Computer Science and Interdisciplinary Research Journal
Volume 1, Issue 2, 2024. ISSN: 3033-1218, DOI: 10.70862/CSIR.2024.0102-08

Classification of News Using Artificial Intelligence Methods

Georgi Gustinov[1], Emil Hadzhikolev[2], Stanka Hadzhikoleva[3], Stoyan Cheresharov[4]

University of Plovdiv Paisii Hilendarski, 236, Bulgaria Blvd, 4027 Plovdiv, Bulgaria

[1] lz3gt@uni-plovdiv.bg, [2] hadjikolev@uni-plovdiv.bg,
[3] stankah@uni-plovdiv.bg, [4] cheresharov@uni-plovdiv.bg

Abstract – Many organizations use analysis and forecasting systems that rely on human labor for data collection and subsequent processing. The use of artificial intelligence algorithms improves the efficiency of these systems by reducing the risk of suboptimal decisions and erroneous judgments. For such systems to be useful, their costs must not exceed the generated revenue.

This article presents experiments on news classification using intelligent methods as part of a larger system for automated forecasting of critical event. A model and experimental framework for a software system utilizing artificial intelligence methods are proposed. A prototype has been developed to experiment with the Multinomial Naive Bayes and Support Vector Machines algorithms, combined with the vectorizers CountVectorizer, HashingVectorizer, and TfidfVectorizer. The tests were conducted using a parallel computer consisting of four Raspberry Pi 4 nodes. The best results were achieved with the Multinomial Naive Bayes method and the CountVectorizer.

Keywords – Critical Events, Forecasting, Classification, Categorization, Parallel Processing.

DOI: https://doi.org/10.70862/CSIR.2024.0102-08

Corresponding author: Emil Hadzhikolev,
University of Plovdiv Paisii Hilendarski
Email: hadjikolev@uni-plovdiv.bg

Received: 12 January 2025.
Revised: 24 January 2025.
Accepted: 18 February 2025.
Published: 22 February 2025.

© 2024 Georgi Gustinov, Emil Hadzhikolev, Stanka Hadzhikoleva & Stoyan Cheresharov. This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

The article is published with Open Access at http://csirjournal.com/

1. Introduction

Innovations in manufacturing, healthcare, energy, transportation, commercial services, and governance in recent years have led to a continuous increase in global data generation. According to Statista, the volume of data created, captured, copied, and consumed worldwide is projected to reach 394 zettabytes by 2028, which is approximately four times more than in 2022 [1]. The accumulation of large volumes of historical data on past processes and events enables more detailed study and analysis. Predicting critical events in a given field is becoming increasingly accessible thanks to the capabilities of advanced data analysis methods, including those powered by artificial intelligence.

A critical event represents a significant turning point that often occurs unexpectedly and has the potential to lead to serious negative consequences. Such events can disrupt or even completely unsettle the normal course of routine human or business activities. Numerous authors have studied critical events across various domains.

Sornette investigates the collapse of financial markets by examining the causes behind the formation of financial "bubbles" throughout history [2]. The author draws on studies of other historical precedents, such as the "tulip mania" in the Netherlands, the Great Crash of October 1929, Black Monday in 1987, and others. Longin applies extreme value theory in finance and its methods to explain critical events in capital markets [3]. Critical analyses related to modeling Extremal Events in Financial Phenomena have been conducted by Focardi and Fabozzi in [4].

The issue of incidents in healthcare remains a pressing topic. The Australian Commission on Safety and Quality in Health Care published the Incident Management Guide, where it defines an incident as an “event or circumstance that resulted, or could have resulted, in unintended or unnecessary harm to a patient or consumer; or a complaint, loss or damage”

[5]. Woloshynowych et al. conducted a literature review on the methods used for investigating incidents in healthcare. They developed guidelines for analyzing critical incidents in the hospital sector, mental health, and primary care [6]. Working in the same field, Walshe defines a critical event as an instance where a patient received poor-quality care [7]. Seys et al. expanded research on adverse events in healthcare, focusing on their impact on healthcare workers, who are often considered second victims [8]. Baldisseri et al. examined critical events from the perspective of intensive care units, defining such events as cases involving natural disasters, biological (pandemic) and chemical disasters, radiation, anthropogenic (terrorist and criminal acts) disasters, and others [9].

