018). Pp 40-44.*

"Sentiment Analysis : Datamation in Python and Weka"

Ms. Puja M. Dadhe[1], Dr. R.N. Jugele[2] and Mr. D.S. Sadhankar[3]
[1] Research Scholar, Department of Computer Science,
Shivaji Science College, Nagpur.
[2]Assosiate Professor, Department of Computer Science,
Shivaji Science College, Nagpur.
[3]Assistant Professor, Department of Computer Science,
SFS College, Nagpur.
poojadadhe@gmail.com[1], rn_jugele@yahoo.com[2],dileep.sadhankar@gmail.com[3]

Abstract: *Data analysis is a process of investigating and analysing data in order to derive some useful information. With the increase in use of internet lot of data is generated every day. About 80 percent of this data is unstructured, which needs a proper measure to be analysed. The objective of such analysis has a wide scope in discovering interesting information in the areas like business, politics, research, science and social science domains. Sentiment analysis plays a very important role in decision making process. It classifies a document, sentences and aspect in to positive and negative sentiments. It employs Supervised and unsupervised approaches to find polarity. The paper presents a comparison between Weka and Python as a tool for sentiment analysis on different datasets and compares the accuracy for each dataset.*

Keywords: **Sentiment analysis, Datasets, Weka, Python, Movie Reviews, Accuracy, Sentiment polarity, NLP, Naïve Bayes, Negative, Positive.**

1. INTRODUCTION

Sentiment Analysis is a branch of Natural language processing which deals with the problem of classification and a method of computing and satisfying a view of a person given in a piece of a text, to identify persons thinking about any topic is positive negative or neutral [4]. It is done on the data that is collected from the Internet and various social media platforms. Organizations, Companies and Governments often use sentiment analysis to understand how the people feel about themselves, products and their policies. The purpose of Sentiment analysis is to classify the polarity of user's sentiment and extract his opinion regarding an interested entity, which help in providing valuable information for decision making [3]. Polarity in sentiment analysis means classifying the sentiments as positive, negative and neutral. Further it is classified into different levels:

- **Document level**: This classifies the whole document text into positive or negative polarity.
- **Sentence level**: This extracts the polarity of each sentence of a document into positive or negative polarity.
- **Aspect/entity level**: This classifies the sentiment polarity of each entity's aspect or feature of a sentence/document [3].

With social media becoming the main platform for expressing feelings, views of a person now days, it is also gaining a lot of exposure for finding credible information. Social media refers to websites and application that are designed to allow people to share content quickly, efficiently and in real time[6]. Twitter and Facebook are most common social media

platforms. In a scenario where analytics uses various methods for data analysis, The primary goal of data analytics is to help companies make more informed business decisions. It is performed by enabling data scientists, predictive modellers and other analytics professionals to analyze large volumes of transaction data. The other forms of data is untapped by conventional Business Intelligence (BI) programs which include Web server logs, Internet clickstream data, social media content, social network activity reports, text from customer emails, survey responses, mobile-phone call detail records and machine data captured by sensors connected to the Internet [6]. Social data plays an important role in online trading, as it depends on the stakeholder‘s interest. The comments or the opinion from the stakeholders play multiple roles. Sentiment analysis or opinion mining has been exploited to process this information. The process of discovering the subjective information using natural language processing, text analysis, and computational linguistics from the social data is known as sentiment analysis [1]. The Data sets used in this paper consists of views of customers, viewers and data collected from social media.

There are two types of techniques used in Sentiment Analysis:

- Machine learning based techniques: Here various machine learning algorithms like Naïve Bayes, Maximum Entropy, SVM, K-means etc are used for classification of sentiments. It plays an important role in designing a tool. Various supervised and unsupervised machine learning algorithms can be used for finding the sentiment analysis [5].
- Lexicon Based technique: In Lexicon, a sentiment dictionary is used with sentiment words for classification of sentiments.

In this paper, the technique Naïve Bayes a has been evaluated for finding accuracy, precision and recall. Naïve Bayes is taken because Naive Bayes is a high bias, low variance classifier, also it can build good model with small data set.

Lots of free and open source tools are available online like NLTK, Weka, Python, Rapid miner, GATE, Open NLP etc. In this paper two tools, Weka and Python has been used to analyse the sentiments collected from different datasets. Weka is a open source software which encompasses data analysis, data integration and reporting in a single suit. It is very easy to use software with lots of features like cross validation, performance vector, split validation. Weka is easy to use with friendly interface. The reason for choosing this as a tool for Sentiment Analysis is due to its GUI and ready to use properties. Python is general purpose and high level programming language use for developing desktop GUI applications. This paper analyses Accuracy of four Datasets employing Naïve Bayes in Weka and Python.

2. DATA SOURCE AND DATA SETS

Following are the data sets used for performing Sentiment Analysis in Weka and Python.

- Dataset1- The first data set used is movie data known as sentiment polarity dataset downloaded from http://www.cs.cornell.edu/people/pabo/movie-review-data. This

dataset contains two data files, pos and neg. each file contains 5000 positive and negative statements respectively. Sentence level sentiment analysis is done on it.

- Dataset2- Second dataset is Next Data set is also movie review data set which contains 1000 positive and 1000 negative text files downloaded from www.Kaggle.com
- Dataset3- Third data set used is imdb-sentiment-2011. This is large dataset consisting of 25000 positive and 25000 negative movie reviews. Downloaded from ai.standford.edu. This dataset is in .arff form. For python it is converted into .csv format.
- Dataset4- Fourth data set used is tf.data.dataset by tensoreflow. It consists of 900 postive and 900 negative files each.

