CENTIVE
Conference on Electrical Engineering, Informatics
, Industrial Technology, and Creative Media
ISBN : 978-602-53004-0-0

Comparison of Ensemble Learning Methods on the IoT-23 Dataset

Syakira Az Zahra[1*], Kurnia Anggriani[2], Agus Susanto[3]
[1,2,3]Informatics Department, Universitas Bengkulu
[1,2,3]Jalan WR. Supratman, Kandang Limun, Bengkulu 38229, Indonesia
[*]syakiraazzahra2021@gmail.com

Abstract — The Internet of Things (IoT) has provided numerous benefits across various sectors, but it also poses significant challenges in cybersecurity, particularly malware threats. Malware on IoT devices has the potential to damage systems, steal data, and disrupt network performance. Previous research has shown that the Naïve Bayes algorithm produces a low accuracy of 0.24, increasing slightly to 0.35 when combined with AdaBoost, and reaching 0.99 when combined with XGBoost using the soft voting method. However, there is still room to explore other ensemble learning methods to obtain more stable results. This research focuses on the application of an alternative ensemble learning method, namely stacking, using the IoT-23 dataset with reference to the CRISP-DM framework. The results show that the stacking method can significantly improve malware detection accuracy from 0.35 to 0.72, thus proving superior to soft voting and can be an effective approach in improving malware detection performance in IoT networks.

Keywords – Ensemble Learning, IoT-23, Malware.

I. INTRODUCTION

Malware is malicious code sent over a network with the aim of infecting systems, conducting exploration, stealing information, or carrying out various actions according to the attacker's wishes [1]. Malware can spread and infect multiple computers through various channels, such as email, internet downloads, or infected programs. Its primary goal is to attack user files and replicate itself. As a result, malware can degrade hard disk and software performance, steal critical data, and damage the operating system on the targeted computer [2]. Malware comes in various forms, such as adware, Trojans, ransomware, and others. During the first half of 2022, Kaspersky, a leading antivirus company, detected and stopped 79,442 malware attacks targeting mobile devices in Indonesia. Furthermore, during the first six months of 2022, Indonesia ranked fourth in the world in terms of threats to mobile devices [2]. Malware threats are not only limited to mobile devices, but also include critical infrastructure that is the backbone of a country's operations [3] Malware can also be present in internet networks. The increasing number of malware attacks on IoT networks in recent years shows that interconnected devices are highly vulnerable to cyber threats. This study utilized the IoT-23 dataset as the primary data source. The dataset used is a preprocessed version available open source through the Kaggle platform (kaggle.com/datasets/engraqeel/iot23preprocesseddata), developed by Aqeel Ahmed. Ensemble learning is a method in machine learning that utilizes the training of multiple models, often referred to as "weak learners," to solve the same problem. These models are then combined with the aim of producing better predictions. The basic concept is that the combination of several models with weak performance can form a final model with a higher level of accuracy [4]. The detection utilizes ensemble learning between Naïve Bayes, AdaBoost, and XGBoost with the Soft Voting Classifier method, which is able to improve the performance of low-performance models such as Naïve Bayes. The result of combining the three is 99.9% [5]. However, the ensemble results between Naive Bayes and AdaBoost models tend to be biased toward class 1, and AdaBoost cannot overcome this. Soft voting of the ensemble between Naive Bayes and AdaBoost can cause problems if one of the base models has very low accuracy. Naive Bayes can pull the ensemble predictions toward the more dominant class, resulting in a sparse confusion matrix and lowering the overall accuracy [4]. In the soft voting method, each model predicts probabilities for each class. For example,

Model 1 produces a probability of 70% for class 1 and 30% for class 0. Model 2 produces a probability of 40% for class 1 and 60% for class 0. Model 3 produces a probability of 90% for class 1 and 10% for class 0. The probabilities from each model are then combined by calculating the average. Therefore, this study uses the Ensemble Learning approach with the stacking method to improve the accuracy of malware detection compared to the soft voting method, with the title "Improving the Accuracy of Ensemble Learning in Malware Detection on the IoT-23 Dataset".