Kadri and colleagues examine critical events caused by natural disasters or technological reasons. They study their impact on critical infrastructure (industrial and residential complexes, transportation infrastructure, etc.), while also considering subsequent cascading effects [10]. In a similar way, Mittelstädt et al. focus on the cascading effects between multiple infrastructures, such as the power grid, communication networks, and road networks. They develop a comprehensive visual analytics system that integrates separate visualizations of different network infrastructures with social media analysis and on-site mobile analytics [11]. Nivolianitou and Synodinou identify factors that contribute to effective emergency management, including natural disasters and critical incidents. They highlight the importance of prevention, coordination, adherence to procedures, and clarity of roles [12]. Mijalkovic and Cvetković explore and analyze the vulnerability of critical infrastructure during major natural disasters. They present various approaches and measures to minimize damages and ensure resilience during natural disasters [13].

Critical events are also a subject of interest in other domains. For instance, Berzin and De Marco study critical events in human development during the transition to adulthood and their relationship with poverty [14]. Road traffic conflicts as critical events are explored in the work of Bagdadi [15]. He proposes a method for assessing the safety-critical events based on risk evaluation of crashes and the assessment of potential consequences. Various techniques are usually employed for analyzing critical events, including regression, logistic regression, cluster analysis, neural networks, decision trees, and others [16].

Critical events have various forms and can be triggered by different causes, including natural, technological, human, social, and others. Natural factors include earthquakes, floods, hurricanes, volcanoes, and wildfires. Technological factors can involve industrial accidents, transportation incidents, cyberattacks, and more. The human factor encompasses criminal and terrorist activities, human errors, and conflicts. Social factors can include epidemics, population migration, overpopulation, and others.

The management of critical events generally involves four stages:

- *Prevention* – identifying potential risks and critical events and taking measures to reduce the likelihood of their occurrence.
- *Preparation* – developing strategies and action plans, training employees, and ensuring the availability of necessary resources to be used in the event of a critical incident.
- *Response* – providing a rapid and effective reaction to the event to minimize damage.
- *Recovery* – restoring normal conditions after the crisis has ended, analyzing the causes and consequences, and improving current strategies and action plans.

Early identification of critical events and a rapid and effective response are essential for successful crisis management. One possible approach to achieving this is through the analysis of real-time data collected from the internet.

As part of the present study, a module for news classification has been developed and analyzed. Its purpose is to classify news content by topics and is part of a larger software system for early identification of critical events. Independent news from the United States Geological Survey (USGS) [17] is used as input data. Through a classifier, incoming news is distributed into thematic categories. Then, the information must be analyzed for specific terms, based on which the importance and type of the critical event are determined. Machine learning methods are used for classification, which increases processing speed and minimizes human intervention in the system.

The present study is inspired by the publication of Zowalla et al. [18]. Their team developed a crawler for the acquisition of the German Health Web, which includes all health-related web content from Germany, Austria, and Switzerland. Using this crawler, they collected and analyzed health-related articles from public internet sources. To classify the publications, they employed the Support Vector Machine (SVM) algorithm, which is highly suitable for text categorization [19]. Zowalla and colleagues found that over 80% of the target data could be successfully retrieved from the internet [18], which provides a strong basis for its use as a primary data source. The choice of the news section of the USGS is driven by its specific structure, independent of the news flow in established news agencies. Typically, USGS news is

used as a primary source in the media space and is republished or cited.

The aim of the current experiment is:

- To create an automatic classifier for articles from multiple media streams;
- To investigate the classification effectiveness, measured using a statistical method;
- To study the flexibility of the model by modifying their parameters.

2. Models and Methods

2.1. Model of a Critical Events System

Every organization that works with information requires an analytical system for its processing. Systems based on the human factor are inert, and even if they are effective and successful initially, at some point, they lead to suboptimal decisions [20] and bias [21]. The main objective of the proposed automated system with artificial intelligence is to optimize activities and eliminate errors and inaccuracies caused by human intervention. The system consists of the following modules (Fig. 1):

1. Data collection from the internet: The data collection module aims to periodically extract internet content from specified web services according to a predefined schedule. Each extracted result must be saved in a text file.

2. Processing of collected data: This module is responsible for parsing the data, where the text is split into words, and they are stored in appropriate structures for further processing.

3. Data storage: The data is stored on scalable file systems in relational or non-relational databases for subsequent extraction and analysis.