3. METHADOLOGY

The main aim of this research is to analyze the accuracy of sentiments polarity using Naive Bayes technique also comparison between these techniques in Weka and Python to find out the best performing one. Diagrammatical representations of process involve in sentiment analysis in both Weka and Python is given below.

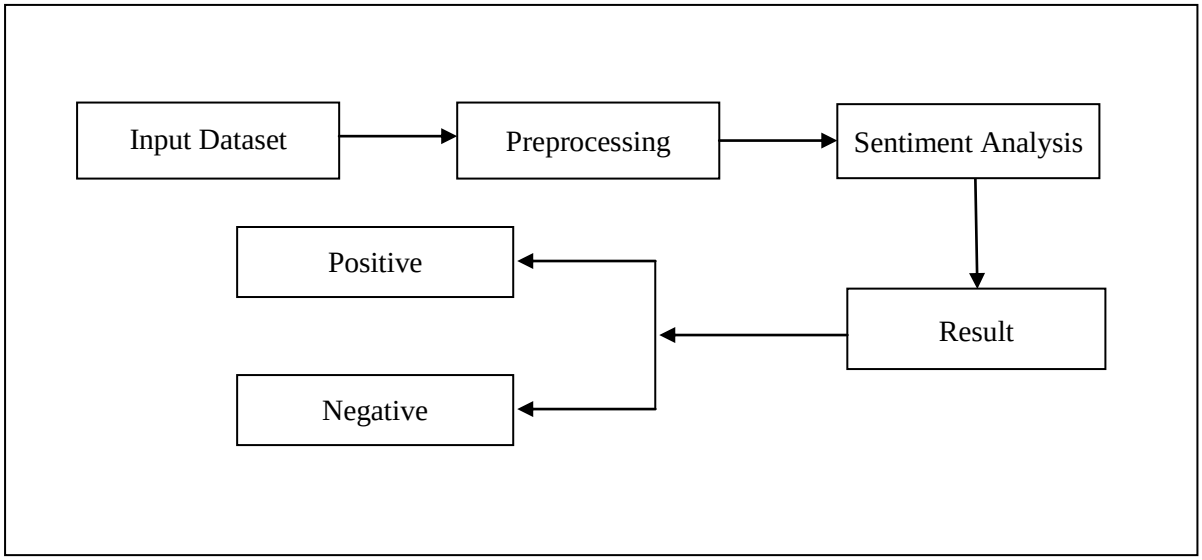


Fig 1: Sentiment Analysis Process

A preliminary Pre-Processing phase and attribute selection is essential for the sentiment classification task to be done which involves the following

- **Text preprocessing and feature extraction:** For the classification task to be done, a preliminary phase of text preprocessing and feature extraction is essential. To build the vocabulary, various operations are typically performed [2].

- **Word parsing and tokenization**: In this phase, each document is analyzed with the purpose of extracting the terms. Separator characters must be defined, along with a

tokenization strategy for particular cases such as accented words, hyphenated words, acronyms, etc [2].

- **Stop-words removal**: A very common technique is the elimination of frequent usage words: conjunctions, prepositions, base verbs, etc [2].
- **Lemmatization:** The lemmatization of a word is the process of determining its lemma. The lemma can be thought of as the "common root" of various related inflectional forms for instance, the words walk, walking and walked all derive from the lemma walk [2].
- **Stemming:** A simple technique for approximated lemmatization is the stemming. It works by removing the suffix of the word, according to some grammatical rules [2].
- **Term selection/feature extraction:** The term set resulting from the previous phases has still to be filtered, since we need to remove the terms that have poor prediction ability (w.r.t the document class) or are strongly correlated to other terms [2].

Paper focus on the **Naïve Bayes**, machine learning technique for both the tools.

Naïve Bayes: It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. This model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods. It is a simple probabilistic classifier based on Bayes' theorem which can build a good model even with a small data set. It is simple to use, computationally inexpensive and is very useful for the case where dimensions of input are high and for a given class as positive or negative, the words are conditionally independent of each other [7].

Naïve Bayes classifier [5] is an approach in which a classification of text (specific attribute) on the bases of appearance or absence of a class c in a given document d.

$$P(c|d) = \frac{\mathrm{P(c)P(d|c)}}{\mathrm{P(d)}}$$

Where c belongs to the positive or negative class and d belongs to the document whose class is being predicted, also P(c) and P(d|c) obtained during training.

To calculate accuracy of datasets following are some key terminology used which includes

Confusion matrix – It is also known as error matrix, is required to compute the ***accuracy*** of the machine learning algorithm in classifying the data into its corresponding labels. Confusion matrix C is a square matrix where Cij represents the number of data instances

which are known to be in group i(true label) and predicted to be in group j(predicted label) [8].

If we consider a binary classification problem,

Coo represents the count of true negative

Coi represents the count of false positive

Cio represents the count of false negative

Cu represents the count of true positive.

Represented as

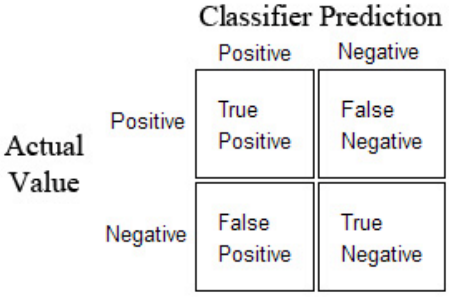


Fig 2: Confusion Matrix

Accuracy represents the number of correctly classified data instances over the total number of data instances calculated as-

$$\text{Accuracy} = \frac{TN+TP}{TN+FP+TP+FN}$$

To get perfect Accuracy following parameters are considered-

Precision refered as positive predictive value calculated as

$$\text{Precision} = \frac{TP}{TP+FP}$$

Precision should ideally be 1 (high) for a good classifier. Precision becomes 1 only when the numerator and denominator are equal i.e $TP = TP + FP$, this also means FP is zero. As FP increases the value of denominator becomes greater than the numerator and *precision* value decreases [8].

