Distributed Classifier for SDGs Topics in Online News using RabbitMQ Message Broker

A Nugroho\textsuperscript{1}, Widyawan\textsuperscript{2}, and S S Kusumawardani\textsuperscript{3}
\textsuperscript{1,2,3}Department of Electrical Engineering and Information Technology, Faculty of Engineering, Universitas Gadjah Mada, Indonesia
E-mail: agung.n2016@mail.ugm.ac.id\textsuperscript{1}, widyawan@ugm.ac.id\textsuperscript{2}, and suning@ugm.ac.id\textsuperscript{3}

Abstract.
Internet data has grown very fast and becomes very large. Thus continuous improvement will always be required to face this challenge. The Sustainable Development Goals (SDGs) are defined by the United Nations (UN) to encourage improvements in the field of life in each country. We proposed a combination of Distributed System (RabbitMQ) and Machine Learning (Naïve Bayes Classifier) as one of the support to measure the achievement level of Sustainable Development Goals (SDGs) in Indonesia. The methods will categorize the Detik.com news into two classes; the relevant to SDGs of Indonesia and the irrelevant to SDGs of Indonesia. Our work shows that the use of the load-balance feature in RabbitMQ could shorten the processing time of the Naïve Bayes Classifier. RabbitMQ as a load-balancer can divide the workload equally, thus reducing the latency time of the Naïve Bayes Classifier classification process by 30.3 percent.

1. Introduction
In the past, newspapers were printed once a day. Current news is created and produced by social media (citizen journalism) and the official press. With the development of information and communication technology, it is possible to read newspaper news in real-time and on-demand, so that we can receive information quickly anytime and anywhere.

Information Retrieval is a process to rediscover the information required by the system. News is not only communicated through printed media, but also online media. Fast technology makes people more up to date with the latest news or information. Detik.com is one of the online news websites that presents a wide range of latest information \cite{1}.

The use of official online news media is preferred than the information from social media, because the official online media is more accountable and complies with the rules of journalism. Detik.com is registered at the Indonesian Pressboard, and Detik.com also one of the popular news websites in Indonesia.

1.1. Sustainable Development Goals
News in the Indonesian official media today is quite reliable. One way to measure the achievement level of the Sustainable Development Goals (SDGs) in a country can be done by collecting information from the Internet news. The Sustainable Development Goals (SDGs) are defined by the United Nations (UN) to encourage improvements in the areas of the life of every country in the world. There are 17 goals in SDGs (Figure 1). The SDGs term in
Indonesia is called the ”Tujuan Pembangunan Berkelanjutan (TPB)” by the Ministry of National Development Planning/National Development Planning Agency of Indonesia.

![Image of 17 SDGs goals](image)

**Figure 1.** 17 goals on SDGs according to the UN.

### 1.2. Machine Learning Classifier

To classify SDGs from news can be done manually by humans and can also be done using Machine Learning Classifiers. Of course, the use of Machine Learning Classifiers requires less effort compared to manual classification by humans. There are two types of Machine Learning Classifiers based on the data character. Batch Classifiers are for the static dataset (like CSV files, log files, etc.), whereas Online Classifiers are for streaming data (such as road traffic, realtime CCTV, Twitter and Instagram posts, etc.). The differences between Batch Classifiers and Online Classifiers are located on how the machine learns and how the machine predicts. Online Classifiers have to deal with data that continues to grow, while its processing is done only once, so the period of observations becomes mandatory [2].

From different types of Machine Learning such as Naïve Bayes, K-Nearest Neighbors [3], Decision Trees, Support Vector Machine (SVM) and also Deep Learning (such as Convolutional Neural Network [4] and Recurrent Neural Network [5]), each of Machine Learning can perform data classification tasks (Classifier). The goal of Machine Learning Classifier is to learn a mapping from input $x$ (dataset) to output $y$, where $y \in \{1, ..., C\}$. $C$ was the number of classes. There were two types of Classification, based on the number of classes:

- **Binary Classification**
- **Multi-label Classification**

One of the most widely used classifier is Naïve Bayes Classifier because it performs very high in real-world scenarios [2]. Once the probability is calculated, then the class that has the highest probability will be selected as the class that will be processed predictions. The training process of Naïve Bayes is a forward flow. Figure 2 shows the conventional design of Naïve Bayes Classifier. The design consists of a dataset as the input, Naïve Bayes Classifier as the process, and the output.
1.3. Distributed System
The conventional design of Machine Learning is usually without the use of a Distributed System. Distributed Systems can be built by combining multiple Machine Learning and by using a Message Queuing System. Message Queuing System is an entity responsible for handling delivery or distribution of messages.

