International Journal of Informatics Engineering and Computing (IJIMATIC)
Vol. 1, No. 1, 2024

Intrusion Detection System in Network Security Using Naive Bayes and Support Vector Machine

Selamet Riadi[1], Muhammad Nur Fawaiq[2]
[12]Universitas AMIKOM Yogyakarta, Yogyakarta, Indonesia

Abstract
An Intrusion Detection System (IDS) is designed to detect suspicious activities or security threats within a network, necessitating continuous advancements in the field. Both implementation techniques and algorithmic research play pivotal roles in enhancing IDS capabilities. This study addresses this need by focusing on the implementation and comparison of two prominent classification models: Naive Bayes and Support Vector Machine (SVM). The study is centered within the domain of Intrusion Detection Systems (IDS) tailored for network security. In the course of this research, a relevant dataset sourced from Kaggle serves as the foundation for training and testing both classification models. The findings of this study underscore the models' efficacy in intrusion detection. The SVM model, in particular, emerges as a standout performer, showcasing an accuracy rate that approaches 100%, thus exemplifying its potential in real-world scenarios. Meanwhile, the Naive Bayes model delivers commendable accuracy, surpassing 88%. This investigation not only contributes to the advancement of intrusion detection methodologies but also highlights the viability of these classification models for bolstering network security against the ever-evolving threat landscape.

Keywords:
Intrusion Detection System, SVM, Naive Bayes, Machine Learning

1. Introduction

Network security has become crucial in the current digital era. With the continuous advancement of information technology, computer networks have become the backbone of connecting devices and essential resources across various sectors, such as business, government, and education. However, as networks continue to evolve, new security challenges arise, particularly concerning intrusion threats that can compromise the integrity, confidentiality, and availability of data. As a solution to these challenges, various methods and technologies have been developed to detect and prevent potential attacks on networks. One effective approach to addressing this issue is by employing an Intrusion Detection System (IDS). IDS functions as a monitoring tool that observes all events or packets taking place within a computer system or network. It then evaluates each of these occurrences to determine if the activity is harmful or not. In cases where the activity is identified as malicious, appropriate actions are taken based on the potential extent of damage that could result from that particular activity. Thus, IDS becomes a crucial component in network security defense, as it is capable of detecting potential threats and taking appropriate preventive actions to safeguard system integrity and availability [1].

Corresponding Author: Selamet Riadi (selametriadi@students.amikom.ac.id)

Currently, IDS has grown increasingly important, especially for highly active and globally connected communication networks [2]. The Intrusion Detection System (IDS) plays a role in identifying the traffic or data circulating within computer networks, where it can assess whether the traffic is safe, suspicious, or even indicative of an attack. Many researchers have researched Intrusion Detection Systems (IDS) using Machine Learning techniques, which undoubtedly is beneficial for the potential advancement of IDS. Another work utilized IDS (Intrusion Detection System) as one of the methods employed to detect suspicious activities within computer network systems. IDS aids in identifying traffic anomalies on Twitter by employing the Support Vector Machine (SVM) method [3].

A study compared several algorithms including Naïve Bayes, Linear SVM, Polynomial SVM, and Sigmoid SVM. The study analyzed the outcomes of each method compared, based on accuracy values in the confusion matrix, precision, recall, and f1 score. The research successfully concluded that Polynomial SVM achieved the highest accuracy value, while Naive Bayes yielded the lowest accuracy result [5]. Another paper conducted research aimed at implementing the K-Nearest Neighbors (K-NN) algorithm in classifying anomaly network traffic datasets in an Intrusion Detection System (IDS) and comparing it with the Naive Bayes algorithm based on metric parameters of accuracy, sensitivity, and specificity. According to the conducted study, the Naive Bayes algorithm achieved an accuracy rate of 70% [6].

Another paper developed a hybrid intrusion detection system for cloud computing environments using machine learning techniques. This system combines signature-based detection with anomaly detection to enhance accuracy. The research evaluates the proposed approach using the UNSW-NB15 dataset and compares its results with previous studies. K-means Clustering is employed in this research to group data into relevant clusters, achieving an accuracy of 0.886. Additionally, a Support Vector Machine (SVM) is used as the classification model in constructing the anomaly detection system, achieving an accuracy of 0.847 [7]. Another work compares K-Nearest Neighbor and Naïve Bayes methods to optimize the detection of computer network attacks. This research analyzes the comparative methods through the classification process using the confusion matrix and ROC curve. The final results of the study demonstrate that the K-Nearest Neighbor (KNN) method achieves an accuracy level of 99.994%, indicating excellent data classification quality. On the other hand, the Naïve Bayes method only achieves an accuracy of 39.885% [8].

