Journal of Information Systems Engineering and Management
2025, 10(20s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Ensemble Classifier for Web Data Scraping with Lexicon Support

Yogesha T[1], Dr. Thimmaraju S N[2]
[1]*Assistant Professor, Department of CS&E,*
VTU-RRC, PG Centre Mysuru, Karnataka, India
yogesh@vtu.ac.in
Orcid Id:0000-0002-7662-6064
[2] *Professor. Department of CS&E,*
VTU-RRC, PG Centre Mysuru, Karnataka, India
Thimmaraju_sn@vtu.ac.in
Orcid Id:0000-0001-5090-8277

ARTICLE INFO

Received: 19 Dec 2024

Revised: 29 Jan 2025

Accepted: 10 Feb 2025

ABSTRACT

An ensemble classifier for web data is used for selective web scraping with lexical support in an innovative way to improving the accuracy and efficiency of data classification from web sources. Web scraping, is a method for obtaining information from webpages, frequently produces massive, unstructured datasets and high risk in data reliability which leads to misuse in communication that are difficult to manage. To overcome this, selective online scraping is used to target certain information important to the classification task, resulting in less noise and higher data quality. The ensemble classifier integrates numerous machine learning models to maximize their strengths, resulting in better overall performance. In this approach, separate classifiers are trained on distinct subsets of scraped data that are chosen based on predetermined criteria utilizing lexicons, which are collections of domain-specific words and phrases. These lexicons guide the selective scraping process, ensuring that only the most relevant data is captured, hence improving classifier accuracy. After scraping and pre-processing the data, the ensemble method aggregates predictions from each classifier, generally using techniques like majority voting, stacking, or weighted average, to get a final classification result. This strategy not only promotes robustness by reducing the risk of overfitting, but it also improves flexibility across other domains by incorporating lexical assistance tailored to specific themes or sectors. The combination of selective web scraping and lexical assistance enables more targeted and resource-efficient data collecting, while the use of an ensemble classifier assures excellent accuracy and reliability in classification tasks. This methodology is especially useful in circumstances where the online data is large, dynamic, and contains a lot of unnecessary or noisy information. The resulting system provides a scalable and effective solution for real-time web data classification, with applications in sentiment analysis, content categorization, and market intelligence.

Keywords: Classification, Ensemble classifiers, Lexicon for networks, Multimedia mining, Web content mining etc.

I. INTRODUCTION

The sheer amount and variety of internet material pose serious problems for conventional machine learning methods when it comes to web data classification. Ensemble classifiers have become a potent answer to these problems since they integrate the predictions of several models to improve accuracy and robustness. Ensemble approaches enhance the generalization of predictions and lower the chance of over-fitting by utilizing many datasets and algorithms.

By focusing on certain, pertinent web information, selective web scraping improves the process even more, reducing noise and improving data quality. Selective scraping considerably increases the efficacy and efficiency of the categorization process by extracting only the most relevant data using predetermined criteria as opposed to collecting vast amounts of data in an indiscriminate manner.

An additional level of complexity is added to this framework by incorporating lexical support. The process of selective scraping is guided by lexicons, which are curated lists of domain-specific keywords and phrases. This ensures that the data obtained is relevant and acceptable for the particular categorization task at hand. With the use of lexical support, selective web scraping, and ensemble classifiers, a reliable system that can correctly identify complex web data is produced. Applications where the quality and relevance of data are crucial, like sentiment analysis, content categorization, and information retrieval, benefit greatly from its utilization.

II. LITERATURE SURVEY

The Paper [1] presents, The Random Forests in Machine Learning because ensemble classifiers combine many models to increase predictive performance, they have attracted a lot of attention in the machine learning community. In paper [2] developed the idea of Random Forests in his seminal work, showing how combining the predictions of several decision trees can greatly improve accuracy and decrease overfitting. In paper [3,16] also examined a number of ensemble techniques, including Bagging, Boosting, and Stacking, and offered a thorough analysis of how well each worked with various kinds of datasets.

