Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 6 Issue 22 (September 2021) PP. 15-24
DOI: 10.35631/JISTM.622002

JOURNAL OF INFORMATION SYSTEM AND TECHNOLOGY MANAGEMENT (JISTM)
www.jistm.com

CLASSIFICATION COMPLEX QUERY SQL FOR DATA LAKE MANAGEMENT USING MACHINE LEARNING

Nurhadi[1*], Rabiah Abdul Kadir[2], Ely Salwana Mat Surin[3]

[1] Institute of IR4.0, Universiti Kebangsaan Malaysia, Malaysia
Email: p91334@siswa.ukm.edu.my
[2] Institute of IR4.0, Universiti Kebangsaan Malaysia, Malaysia
Email: rabiahivi@ukm.edu.my
[3] Institute of IR4.0, Universiti Kebangsaan Malaysia, Malaysia
Email: elysalwana@ukm.edu.my
[*] Corresponding Author

Article Info:

Article history:
Received date: 10.06.2021
Revised date: 15.07.2021
Accepted date: 20.08.2021
Published date: 01.09.2021

To cite this document:

Nurhadi, Kadir, R. A., & Mat Surin, E. S. (2021). Classification Complex Query SQL for Data Lake Management using Machine Learning. *Journal of Information System and Technology Management, 6* (22), 15-24.

DOI: 10.35631/JISTM.622002.

Abstract:

A query is a request for data or information from a database table or a combination of tables. It allows for a more accurate database search. SQL queries are divided into two types, namely, simple queries and complex queries. Complex SQL is the use of SQL queries that go beyond standard SQL by using the SELECT and WHERE commands. Complex SQL queries often involve the use of complex joins and subqueries, where the queries are nested in a WHERE clause. Complex SQL queries can be grouped into two types of queries, namely, Online Transaction Processing (OLTP) and Online Analytical Processing (OLAP) queries. In the implementation of complex SQL queries in the NoSQL database, a classification process is needed due to the varying data formats, namely, structured, semi-structured, and unstructured data. The classification process aims to make it easier for the query data to be organized by type of query. The classification method used in this research is the Naive Bayes Classifier (NBC) which is generally often used in text data, and the Support Vector Machine (SVM), which is known to work very well on data with large dimensions. The two methods will be compared to determine the best classification result. The results showed that SVM was 84.61% accurate in terms of classification, and comparatively, NBC was at 76.92%.

Keywords:

Classification, Machine Learning, SVM, NBC.

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 6 Issue 22 (September 2021) PP. 15-24
DOI: 10.35631/JISTM.622002

Introduction

Smart Cities use data lake technology to store data in enormous capacity and there are many types of databases. One of the main and key features of big data technologies is the NoSQL database (Gupta and Giri, 2018), (John and Misra, 2017). NoSQL database is able to store structured, semi-structured, and unstructured data regardless of type or format, with 4Vs features: volume, velocity, variety, veracity (Bengfort and Kim, 2016), (Nguyen and Le-Thanh, 2014). However, in its application, the NoSQL Database has weaknesses, one of which is not being able to fully support the trigger function (Kim and Kim, 2019). Trigger functions are part of a complex SQL query, in that they allow more accurate database searches. The SQL Queries are themselves divided into two types, namely, simple queries and complex queries. Complex SQL is the use of SQL queries that go beyond standard SQL, by using the SELECT and WHERE commands. Complex SQL queries often involve the use of complex joins and subqueries, where the queries are nested in a WHERE clause. Complex SQL queries can be further grouped into two types of queries, namely, Online Transaction Processing (OLTP) and Online Analytical Processing (OLAP).

In implementing complex SQL queries in NoSQL databases, a classification process is required to facilitate the translation process from SQL queries to NoSQL due to the varying data formats, namely, structured, semi-structured, and unstructured data. The classification process aims to make it easier for the query data to fall into the appropriate type of query classification. Complex Query Filtering is a text classification technique that is proving to be a powerful technique for overcoming this problem. Currently, there are several classification methods that are accurate and efficient for classifying data, such as Support Vector Machine (SVM), Naive Bayes Classifier (NBC), Logistic Regression, Decision Tree, and K-Nearest Neighbors. Based on the type, data format, and complexity of the SQL queries, the SVM and NBC methods are considered to be the most accurate and efficient in the complex process of classifying SQL queries (Kusumawati, D'Arofah and Pramana, 2019). As such, it is necessary to know the level of accuracy of each data classification method, namely, the Support Vector Machine (SVM) and the Naive Bayes Classifier (NBC) methods.