Recall is also known as sensitivity or true positive rate and is defined as follows:

$$\text{Recall} = \frac{TP}{TP+FN}$$

Recall should ideally be 1 (high) for a good classifier. Recall becomes 1 only when the numerator and denominator are equal i.e $TP = TP + FN$, this also means FN is zero. As FN increases the value of denominator becomes greater than the numerator and recall value decreases [8].

So ideally in a good classifier, both precision and recall to be one which also means FP and FN are zero. Therefore we need a metric that takes into account both precision and recall. F1-score is a metric which takes into account both precision and recall and is defined as follows:

$$\text{F1 Score} = 2 * \frac{Precision * Recall}{Precision + Recall}$$

F1 Score becomes 1 only when precision and recall are both 1. F1 score becomes high only when both precision and recall are high. F1 score is the harmonic mean of precision and recall and is a better measure to find more perfect accuracy [8]. All these are evaluated on different dataset in weka and python and accuracy is been compared.

4. RESULTS

In this paper all four datasets were evaluated for accuracy in both the tools Weka and Python and following table shows the results procured by the tools.

Sr.no	Datasets	Positive/Negative files	Accuracy in weka	Accuracy in Python
1	Rt-polarity	5000	100%	77%
2	Txt-sentoken	1000	81.45%	78%
3	Imdb-sentiment-2011	25000	81.2%	82.%
4	Tf-data.dataset	900	81.33%	90.5%

Table 1: Results Showing the Accuracy values

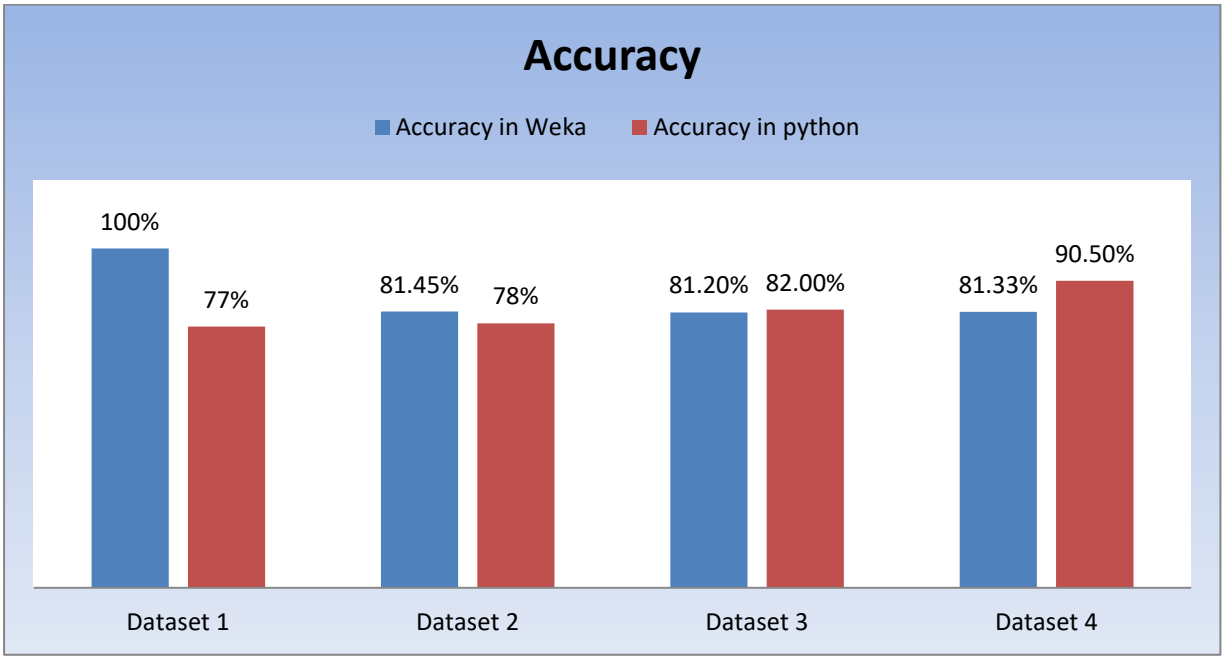


Chart 1: Comparison of Accuracy of Datasets in Weka and Python

5. CONCLUSION

With the Data set1 of 5000 text, Naïve Bayes in weka showed 100% accuracy whereas in Python it come out to be 77%, less compared to weka. For Dataset2 which consist of 1000 files again Weka Performed better than Python. But for Dataset3 and Dataset4 Python Accuracy is higher as compared to Weka. In Python processing is faster as compared to weka. Weka is little incompatible with large dataset. Time taken by Python to produce result is higher than Weka as observed while performing classification with both the tools. In this paper Naïve Bayes has been compared and observed for both the tools which show both the tools perform well for Sentiment Analysis.

In future these different data sets and methods can be taken to find out accuracy or comparison between different tools available. One can make use of rapid miner tool and R programming to find out how they work for sentiment Analysis. Also in this paper Naïve bayes has been used but the experiment can be also done with methods like SVM, Maximum Entropy etc. Instead of Datasets, Twitter data can be taken for classifying Sentiments.