A. Ensemble Learning

a) Soft Voting and Hard Voting Soft Voting Classifier, an approach in which predictions from several different models are combined by averaging the probabilities of each model or assigning a specific weight to each model. The final prediction is determined based on the majority of the predictions from the models used. In other words, the ensemble method using soft voting integrates several different classification algorithms to produce more accurate decisions [6]. Hard Voting Classifier is a model with an ensemble method or a combination of several prediction algorithms [7]. Each classifier generates its own prediction, and the ensemble determines the final output based on majority voting. For instance, if three classifiers classify an image as a cat while one classifier identifies it as a deer, the ensemble result will be a cat.

b) Stacking Ensemble stacking is a machine learning method that integrates predictions from several base models through the use of an additional model, known as a meta-learner, to generate the final prediction [8]. The stacking framework operates in two levels: the base learners at level zero and the meta learner at level one. At level zero, various models are trained on the dataset, and their outputs are combined to form a new dataset. In this newly constructed dataset, each instance is aligned with the true target value. The meta learner at level one then processes this dataset to produce the final prediction [9].

B. IoT-23

This research utilizes network traffic data collected from IoT devices. The dataset, known as IoT-23, was created by Sebastian Garcia, Agustin Parmisano, and Maria Jose Erquiaga in collaboration with Avast, and contains real-world examples of both normal and malicious IoT network traffic [10]. It includes 20 malware samples and 3 benign samples. The benign traffic was obtained from actual IoT devices such as the Somfy door lock, Philips Hue, and Amazon Echo, while the malicious traffic was generated using Raspberry Pi. After the IoT-23 dataset produced the .pcap files, these were processed with the Zeek Network Analyzer to generate log files that capture details of communication between endpoints. The .pcap files were then manually examined to determine specific property labels, and a Python script was applied to assign labels based on the analysis. Since the malware files varied widely in size, from just a fewkilobytes to nearly 10 gigabytes, the flow was chosen as the primary unit of analysis [11].

II. RESEARCH METHOD

This study adopts the CRISP-DM (Cross Industry Standard Process for Data Mining) framework, a data mining methodology developed to ensure that data processing is carried out systematically, with welldefined steps, and in an efficient manner [12]. The framework is divided into six phases, namely Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment. The explanation of each phase is described in Fig. 1.

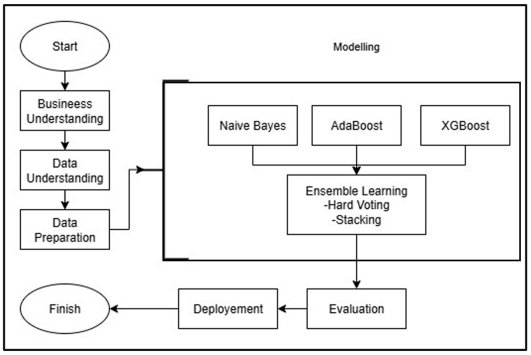

Fig. 1. Method of the research.

This study is a study that aims to show the performance of Ensemble learning techniques on the Naïve Bayes, XGBoost, and AdaBoost Algorithm models in detecting malware on IoT networks in the IoT-23 dataset. This study was chosen because it allows the author to conduct controlled testing of certain variables and analyze their influence on the results. The data in this study are secondary data, namely data collected by others. The data set consisting of data with 21 columns or features about malware and its types was collected from Kaggle. Kaggle is a platform for data science and artificial intelligence that hosts competitions organized by large companies and organizations, often offering cash prizes to participants. CRISP-DM offers a structured and comprehensive approach, making it suitable for use in various industrial sectors. This approach is also in managing and analyzing data, so that it can produce useful insights to support appropriate and strategic decision making.

A. Businesss Understanding

The first stage in the CRISP-DM method used in this study is Business Understanding. This stage aims to understand the background, needs, and objectives of the research in the context of business or real-world problems to be solved. The main problem faced is the increase in malware attacks on Internet of Things

Table 1. One line data

Parameter	Value
ts	1.536.227.023.384.670
uid	CeqqKl3hyLQmO8LK98
id.orig_h	192.168.100.111
id.orig_p	17576.0
id.resp_h	78.1.220.212
id.resp_p	8081.0
proto	tcp
service	-
duration	$3E-06$
orig_bytes	0
resp_bytes	0
conn_bytes	S0
local_orig	-
local_resp	-
missed_bytes	0.0
history	S
orig_pkts	2.0
orig_ip_bytes	80.0
resp_pkts	0.0
resp_ip_bytes	0.0
label	PartOfAHorizontalPortScan

(IoT) networks, which pose risks such as data leaks and damage to the devices used. Along with the rapid development of IoT devices, threats to network security are also increasingly security systems, making them vulnerable to malware infiltration that can potentially lead to data theft, device damage, or unauthorized access to the network. The main objective of this research is to strengthen IoT network security by developing a machine learningbased malware detection system capable of quickly and accurately identifying suspicious activity.