Related Work on Classification

Classification here refers to the organization of new data based on training data. Classification is done in data mining to predict Class labels, where classified data is based on training data, and class label values are used in classifying and using attributes when classifying new data (Becker, Moreira and dos Santos, 2017),(Sari, 2017). Classification is the process of considering each instance of a dataset and assigning it to a particular class in normal and abnormal ways. It categorizes data sets into predefined sets (Shilpashree, 2021).

Naive Bayes Classifier (NBC)

NBC is a statistical classifier that can be used to predict the probability of membership of a class. NBC is a probabilistic learning algorithm, and its simplicity is rooted in the assumption that the features of the underlying data are independent of each other (Mocherla, Danehy and Impey, 2017). NBC is proven to have high accuracy and speed when applied to databases with large data (Learning, 2018). The Bayes Theorem is a theorem that refers to the concept of conditional probability (DIxit *et al.*, 2018). The NBC method, however, is a method that can classify text. The advantage of NBC is that the algorithm is simple but has high accuracy

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 6 Issue 22 (September 2021) PP. 15-24
DOI: 10.35631/JISTM.622002

(Arafiyah *et al.*, 2018). Additionally, an alternative form of Bayes Theorem is generally encountered when looking at two competing forces statements or hypotheses:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B|A)P(A)+P(B|A')P(A')} \quad (1)$$

$P(A')$ is the corresponding probability of the initial degree of belief against A, where $P(A') = 1 - P(A)$. For some partition $\{A_i\}$ of the sample space, the extended form of Bayes Theorem is (Zhang and Sakhanenko, 2019):

$$P(A|B) = \frac{P(B|A)P(A)}{\sum_i P(B|A_i)P(A_i)} \quad (2)$$

In the NBC algorithm, each Document (Doc) is represented by a pair of attributes A1, A2, A3, ..., An. where A1 is the first word, A2 is the second word, and so on.

$$P(V_j) = \frac{|Doc_j|}{|Training|} \quad (3)$$

Where | Doc_j | is the number of documents that have category *j* in training. Whereas | training | is the number of documents in the example used for training.

Support Vector Machines (SVM)

The SVM is a supervised classifier that has proven to be very effective in solving a variety of pattern recognition and computer vision problems. Currently, in the era of big data, the machine learning community faces new challenges regarding implementing SVM in real life scenarios resulting from variations in data, volume, velocity, and veracity (Nalepa and Kawulok, 2019), (Sarkar, 2019).

- Linier SVM

Linear SVM separates data in a dimensional input space with the use of a decision hyperplane which is defined as;

$$f(x) : w^T x + b = 0 \quad (4)$$

where w is the hyperplane normal vector, w ∈ RD, and b/ ||w|| is the perpendicular distance positioned such that the distance between the closest vectors of the opposite classes to the hyperplane is maximal.

- Non-linear SVM

Many real-life recognition problems cannot be solved linearly and as such require non-linear decision functions. A kernel trick was introduced to get non-linear hyperplane in SVM (Yuan *et al.*, 2018). It consists in defining a kernel function which must satisfy the conditions that compute the inner product of two feature vectors in deriving non-linear feature space;

$$K(\boldsymbol{a}, \boldsymbol{a}') = \emptyset(\boldsymbol{a})^T \emptyset(\boldsymbol{a}') \quad (5)$$

where $\emptyset : R^D \rightarrow F$ is a vector that maps $\boldsymbol{a}$ from an input to a non-linear (possibly infinitely dimensional) feature space F, in which vectors are linearly separable, and K: R^Dx $R^D \rightarrow R$. The kernel does not require calculating the $\emptyset$ mapping explicitly (the kernel matrix which

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 6 Issue 22 (September 2021) PP. 15-24
DOI: 10.35631/JISTM.622002

contains all of the kernel values computed between every pair of *T* vectors is of **a** *T* x *T* size). The non-linear decision function is as follows (Afifi, GholamHosseini and Sinha, 2020);

$$f(a) = sgn\left(\sum_{i=1}^{t} \alpha i\ yi\ K(x_i^T\ a) + b\right) \quad (6)$$

where α_i is a Lagrange multiplier. To determine the SVM response in a non-linear kernel space, it is not necessary to calculate the mapping ∅ of any vector, given the kernel function K (Nalepa and Kawulok, 2019) to specify one more measure of performance accuracy. This measure is simply the average of the correct predictions over the test set (Bouza, Altieri and Galarza, 2019).