Some researchers called Message Queuing System as Message Broker. Dobbelaeere classifies RabbitMQ and Apache Kafka are equally in the Publish-Subscribe System (Pub/Sub System) [6]. Whereas Tennant classifies that there are three groups of Message Queuing System [7], which are as follows:

- Message Queue System: Kestrel and RabbitMQ
- Publish-Subscribe System: Kestrel and Apache Kafka
- Log System: Flume and Scribe

1.4. Related Works
It was found that the performance of RabbitMQ and ActiveMQ Message brokers had a difference during message transmission and also had performance differences when receiving messages [8]. Evaluation of performance performed by other researchers [10], RabbitMQ was implemented with Advanced Message Queuing Protocol (AMQP) with the concept of High Availability (HA). The study aims to see the performance of the RabbitMQ Cluster nodes and the Mirrored queue system. Also, the Ad-hoc evaluation of functionality and performance has been undertaken. A Pub/Sub System implementation model was created to facilitate Ad-hoc testing of Multiagent System (MAS) [11].

Complete performance testing followed by involving a variety of popular Pub/Sub Systems, such as RabbitMQ (AMQP), Mosquitto (MQTT), Ejabberd (XMPP), and ZeroMQ. It is known that the main features of each Pub/Sub System have similarities, but there are also differences in filtering capabilities, semantic warranty, and encoding. The differences are found to have a significant impact on the throughput and delay of the Cloud-based IoT platform [12].

On the other hand, there is research on the use of RabbitMQ and RESTful APIS as Message-oriented-Middleware for Microservice Web applications, indicating that when the number of users is very much at a time, RabbitMQ able to handle it stably. The RESTful API shows lower performance in the same case [9].

Furthermore, the holistic approach by comparing the main functions of the Pub/Sub System framework has also been researched [6]. RabbitMQ and Kafka are compared and measured according to qualitative value and quantitative value. The results of the RabbitMQ or Kafka implementation selection guidelines are based on the characteristics of the user needs.

Our work has a distinction with Freund, which uses an Ensemble method on Adaboost [14]. Adaboost uses several types of classification, and then the result of every classification will be voted to get the final result. The materials and the methods that we used to do the work are explained in next section.
2. Materials and Methods

2.1. Data Collection

The dataset for this study was gathered from the Detik.com news title. It has satisfy our research requirement; that is classifying news related to Sustainable Development Goals (SDGs) in Indonesia. A set of codes in Python created to run the scrapping/HTML parsing process. We use the request and the BeautifulSoup modules to get certain data from the web page.

Moreover, the flow of the code is visualized in Figure 3. First, the data retrieval time range is determined, followed by access to the news index page Detik.com which is located at https://news.detik.com/indeks. This index page contains a list of news stories published on a given day. The Detik.com index page display is subdivided into multiple pages that are indicated by page navigation at the bottom of the index page. Repetition is needed until the last page to fetch all news title on the daily news index page (Figure 3). The first page contains the most recent news of the day (night time), while the latest page contains the earliest news of the day (early morning), as seen in Figure 4. The details of page navigation can be seen at the bottom of Figure 4, ”≪ 1 2 3 4 5 ... 12 ▷” where ”1” is the first page, and ”12” is the last page.
2.2. Dataset Processing
Labeling manually is a must on Supervised Learning like Naïve Bayes Classifier. For training purpose, the news is collected into a dataset, then we assigned the label manually for each news. There are two types of labels given:

- **yes**: if the news has a relationship (relevant) with the Sustainable Development Goals (SDGs) in Indonesia
- **no**: if the news has no relationship (irrelevant) to the Sustainable Development Goals (SDGs) in Indonesia

Furthermore, data in the form of a text cannot be directly processed by Machine Learning. The text data should be processed first into a vector to be successfully classified by the Naïve Bayes Classifier. We chose TF-IDF vectorizer to handle this task. With TF-IDF every word there will be given a kind of weight/value, then TF-IDF will measure its relevance [15].