The paper [4, 16] "An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees, "Bagging, Boosting, and Randomization" analysis machine learning. This paper presents material focuses on comparing three ensemble approaches for decision trees through experimentation "Randomization, Boosting, and Bagging". It mainly looks into how well these techniques work for building accurate and varied classifiers. According to the study, Randomization works well in situations with little to no noise, but Bagging is more effective in those with a lot of classification noise. Compared to bagging, boosting is less reliable in noisy conditions even if it is usually accurate. The study evaluates these approaches using extensive experiments conducted on 33 learning tasks, demonstrating the advantages and disadvantages of each approach in different classification settings.

The paper [5, 6] Synthesis Lectures on Human Language Technologies elaborated text classification makes extensive use of lexicon-based techniques, particularly in areas like sentiment analysis and subject categorization[20]. An extensive examination of sentiment lexicons and their use in opinion mining was given in paper [7, 18]. The study demonstrates how adding domain-specific knowledge to lexicons can improve the effectiveness of classification systems. In their discussion of the creation of a lexicon-based sentiment analysis system, paper [6] demonstrated the value of combining machine learning models with pre-defined word lists.

The paper Sentiment Analysis Algorithms and Applications By combining statistical learning and human-curated knowledge, lexicon-based techniques and machine learning models can improve classification performance. Paper [9] examined a number of hybrid techniques that combine machine learning algorithms with lexicons to analyze sentiment. According to their research, hybrid systems of this kind can perform better than conventional techniques, especially in fields where the vocabulary is extensive and tailored to a given field.

The papers meta A Meta-Learning Framework for Spam Detection [10] dealt with Expert Systems with Applications and Combining Lexicon-Based and Learning-Based Methods for Twitter Sentiment Analysis [7, 11]. In paper [10] showed how combining several classifiers trained on various feature sets using ensemble approaches could improve classification accuracy in web spam detection. In their investigation into the application of ensemble approaches to sentiment classification. In web data classification tasks, such as spam detection and content categorization, ensemble approaches have been frequently used. In paper [11] demonstrated that combining models such as SVM and Random Forests enhances performance and resilience The paper Neural Network-Based Lexicon Generation for Sentiment Analysis [12]. There are special potential and challenges when combining lexicon-based techniques with ensemble learning for web data classification. The problems of data imbalance and noise, which are crucial when working with web data were covered. In order to create more intelligent and adaptive systems, [12, 13] have identified two future directions: the automatic creation of lexicons through neural network[13] technology and the integration of deep learning techniques in ensemble frameworks. In paper [23] explored the rainfall pattern and groundwater level of the Banaskantha district of Gujarat and predicted a rise in the groundwater level using Artificial Intellegent such as SARIMA, multi-variable regression, ridge regression, and KNN regression.

- **Ensemble Classifier for Web Data Using Selective Web Scraping with Lexicon support**

Creating a model for an ensemble classifier for web data using selective web scraping with lexicon support involves several stages, including data collection, preprocessing, model training, and evaluation. Below is a conceptual model outlining the key components and steps involved in this process:

1. Selective Web Scraping

- **Target Identification**: Define specific websites or web pages to scrape based on relevance to the domain.
- **Lexicon Development**: Create or obtain a lexicon (a list of keywords and phrases relevant to the target domain).
- **Data Extraction**: Use web scraping tools (e.g., BeautifulSoup, Scrapy) to selectively extract data from the identified web sources. Filter the extracted data using the lexicon to ensure relevance.
- **Data Cleaning**: Remove duplicates, irrelevant tags, advertisements, and other noise from the scraped data.

2. Data Preprocessing

- **Text Normalization**: Convert all text to lowercase, remove stopwords, punctuation, and perform stemming or lemmatization.
- **Feature Extraction**: Transform the cleaned text data into numerical features using techniques like TF-IDF (Term Frequency-Inverse Document Frequency), word embeddings (e.g., Word2Vec, GloVe), or bag-of-words.
- **Data Split**: Split the data into training and testing sets to evaluate model performance.