Term Frequency Inverse Document Frequency

Term Frequency Inverse Document Frequency (TF-IDF) Weighting is done after extracting data articles. Assuming TF-IDF weighs down unimportant features, we might get better-performing models (Sarkar, 2019). The formula for finding the weight with TF-IDF is as follows:

$$Wij = tfif\ x\ idf = log\left(\frac{N}{df\,j}\right) \quad (7)$$

Where *Wij* is the weight of the word *i* in article *j*, *N* is the total number of documents, *tfif* is the number of occurrences of the word *i* in document *j*, *dfj* is the number of articles j containing the word *i*. TF-IDF is done so that data can be analyzed using a support vector machine.

Performance Measurement

Performance measurement is done to see the results obtained from the classification. There are several ways to measure performance; some of the ways that are often used involve calculating accuracy, recall, precision, and F-measure.

$$\text{Accuracy} = \frac{\text{Total Classification Correct}}{\text{Number of Test Documents}}\ x\ 100\% \quad (8)$$

$$Recall = \frac{|\{Relevant\ Document\} \cap \{Retrieved\ Document\}|}{\{|Relevant\ Document|\}} \quad (9)$$

$$Precision = \frac{|\{Relevant\ Document\} \cap \{Retrieved\ Document\}|}{\{|Retrieved\ Document|\}} \quad (10)$$

$$F = \frac{2\ x\ Recall\ x\ Precision}{Recall + Precision} \quad (11)$$

Classification of OLAP and OLTP Query

This section presents the research method for Classification Query OLAP and OLTP for Smart City Data Lake Management using Machine Learning. In this study, the research method was implemented based on the framework as shown in Figure 1.

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 6 Issue 22 (September 2021) PP. 15-24
DOI: 10.35631/JISTM.622002

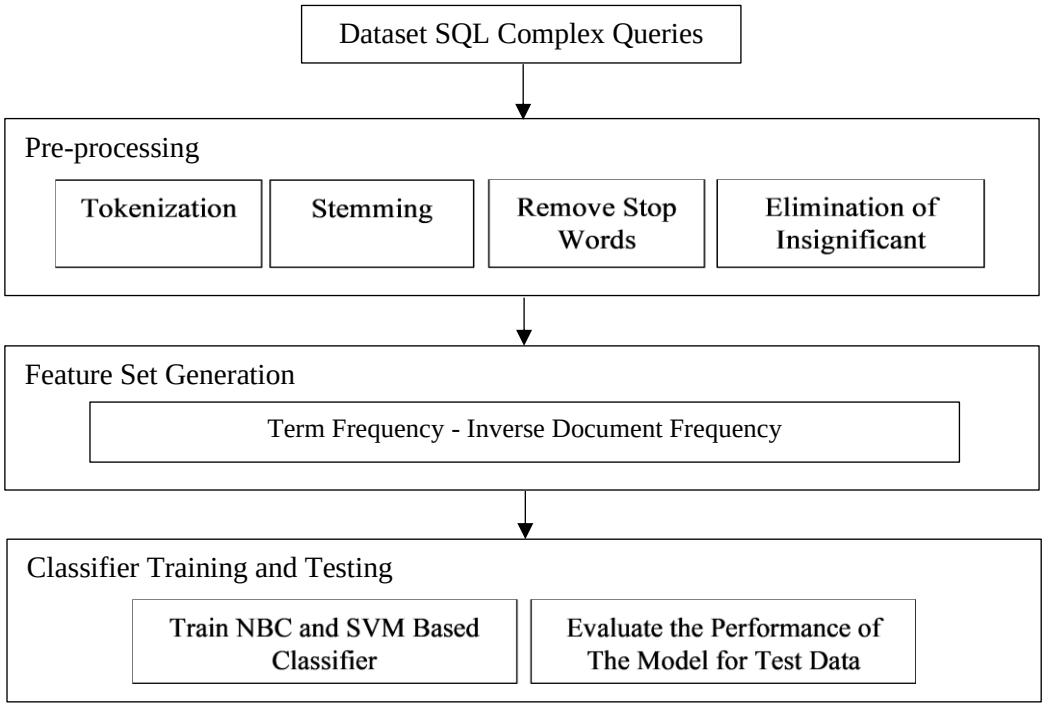

Figure 1: Framework of Research Evaluation

Source: Design Result, 2021

In this experiment, 41 complex SQL queries consisting of 5,000 words were tested with a combination of functions and operations. In this study, the methods used were SVM and NBC methods for classification. Classification is the process of finding a model or function that describes or differentiates a concept or data class with the aim of estimating the unknown class of an object. In general, the classification process has two sub-processes, namely:

1. The training process: the training process uses labeled training sets to build a model or function.
2. Testing process: to see the accuracy of the model or function that will be built in the training process, data called for testing is used to predict labels.