4. Data analysis: Data analysis can be performed using various artificial intelligence methods, and for real-time analysis and classification of information, there are suitable libraries for natural language processing. The goal is to classify the output words according to their belonging to specific linguistic sets, resulting in classified individual words or phrases ranked by importance. Another responsibility of this module is the identification of potential critical events, which is based on expressions with a high significance coefficient or recurrence in the input data.

5. Decision making: This module aims to generate signals for potential critical events, which are to be sent to systems specialized in processing signals of the respective type.

6. Visualization: The results for critical events recorded in the graph database must be visualized for users in an appropriate manner.

After conducting research, the technologies presented in Table 1 were selected as suitable for the initial experiments.

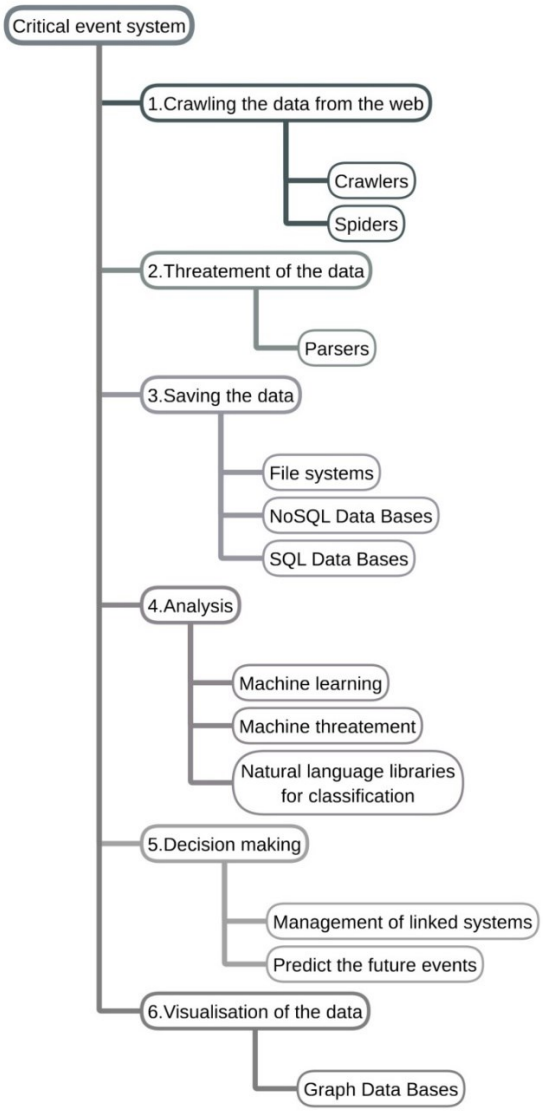


Figure 1. Model of a system for identifying, analyzing, and processing critical events.

Table 1. Modules and suitable technologies for their implementation.

Module	Technologies
Data collection	Crawler – Apache Nutch
Data processing	Parser – Apache Tika
Data storage	File storage – Apache Hadoop
Data analysis	Spark, NLP, NLTK
Decision-making	System Functions
Visualization	Standard Visualization Tools for Neo4j on Hadoop

2.2. Data Collection and Preparation

In the initial experiments, the news from the USGS website (page https://www.usgs.gov/news/news-releases/all-news-releases) was collected manually. The subsequent categorization and filtering of the content were performed automatically. Separating meaningful content from repetitive phrases used for advertisements and links to other websites is crucial

for the categorization of the news. The steps in the text processing workflow are as follows:

1. Open the news page and select the desired category of news from the Topic section.
2. In the Year section, select a specific year and sort the articles in descending order.
3. Download all news articles from the selected year and save them in text format.
4. Edit the saved files by removing all unnecessary links or text located before the Breadcrumb tag, as well as after the tags ### or Get Our News, depending on the content.
5. Store the cleaned files without extensions, assigning them sequential numerical names in the current dataset.
6. Once the addition of news articles to the dataset is complete, compress the dataset in ZIP format.
7. The resulting compressed dataset is used as input for the classifier executed in the research environment Jupyter [22].

The technique used for text content analysis and classification is Bag of Words (BoW). This technique represents a document as a BoW vector, where each element corresponds to the normalized count of occurrences of a given word in the document [23, 24]. The counting process requires exact matching, which is treated as a strict match between the words in the text and the predefined words or expressions. The main disadvantage of this approach is the inability to perform higher-order semantic analysis; consequently, in the current implementation, the system cannot recognize nuances in the analyzed documents. Despite these limitations, this method was chosen due to its speed. Experiments were conducted using three types of vectorizers — CountVectorizer, HashingVectorizer, and TfidfVectorizer - available in the scikit-learn library [25, 26].