B. Data Understanding

In research, the quality of the dataset plays a vital role in achieving accurate outcomes. The Data Understanding phase focuses on identifying the patterns and structure within the data [13]. The process begins with examining the IoT-23 dataset, which consists of 20 features, one target variable, and a total of 1,048,575 records. IoT-23 is a relatively recent dataset containing network traffic from Internet of Things (IoT) devices, collected between 2018 and 2019 by Avast AIC laboratories in collaboration with CTU University in the Czech Republic. The traffic data was generated using three IoT devices: the Amazon Echo Home, Philips HUE Smart LED, and Somfy Smartdoorlock [14]. At this stage, the researchers examined the description of each feature and target, while also checking for missing or duplicate data. Missing values may not only refer to incomplete records but can also manifest as outliers, inconsistent values, or abnormal entries [15]. Such missing or duplicated data can negatively affect the effectiveness of the model's performance.

C. Data Preparation

At this stage, the dataset is prepared to be ready for the modeling process. The IoT-23 dataset consists of 13 features and 1 non-numeric target, which therefore requires encoding into a numeric format. In addition, the dataset undergoes a balancing process to address class imbalance. Imbalanced data can lead to suboptimal classification performance, as models often place greater emphasis on the majority class during prediction [**?**]. To resolve this issue, an undersampling technique is applied, reducing instances from the majority class. As a result, the original dataset of 1,048,575 entries was reduced to 34,415. The final step in data preparation before entering the modeling phase involved splitting the dataset. From the 34,415 processed entries, 60% (20,649) were allocated for training and 40% (13,766) for testing. This division follows the study Ideal Dataset Splitting Ratios in Machine Learning Algorithms: General Concerns for Data Scientists and Data Analysts by Ismail Olaniyi Muraina, which examined the impact of different splitting ratios on accuracy. The study suggests that for dataset sizes ranging between 100 and 1,000,000, a 60:40 split—60% for training and 40% for testing—is an appropriate ratio [16].

D. Modeling

In this stage, the dataset is initially trained using three individual models: Naive Bayes, a probabilistic classifier based on Bayes' Theorem that assumes feature independence; AdaBoost, a boosting technique that integrates multiple weak learners into a stronger model; and XGBoost, a gradient boosting implementation optimized for speed, regularization, and computational efficiency. Following this, the models are combined in the second stage through a Soft Voting ensemble, continued in the third stage with a Hard Voting ensemble, and finalized with a Stacking ensemble approach.

E. Evaluation

The developed model is evaluated to measure its performance in solving the initially identified problem. The evaluation includes analyzing accuracy, matching the results to business or research needs, and identifying potential deficiencies or unanswered questions.

F. Deployment

After the models are evaluated, the model with the best performance is saved for use in production. This can be done by saving the model to a specific file format (e.g., .pkl for Python).

III. RESULT

A. Ensemble Naïve Bayes + AdaBoost

The results of Ensemble Learning modeling between Naïve Bayes and AdaBoost using the HardVoting method with results of 45% and Stacking & 72%. Figures 2-4 below are each of the recall, precision, and F1 matrices.

B. Ensemble Naïve Bayes + XGBoost

The results of Ensemble Learning modeling between Naïve Bayes + XGBoost using the HardVoting Classifier method are 47% while using stacking is 100%. Figures 5-7 below are each of the recall, precision, and F1 matrices.

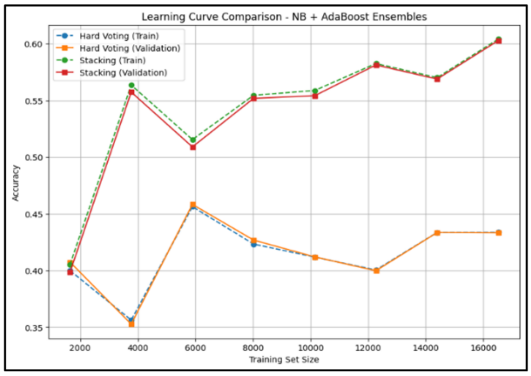

Fig. 2. Learning Curve Ensemble Naive Bayes + AdaBoost.

Classification Report for Stacking Ensemble:

	precision	recall	f1-score	support
0	0.88	0.36	0.52	2749
1	0.52	1.00	0.68	2780
2	0.98	0.89	0.94	2703
3	0.74	1.00	0.85	2724
4	0.84	0.34	0.48	2810
accuracy			0.72	13766
macro avg	0.79	0.72	0.69	13766
weighted avg	0.79	0.72	0.69	13766

Fig. 3. Stacking Matrix of Naive Bayes + AdaBoost.