A study explored Chi-square (Chi2) for feature selection and Synthetic Minority Oversampling Technique (SMOTE) for class balance. Results favor XGBoost over other methods. The research presents an Intrusion Detection System (IDS) for Vehicle Ad Hoc Networks (VANET) using the ToN-IoT network dataset. The model incorporates key elements, addressing class imbalance and missing values in ToN-IoT records. Chi2 reduces features to 20, improving training time and model complexity while maintaining the best performance. SMOTE balances classes, mitigating bias, overfitting, and enhancing performance. Evaluation metrics (accuracy, precision, recall, F1-score, FPR, confusion matrix) conclude that XGBoost outperforms all methods for binary and multi-class classification [9].

This research focuses on comparing the usage of popular classification methods in the field of machine learning, namely Naive Bayes and Support Vector Machine (SVM), to detect abnormal or suspicious data communications. With information on the presence of suspicious or abnormal data communications, the system firewall can promptly take preventive action by blocking network activities.

2. Proposed Method

This research is conducted following a pre-designed research framework. The objective is to organize the study and establish the path to be followed, facilitating the achievement of research goals. The research framework can be seen in Figure 1 for a clearer understanding of its structure.

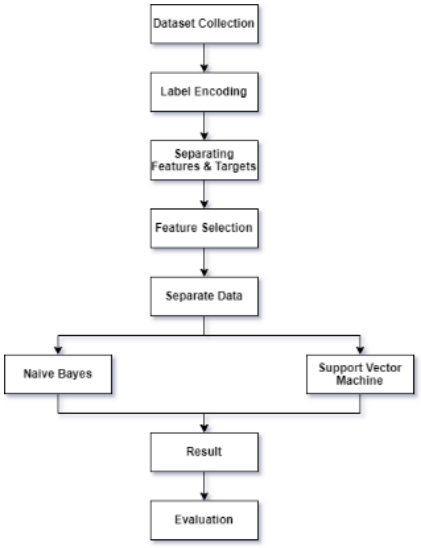

Fig 1. Research Flowchart

1. Dataset Collection

The dataset utilized in this study is a public dataset sourced from Kaggle, known as the "Network Intrusion Detection" dataset. This dataset is specifically tailored for intrusion detection in Vehicular ad hoc networks (VANETs). Its primary purpose is to determine whether network activities are classified as normal or anomalies (intrusions). The "Network Intrusion Detection" dataset is meticulously designed and curated to encompass various types of attacks that could occur within the VANET environment. Thus, it offers a realistic and pertinent representation to evaluate the Intrusion Detection System (IDS) performance for VANETs. Leveraging this dataset provides an opportunity to assess the accuracy of machine learning algorithms in precisely detecting intrusions in autonomous vehicle networks. By utilizing a trusted and pertinent dataset, this research aims to deliver valid and reliable outcomes to address the intricate security challenges within VANET environments.

2. Label Encoding

Label Encoding is the process of converting labels into numeric form [10]. The Label Encoding stage in this research is a part of the preprocessing phase. This process involves transforming string-type labels in the dataset into numerical values. The purpose is to

facilitate inputting numeric data into the employed machine learning algorithms, enhancing their compatibility.

3. Separating Features and Targets

At this stage, the separation of features and targets is performed to prepare the data in a suitable form for machine learning model training. This separation is necessary because machine learning models require inputs in the form of features to learn and discover patterns present in the data. Meanwhile, the target (label) functions as the desired answer, enabling the model to learn how to associate these features with the corresponding target

4. Feature Selection

The pre-data processing technique often utilized in data collection aims to reduce data by eliminating irrelevant attributes. By employing appropriate algorithms, this technique can enhance accuracy in data analysis and modeling [11]. One of the techniques used for feature selection is the Extra Trees Classifier. This technique is an ensemble method based on the Decision Tree algorithm. The Extra Trees Classifier operates by calculating how frequently a feature appears as a splitter in various constructed Decision Trees. Features that frequently emerge as splitters in diverse decision trees will be considered more important.