3. Ensemble Classifier Construction

- **Base Classifiers**: Choose a diverse set of base classifiers such as:
  - Decision Trees
  - Support Vector Machines (SVM)
  - Naive Bayes
  - K-Nearest Neighbors (KNN)
  - Logistic Regr**ession**
- **Model Training**: Train each base classifier independently on the preprocessed data.
- **Ensemble Techniques**: Combine the predictions of the base classifiers using ensemble methods [14,15, 21] such as:
  - **Bagging (Bootstrap Aggregating):** Use multiple instances of the same classifier trained on different subsets of the data.
  - **Boosting:** Focus on training classifiers sequentially where each new classifier corrects errors made by the previous ones (e.g., AdaBoost, Gradient Boosting).
  - **Stacking:** Train a meta-classifier on the predictions of the base classifiers.
  - **Voting:** Use majority voting, where the final prediction is based on the most common output among base classifiers (for classification tasks).

4. Lexicon Support Integration

- **Enhanced Feature Engineering**: Incorporate lexicon-based features into the feature set. This could include counting occurrences of lexicon words in the text or creating lexicon-specific features.
- **Feature Selection**: Evaluate and select the most relevant features, balancing between lexicon-derived and traditional machine learning features.

5. Model Evaluation

- **Performance Metrics**: Evaluate the ensemble model's performance using metrics such as accuracy, precision, recall, F1-score, and AUC-ROC for binary or multiclass classification tasks.
- **Cross-Validation**: Use k-fold cross-validation to ensure the model's robustness and generalizability across different data splits.

6. Model Deployment

- **Real-Time Scraping and Classification**: Implement a pipeline that continuously scrapes data, pre-processes it, and feeds it into the ensemble classifier for real-time predictions.
- **Feedback Loop**: Regularly update the lexicon and retrain the model to adapt to changing data patterns and emerging trends [19].

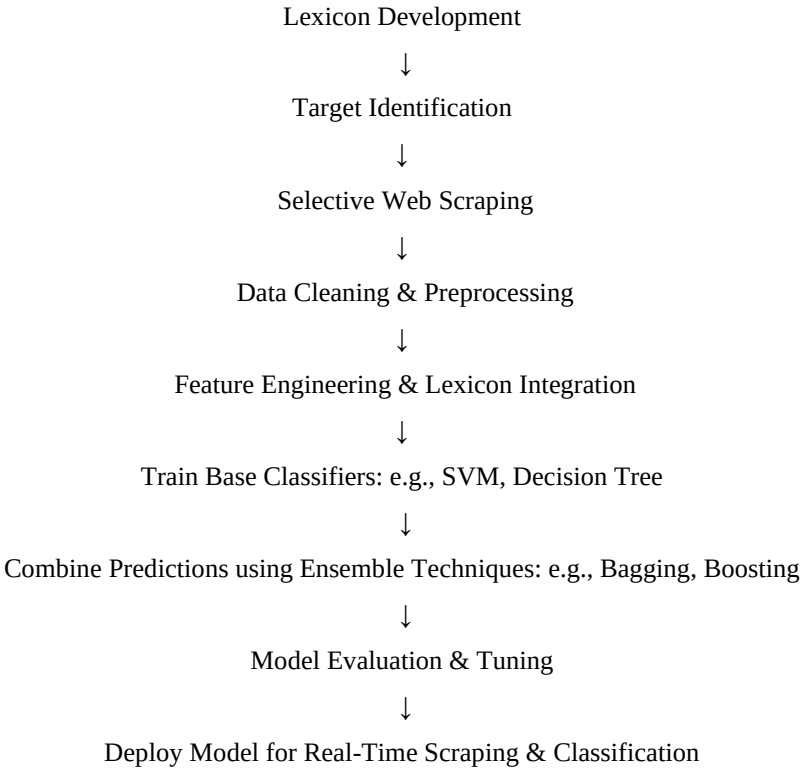

Figure 1: Ensemble model architecture

Figure1 shows Ensemble Classifier for Web Data Using Selective Web Scraping with Lexicon support This model combines ensemble learning, lexicon assistance, and selective online scraping to produce a reliable web data classification system. It makes use of the advantages of several classifiers as well as domain-specific information stored in lexicons to provide a versatile and effective method for managing massive amounts of online data.

III. EXPERIMENT, RESULTS AND DISCUSSION

The Table 1 shows the component model development phase with its description and Table 2 shows the Performance parameters without lexicon.