2.3. Load-balance Method for Naïve Bayes Classifier
We propose a Load-balance architecture for the Naïve Bayes Classifier. As shown on Figure 5, the Load-balance feature can be implemented by adding a Pub/Sub System to handle the distribution of messages. Every row in the dataset will be processed one by one as a message. Publisher serves as a Message Broker responsible for organizing message delivery, while Subscriber plays a role in receiving messages sent by Publisher. The Publisher will distribute the message evenly to each Subscriber without any redundancy of the message.

The first message will be sent to the first RabbitMQ Subscriber. The second message will be sent to the second RabbitMQ Subscriber. Then, the third message will be sent to the first RabbitMQ Subscriber, and so on until the end of the dataset. Every message that has been received at each Subscriber will be classified directly by the respective Naïve Bayes Classifier into classes as output. To be more specific, the Multinomial Naïve Bayes method is used for the Naïve Bayes Classifier. This method is supported by Scikit-learn library from Python.
Finally, to measure the performance of the classification process, we use this formula to count latency deviation rate:

\[ dv = (1 - \frac{t_1}{t_2}) \times 100 \] (1)

The latency deviation rate \( dv \) is in percent value. \( t_1 \) is the mean latency of Naïve Bayes Classifier with load-balance (in seconds), whereas \( t_2 \) is the mean latency of Naïve Bayes Classifier (in seconds).

3. Results

In the previous section, we describe the materials and the methods used to do the work. While in this section will be shown the results from our experiment.

The news on Detik.com reached hundreds of titles per day. The scrapping process itself is using the news index page (https://news.detik.com/indeks) as a source. Each page of the news index has 20 news titles and also has additional page navigation, information retrieved is obtained from dozens of pages starting from the first page to the last page of page navigation.

We use training data that consists of 600 labeled data to train each Naïve Bayes Classifier. The training dataset distribution was 300 relevant news to Indonesian SDGs and 300 irrelevant news to Indonesian SDGs (Figure 6). The testing process is done using different datasets. The testing dataset consists of 11,741 Detik.com news titles during October 2019 (Figure 7).

Our experiment runs on three Virtual Machine. Virtual Machine number one acts as the Server (RabbitMQ Publisher), which has the testing dataset (Figure 7). Two other Virtual Machine (RabbitMQ Subscriber) act as a pair of Naïve Bayes Classifier (RabbitMQ Subscriber number one and RabbitMQ Subscriber number two) which is doing the training task of training dataset (Figure 6) and the classification task of the testing dataset (Figure 7) that received from the server (RabbitMQ Publisher).

The Naïve Bayes Classifier are trained separately using training dataset (Figure 6) on each Virtual Machine to build a Machine Learning model. The testing dataset will be distributed by the Server (RabbitMQ Publisher) to the RabbitMQ Subscribers on load-balanced style. The first message sent to the RabbitMQ Subscriber number one, the second message sent to RabbitMQ Subscriber number two. Then, the third message send to RabbitMQ Subscriber number one, and so on until the end of the testing dataset. Lastly, we measure the amount time needed (latency) for each classification process scenario (Table 1).
After four testing experiment, we calculate the mean latency of Naïve Bayes Classifier and Load-balance Naïve Bayes Classifier (Table 1). Subsequently, the deviation ($dv$) are calculated using formula that has been explained in the Section 2.3 on Equation (1). As a result of our experiment, the Load-balance Naive Bayes Classifier shows a 30.3 percent time reduction in binary classification latency.

| Classifier                  | Latency #1 (in seconds) | Latency #2 (in seconds) | Latency #3 (in seconds) | Latency #4 (in seconds) | Mean (in seconds) |
|-----------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------|
| Naïve Bayes Classifier      | 8.47                    | 8.85                    | 9.13                    | 9.05                    | 8.88              |
| Load-balance Naïve Bayes Classifier | 6.02                    | 6.23                    | 6.38                    | 6.12                    | 6.19              |