6. REFRENCES

1. G. Ramajayam, 2Dr. V. Radhika, A Survey on Role Of Sentiment Analysis In Stakeholer's Statisfaction, International Journal of Pure and Applied Mathematics, Volume 119 No. 18 2018, 3021-3027.
2. Jincymol joseph, J R jeba, Information Extraction Using Tokenisation And Clustering Methods, International Journal of Recent Technology and Engineering(IJRTE), Volume-8 Issue-4,November 2019.
3. Ms. Puja M. Dadhe, Dr. R.N. Jugele, Inspection of Retrospection : Challenges of Sentiment Analysis, Compliance Engineering Journal,Volume 11,Issue 1,2020.
4. Rupinder Kaur et al, A Review on Sentimental Analysis on Facebook Comments by using Data Mining Technique International Journal of Computer Science and Mobile Computing, Vol.8 Issue.8, August- 2019, pg. 17-21.
5. Shilpa Singh Hanswal, Astha Pareek, Twitter Sentiment Analysis using Rapid Miner Tool, International Journal of Computer Applications (0975 – 8887) Volume 177 – No. 16, November 2019.
6. https://www.digitalvidya.com/blog/big-data-applications/
7. https://www.analyticsvidya.com/blog/2017/09/naive-bayes-explained/
8. https://www.geeksforgeeks.org/confusion-matrix-machine-learning/

Identifying Analytics of Sentiment Analysis on Twitter Data

Ms. Puja M. Dadhe[1], Dr. R.N. Jugele[2] and Dr. D.S. Sadhankar[3]
[1] Research Scholar, Department of Computer Science,
Shivaji Science College, Nagpur.
[2]Assosiate Professor, Department of Computer Science,
Shivaji Science College, Nagpur.
[3]Assistant Professor, Department of Computer Science,
SFS College, Nagpur.

Abstract: *Social Media becoming the most reliable platform for people to express their views, Sentiment Analysis has gained major popularity in research Area. Twitter is the most popular social media, people tend to put up their views very often. Corona Pandemic has raised various issues in human life, that is why social media has become a most convenient way to spread a word about opinions. This paper analyses corona tweets for Sentiments using Textblob and Vader libraries of Python. This work is based on real time Sentence level Sentiment Analysis (SLSA) which predicts the sentiment of Sentences mentioned in the Tweets.*

Keywords: Corona, Sentiments, Lexicon Approach, Textblob, Vader, Tweets.

1. INTRODUCTION:

Sentiment analysis is a problem in natural language processing (NLP) that involves machine learning to train a system to interpret human opinion derived from various sources. Sentiment analysis, often known as opinion mining, is a text analytic technique that studies people's feelings, emotions, sentiments and attitudes about various items or entities using written language[1]. Sentiment analysis is a technique for analyzing the opinions of individuals or groups, such as a subset of a brand's followers or a single consumer in contact with a customer service agent. Sentiment analysis is a technique for determining whether or not articulation is positive, negative or neutral and to what extent[2]. Sentiment Analysis involves mechanisms to categories text in to positive, negative and neutral polarity. levels of sentiment analysis are

- **Document level**: This classifies the whole document text into positive or negative polarity.
- **Sentence level**: This extracts the polarity of each sentence of a document into positive or negative polarity.
- **Aspect/entity level**: This classifies the sentiment polarity of each entity's aspect or feature of a sentence/document[3].

Social networking connects people. Multiple social networking sites are available for users to post their views. Twitter, one of the most popular social networking sites, has recently garnered investigation due to its rapid expansion. Twitter began as an online microblogging service in March 2006, allowing users to create status messages known as tweets.

A user can also follow other users and watch their tweets, as well as retweet them to their followers. Twitter's user-generated content covers a wide range of themes, including products, events, people, and current events. It can be beneficial in the decision-making process of businesses and other communities[4].

To answer the challenge of sentiment analysis, Twitter is widely used and tweets are readily available. These tweets can be fed into a variety of sentiment analysis algorithms and labelled as positive, negative or neutral. Because of the following factors, Twitter messages are a valuable source for sentiment analysis:

1. Tweets are 140-character long and have a more abstract tone.

2. Real-time analysis on tweets is possible.

3. Access to a large quantity of tweets for analysis.

The goal of sentiment analysis is to find and extract opinions from user-generated content. From review sites to microblogs, there has been progress in the field of sentiment analysis. It can be beneficial in the decision-making process of businesses and other communities.

Two basic techniques for sentiment analysis

Supervised : Machine Learning (ML) based sentiment analysis. Supervised learning is the process of inferring a function from labelled training data. The training data consist of a set of training examples. Supervised learning is a widely used solution for classification purpose and is been used in most of the sentiment classification techniques. Techniques for sentiment classification include SVM, Neural Network and Decision tree Classifiers. Other commonly used algorithms include K-Nearest Neighbour, Bayesian Network.

Unsupervised : Rule-based sentiment analysis, uses a dictionary of words labelled by sentiment to determine the sentiment of a sentence. Unsupervised learning is used to find inferences from datasets consisting of input data without labelled responses. In most of the classification techniques, especially text data it is very difficult to create training labelled data and it requires much of the human effort. Making use of unsupervised techniques can help overcome the disadvantage.

In this paper Lexicon based techniques are employed to determine the polarity of 5000 corona tweets.

2. METHODOLOGY :

The two main areas of research in sentiment classification are lexical and machine learning approaches For the lexical approach, a dictionary is prepared to store the polarity values of lexicons. For calculating polarity of a text, polarity score of each word of the text, if present in the dictionary, is added to get an ‘overall polarity score’. For example, if a lexicon matches a word marked as positive in the dictionary, then the total polarity score of the text is increased. If the overall polarity score of a text is positive, then that text is classified as positive, otherwise it is classified as negative. Though this approach seems very basic, variants of this lexical approach have been reported to have considerably high accuracy. Since the polarity of the text depends on the score given to each lexicon, there has been a large volume of work dedicated to discovering which lexical information is most efficient[5].