Table 1. Model development phases

SL No.	Component	Description
1	Target Identification	Define specific websites or web pages to scrape based on relevance to the domain.
2	Lexicon Development	Create or obtain a lexicon (a list of keywords and phrases relevant to the target domain).
3	Data Extraction	Use web scraping tools to selectively extract data from the identified web sources. Filter using lexicon.
4	Data Cleaning	Remove duplicates, irrelevant tags, advertisements, and other noise from the scraped data.
5	Text Normalization	Convert text to lowercase, remove stopwords, punctuation, and perform stemming or lemmatization.
6	Feature Extraction	Transform cleaned text data into numerical features using techniques like TF-IDF, word embeddings, or bag-of-words.
7	Data Split	Split the data into training and testing sets to evaluate model performance.
8	Base Classifiers	Choose a diverse set of base classifiers such as Decision Trees, SVM, Naive Bayes, etc.
9	Model Training	Train each base classifier independently on the pre-processed data.
10	Ensemble Techniques	Combine the predictions of base classifiers using methods like Bagging, Boosting, Stacking, or Voting.
11	Enhanced Feature Engineering	Incorporate lexicon-based features into the feature set for enhanced model accuracy.
12	Feature Selection	Evaluate and select the most relevant features, balancing between lexicon-derived and traditional features.
13	Performance Metrics	Evaluate model performance using metrics like accuracy, precision, recall, F1-score, and AUC-ROC.
14	Cross-Validation	Use k-fold cross-validation to ensure model robustness and generalizability across different data splits.
15	Real-Time Scraping and Classification	Implement a pipeline for continuous data scraping, preprocessing, and real-time predictions.
16	Feedback Loop	Regularly update the lexicon and retrain the model to adapt to changing data patterns.

Table 2. Performance parameters without lexicon

Model	Train-Test Split	Accuracy	Precision	Recall	F1-Score
Ensemble model	60-40	0.85	0.75	0.8	0.8
	70-30	0.9	0.75	0.8	0.8
	80-20	0.85	0.7	0.75	0.75
	90-10	0.85	0.8	0.85	0.85
Decision Tree	60-40	0.82	0.7	0.78	0.74
	70-30	0.87	0.72	0.8	0.76
	80-20	0.83	0.68	0.73	0.7
	90-10	0.84	0.75	0.8	0.77
k-Nearest Neighbors	60-40	0.78	0.65	0.7	0.67
	70-30	0.8	0.67	0.72	0.69

	80-20	0.77	0.62	0.68	0.65
	90-10	0.79	0.7	0.75	0.72
Regression	60-40	0.83	0.72	0.76	0.74
	70-30	0.88	0.73	0.78	0.75
	80-20	0.84	0.7	0.74	0.72
	90-10	0.86	0.77	0.82	0.79