For the creation of the initial dataset, news from USGS in the form of HTML documents [17] was used, categorized into Biology, Climate, and Energy. The process of creating a word vector corresponding to a single news item involves several stages (Fig. 2):

1. Initial processing of the text document: At this stage, duplicate titles and links to similar articles are removed from the text files.

2. Applying a regular expression (REGEX): This removes prepositions, punctuation marks, whitespaces, and others.

3. Tokenizing: By applying the regular expression, a list of key words in the text is generated.

4. Removing stop words: Optionally, parasitic connecting words with no weight in the current text (known as stop words) such as if, then, also, are, have, etc., can be removed.

5. Applying a Vectorizer: This step provides a vectorized numerical representation of the word list.

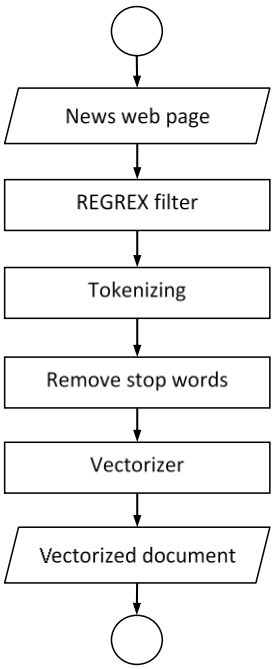


Figure 2. General process of creating a vectorized document.

Different vectorizers use different algorithms to determine the significance of words, which is subsequently important for machine learning classification methods.

2.3. Classification Methods

The classification of vectorized documents is performed using the algorithms Multinomial Naive Bayes [27, 28] and Support Vector Machines [29] with varying kernels. These algorithms operate on word vectors and calculate the percentage ratio of specific expressions in the text relative to all expressions within it. Based on these calculations, the category to which the currently processed document belongs is determined.

The input data for training and testing the models is split into a training and test dataset in a 75:25 ratio.

The training dataset X_{train} consists of document vectors and their corresponding target categories and can be represented as:

$$X_{train} = \{(DocumentVector_i, TargetCategory_i)\}_{i=1}^{n}, n \in N.$$

Similarly, the test dataset X_{test} also contains document vectors and their corresponding real categories, represented as:

$$X_{test} = \{(DocumentVector_j, RealCategory_j)\}_{j=1}^{m}, m \in N.$$

The model M is trained on the training dataset X_{train}. After training, the model is applied to the document vectors in the test dataset to evaluate its performance. During testing, the model predicts a category for each vector in the test dataset:

$$PredictedCategory_j = M(DocumentVector_j),$$
$$j = 1, \dots, m.$$

Various evaluation methods compare the real and predicted categories – $RealCategory_j$ and $PredictedCategory_j$ – for each document vector and its original news content.

The overall classification process includes the parallel training of different classification methods on vectorized documents (Figure 3).

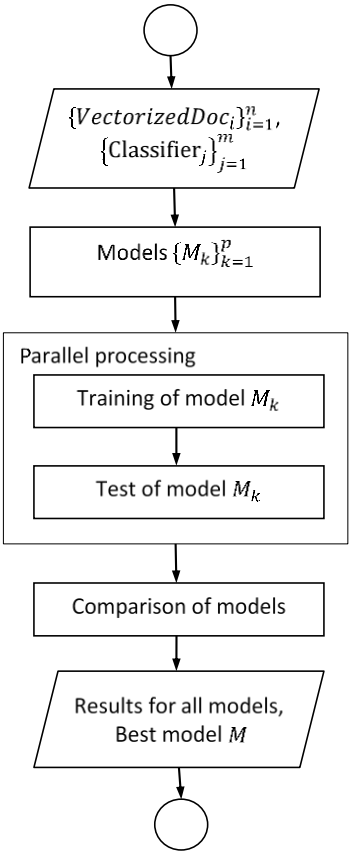


Figure 3. General diagram of the process for training, testing, and evaluating classification models.