The dataset at the pre-processing stage must go through 4 processes, namely:

a). Tokenization

In this process, words that have punctuation marks, or symbols which are not letters, will be deleted. The system will perform tokenization on the entered query by separating it into single words, where each word represents a token (Approach, 2015).

b). Steeming

The process of removing prefixes and suffixes to form root words.

c). Remove Stop Words

The process of deleting words that are frequently displayed in a document.

d). Elimination of Insignificant

The process of eliminating unnecessary words in a document.

After passing through these four processes, the next stage is the Feature Set Generation, where there is a Frequency-Inverse Document Frequency (TF-IDF) process, which is a process to measure the frequency of appearance of a term in a document. Then the final process is carried out, which is the process of training and testing data to measure the level

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 6 Issue 22 (September 2021) PP. 15-24
DOI: 10.35631/JISTM.622002

of accuracy, performance, and evaluation of complex SQL query data using the SVM and NBC methods.

Results and Analysis

The experimental process that the author utilized used the Python 3.7 programming language with IDE Tools Spyder (Anaconda 3.0), and included the following packages: nltk, re, datetime, nltk.tokenize, nltk.tag, nltk.stem, nltk.corpus, sklearn.preprocessing, sklearn.feature, sklearn.metrics (Uma *et al.*, 2019). For models, the testing is done using a complex query dataset. The test steps to classify complex SQL queries are as follows:

1. Prepare a dataset for experiments with known labels
2. Design an NBC and an SVM algorithm
3. Perform training and testing of the NBC and SVM algorithms, and record accuracy results.

The testing of the SVM and the NBC algorithms obtained accuracy values of 84.61% and 76.92%, respectively. The total number of completed queries was 41 data, consisting of 26 OLTP query data and 15 OLAP query data. The number of word vectors that were tested on the training data were 1000,1500, 2000, 2500, 3000, 3500, 4000, 4500, and 5000.

Table 1: Classification Accuracy (%) NBC Model Formation Using Training Data

No.	Word Vector	Classification Accuracy
1	1000	90.1892
2	1500	92.5129
3	2000	93.2897
4	2500	93.2098
5	3000	94.7896
6	3500	94.8796
7	4000	94.8887
8	4500	94.899
9	5000	94.998

Source: Research Result, 2021

The table above shows that using a word vector of 5000 will produce the best classification level. Classification accuracy tends to increase with an increase in word levels; except for the word vector 2500, where the accuracy has decreased, then continues to increase until the last word vector.

Table 2: Accuracy of Classification (%) NBC Using Data Testing

No.	Word Vector		Classification Accuracy (%)
	Data Training	Data Testing	
1	1000	1000	75.16
2	1500	1500	75.34
3	2000	2000	75.58
4	2500	2500	75.79
5	3000	3000	75.98
6	3500	3000	76.19
7	4000	3000	76.92

Journal of Information System and Technology Management
EISSN: 0128-1666
Volume 6 Issue 22 (September 2021) PP. 15-24
DOI: 10.35631/JISTM.622002

8	4500	3000	76.92
9	5000	3000	76.92

Source: Research Result, 2021

Based on the table above, which shows the highest accuracy for the classification accuracy of complex SQL queries, it can be seen that the classification accuracies for the word vectors 4000, 4500, and 5000 give the best results and produce the same results, namely 76.9230%.

Table 3: Results of Accuracy, Precision, Recall, and F-Measure NBC on Data Testing

No.	Category	Accuracy	Precision	Recall	F-Measure
1	OLTP	76.86%	77.71%	76.88%	76.98%
2	OLAP	76.98%	77.9%	76.95%	76.99%
	Average	76.92%	77.81%	76.92%	76.99%

Source: Research Result, 2021

The results of Accuracy, Precision, Recall, and F-Measure NBC on the data testing table above show a fairly good performance with the respective values of accuracy, precision, recall, and F-Measure being 76.92%, 77.81%, 76.92%, and 76.99%. The OLAP category is the category with the highest level of accuracy, namely 76.98%.