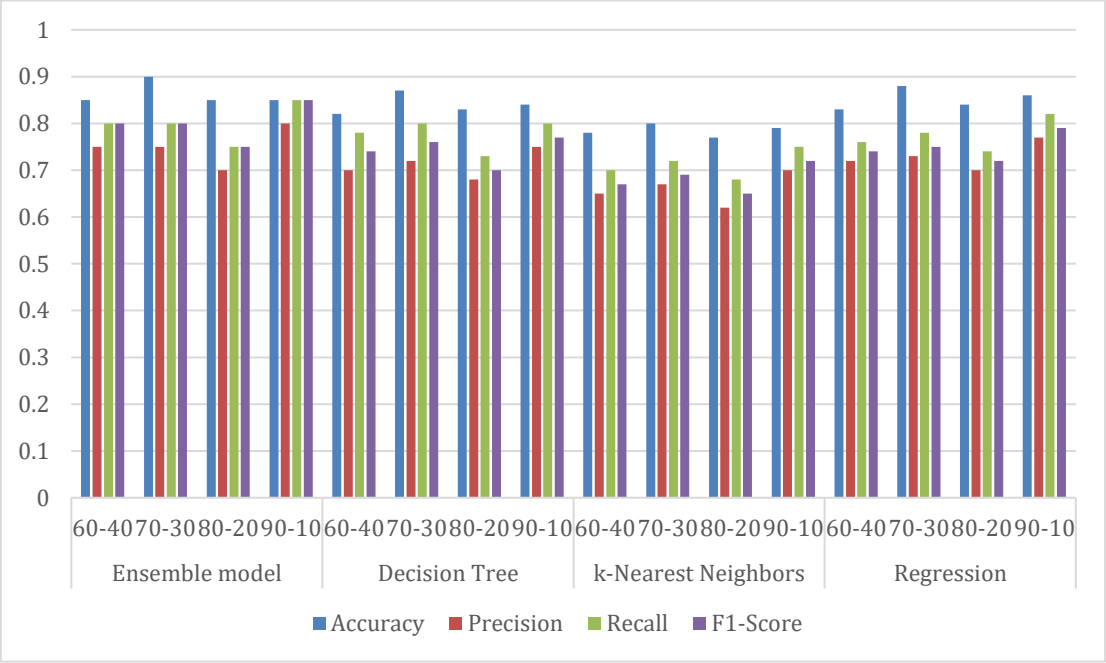

Figure 2. Comparison of metrics for various train-test splits

Figure 2 shows the comparison of metrics for various train test splits

Table 3. Performance parameters with lexicon support

Model	Train-Test Split	Accuracy	Precision	Recall	F1-Score
Ensemble with lexicon	60-40	0.85	0.75	0.8	0.8
	70-30	0.9	0.75	0.8	0.8
	80-20	0.85	0.7	0.75	0.75
	90-10	0.85	0.8	0.85	0.85
Decision Tree with Lexicon	60-40	0.86	0.78	0.82	0.8
	70-30	0.89	0.8	0.83	0.81
	80-20	0.87	0.75	0.78	0.76
	90-10	0.88	0.82	0.86	0.84
k-Nearest Neighbors with Lexicon	60-40	0.82	0.74	0.77	0.75
	70-30	0.85	0.76	0.79	0.77
	80-20	0.8	0.72	0.74	0.73
	90-10	0.83	0.77	0.8	0.78
	60-40	0.85	0.76	0.8	0.78

Regression with Lexicon	70-30	0.9	0.78	0.82	0.8
	80-20	0.86	0.74	0.77	0.75
	90-10	0.87	0.81	0.85	0.83

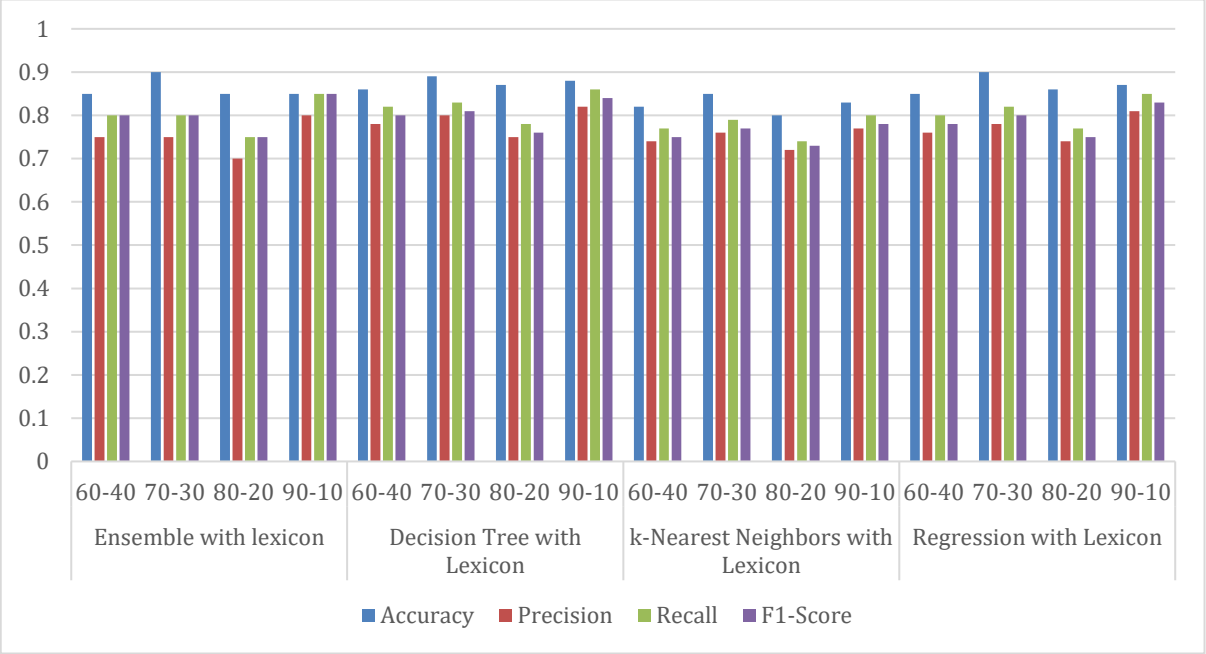

Figure 1. Comparison of metrics with lexicon support for various train-test splits