Classification Report for Hard Voting Ensemble:

	precision	recall	f1-score	support
0	0.99	0.91	0.95	2749
1	0.29	0.99	0.45	2780
2	1.00	0.17	0.29	2703
3	0.41	0.19	0.26	2724
4	0.00	0.00	0.00	2810
accuracy			0.45	13766
macro avg	0.54	0.45	0.39	13766
weighted avg	0.53	0.45	0.39	13766

Fig. 4. Hard Voting Matrix of Naive Bayes + AdaBoost.

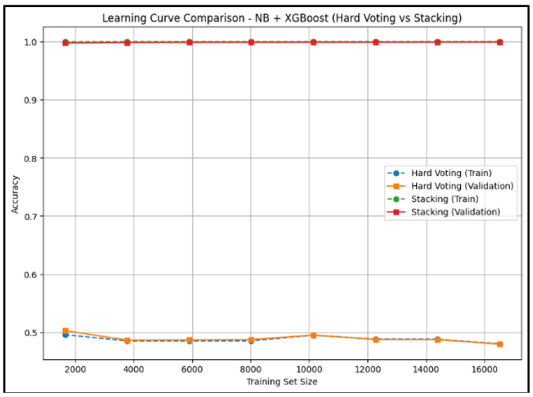

Fig. 5. Learning Curve Naive Bayes + XGBoost.

Classification Report for Hard Voting Ensemble:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	2749
1	0.29	1.00	0.46	2780
2	1.00	0.17	0.29	2703
3	0.47	0.19	0.27	2724
4	0.00	0.00	0.00	2810
accuracy			0.47	13766
macro avg	0.55	0.47	0.40	13766
weighted avg	0.55	0.47	0.40	13766

Fig. 6. Stacking Matrix of Naive Bayes + XGBoost.

Classification Report for Stacking Ensemble:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	2749
1	1.00	1.00	1.00	2780
2	1.00	1.00	1.00	2703
3	1.00	1.00	1.00	2724
4	1.00	1.00	1.00	2810
accuracy			1.00	13766
macro avg	1.00	1.00	1.00	13766
weighted avg	1.00	1.00	1.00	13766

Fig. 7. Hard Voting Matrix of Naive Bayes + XGBoost.

C. *Ensemble AdaBoost + XGBoost*

The results of Ensemble Learning modeling between AdaBoost + XGBoost using the HardVoting Classifier method are 100% while using stacking is 100%. Figures 8-10 below are each of the recall, precision, and F1 matrices.

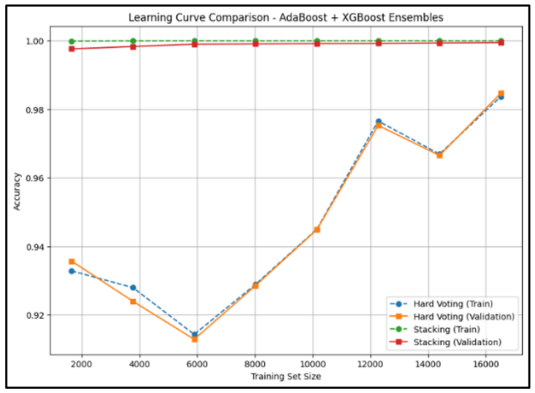

Fig. 8. Learning curve AdaBoost + XGBoost.

Classification Report for Hard Voting Ensemble:

	precision	recall	f1-score	support
0	0.99	1.00	0.99	2749
1	1.00	0.99	0.99	2780
2	1.00	1.00	1.00	2703
3	1.00	1.00	1.00	2724
4	1.00	1.00	1.00	2810
accuracy			1.00	13766
macro avg	1.00	1.00	1.00	13766
weighted avg	1.00	1.00	1.00	13766

Fig. 9. Stacking Matrix of AdaBoost + XGBoost.

Classification Report for Stacking Ensemble:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	2749
1	1.00	1.00	1.00	2780
2	1.00	1.00	1.00	2703
3	1.00	1.00	1.00	2724
4	1.00	1.00	1.00	2810
accuracy			1.00	13766
macro avg	1.00	1.00	1.00	13766
weighted avg	1.00	1.00	1.00	13766

Fig. 10. Hard Voting Matrix of AdaBoost + XGBoost.