5. Separate Data

The next step in this research is data splitting. In this stage, the previously prepared data will be divided into two subsets: the training data and the testing data. The ratio is 80% for training data and 20% for testing data. The data splitting is conducted to evaluate and test the machine learning model. The training data is used to train the model and aid it in understanding patterns and relationships between features and targets. On the other hand, the testing data is employed to assess the performance of the trained model and to gauge how well the model can make predictions on unseen data.

6. Naive Bayes

Naive Bayes is an algorithm that exhibits flexibility in accommodating various forms of input data and is also renowned for its high-speed data processing capabilities [12]. The Naive Bayes classifier boasts numerous advantages, including high accuracy, speed, and simplicity [13]. Through the Naive Bayes method, we can effortlessly and swiftly classify data into various categories based on their features. Below is the equation for the Naive Bayes Classifier [14]:

$$\boldsymbol{P(H|X)} = \frac{\boldsymbol{P(H|X).P(H)}}{\boldsymbol{P(X)}} \qquad (1)$$

Information:
X: Data with unknown class.
H: The data hypothesis is class-specific.
P(H|X): Probability of H based on condition X.
P(H): Probability of H.
P(X|H): Probability of X based on the conditions of H.

7. Support Vector Machine

The Support Vector Machine (SVM) is an algorithm that employs nonlinear mapping to transform training data into a higher-dimensional space. In this new dimension, SVM seeks a hyperplane to separate two classes either linearly or with an appropriate nonlinear mapping. Through this mapping, data from two classes can always be separated using the hyperplane [15]. SVM can be regarded as one of the popular algorithms in classification and regression, with the flexibility to handle inherently nonlinear data. Below is the equation for SVM [16] :

$$f(x) = w.x + b$$

Or (2)

$$f(x) = \sum_{i=1}^{m} a_i y_i K(x_i x_i) + b$$

Information:
w: the hyperplane parameter being sought (the perpendicular line between the hyperplane line and the support vector point)
x: Support Vector Machine input data point
ai: weight value of each data point
$K\ (x,\ xi)$: kernel function
b: parameter of the hyperplane being searched (bias value)

4. Result and Analysis

Many The dataset utilized in this research comprises a total of 22,545 instances, wherein there exist 2 classes or labels, namely normal and anomaly. There are 13,449 instances for the normal class and 11,743 instances for the anomaly class. For a clearer insight, please refer to Figure 2.

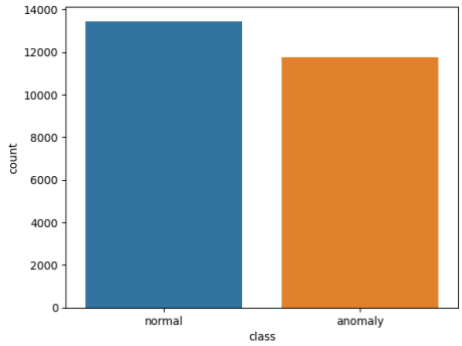

Fig 2. Class Data

3.1. Calculating the Count of Null.

Counting the number of nulls is conducted to identify the presence of missing values in each data column, enabling appropriate actions such as filling empty values with specific data or performing other data preprocessing steps. For further clarification, please refer to Table 1.

Table 1. Number of Null Values

Nama Kolom/attribute	Number of Null Values
duration	0
protocol_type	0
service	0
flag	0
src_bytes	0
dst_bytes	0
land	0
wrong_fragment	0
urgent	0
hot	0
num_failed_logins	0
logged_in	0
num_compromised	0
root_shell	0
su_attempted	0
num_root	0
num_file_creations	0
num_shells	0
num_access_files	0
num_outbound_cmds	0
is_host_login	0
is_guest_login	0
count	0
srv_count	0
serror_rate	0

srv_serror_rate	0
rerror_rate	0
srv_rerror_rate	0
same_srv_rate	0
diff_srv_rate	0
srv_diff_host_rate	0
dst_host_count	0
dst_host_srv_count	0
dst_host_same_srv_rate	0
dst_host_diff_srv_rate	0
dst_host_same_src_port_rate	0
dst_host_srv_diff_host_rate	0
dst_host_serror_rate	0
dst_host_srv_serror_rate	0
dst_host_rerror_rate	0
dst_host_srv_rerror_rate	0
class	0

3.2. Feature Selection

The feature selection in this study employs the Extra Trees Classifier (ETC), a proven algorithm for effectively choosing the best features. The Extra Trees Classifier excels at performing optimal feature selection based on their importance scores, thereby retaining only the most relevant features. The top 4 selected features by the Extra Trees Classifier can be observed in Table 2.