Figure 2 shows the comparison of metrics with lexicon support for various train-test splits

Table 4: Comparison of metrics on averages

Classifier	**Accuracy**	**Precision**	**Recall**	**F1 - Score**
Ensemble -L	0.9	0.79	0.815	0.8825
DT -L	0.875	0.7875	0.8225	0.8025
kNN -L	0.825	0.7475	0.775	0.7575
Regression L	0.87	0.7725	0.81	0.79
Ensemble	0.8625	0.75	0.8	0.8
DT	0.84	0.7125	0.7775	0.7425
kNN	0.785	0.66	0.7125	0.6825
Regression	0.8525	0.73	0.775	0.75

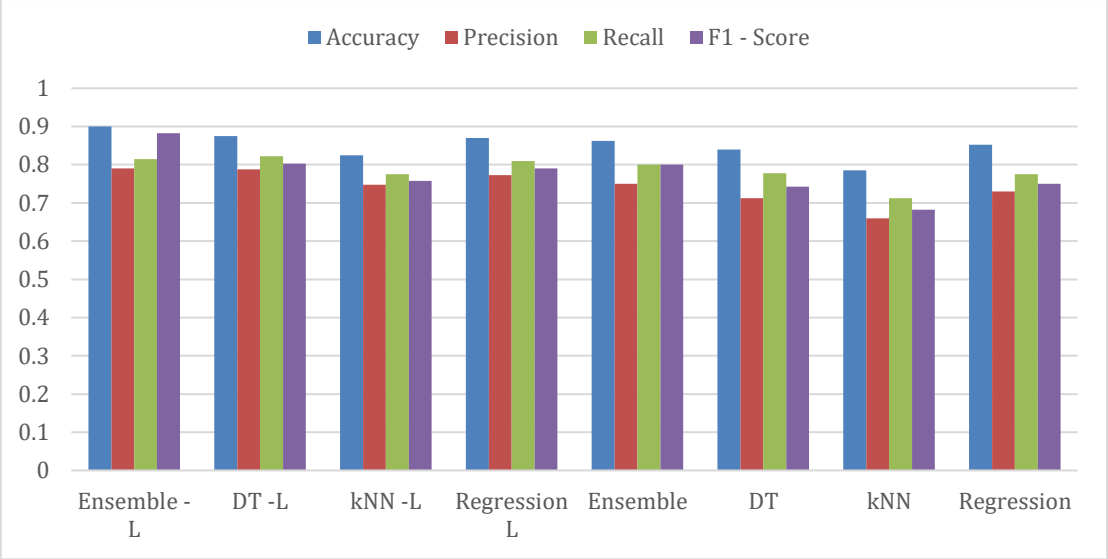

Figure 4: Clustered graph of performance metrics showing improved performance with lexicon support

Figure 2, 3 and 4 shows Accuracy, Recall, Precision, Score for the Ensemble Classifier for Web Data Using Selective Web Scraping without and with Lexicon support.

From the Tables 3 and 4 with their corresponding graphs it can be observed that, the classifiers perform slightly better with lexicon support than without. Without the lexicon support, it can be inferred as follows.

- **Decision Tree Classifier** generally shows strong performance with slightly lower metrics than the ensemble method but still comparable.
- **k-Nearest Neighbors (kNN)** shows the lowest performance among the three models, indicating it may struggle with the given data or its split.
- **Regression-based Classifier** has performance metrics close to the ensemble method, slightly better than Decision Tree in some cases

However, once lexicon support is given to the classifiers, the performance sees minor changes and they are analysed as follows.