4. Discussion

The focus of this paper is to get an improvement in the processing latency of Naïve Bayes Classification. A combination of a Distributed System (RabbitMQ) and Machine Learning (Naïve Bayes Classifier) could support measurement of the Sustainable Development Goals (SDGs) achievement level in Indonesia. The Detik.com news used for the data source (as training and testing dataset) and successfully categorized into two classes; the relevant to SDGs Indonesia and the irrelevant to the SDGs of Indonesia. Our proposed methods can reduce the amount of classification (latency) time by 30.3 percent.
We use two Subscribers that serve as the Naïve Bayes Classifier, and one Publisher act as a Message Broker (to divide the workload equally). RabbitMQ with the load-balance configuration could prune the processing time of the Naïve Bayes Classifier. Finally, the result of this study brings a positive outcome to overall performance.

5. Conclusion
This study only measures the latency time of the Naïve Bayes classification process. Hence, the accuracy of the classification process has not been measured. Moreover, about the time range, news information on the news index page of the Detik.com is only available from 29 March 2004 until the present. Therefore the news before 29 March 2004 on Detik.com could not be obtained using our methods. The Sustainable Development Goals (SDGs) relevant news from the Detik.com website needs to be classified into 17 classes in the SDGs to become the next research opportunity. The optimum number of subscribers acting as the Naïve Bayes Classifier is beyond our scope, so further studies are required on this matter. The load-balance implementation for another classifier (besides Naïve Bayes) also needs to be researched.

Acknowledgments
The first author gratefully appreciate the financial support of Saintek scholarship from the Indonesian Ministry of Research, Technology, and Higher Education.

References
[1] Arfiani Nur Khusna I A 2018 2018 12th International Conference on Telecommunication Systems, Services, and Applications (TSSA)
[2] Bahri M, Maniu S and Bifet A 2019 2018 IEEE International Conference on Big Data (Big Data) 604–613
[3] Kwon O W and Lee J H 2003 Information Processing & Management 39 25 – 44 ISSN 0306-4573
[4] Chen T, Xu R, He Y and Wang X 2017 Expert Systems with Applications 72 221–230 ISSN 09574174 (Preprint 1404.7828)
[5] Kim J and Hastak M 2018 International Journal of Information Management 38 86–96 ISSN 02684012
[6] Dobbelaere P and Esmaili K S 2017 Industry Paper : Kafka versus RabbitMQ A comparative study of two industry reference publish / subscribe implementations Proceedings of the 11th ACM International Conference on Distributed and Event-based Systems ISBN 9781450350655
[7] Tennant M, Stahl F, Rana O and Gomes J B 2017 Future Generation Computer Systems 75 187–199 ISSN 0167739X URL http://dx.doi.org/10.1016/j.future.2017.03.026
[8] Ionescu V M 2015 2015 14th RoEduNet International Conference - Networking in Education and Research, RoEduNet NER 2015 - Proceedings 132–137
[9] Hong X J, Sik Yang H and Kim Y H 2018 9th International Conference on Information and Communication Technology Convergence: ICT Convergence Powered by Smart Intelligence, ICTC 2018 257–259
[10] Rostański M, Grochla K and Seman A 2014 Proceedings of the 2014 Federated Conference on Computer Science and Information Systems 2 879–884
[11] Nascimento N, Viana C J, Staia A and Lucena C 2017 Proceedings of the 29th International Conference on Software Engineering and Knowledge Engineering 2017 521–526
[12] Happ D, Karowski N, Menzel T, Handziski V and Wolisz A 2017 Annales des Telecommunications/Annals of Telecommunications 72 41–52 ISSN 19589395 URL http://dx.doi.org/10.1007/s12243-016-0537-4
[13] Patro S, Potey M and Golhani A 2017 Proceedings of the 2017 2nd IEEE International Conference on Electrical, Computer and Communication Technologies, ICECCT 2017
[14] Freund Y and Schapire R E 1997 Journal of Computer and System Sciences 55 119–139 ISSN 0022-0000
[15] Ponte J M and Croft W B 1998 SIGIR Forum (ACM Special Interest Group on Information Retrieval) 275–281 ISSN 01635840