In the mathematical model, the total number of vectorized documents is denoted as a set of n elements, and the classifiers are denoted as a set of m elements:

$$VectorizedDocuments = \{VectorizedDocument_i\}_{i=1}^{n}$$
$$Classifiers = \{Classifier_j\}_{j=1}^{m}, n, m \in N.$$

The maximum number of training models, combining classifiers and vectorized documents, is $n \times m$:

$$Models = \{M_k\}_{k=1}^{p} \subseteq \{M_{i,j}\}_{i=1,j=1}^{i=n,j=m}, n, m, p \in N.$$

To identify the most effective model, all models can be trained in parallel, or experiments can be conducted with only p selected models. Each of the selected models is trained and tested sequentially. Important steps in the process include presenting the performance of all models and comparing them to determine the model with the best results.

2.4. Hardware and Software Tools

The research utilized the following tools and systems:

- *Parallel Computer*: A cluster consisting of four Raspberry Pi 4 nodes with quad-core processors and 8 GB of RAM [30], designed to minimize power consumption (approximately 4x2.5 watts compared to 200–400 watts for a standard desktop computer) while providing acceptable performance.
- *Operating system* Ubuntu 22 Server [31].
- *Remote research environment Jupyter* [22].
- *Programming language Python 3* [32].

To accelerate computations when processing large datasets, IPython Interactive Computing [33] was employed. This setup utilizes a parallel cluster of Python 3 instances distributed across each node (four instances, one for each core). While this configuration was tested for performance evaluation, it was not deemed necessary for the specific case. However, the need for further exploration of this parallel system as a cost-effective option for 24/7 parallel text analysis was noted.

The parallel computer was equipped with a HADOOP distributed file system [34] and the SPARK analytics engine [35]. These software systems will be utilized for data storage and analysis when the volume of processed information is expanded.

3. Experiments and Results

The training data includes a total of 177 news articles, collected manually and divided into a ratio of 75% for training and 25% for testing. The news articles include a title, date, content, and a concluding section. The distribution of the news articles by category and group is presented in Table 2.

Table 2. Number of articles by categories and data groups

Category	Total Articles	Training Articles	Test Articles
Biology	62	47	15
Climate	40	30	10
Energy	75	55	20

In the initial experiments, the vectorizers CountVectorizer, HashingVectorizer, and TfidfVectorizer were tested in combination with the algorithms Multinomial Naive Bayes and Support Vector Machines with a linear kernel. To extract meaningful words, the expression REGEX = ur"\b\w\w+\b" was used, which identifies whole words with at least two characters in the text. Based on the obtained results (Table 3), the following combinations were selected as the most suitable for further experiments:

- Multinomial Naive Bayes with CountVectorizer;
- Support Vector Machines with a linear kernel combined with TfidfVectorizer.

A cross-validation coefficient of $K = 6$ was chosen to ensure balanced use of the data for training and testing.

Table 3. Precision using the regular expression REGEX = ur"\b\w\w+\b" for different combinations of algorithms and vectorizers.

Algorithm	Count Vectorizer	Hashing Vectorizer	Tfidf Vectorizer
MNBayes	0.712	0.441	0.621
SVM (Linear Kernel)	0.689	0.593	0.717

In subsequent experiments, the REGEX: ur"\b[a-z0-9_-.]+[a-z][a-z0-9_-.]+\b" was tested, where strings containing special characters such as hyphens (-), dots (.), digits (0-9), etc., are also considered as words. Additionally, frequently repeated parasitic words (stop words) [36] such as “and”, “is”, “are”, “then”, etc., were removed. The results are presented in Table 4.

Based on the previous studies, experiments were conducted with several variants of the classifiers (Table 5). It is observed that the removal of stop words has a positive impact on the precision of the classifiers.

Table 4. Precision of selected algorithms using REGEX = ur"\b[a-z0-9_-.]+[a-z][a-z0-9_-.]+\b" and stop words.

Algorithm	REGEX Only	REGEX + Stop Words
MNBayes + CountVectorizer	0.718	0.701
SVM (Linear Kernel) + TfidfVectorizer	0.706	0.717

Table 5. Precision of selected classifiers.

Classifier	Training Dataset	Test Dataset
CountVectorizer, MNBayes, REGEX, alpha, stop words	0.955	0.778
CountVectorizer, MNBayes	0.955	0.800
TfidfVectorizer, SVM Linear, REGEX, stop words	0.947	0.756
TfidfVectorizer, SVM Rbf, REGEX, stop words	0.947	0.689
TfidfVectorizer, SVM Linear	0.932	0.756

The confusion matrix for Multinomial Naive Bayes + CountVectorizer on the test dataset (Table 6) shows an accuracy of 0.80. The confusion matrix for SVM Linear Kernel + TfidfVectorizer on the test dataset (Table 7) shows an accuracy of 0.756.