Table 2. Top 4 selected features

no	Feature
1	same_srv_rate
2	dst_host_srv_count
3	logged_in
4	dst_host_same_srv_rate

3.3. Classification Model Performance

The accuracy results obtained in this study using the Naive Bayes and Support Vector Machine algorithms yield different percentage outcomes. For further clarity, please refer to Table 3.

Table 3. Accuracy result

Algorithm	accuracy
Naive Bayes	88.21%
Support Vector Machine	99.09%

In this study, the performance of each defined algorithm is assessed using the Confusion Matrix (CM). For a clearer understanding, the CM for Naive Bayes and SVM can be observed in Figures 3 and 4, respectively.

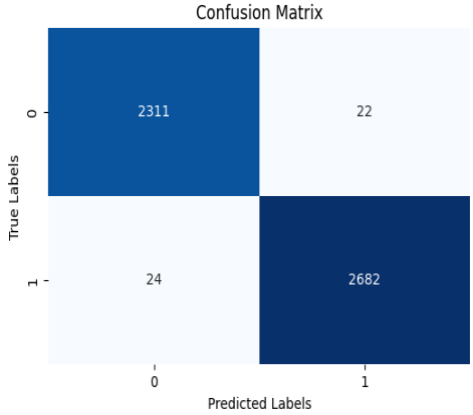

Fig 3. Naive Bayes Confusion Matrix

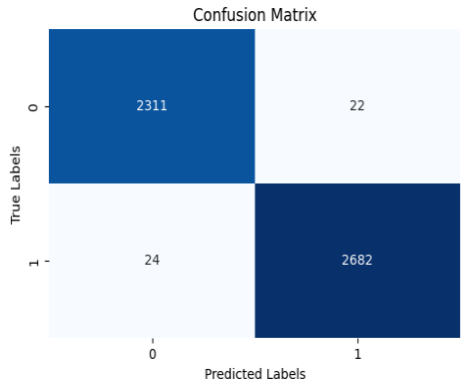

Fig 4. Support Vector Machine Confusion Matrix

Using the classification report evaluation, the results obtained from each of the employed algorithms, namely Naive Bayes and Support Vector Machine, can be seen in Tables 3 and 4, respectively.

Table 4. Classification Report NB

	Precision	Recall	F1-Score	Support
Class 0 (anomaly)	0.88	0.86	0.87	2333
Class 1 (Normal)	0.88	0.90	0.89	2706
Accuracy			0.88	5039
Macro Avg	0.88	0.88	0.88	5039
Weighted Avg	0.88	0.88	0.88	503

The evaluation results from the Naive Bayes method indicate a fairly strong performance in detecting network intrusions. The table presents several evaluation metrics used to assess the model's performance, including Precision, Recall, and F1-Score. Precision measures how accurately the model classifies positive data. The evaluation results show that the model has a precision of 0.88 for both the "anomaly" and "Normal" classes. This means that among the data predicted as "anomaly" or "Normal," approximately 88% of them are correctly categorized in the respective class.

Recall gauges how well the model identifies positive data overall. In the evaluation results, the model has a recall of 0.86 for the "anomaly" class and 0.90 for the "normal" class. This suggests that the model tends to be better at identifying positive data for the "normal" class compared to the "anomaly" class. F1-Score is the harmonic mean of precision and recall, providing an overall picture of the model's performance. The evaluation results show that the model has an F1-Score of 0.87 for the "anomaly" class and 0.89 for the "normal" class. A higher F1-Score indicates a better balance between precision and recall in classifying data.

Furthermore, the overall accuracy of the model is 0.88, representing the percentage of total data correctly predicted by the model. These evaluation results indicate that the Naive Bayes model can provide accurate predictions in detecting network intrusions with a good level of accuracy. However, it's important to note that this evaluation is based on the utilized dataset, and results may differ when applied to other datasets

Table 5. SVM Classification Report

	Precision	Recall	F1-Score	Support
Class 0 (anomaly)	0.99	0.99	0.99	2333
Class 1 (Normal)	0.99	0.99	0.99	2706
Accuracy			0.99	5039
Macro Avg	0.99	0.99	0.99	5039
Weighted Avg	0.99	0.99	0.99	5039

The evaluation results from the Support Vector Machine (SVM) method showcase an exceptional performance in detecting network intrusions. Within the evaluation table, several crucial metrics are employed to assess the SVM model's performance. The SVM model achieves an accuracy of 99.09%, signifying an impressively high level of precision in data classification. This accuracy level indicates that approximately 99.09% of the entire dataset is accurately predicted by the model.