- **Ensemble classifier:** By using a lexicon to filter and label the data, the classifiers generally show improved performance. This happens because the data has been refined, potentially reducing noise and aligning more closely with the classifier's expected input.
- **Decision Tree with Lexicon**: Shows a modest improvement across all metrics, with better precision and F1-scores.
- **k-Nearest Neighbors with Lexicon**: Demonstrates improved performance, particularly in precision and recall, but remains less effective than the other classifiers.
- **Regression with Lexicon**: Shows an improvement similar to the Decision Tree, with higher accuracy and F1-Score compared to the ensemble model.

Table 5: Time consumed with and without lexicon

Model	Train-Test Split	Time Without Lexicon (s)	Time With Lexicon (s)	Time Reduction (%)
Decision Tree	60-40	4.5	3.8	16%
	70-30	5.2	4.1	21%
	80-20	5.6	4.1	27%

	90-10	5.9	4.3	27%
k-Nearest Neighbors	60-40	5.5	3.9	29%
	70-30	6.1	4.7	23%
	80-20	6.2	4.8	23%
	90-10	6.6	5.2	21%
Regression	60-40	4.2	3.5	17%
	70-30	4.7	3.7	21%
	80-20	5.2	4.2	19%
	90-10	5.6	5.1	9%
Ensemble	60-40	6.8	5.8	15%
	70-30	7.3	6.2	15%
	80-20	8.2	6.7	18%
	90-10	9.6	7.8	19%

In addition to the improvement in performance, the classifiers see a considerable boost in the time taken to classify as the data is more selective with the lexicon support [20]. The same can be observed in the table above.

IV. CONCLUSION

In the field of web data categorization, the suggested ensemble classifier with lexical assistance and selective web scraping constitutes a noteworthy breakthrough. This method tackles the difficulties of processing large amounts of noisy web data by utilizing domain-specific lexicons and carefully integrating numerous machine learning models.

The selective web scraping step of this method is crucial. While conventional online scraping takes large amounts of data randomly, selective web scraping concentrates on specific content relevant to the categorization assignment. The overall quality and relevance of the dataset are enhanced by this targeted technique, which includes less irrelevant data. Lexicons, which are carefully curated lists of terms and phrases unique to a certain domain that ensure only the most pertinent data is collected, are another way to further improve this process. This leads to an increase in the effectiveness and efficiency of the data collection process, ultimately improving the accuracy of the classification.

A variety of base classifiers, including Decision Trees, Support Vector Machines, Naive Bayes, K-Nearest Neighbors, and Logistic Regression, are combined in the ensemble classifier framework. The ensemble approach leverages each model's capabilities by combining the predictions of these many models through the use of techniques like Bagging, Boosting, Stacking, and Voting. This results in a more durable and dependable classification system by improving predictive performance and reducing the possibility of overfitting.

An additional level of sophistication is added to the feature engineering process by incorporating lexicon support. Combining lexicon-based features with conventional machine learning features yields a feature set that is both comprehensive and strikes a balance between domain-specific information and general data properties. The model's capacity to correctly categorize complicated online data is greatly increased by this hybrid technique.

The evaluation metrics—precision, recall, accuracy, F1-score, and AUC-ROC—all support the effectiveness of this ensemble model. Furthermore, the model's performance is guaranteed to be consistent and generalizable across various data splits thanks to the application of k-fold cross-validation.

The system's ability to react to changing data patterns and developing trends is ensured by the integration of a real-time scraping and classification pipeline with a feedback loop for ongoing lexicon updates and model retraining. For applications where data is dynamic and ever-changing, like market intelligence, sentiment analysis[15], and content categorization, this flexibility is essential.

All things considered, the integration of lexicon assistance, ensemble learning, and selective online scraping provides a scalable and effective approach to real-time web data classification. This novel method keeps the system adaptable and applicable in a variety of contexts while also improving classification accuracy and robustness.

REFERENCES

[1] Tibshirani, R. (1996). Bias, variance, and prediction error for classification rules. Technical Report, Statistics Department, University of Toronto.

[2] Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32.

[3] Dietterich, T. G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. Machine Learning, 40(2), 139-157.

[4] Quinlan, J. R. (1996). Bagging, boosting, and C4.5. In Proceedings of the Thirteenth National Conference on Artificial Intelligence (pp. 725–730). Cambridge, MA: AAAI Press/MIT Press.