Table 4: Accuracy and Time of Linear SVM Kernel Classification Using Training Data

No.	Word Vector	Classification Accuracy (%)						
		0.01	0.1	1	10	100	1000	5000
1	1000	99.892	100	100	100	100	100	100
2	1500	99.892	100	100	100	100	100	100
3	2000	99.892	100	100	100	100	100	100
4	2500	99.892	100	100	100	100	100	100
5	3000	100	100	100	100	100	100	100
6	3500	100	100	100	100	100	100	100
7	4000	100	100	100	100	100	100	100
8	4500	100	100	100	100	100	100	100
9	5000	100	100	100	100	100	100	100

Source: Research Result, 2021

The table above shows the results for SVM, where using a linear kernel for each word vector in the training data yielded an accuracy value of 100% except for the word vectors 1000, 1500, 2000, and 2500, in the parameter 0.01.

Table 5: SVM Classification Accuracy Using Data Testing

No.	Word Vector		Classification Accuracy (%)	
	Training	Testing	RBF	Linear
1	1000	1000	82.7678	82.5677
2	1500	1500	83.8685	83.7699
3	2000	2000	83.9888	83.8878
4	2500	2500	84.3211	84.3211
5	3000	3000	83.4555	83.4555
6	3500	3000	83.9888	83.8777

Journal of Information System and Technology Management
EISSN: 0128-1666
Volume 6 Issue 22 (September 2021) PP. 15-24
DOI: 10.35631/JISTM.622002

7	4000	3000	83.9989	83.9989
8	4500	3000	84.5079	84.5079
9	5000	3000	84.6134	84.6134

Source: Research Result, 2021

The table above shows the RBF kernel results of word vectors 1000, 1500, 2000, and 3500, which have yielded higher classification accuracy compared to linear kernels. When the number of word vectors is added to 4000 or 5000, the classification accuracy between the RBF and linear kernels measures the same.

Table 6: Results of SVM Accuracy, Precision, Recall, and F-Measure on Data Testing

No.	Category	Accuracy	Precision	Recall	F-Measure
1	OLTP	84.51%	85.61%	84.51%	84.87%
2	OLAP	84.71%	85.81%	84.71%	84.95%
	Average	84.61%	85.71%	84.61%	84.91%

Source: Research Result, 2021

The results of the testing data classification use linear SVM. The table above shows a fairly good performance with the respective values of accuracy, precision, recall, and F-Measure being 84.61%, 85.71%, 84.61%, and 84.91%. The OLAP category is the category with the highest level of accuracy, which is 84.71%.

Table 7: Comparison of Classification Accuracy Between NBC and SVM

No.	Method	Accuracy	Precision	Recall	F-Measure
1	Naive Bayes Classifier (NBC)	76.92%	77.81%	76.92%	76.99%
2	Support Vector Machine (SVM)	84.61%	85.71%	84.61%	84.91%

Source: Research Result, 2021

From the table above, in all ways of measuring performance such as accuracy, precision, recall, and F-Measure, the SVM linear kernel fares better than NBC. Additionally, using the SVM Classification is much faster for obtaining results, as compared to NBC.

Conclusion

After previously obtaining the results and carrying out a discussion on the classification of complex SQL queries for the OLTP and OLAP transaction data categories using the NBC and SVM methods, the results obtained show that SVM has higher accuracy, of about 84.61%, as compared to NBC's 76.92%. As such, for further research on the complex SQL query classification process, the SVM method is the right choice and has a high accuracy value in classifying the OLAP and OLTP categories in the implementation, especially in the NoSQL Database. It is hoped that this SVM machine learning method can provide a great solution for complex SQL classification queries for Data Lake management.

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 6 Issue 22 (September 2021) PP. 15-24
DOI: 10.35631/JISTM.622002

Acknowledgement
The authors would like to express gratitude to the National University of Malaysia (UKM) for providing the opportunity and funding under the Research Project Code TAP-K020558.

References

Afifi, S., GholamHosseini, H. and Sinha, R. (2020) 'FPGA Implementations of SVM Classifiers: A Review', *SN Computer Science*. Springer Singapore, 1(3). doi: 10.1007/s42979-020-00128-9.

Approach, P. (2015) 'Conversion of Natural Language Statement into SQL Query using', *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*. IEEE, (Iceca), pp. 488–491.

Arafiyah, R. *et al.* (2018) 'Classification of Dengue Haemorrhagic Fever (DHF) using SVM, naive bayes and random forest', *IOP Conference Series: Materials Science and Engineering*, 434(1). doi: 10.1088/1757-899X/434/1/012070.