Moreover, the precision and recall values for both the "anomaly" and "Normal" classes are also exceptionally high, reaching 0.99. Precision reflects how accurately the model classifies positive data, while recall signifies how well the model identifies positive data overall. This high precision and recall values indicate that the SVM model delivers remarkably accurate predictions and excels in recognizing positive data.

The F1-Score, a harmonic mean of precision and recall, also attains 0.99 for both classes. This F1-Score portrays the overall performance of the model in providing balanced predictions between precision and recall.

With the exceedingly high levels of accuracy, precision, recall, and F1-Score, it can be concluded that the SVM method excels in detecting network intrusions. This outstanding

performance underscores the reliability of the SVM model in addressing security challenges within complex network environments. Nevertheless, it's important to bear in mind that these evaluation results are based on the dataset used in the research, and the model's performance may vary when applied to different datasets.

4. Conclusion

Neural The conducted research has successfully implemented and compared two classification models, namely Naive Bayes and Support Vector Machine (SVM), for intrusion detection in network security systems. The evaluation results demonstrate that both models exhibit good performance, but SVM stands out with an exceptionally high accuracy level, almost reaching 100%.

The Naive Bayes model achieved an accuracy of 88.21%. Accuracy depicts the overall correctness of the model's predictions, indicating that approximately 88.21% of the data is accurately predicted by the model. The precision for class 0 (anomaly) and class 1 (normal) is 0.88, indicating how accurately the model classifies positive and negative data. The recall for class 0 is 0.86, illustrating how well the model identifies positive data overall. Meanwhile, the recall for class 1 is 0.90. The F1-Score for class 0 is 0.87, and for class 1, it is 0.89. All these evaluation metric values suggest that the Naive Bayes model performs quite well in detecting intrusions within the network data that was utilized.

Meanwhile, the SVM model demonstrates exceptional performance with an accuracy of 99.09%. This figure indicates that around 99.09% of the data is correctly predicted by the SVM model. The precision and recall values for both class 0 and class 1 also reach 0.99, showcasing the SVM model's excellent ability to classify positive and negative data. The F1-Score for both classes reach 0.99, signifying outstanding performance in network intrusion detection.

From these evaluation results, it can be concluded that both models exhibit good performance in intrusion detection. However, the SVM model showcases nearly perfect accuracy, making it a superior choice for intrusion detection in network security. The remarkable performance of the SVM model positions it as a highly promising option for addressing security challenges within complex network environments.

Although the Naive Bayes model demonstrates lower performance compared to SVM, it remains a viable alternative with satisfactory results. The choice of the appropriate model depends on the specific needs and characteristics of the data at hand. This research contributes significantly to enhancing network security by comparing the performance of two distinct classification models and offering valuable insights for further advancements in intrusion detection within network security systems.

For future research, it could be considered to test the SVM model with larger and diverse datasets, as well as identify data classes that might contribute to the data imbalance issue in the Naive Bayes model. Additionally, exploring alternative techniques to enhance model performance could be an intriguing avenue to explore. By continuously developing and refining intrusion detection models, it is anticipated that network security levels can be further elevated to safeguard crucial information and data from potential intrusion threats.

References

[1] R. Patgiri, U. Varshney, T. Akutota, and R. Kunde, “An Investigation on Intrusion Detection System Using

Machine Learning," *Proc. 2018 IEEE Symp. Ser. Comput. Intell. SSCI 2018*, pp. 1684–1691, 2019, doi: 10.1109/SSCI.2018.8628676.

[2] S. Budiman, A. Sunyoto, and A. Nasiri, "Performance Analysis of Feature Selection Usage for Detecting Intrusion Detection Systems with the Random Forest Classifier Algorithm," *Sistemasi*, vol. 10, no. 3, p. 753, 2021, doi: 10.32520/stmsi.v10i3.1550.

[3] A. Prasetyo, L. Affandi, and D. Arpandi, "I Implementation of Naive Bayes Method for Intrusion Detection System (IDS)," *J. Inform. Polinema*, vol. 4, no. 4, p. 280, 2018, doi: 10.33795/jip.v4i4.220.