D. *Ensemble Naïve Bayes + AdaBoost + XGBoost*

The results of Ensemble Learning modeling between Naïve Bayes +AdaBoost+ XGBoost using the HardVoting Classifier method are 92% while using stacking is 100%. Figures 11-13 below are each of the recall, precision, and F1 matrices.

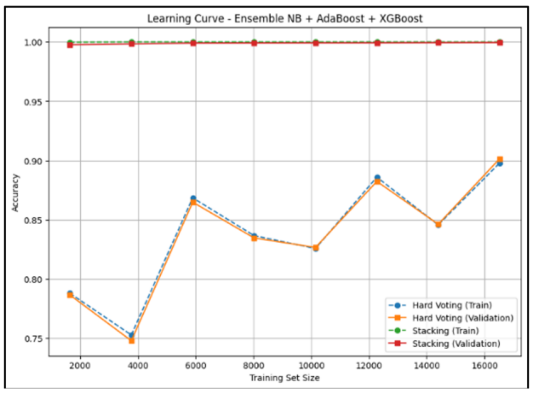

Fig. 11. Learning curve Ensemble Naive Bayes + AdaBoost + XGBoost.

Classification Report for Stacking Ensemble:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	2749
1	1.00	1.00	1.00	2780
2	1.00	1.00	1.00	2703
3	1.00	1.00	1.00	2724
4	1.00	1.00	1.00	2810
accuracy			1.00	13766
macro avg	1.00	1.00	1.00	13766
weighted avg	1.00	1.00	1.00	13766

Fig. 12. Stacking Matrix of Naive Bayes + AdaBoost + XGBoost.

IV. DISCUSSION

The following Table 2 is a modeling results from the 4 ensemble models. The table shows a comparison of the performance of three ensemble learning methods: Hard Voting, Soft Voting, and Stacking, combined with the base models Naive Bayes (NB), AdaBoost (AB), and XGBoost (XGB). Hard Voting tends to have low accuracy (4592%) because the final prediction is determined by a majority vote, so weak models like NB often influence the results. Conversely, Soft Voting achieves very high performance, even reaching 100% in almost all combinations, because it considers prediction probabilities, allowing strong models like AB and XGB to dominate the final result and offset the weaknesses of NB. Stacking, which uses a meta-model to learn patterns between base model predictions, consistently achieves high accuracy (100% in most combinations) because the meta-model is able to determine when to trust a particular model. However, in the NB+AB combination, stacking only achieves 72% because both are relatively weaker than when

Table 2. Accuracy Model

Model	Soft Voting Classifier [?]	Hard Voting	Stacking
Naive Bayes + AdaBoost	35%	45%	72%
Naive Bayes +XGBoost	99%	47%	100%
AdaBoost + XGBoost	99%	100%	100%
Naive Bayes + AdaBoost + XGBoost	99%	92%	100%

Classification Report for Hard Voting Ensemble:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	2749
1	0.71	1.00	0.83	2780
2	1.00	1.00	1.00	2703
3	1.00	0.59	0.74	2724
4	1.00	1.00	1.00	2810
accuracy			0.92	13766
macro avg	0.94	0.92	0.91	13766
weighted avg	0.94	0.92	0.91	13766

Fig. 13. Hard Voting Matrix of Naive Bayes + AdaBoost + XGBoost.

XGB is involved. This study demonstrates that the ensemble stacking method provides better performance than soft voting on the IoT-23 dataset. These findings complement previous studies by providing a new contribution to machine learning-based malware detection. The following Table 3 gives the comparison of this research with the previous research.

Table 3. Comparasion with other research

Author	Year	Dataset	Algorithm
Enhancing Malware Detection in IoT Networks using Ensemble Learning on IoT-23 Dataset [?]	2025	IoT-23	Ensemble Naïve Bayes+ AdaBoost + XGBoost = 99% (Soft Voting Classifier)
Machine Learning for Anomaly Detection in IoT networks: Malware analysis on the IoT-23 Data set [17]	2024	IoT-23	Random Forest = 99.5%
An Efficient Android Malware Prediction Using Ensemble machine learning algorithms [18]	2021	Drebin Dataset	LightGBM = 99.5%
Comparison of ensemble learning methods, on the IoT-23	2025	IoT-23	Ensemble Naïve Bayes+ AdaBoost + XGBoost = 99.9% (Stacking)

ACKNOWLEDGMENT

This research is supported by the Informatics Study Program, University of Bengkulu.

REFERENCES

[1] R. Chiwariro and L. Pullagura, "Malware detection and classification using machine learning algorithms," *International Journal of Research in Applied Science and Engineering Technology*, vol. 11, no. 8, pp. 1727–1738, 2023.