This paper aims to analyze the performance of Textblob and Vader of Python for polarity detection (positive, negative and neutral) of textual data. 5000 corona tweets are fetched using rest Api of Twitter and reports the implementation of the Twitter sentiment analysis. Tweepy is a library of Twitter API for fetching the tweets directly from Twitter that are tweeted by the different Twitter user. The real time

tweets fetched are then saved into CSV files for performing sentiment analysis. There are great works and tools focusing on text mining on social networks. In this study, the wealth of available libraries has been used. The approach to extract sentiment from tweets is as follows:

1. Create Twitter Account and Import Tweepy for creating the connection with Twitter API.
2. Tweets related to corona are fetched and are then saved in CSV file.
3. Pre-processing of tweets is done for removing the stop words, punctuations, #tags, etc.
4. Processed dataset is saved.
5. Textblob and Vader libraries are used to classify the tweets as positive, negative and neutral.

Textblob:

TextBlob is a Python 2 and Python 3 library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation and more.

Features of Textblob includes Noun phrase extraction,Part-of-speech tagging,Sentiment analysis,Classification (Naive Bayes, Decision Tree)Tokenization, (splitting text into words and sentences), Word and phrase frequencies, Parsing,n-grams,Word inflection (pluralization and singularization) and lemmatization, Spelling correction etc[6].

VADER (Valence Aware Dictionary and sEntiment Reasoner)

VADER is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media. VADER uses a combination of A sentiment lexicon is a list of lexical features (e.g., words) which are generally labeled according to their semantic orientation as either positive or negative. VADER not only tells about the Positivity and Negativity score but also tells us about how positive or negative a sentiment is[7].

Following code snippet is for connecting to twitter by importing Tweepy and the keys required for secure connection.

```
In [1]: import tweepy
        import pandas as pd
        import numpy as np
        import re

        import matplotlib.pyplot as plt

        from nltk.tokenize import TweetTokenizer

        from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
        from textblob import TextBlob
```

```
In [2]: consumer_key = '[redacted]
        consumer_secret = [redacted]
        access_token = [redacted]
        access_token_secret = '[redacted]
```

Fig 1: Libraries imported and Secret Keys for Twitter API

Once the connection is successful, 5000 tweets are fetched and saved in corona.csv file.

```
In [8]: tweets_df = tweet_search(key_word)
        tweets_df

        Tweets downloaded: 4999 / 5000253 / 5000 496 / 50001277 / 50005000 1872 / 5000 2428 / 5000 3217 / 5000 3506 / 5000 3888 / 5000
        4308 / 5000 4392 / 5000 / 5000
```

Out[8]:

Datetime	Tweet	Username	Retweets	Followers	CleanTweet
2022-02-23 11:12:15	@JoeSmithSDK @2NjoyMore @MattH_4America also n...	gregorylent	0	4285	also, note, five-eyes, is, having, the, most, ...
2022-02-23 11:12:14	RT @VarunKrRana: Corona report of Nawab Malik ...	KapilSGalav	65	823	corona, repoof, nawab, malik, came, negative, ...
2022-02-23 11:11:19	@bent_shamo Corona Outlet 🤠	ClassicalQ8	0	704	corona, outlet, 🤠
2022-02-23 11:11:03	RT @EverythingEnch1: #Newpost 🍦 My journey fr...	DeliciouslySavv	18	33768	💪, □, my, journey, from, london, uk, to, india...
2022-02-23 11:10:55	RT @INC_akhter: To compensate for the loss in ...	kushal_gehlot	14	3009	to, compensate, for, the, loss, in, education,...
...	...	...	...	...	...
2022-02-21 21:59:32	RT @naturalphoton: Dear @UNGeneva - you should...	Amrita1224	16	532	dear, -, you, should, go, one, step, back, and...
2022-02-21 21:58:23	RT @LawrenceSellin: Stolen elections have cons...	GoneWit59335088	19	739	stolen, elections, have, consequences, the, us...

Fig 2: Tweets fetched.

Tweets fetched are then preprocessed and then Textblob and vader libraries are used to classify these tweets as positive negative and neutral.

```
In [9]: def vader_compound_score(tweet):
            vader = SentimentIntensityAnalyzer()
            if vader.polarity_scores(tweet)['compound'] >= 0.05:
                return 'Positive'
            elif vader.polarity_scores(tweet)['compound'] <= -0.05:
                return 'Negative'
            else:
                return 'Neutral'

        def textblob_sentiment(tweet):
            analysis = TextBlob(tweet)
            if analysis.sentiment.polarity > 0:
                return 'Positive'
            elif analysis.sentiment.polarity == 0:
                return 'Neutral'
            else:
                return  'Negative'
```

Fig 3: Python code for sentiment Analysis using Textblob and Vader.

3. RESULTS:

The result is visualized by different plot methods using matplotlib that is the most popular library in python to the visualization of a result like a bar chart, histrogram, pie chart etc.

The result of tweets for ‘coronavirus’ based on 5000 tweets from Twitter.

Textblob shows Positive tweets percentage 40.3% Negative tweets percentage 30.6% Neutral tweets percentage 29.1%.

Vader shows Positive tweets percentage 40.2 % Negative tweets percentage 21.1% Neutral tweets percentage 38.7%.

Following Table shows number of positive, negative and neutral tweets evaluated by Textblob and vader.