[4] I. Riadi, R. Umar, and F. D. Aini, "Comparison Analysis of Anomaly Traffic Detection Using Naive Bayes and Support Vector Machine (SVM) Methods," *Ilk. J. Ilm.*, vol. 11, no. 1, pp. 17–24, 2019, doi: 10.33096/ilkom.v11i1.361.17-24.

[5] M. Fluorida Fibrianda and A. Bhawiyuga, "Comparison Analysis of Accuracy in Detecting Attacks on Computer Networks Using Naïve Bayes and Support Vector Machine (SVM) Methods," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 9, pp. 3112–3123, 2018, [Online]. Available: http://j-ptiik.ub.ac.id

[6] A. D. Afifaturahman and F. MSN, "Comparison of K-Nearest Neighbor (KNN) and Naive Bayes Algorithms in Intrusion Detection System (IDS)," *Innov. Res. Informatics*, vol. 3, no. 1, pp. 17–25, 2021, doi: 10.37058/innovatics.v3i1.2852.

[7] I. Aljamal, A. Tekeoglu, K. Bekiroglu, and S. Sengupta, "Hybrid intrusion detection system using machine learning techniques in cloud computing environments," *Proc. - 2019 IEEE/ACIS 17th Int. Conf. Softw. Eng. Res. Manag. Appl. SERA 2019*, pp. 84–89, 2019, doi: 10.1109/SERA.2019.8886794.

[8] M. Iqbal, R. RohmatSaedudin, and M. Fathinuddin, "COMPARATIVE ANALYSIS OF ACCURACY BETWEEN K-NEAREST NEIGHBOR AND NAÏVE BAYES FOR CLASSIFYING COMPUTER NETWORK ATTACK DATA." vol. 9, no. 3, pp. 920–929, 2022.

[9] A. R. Gad, A. A. Nashat, and T. M. Barkat, "Intrusion Detection System Using Machine Learning for Vehicular Ad Hoc Networks Based on ToN-IoT Dataset," *IEEE Access*, vol. 9, pp. 142206–142217, 2021, doi: 10.1109/ACCESS.2021.3120626.

[10] N. Amini, T. H. Saragih, M. R. Faisal, A. Farmadi, and F. Abadi, "Implementation of Genetic Algorithm for Feature Selection in Music Genre Classification Using the Random Forest Method," *J. Inform. Polinema*, vol. 9, no. 1, pp. 75–82, 2022, doi: 10.33795/jip.v9i1.1028.

[11] C. Cahyaningtyas, D. Manongga, and I. Sembiring, "Algorithm Comparison and Feature Selection for Classification of Broiler Chicken Harvest," *J. Tek. Inform.*, vol. 3, no. 6, pp. 1717–1727, 2022, doi: 10.20884/1.jutif.2022.3.6.493.

[12] M. Ridho Handoko and Neneng, "Expert System for Pregnancy-Related Disease Diagnosis Using Naive Bayes Method in a Web-Based Platform," *J. Teknol. dan Sist. Inf.*, vol. 2, no. 1, pp. 50–58, 2021, [Online]. Available: http://jim.teknokrat.ac.id/index.php/JTSI

[13] E. Indrayuni, "Text Mining Classification of Cosmetic Product Reviews in Indonesian Language Using Naive Bayes Algorithm," *J. Khatulistiwa Inform.*, vol. 7, no. 1, pp. 29–36, 2019, doi: 10.31294/jki.v7i1.1.

[14] P. S. M. Suryani, L. Linawati, and K. O. Saputra, "Utilizing Naive Bayes Classifier Method for Sentiment Analysis on Indonesian Language Facebook Posts," *Maj. Ilm. Teknol. Elektro*, vol. 18, no. 1, p. 145, 2019, doi: 10.24843/mite.2019.v18i01.p22.

[15] A. S. Ritonga and E. S. Purwaningsih, "Application of Support Vector Machine (SVM) Method in Classification of SMAW Welding Quality (Shield Metal Arc Welding)," *Ilm. Edutic*, vol. 5, no. 1, pp. 17–25, 2018.

[16] U. Rofiqoh, R. S. Perdana, and M. A. Fauzi, "Sentiment Analysis of User Satisfaction Levels for Indonesian Mobile Telecommunication Service Providers on Twitter using the Support Vector Machine Method and Lexicon-Based Features. Twitter Event Detection - View Project. Human Detection and Tracking - View Project," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 1(12), no. October, pp. 1725–1732, 2017, [Online]. Available: https://www.researchgate.net/publication/320234928