Table 6. Confusion matrix for Multinomial Naive Bayes + CountVectorizer on the test dataset.

	Biology	Climate	Energy
Biology	14	0	1
Climate	2	6	2
Energy	0	4	16

Table 7. Confusion matrix for SVM Linear Kernel + TfidfVectorizer on the test dataset.

	Biology	Climate	Energy
Biology	14	0	1
Climate	5	3	2
Energy	1	2	17

In both cases, classification errors in the Climate category can be attributed to the smaller amount of training data available for this category.

The best classifier in the current experiments is the CountVectorizer + Multinomial Naive Bayes model, which achieved an accuracy of 0.95 on the training dataset and 0.80 on the test dataset.

The evaluation metrics – Precision, Recall, and F1-Score – for assessing the quality of the classification model are presented in Table 8.

Table 8. Evaluation of the CountVectorizer + Multinomial Naive Bayes Classifier.

Category	Precision	Recall	F1-Score
Biology	0.88	0.93	0.90
Climate	0.60	0.60	0.60
Energy	0.84	0.80	0.82

4. Conclusion

This study presents a news classification module, which is part of a system for predicting critical events. The hardware and software tools used for its implementation are described. Experiments were conducted to utilize artificial intelligence algorithms for classifying news into different categories. The most effective model, achieving 80% accuracy, was obtained using the combination of the Multinomial Naive Bayes method and the CountVectorizer. Improved model performance can be expected with larger test datasets. The use of Large Language Models for semantic analysis and prediction will be the subject of future experiments.

References

[1] Statista (2024) Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2023, with forecasts from 2024 to 2028, Retrieved from

https://www.statista.com/statistics/871513/worldwide-data-created/

[2] Sornette, D. (2009). *Why stock markets crash: critical events in complex financial systems*. Princeton university press. https://doi.org/10.1515/9781400829552

[3] Longin, F. (Ed.). (2016). *Extreme events in finance: A handbook of extreme value theory and its applications*. John Wiley & Sons. http://dx.doi.org/10.1002/9781118650318

[4] Focardi, S. M., & Fabozzi, F. J. (2003). Fat tails, scaling, and stable laws: a critical look at modeling extremal events in financial phenomena. *The Journal of Risk Finance*, *5*(1), 5-26. https://doi.org/10.1108/eb022976

[5] Australian Commission on Safety and Quality in Health Care. (n.d.). *Incident management guide*. Retrieved from https://www.safetyandquality.gov.au/publications-and-resources/resource-library/incident-management-guide

[6] Woloshynowych, M., Rogers, S., Taylor-Adams, S., & Vincent, C. (2005). The investigation and analysis of critical incidents and adverse events in healthcare. http://dx.doi.org/10.3310/hta9190

[7] Walshe, K. (2000). Adverse events in health care: issues in measurement. *BMJ Quality & Safety*, *9*(1), 47-52. https://doi.org/10.1136/qhc.9.1.47

[8] Seys, D., Scott, S., Wu, A., Van Gerven, E., Vleugels, A., Euwema, M., ... & Vanhaecht, K. (2013). Supporting involved health care professionals (second victims) following an adverse health event: a literature review. *International journal of nursing studies*, *50*(5), 678-687. https://doi.org/10.1016/j.ijnurstu.2012.07.006

[9] Baldisseri, M., Reed, M. J., & Wax, R. (Eds.). (2019). *Intensive Care Unit in Disaster, An Issue of Critical Care Clinics* (Vol. 35, No. 4). Elsevier Health Sciences.

[10] Kadri, F., Birregah, B., & Châtelet, E. (2014). The impact of natural disasters on critical infrastructures: A domino effect-based study. *Journal of Homeland Security and Emergency Management*, *11*(2), 217-241. https://doi.org/10.1515/jhsem-2012-0077

[11] Mittelstädt, S., Wang, X., Eaglin, T., Thom, D., Keim, D., Tolone, W., & Ribarsky, W. (2015). An integrated in-situ approach to impacts from natural disasters on critical infrastructures. In *2015 48th Hawaii International Conference on System Sciences* (pp. 1118-1127). IEEE. https://doi.org/10.1109/HICSS.2015.136

[12] Nivolianitou, Z., & Synodinou, B. (2011). Towards emergency management of natural disasters and critical accidents: The Greek experience. *Journal of environmental management*, *92*(10), 2657-2665. https://doi.org/10.1016/j.jenvman.2011.06.003

[13] Mijalković, S., & Cvetković, V. (2013). Vulnerability of critical infrastructure by natural disasters. *national critical infrastructure protection–regional perspective, University of Belgrade– Faculty of Security Studies, Institute for Corporative Security Studies in Ljubljana, Belgrade*, 91-102.