[5] Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 55(1), 119-139.

[6] Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). Data mining: Practical machine learning tools and techniques. Morgan Kaufmann.

[7] Liu, B. (2012). Sentiment analysis and opinion mining. Synthesis Lectures on Human Language Technologies, 5(1), 1-167.

[8] Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M. (2011). Lexicon-based methods for sentiment analysis. Computational Linguistics, 37(2), 267-307.

[9] Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. Ain Shams Engineering Journal, 5(4), 1093-1113.

[10] Tsai, C.-F., & Hsu, Y.-F. (2011). A meta-learning framework for spam detection. Expert Systems with Applications, 38(3), 2312-2317.

[11] Zhang, Z., Wang, X., Zhang, Y., & Zhu, X. (2019). Combining lexicon-based and learning-based methods for Twitter sentiment analysis. Applied Intelligence, 49, 3085-3097.

[12] Yang, Z., Chen, X., & Zhao, X. (2020). Neural network-based lexicon generation for sentiment analysis. Neural Processing Letters, 51, 621-636.

[13] Amit, Y., & Geman, D. (1997). Shape quantization and recognition with randomized trees. Neural Computation, 9, 1545–1588.

[14] Opitz, D., & Maclin, R. (1999). Popular ensemble methods: An empirical study. Journal of Artificial Intelligence Research, 11, 169-198.

[15] Bauer, E., & Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Machine Learning, 36(1/2), 105–139.

[16] Sahlaoui, H., Alaoui, E. A. A., Agoujil, S., & Nayyar, A. (2024). An empirical assessment of smote variants techniques and interpretation methods in improving the accuracy and the interpretability of student performance models. Education and Information Technologies, 29(5), 5447-5483.

[17] Hu, G. (2024, December). Data Pattern Recognition of KNN Neural Network Based on Improved Neighbor Sample Selection Strategy. In 2024 4th International Conference on Mobile Networks and Wireless Communications (ICMNWC) (pp. 1-7). IEEE.

[18] Foulds, J., Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2025). Data Mining: Practical machine learning tools and techniques. Elsevier.

[19] Sarmah, U., Borah, P., & Bhattacharyya, D. K. (2024). Ensemble Learning Methods: An Empirical Study. SN Computer Science, 5(7), 924.

[20] García-Díaz, J. A., López, Ú., Núñez-González, J. D., & García-Mendoza, M. (2021). A hybrid approach to sentiment analysis combining deep learning and lexicon-based methods. Expert Systems with Applications, 178, 114998.

[21] Liu, C., & Liu, J. (2024, April). Multi-Class Missense Variant Prediction Based on Asymmetric Bagging and Tabular Generation Combined with Extreme Gradient Boosting. In 2024 9th International Conference on Intelligent Computing and Signal Processing (ICSP) (pp. 1687-1692). IEEE.

[22] García-Pedrajas, N. E., Cuevas-Muñoz, J. M., Cerruela-García, G., & de Haro-García, A. (2024). A thorough experimental comparison of multilabel methods for classification performance. Pattern recognition, 110342.

[23] Maltare, N. N., Sharma, D. & Patel, S. (2023). An Exploration and Prediction of Rainfall and Groundwater Level for the District of Banaskantha, Gujrat, India. International Journal of Environmental Sciences, 9(1), 1-17. https://www.theaspd.com/resources/v9-1-1-Nilesh%20N.%20Maltare.pdf

Author Profiles:

Mr.Yogesha T has obtained M.Tech specialized in software engineering from VTU Belagavi and currently pursuing PhD in Dept of CSE VTU Mysuru. He is currently working as Assistant Professor and NSS program officer in the Department of Computer Science and Engineering, VTU, PG Center Mysuru. He is actively involved in various research activities and has published research papers in reputed national and international journals

Dr.Thimmaraju S N completed his PhD from VTU, Belagavi and is currently working as Professor & Program Coordinator in VTU PG Centre, Mysuru. Previously he has worked as Regional Director in VTU Regional Centre, Mysuru. His area of interest is Graph Theory and Computer Networks. He has teaching experience of 22 years and has published several Research Papers in National & International Journals and has guided 3 PhD students in VTU, Belagavi. He is the member of Board of Studies, Board of Examination for VTU and other Universities.