	No. of Positive tweets	No. of Negative tweets	No. of Neutral tweets
Textblob	2010	1055	1935
Vader	2015	1530	1455

Table 1: Result

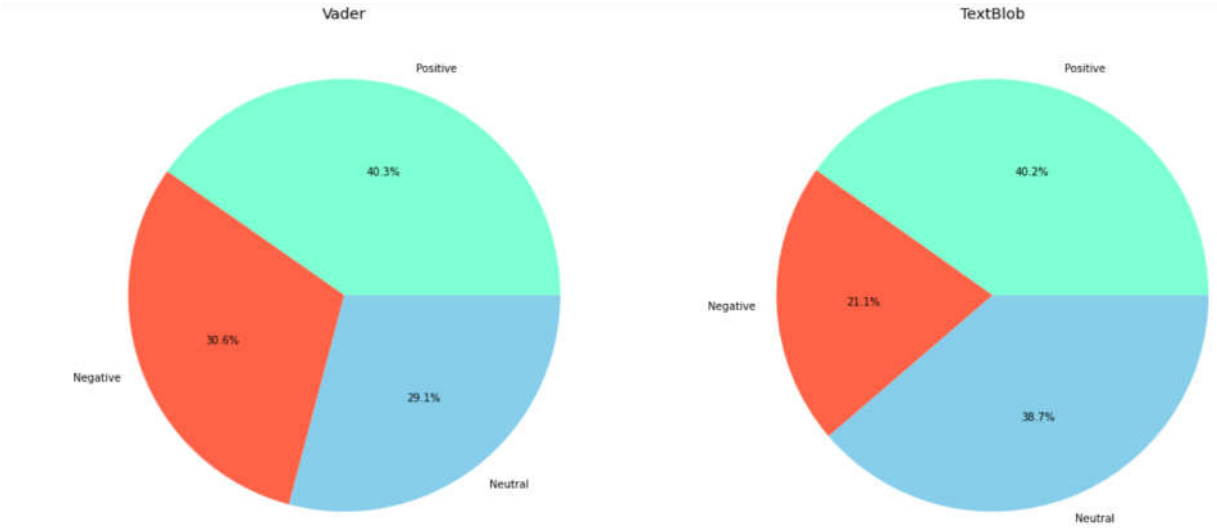


Fig 4: Percentage of Polarized Tweets

4. CONCLUSION:

With the Advancement in the field of Natural language processing algorithms, understanding and managing text based data is now easy. For the analysis of attitudes with data, the algorithm has a greater accuracy rate. Both Textblob and Vader have a plethora of features and functionalities — plotting the graph for the comparison revealed that Vader is primarily developed for social media platforms and can provide superior results as well as intensity when used with data from social media platforms such as Twitter. The rule-based methodology for sentiment analysis sometimes has a flaw in that it focuses just on individual words while completely ignoring the context in which they are used. The result indicated that VADER performs well than Text blob in sentiment analysis of tools.

5. REFERENCES:

1. Shubham V. Pandey, A. V. Deorankar, “A Study of Sentiment Analysis Task and It's Challenges”, 978-1-5386-8158-9/19/$31.00©2019IEEE.
2. Abdullah Alsaeedi1 , Mohammad Zubair Khan, “A Study on Sentiment Analysis Techniques of Twitter Data”, International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 10, No. 2, 2019.
3. Ms. Puja M. Dadhe, Dr. R.N. Jugele, “Inspection of Retrospection - Challenges in Sentiment Analysis”, Compliance Engineering Journal,Volume 11,Issue 1,2020.

4. K. Revathy, Dr. B. Sathiyabhama, “A Hybrid Approach for Supervised Twitter Sentiment Classification”, International Journal of Computer Science and Business Informatics, Vol. 7, No. 1. NOVEMBER 2013.
5. Chetashri Bhadanea ,Hardi Dalalb , Heenal Doshi, “Sentiment analysis: Measuring opinions” International Conference on Advanced Computing Technologies and Applications (ICACTA2015) volume 45, 2015, Pages 808-814.
6. https://textblob.readthedocs.io/en/dev/
7. https://www.geeksforgeeks.org/python-sentiment-analysis-using-vader/

Inspection of Retrospection - Challenges in Sentiment Analysis

Ms. Puja M. Dadhe[1] and Dr. R.N. Jugele[2]
[1] Research Scholar, Department of Computer Science,
Shivaji Science College, Nagpur.
[2]Assosiate Professor, Department of Computer Science,
Shivaji Science College, Nagpur.
poojadadhe@gmail.com[1], rn_jugele@yahoo.com[2]

Abstract: *With the advancement in social media and technology, social media had became a platform for millions of users' to share views and opinion related to products, issues and policies. Data generated through this media is an important source for analysis and opinion mining for extracting sentiments. Performing such analysis is not an easy task, it encounters multiple challenges which need to be addressed. This paper presents a retrospection of various challenges of sentiment analysis.*

Keywords: **Sentiment analysis, Social media, opinion mining, Sentiment polarity, NLP, summarization, Negative, Positive.**

1. INTRODUCTION

Sentiment Analysis [12] is a branch of Natural language Programming intended to mine various sources of data for opinions. It is defined as a computational study of human's thoughts or opinions, emotions and attitudes toward an object [2]. It is done on the data that is collected from the Internet and various social media platforms. Organizations, Companies, and Governments often use sentiment analysis to understand how the people feel about themselves, products and their policies. The purpose of Sentiment analysis is to classify the polarity of user's sentiment and extract his opinion regarding an interested entity, which help in providing valuable information for decision making [3]. Polarity in sentiment analysis means classifying the sentiments as positive, negative and neutral. Sentiment analysis has been classified into different levels:

- **Document level**: This classifies the whole document text into positive or negative polarity.
- **Sentence level**: Which extracts the polarity of each sentence of a document into positive or negative polarity.
- **Aspect/entity level**: Which classify the sentiment polarity of each entity's aspect or feature of a sentence/document [4].

There are two kinds of information in a particular sentence:
Objective- An objective sentence states factual information about the world.
Subjective- A subjective statement expresses some personal feeling, belief or view.
The task of determining whether a sentence is subjective or objective is called subjectivity classification.