Becker, K., Moreira, V. P. and dos Santos, A. G. L. (2017) 'Multilingual emotion classification using supervised learning: Comparative experiments', *Information Processing and Management*. Elsevier Ltd, 53(3), pp. 684–704. doi: 10.1016/j.ipm.2016.12.008.

Bengfort, B. and Kim, J. (2016) *Data analytics with Hadoop : an introduction for data scientists*.

Bouza, M., Altieri, A. O. and Galarza, C. G. (2019) 'UWB target classification using SVM', *2018 IEEE Biennial Congress of Argentina, ARGENCON 2018*. IEEE, pp. 1–8. doi: 10.1109/ARGENCON.2018.8646072.

DIxit, M. *et al.* (2018) 'Naive Bayes and SVM based NIDS', *Proceedings of the 3rd International Conference on Inventive Computation Technologies, ICICT 2018*. IEEE, pp. 527–532. doi: 10.1109/ICICT43934.2018.9034411.

Gupta, S. and Giri, V. (2018) *Practical Enterprise Data Lake Insights, Practical Enterprise Data Lake Insights*. doi: 10.1007/978-1-4842-3522-5.

John, T. and Misra, P. (2017) *Data Lake for Enterprises.*

Kim, K. J. and Kim, H.-Y. (2019) *Lecture Notes in Electrical Engineering 621 Information Science and Applications*. Available at: http://www.springer.com/series/7818.

Kusumawati, R., D'Arofah, A. and Pramana, P. A. (2019) 'Comparison Performance of Naive Bayes Classifier and Support Vector Machine Algorithm for Twitter's Classification of Tokopedia Services', *Journal of Physics: Conference Series*, 1320(1). doi: 10.1088/1742-6596/1320/1/012016.

Learning, S. (2018) 'LINEAR CLASSIFICATION , PERCEPTRON , LOGISTIC REGRESSION , SVM , NAÏVE BAYES Linear vs non linear classifiers'.

Mocherla, S., Danehy, A. and Impey, C. (2017) 'Evaluation of Naive Bayes and Support Vector Machines for Wikipedia', *Applied Artificial Intelligence*. Taylor & Francis, 31(9–10), pp. 733–744. doi: 10.1080/08839514.2018.1440907.

Nalepa, J. and Kawulok, M. (2019) 'Selecting training sets for support vector machines: a review', *Artificial Intelligence Review*. Springer Netherlands, 52(2), pp. 857–900. doi: 10.1007/s10462-017-9611-1.

Nguyen, T. H. H. and Le-Thanh, N. (2014) *Beyond Databases, Architectures, and Structures, Communications in Computer and Information Science*. doi: 10.1007/978-3-319-58274-0.

Sari, R. (2017) 'Komparasi Algoritma Support Vector Machine, Naïve Bayes Dan C4.5 untuk Klasifikasi SMS', *IJCIT(Indonesia Journal on Computer and Infomation Technology)*, 2(2), pp. 7–13.

Journal of Information System and Technology Management
EISSN: 0128-1666
JISTM
Volume 6 Issue 22 (September 2021) PP. 15-24
DOI: 10.35631/JISTM.622002

Sarkar, D. (2019) *Text Analytics with Python*, *Text Analytics with Python*. doi: 10.1007/978-1-4842-4354-1.

Shilpashree, S. (2021) 'Evaluation of Supervised Machine Learning Algorithms for Intrusion Detection in Wireless Network using KDDCUP ' 99 and NSLKDD datasets', (April 2020).

Uma, M. *et al.* (2019) 'Formation of SQL from natural language query using NLP', *ICCIDS 2019 - 2nd International Conference on Computational Intelligence in Data Science, Proceedings*. IEEE, pp. 1–5. doi: 10.1109/ICCIDS.2019.8862080.

Yuan, Y. *et al.* (2018) 'A Comparative Analysis of SVM, Naive Bayes and GBDT for Data Faults Detection in WSNs', *Proceedings - 2018 IEEE 18th International Conference on Software Quality, Reliability, and Security Companion, QRS-C 2018*, 0, pp. 394–399. doi: 10.1109/QRS-C.2018.00075.

Zhang, Y. C. and Sakhanenko, L. (2019) 'The naive Bayes classifier for functional data', *Statistics and Probability Letters*. Elsevier B.V., 152, pp. 137–146. doi: 10.1016/j.spl.2019.04.017.