[14] Berzin, S. C., & De Marco, A. C. (2010). Understanding the impact of poverty on critical events in emerging adulthood. *Youth & Society*, *42*(2), 278-300. https://doi.org/10.1177/0044118X09351909

[15] Bagdadi, O. (2013). Estimation of the severity of safety critical events. *Accident Analysis & Prevention*, *50*, 167-174. https://doi.org/10.1016/j.aap.2012.04.007

[16] Acito, F. (2023). Predictive analytics with KNIME. *Analytics for citizen data scientists. Switzerland: Springer*. https://doi.org/10.1007/978-3-031-45630-5

[17] U.S. Geological Survey. (n.d.). *All news releases*. Retrieved from https://www.usgs.gov/news/news-releases/all-news-releases

[18] Zowalla, R., Wetter, T., & Pfeifer, D. (2020). Crawling the german health web: Exploratory study and graph analysis. *Journal of medical Internet research*, *22*(7), e17853. https://doi.org/10.2196/17853

[19] Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning* (pp. 137-142). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/BFb0026683

[20] Alós-Ferrer, C., Hügelschäfer, S., & Li, J. (2016). Inertia and decision making. *Frontiers in psychology*, *7*, 169. https://doi.org/10.3389/fpsyg.2016.00169

[21] Samadi, A. H., Panahi, M., & Raanaei, A. (2024). The roots of cognitive inertia: an introduction to institutional changes. In *Institutional Inertia: Theory and Evidence* (pp. 133-153). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-51175-2_6

[22] Project Jupyter. (n.d.). *Project Jupyter*. Retrieved from https://jupyter.org

[23] Zhao, R., & Mao, K. (2017). Fuzzy bag-of-words model for document representation. *IEEE transactions on fuzzy systems*, *26*(2), 794-804. https://doi.org/10.1109/TFUZZ.2017.2690222

[24] Wu, L., Hoi, S. C., & Yu, N. (2010). Semantics-preserving bag-of-words models and applications. *IEEE Transactions on Image Processing*, *19*(7), 1908-1920. https://doi.org/10.1109/TIP.2010.2045169

[25] Scikit-learn. (n.d.). *Scikit-learn: Machine learning in Python — scikit-learn 1.5.1 documentation*. Retrieved from https://scikit-learn.org/

[26] Scikit-learn. (n.d.). *Sklearn.feature_extraction*. Retrieved from https://scikit-learn.org/stable/api/sklearn.feature_extraction.html

[27] Scikit-learn. (n.d.). *1.9. Naive Bayes*. Retrieved from https://scikit-learn.org/stable/modules/naive_bayes.html

[28] Wickramasinghe, I., & Kalutarage, H. (2021). Naive Bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation. *Soft Computing*, *25*(3), 2277-2293.

[29] Scikit-learn. (n.d.). *1.4. Support Vector Machines*. Retrieved from https://scikit-learn.org/stable/modules/svm.html

[30] Raspberry Pi Ltd. (n.d.). *Raspberry Pi 4 Model B specifications*. Retrieved from https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/

[31] Ubuntu. (n.d.). *Install Ubuntu on a Raspberry Pi*. Retrieved from https://ubuntu.com/download/raspberry-pi

[32] Python Software Foundation. (n.d.). *Welcome to Python.org*. Retrieved from https://www.python.org/

[33] IPython Development Team. (n.d.). *Using IPython for parallel computing — IPython 3.2.1 documentation*. Retrieved from https://ipython.org/ipython-doc/stable/parallel/index.html

[34] Apache Software Foundation. (n.d.). *Apache Hadoop*. Retrieved from https://hadoop.apache.org/

[35] Apache Software Foundation. (n.d.). *Apache Spark - Unified engine for large-scale data analytics*. Retrieved from https://spark.apache.org/

[36] xldrkp. (n.d.). *English Stopwords*. Retrieved from https://gist.github.com/xldrkp/4a3b1a33f10d37bedbe0068f2b4482e8