The resulting subjective sentences are further classified as:
Expressing positive or negative opinions. This is called as sentence level sentiment classification [11].

Sentiment Analysis involves various aspects as shown in Figure 1. First phase involves data collection, various data from different sources like blogs, reviews and microbloging (Twitter, Facebook) which act as an input for Sentiment Analysis. Pre-Processing phase involves cleaning and Scraping of data. Feature Extraction identifies aspect which are being referred in particular sentence, document or comment by customer. Sentiment Analysis approaches are the techniques used for classification of Sentiment. Last phase is to generate results.

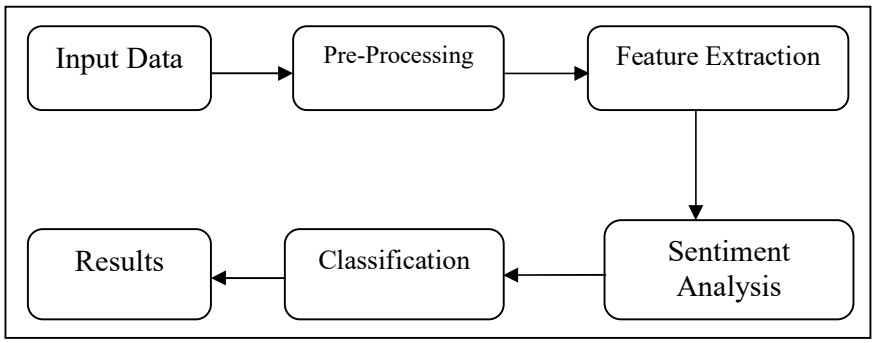


Figure 1- An Aspect of Sentiment Analysis.

Standard Structure of Sentimental Analysis:
Sentiment Analysis concludes whether users' view is positive, minus/neutral about a product, issue, event, etc. It is describe in three primary steps [1].

- **Data Retrieval** - It is the procedure of collecting review text from review sites. Different review websites contain reviews for products, movies, hotels and news. Also include Information retrieval -Techniques such as web crawler can be employed to collect the review text data from many sources and store them in a database. This step involves retrieval of reviews, micro-blogs and comments by user [1].

- **Sentiment Classification** - Primary steps in sentiment analysis are a classification of review text. Given a review document M = {M1….. M1} and a predefined category set K = {positive, negative}, sentiment classification is to classify each day in M, with a label expressed in K. The approach involves classifying review text into two forms namely positive and negative. Machine learning and dictionary based approach is more popular [1].

- **Sentiment Summarization** - Summarization of Sentiment is a major character in the Sentiment Analysis process. Summary of reviews should be based on features or subtopics that are mentioned in the reviews. Many works have been done on summarization of product reviews [1].

2. Techniques of Sentiment Analysis

The techniques of Sentiment classification [8] is divided into:

- Lexicon based approach.
- Machine learning approach.
- Hybrid approach.

2.1 Supervised Machine Learning:

Classification is most frequently used popular data mining technique. It is used to predict the possible outcome from given data set on the basis of defined set of attributes and a given predictive attributes. The given dataset is found to be the training dataset consist on independent variables (dataset related properties) and a dependent attribute (predicted attribute). A training dataset created model test on test corpus contains the same attributes but no predicted attribute. Accuracy of model checks that how accurate it is to make prediction. Product features and sentenced words are extracted using Double Propagation Algorithm [8].

2.2 Unsupervised Learning

In contrast of supervised learning, unsupervised learning has no explicit targeted output associated with input. Class label for any instance is unknown so unsupervised learning is about to learn by observation. Clustering is technique used in unsupervised learning. The process of gathering objects of similar characteristics into a group is called clustering. Objects in one cluster are dissimilar to the objects in other clusters [8].

2.3 Case Based Reasoning

Case based reasoning is an emerging Artificial Intelligence supervised technique. It is a powerful tool of computer reasoning and solve the problems (cases) which is closest to real time scenario. It is a problem solving technique in which knowledge is personified as past cases in library and it does not depend on classical rules. The solutions are stored in CBR repository called Knowledge base or Case base [8].

3. Literature Review

In 2015, P.Kalarani, Dr.S. Selva Brunda, examined various Sentiment Analysis challenges. They discussed about the Challenges and application area in opinion mining and the techniques and tools used for opinion mining. Challenges discussed by them included detection of spam, fake reviews, limitation of classification filtering, asymmetry in availability of opinion mining software, incorporation of opinion with implicit and behaviour data, domain-independence and natural language processing overheads[8].

In 2016, Mohey El-Din, Doaa, discussed the importance and effects of sentiment analysis challenges in sentiment evaluation based on two comparisons among forty-seven papers they reviewed. They recognised the sentiment challenge of domain-dependence. They also suggested that the negation challenge was the popular in all types of review structure. They found the relationship between the

proportion of sentiment techniques usage in theoretical and technical types to solve sentiment challenges [6].

In 2017, Osamah A.M Ghaleb,Anna Saro Vijendran, out looked concept of Sentiment Analysis and presented multiple challenges which includes
1. Big Data-related Issues like data collection, data pre-processing and storage.
2. Language-oriented Issues - lack of lexicon, different writing style and different word meaning.
3 Fake opinions- It includes Fake Positive or Fake Negative opinions.
4. Text related issues [7].

In 2018, Syed Saood Zia , Sana Fatima , IdrisMala , M. Sadiq Ali Khan , M. Naseem , Bhagwan Das, published the biggest challenge faced in sentiment analysis is the domain specific nature of opinionated words. They may perform well in one domain but works poor in other domain. Since every human being has a different nature so it is very hard to correctly classify users provided input belongs to a particular entity [10].