[2] Sharipuddin, R. S. Putra, M. F. Aulia, S. A. Maulana, and P. A. Jusia, "Android security: Malware detection with convolutional neural network and feature analysis," *Media Journal of General Computer Science*, vol. 1, no. 1, pp. 7–13, 2023.

[3] D. Arianyah and I. V. Paputungan, "Jurnal sains, nalar, dan aplikasi teknologi informasi," *Jurnal Sains, Nalar, dan Aplikasi Teknologi Informasi*, vol. 3, no. 2, pp. 50–57, 2024.

[4] M. Kumar, K. Bajaj, B. Sharma, and S. Narang, "A comparative performance assessment of optimized multilevel ensemble learning model with existing classifier models," *Big Data*, vol. 10, no. 5, pp. 371–387, 2021.

[5] K. Anggriani, S. A. Zahra, and A. Susanto, "Enhancing malware detection in iot networks using ensemble learning on iot-23 dataset," *Jurnal Teknik Informatika (Jutif)*, vol. 6, no. 4, pp. 1985–2000, 2025.

[6] S. Joses, S. Quinevera, R. Mardianto, D. Yulvida, and A. M. Shiddiqi, "Pendekatan metode ensemble learning untuk deteksi serangan ddos menggunakan soft voting classifier," *Jurnal Edukasi dan Penelitian Informatika*, vol. 10, no. 1, p. 79, 2024.

[7] M. I. Anugrah, J. Zeniarja, and D. S. Setiawan, "Peningkatan performa model hard voting classifier dengan teknik oversampling adasyn pada penyakit diabetes," *Edumatic: Jurnal Pendidikan Informatika*, vol. 8, no. 1, pp. 290–299, 2024.

[8] A. Munna and E. Zuliarso, "Interpretasi model stacking ensemble untuk analisis sentimen ulasan aplikasi pinjaman online menggunakan lime," *AITI*, vol. 21, no. 2, pp. 183–196, 2024.

[9] A. K. Putri and H. Suparwito, "Uji algoritma stacking ensemble classifier pada kemampuan adaptasi mahasiswa baru dalam pembelajaran online," *KONSTELASI: Konvergensi Teknologi dan Sistem Informasi*, vol. 3, no. 1, pp. 1–12, 2023.

[10] D. Vijayanand and R. K. Singh, "Guardians of iot: Malware analysis of iot devices using machine learning," *TuijinJishu/Journal of Propulsion Technology*, vol. 45, no. 1, pp. 911–924, 2024.

[11] N. Abdalgawad, A. Sajun, Y. Kaddoura, I. A. Zualkernan, and F. Aloul, "Generative deep learning to detect cyberattacks for the iot-23 dataset," *IEEE Access*, vol. 10, pp. 6430–6441, 2022.

[12] M. Ihsan, R. K. Niswatin, and D. Swanjaya, "Deteksi ekspresi wajah menggunakan tensorflow," *Joutica*, vol. 6, no. 1, p. 428, 2021.

[13] F. Abdusyukur, "Penerapan algoritma support vector machine (svm) untuk klasifikasi pencemaran nama baik di media sosial twitter," *Komputa: Jurnal Ilmiah Komputer dan Informatika*, vol. 12, no. 1, pp. 73–82, 2023.

[14] C. V. Oha, F. S. Farouk, P. P. Patel, P. Meka, S. Nekkanti, B. Nayini, S. X. Carvalho, N. Desai, and M. Patel, "Analysis of iot-23 datasets and machine learning models for malicious traffic detection," 2021.

[15] I. J. Fadillah and C. D. Puspita, "Pemanfaatan metode weighted k-nearest neighbor imputation (weighted knni) untuk mengatasi missing data," in *Seminar Nasional Official Statistics*, vol. 2020, pp. 511–518, 2021.

[16] I. O. Muraina, "Ideal dataset splitting ratios in machine learning algorithms: General concerns for data scientists and data analysts," in *7th International Mardin Artuklu Scientific Research Conference*, pp. 496–504, 2022.

[17] N. A. Stoian, "Machine learning for anomaly detection in iot networks: Malware analysis on the iot-23 data set," 2020.

[18] N. Al Sarah, F. Y. Rifat, M. S. Hossain, and H. S. Narman, "An efficient android malware prediction using ensemble machine learning algorithms," *Procedia Computer Science*, vol. 191, pp. 184–191, 2021.