In 2019 S. V. Pandey and A. V. Deorankar, covered different levels of sentiment analysis and discussed aspect-based sentiment analysis. They proposed important challenges to this research area like named entity recognition, sentiment polarity detection, subjectivity detection etc. described with suitable example. They used Stanford Core NPL tools to visualize the result of some basic operation of NLP which can be used for sentiment analysis [9].

4. Sentiment Analysis Challenges

Sentiment Analysis seemed to be just classification problem but more we dive in it more challenging it appears. Challenges can be categorized as linguistic challenges and accuracy based challenges.
4.1 **Linguistic Challenges**- These challenges are related to the language (basically English) used by the author [13]
Feature recognition- This challenge is more related to lack of context in the piece of data being analyzed. For example the statement “it was beautiful” lacks the subject in discussion.
Sarcasm- Sarcastic statements are those in which people express negative thoughts using positive words. This situation occurs usually on social media where one put up opinion in much sarcastic ways which can easily confuse sentiment analysis model.
Contradiction- people can be contradictory in the way they review any product or give their opinion.
Slangs- People may use words like “OMG” ,“TC” or “LOL” to express their response. Identifying these words needs extra efforts to train our network to correctly identify the sentiment.
Jokes- Human brain can easily understand jokes but it is harder for a computer to parse.
Irony- Similar to sarcasm irony statement also causes misinterpretation of data and poses problems to natural language processing.

Emoticons- Massive use of emoticons (emotion icon, emoji) that is pictorial representation of human expression using graphical icon used to express mood or feelings are hard to interpret.

4.2 **Accuracy Based** [6] – This involves challenges which affects to the accuracy of models used to classify sentiments. These are sub-divided into two types of challenges theoretical and technical challenges.

Huge lexicon- This is theoretical challenge, require large amount of lexicon resources. With the advent new words every day it has became hard to handle huge lexicon.

Bi-polar- This is technical challenge, words can be positive and negative at the same time with regards to the context been used.

NLP overhead(emotions)- This is technical challenge, The natural language overhead like ambiguity, co-reference, Implicitness, inference etc. created hindrance in sentiment analysis too[8].

Domain Dependence- This is theoretical challenge, which requires networks to be trained on the specific domain they are analyzed for.

Spam and Fake- This is theoretical challenge, duplicate and similar reviews are fake reviews. This results in wrong and invalid analysis of data for decision making.

Negation- This is theoretical challenge, Most traditional text processing relies on the fact that small differences between two pieces of text don't change the meaning very much. In Sentiment analysis, however, "the picture was great" is very different from "the picture was not great"[5].

5. Conclusion

Sentiment Analysis field has been proven to be most promising research field in data analytics. Providing ranges of solutions for decision making and become popular platform in mining domain. Sentiment Analysis job involves number of challenges which need to be properly addressed for to perform accurately. This paper retrospect's the concept and challenges related to Sentiment Analysis. More Research work in foreign languages other than English is essential. Many challenges related to slang, inferences, ambiguity and contradiction needs more précised techniques to be overcome properly. These challenges provide new areas for further research.

REFERENCES

Journal Article

*[1] Abhishekh Kaushik,Anchal Kaushik,Sudhanshu Naithani, "A Study on Sentiment Analysis: Methods and Tools", International Journal of Science and Research, Volume 4 Issue 12,**2015.***

*[2] Chandni, Navchandra, "Sentiment Analysis and its Challenges", International Journal of Engineering Research and Technology, Vol 4,issue 0, March **2015**.*

*[3] Haseena Rahmath P, "Opinion Mining and Sentiment Analysis -Challenges and Applications", International Journal of Application or Innovation in Engineering & Management (IJAIEM), Volume 3, Issue 5, May **2014**.*

*[4] Doaa mohey el-din Mohamed Hussein,"A survey on sentiment analysis challenges", Journal of King Saud University - Engineering Sciences, volume 30,issue 4, October **2018**, pages 330-338.*

*[5] Dr. P. Sumathy, S. M. Muthukumari."Sentiment Analysis of Twitter Data Using Multi Class Semantic Approach",International Journal of Scientific Research in Computer Science, Engineering and Information Technology, Volume 3,Issue 6, **2018**.*

[6] Mohey El-Din, Doaa." A Survey on Sentiment Analysis Challenges", Journal of King Saud University - Engineering Science. 10.1016/j.jksues.2016.04.002.

*[7] Osamah A.M Ghaleb ,Anna Saro Vijendran, "The Challenges Of Sentiment Analysis On Social Web Communities", International Journal of Advance Research in Science and Engineering,Voulme No.06,Issue No. 12, **2017**.*

*[8] P.Kalarani, Dr.S. Selva Brunda," An Overview on Research Challenges in Opinion Mining and Sentiment Analysis", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 3, Issue 10, October **2015**.*

*[9] S. V. Pandey and A. V. Deorankar, "A Study of Sentiment Analysis Task and It's Challenges", 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), Coimbatore, India, **2019**.*

*[10] Sana Fatima, Syed Saood Zia , IdrisMala , M. Sadiq Ali Khan , M. Naseem , Bhagwan Das,"A Survey on Sentiment Analysis, Classification and Applications", International Journal of Pure and Applied Mathematics Volume 119 No. 10 **2018**.*

*[11] Sweta Morajkar , Maria Christina Barretto1, "Sentiment Analysis Tools and Techniques: A Comprehensive Survey", International Journal for Research in Applied Science & Engineering Technology (IJRASET),Volume 5 Issue XI November **2017**.*

Links

[12] https://en.wikipedia.org/wiki/Sentiment_analysis

[13] https://www.toptal.com/deep-learning/4-sentiment-analysis-accuracy-traps