Received June 8th, 2025; accepted June 29th, 2025. Date of publication June 30th, 2025
Digital Object Identifier: https://doi.org/10.25047/jtit.v12i1.448

Efficient Intrusion Detection System Utilizing Ensemble Learning and Statistical Feature Selection in Agricultural IoT Networks

AHMAD FAHRIYANNUR ROSYADY[1], BEKTI MARYUNI SUSANTO[2], AGUS HARIYANTO[3], MUKHAMMAD ANGGA GUMILANG[4]

[1]Jurusan Teknologi Informasi Politeknik Negeri Jember, Jember, Indonesia
[2]Jurusan Teknologi Informasi Politeknik Negeri Jember, Jember, Indonesia
[3]Jurusan Teknologi Informasi Politeknik Negeri Jember, Jember, Indonesia
[4]Jurusan Teknologi Informasi Politeknik Negeri Jember, Jember, Indonesia

CORESPONDING AUTHOR: AHMAD FAHRIYANNUR ROSYADY(email:ahmad.fahriyannur@polije.ac.id)

ABSTRACT To enhance agricultural processes, smart agriculture combines a variety of devices, protocols, computing paradigms, and technologies. The cloud, edge computing, big data, and artificial intelligence all offer tools and solutions for managing, storing, and analyzing the vast amounts of data produced by various parts. Smart agriculture is still in its infancy and lacks several security measures, brought in the creation of numerous networks that are vulnerable to cyberattacks. The most well-known cyberattack is called a denial of service (DoS) attack, in which the attackers overwhelm the network with massive amounts of data or requests, preventing the nodes from accessing the various services that are provided in that network. Intrusion Detection Systems (IDS) have shown to be effective defense mechanisms in the event of a cyberattack. The implementation of conventional intrusion detection systems (IDS) approaches in Internet of Things (IoT) devices was hindered by resource constraints, such as reduced computing capacity and low power consumption. In this paper, we used an ensemble learning and statistical based feature selection strategy to create a lightweight intrusion detection solution. The results show that the stacking ensemble method is able to improve the performance of single machine learning in the classification of anomalous events even though the computation time required is quite large compared to the computation time of single machine learning..

KEYWORDS: Intrusion Detection Systems, Ensemble Learning, Agriculture Internet of Things

I.INTRODUCTION

As technology has advanced, the Internet of Things (IoT) has expanded rapidly. By offering and improving connection that supports the automation parts of various human services, IoT makes people's life easier. IoT is used by millions of networked devices to exchange, collect, and analyze data across several areas [2]. Although the internet plays a significant role in the technological field, it also provides a new opportunity for cybercriminals [3]. There will likely be roughly 41.6 billion linked devices in the IoT ecosystem by 2025, as stated in [4]. Problems with cybersecurity are on the rise due to the exponential growth of integrated devices. Thus, a secure and dependable IoT infrastructure will be achieved by improving security and integrating innovations and artificial intelligence technology.

The three primary components of the Internet of Things (IoT) design are applications, networks, and user experiences. Every operation, from sensor use to data collection, falls under the purview of the Perception layer, which also happens to be the most susceptible to assaults. Attacks involving physical force against infrastructure, sensor-equipped systems, and other such targets are prevalent. Wireless technologies like Wi-Fi, 3G, and 4G enable sensor-equipped devices to communicate and exchange data with gateways and other Internet of Things devices at the Network layer. The Network layer is vulnerable to many types of assaults, the most prevalent of which include DOS, information

theft, gateway attacks, Man in the middle, and Distributed Denial of Service (DDoS) [5].

The Internet of Things faces a new breed of cunning cybercriminals despite relying on the Internet as its primary communication network. The Internet of Things (IoT) consists of a network of interconnected nodes that perpetually exchange information and process data using various network protocols. The vulnerability of these protocols to exploitation considerably endangers the confidentiality of the sent data [6].

In order to identify and thwart these kinds of assaults, researchers have created and uncovered new security methods that rely on AI technologies like ML. Big data settings are no match for ML, unlike firewalls and conventional detection methods [7]. Moreover, machine learning is a recognized approach for tackling attack detection and categorization issues. Machine learning may be regarded as the most appropriate approach to protect and stabilize Internet of Things network traffic [8]. Machine learning encompasses various techniques, including regression and classification [9]. With its arsenal of supervised, unsupervised, and reinforcement approaches and algorithms, ML not only identifies and prevents new attacks, but also offers a suitable strategy to secure IoT networks [10].

Prevention methods, such intrusion detection systems (IDS), are among the newest machine learning approaches. Detecting and reporting on typical or unusual traffic message behavior is the primary role of an intrusion detection system (IDS) [2]. Furthermore, because IoT devices depend on wireless communication protocols, they are susceptible to attacks [11]. Assaults on IoT systems impact all elements of an IoT network, in contrast to local networks, where assaults target specific nodes [12]. Furthermore, the reliability and efficiency of many machine learning algorithms for constructing dynamic intrusion detection systems remain ambiguous. IoT Intrusion Detection Systems have been the subject of considerable investigation, employing numerous machine learning and deep learning methodologies across diverse datasets to assess their effectiveness [13]. An efficient system to prevent attacks on the Internet of Things (IoT) requires time, thus thinking about ways to create and train the system faster is crucial. One way to achieve this is to make the intrusion detection system use less computing power.

The ML model scenario currently exhibits inaccuracies and unsatisfactory results. To address this issue, the ensemble method is implemented, as it is crucial to integrate multiple approaches to mitigate instability [1]. The primary objective of this ensemble is to enhance efficiency through the integration of multiple fundamental machine learning classifiers. When the performance of the machine learning classifier is deemed inadequate, the ensemble approach is employed to amalgamate weak classifiers, thereby constructing a robust prediction model and enhancing performance [14]. Ensemble learning has been utilized across several datasets and is frequently employed to develop intrusion detection methodologies [14].

This research intends to implement ensemble learning in the intrusion detection system of the Internet of Things network. Ensemble learning combines several classifiers to obtain a higher level of accuracy. The combination of several classifiers will cause higher computation time, therefore filter-based feature selection is applied to select attributes that have a low correlation level to the classification class. The dataset used in this study is a public dataset. The filter-based feature selection used in this study is Chi-Square. Chi-square (X^2) feature selection is a statistical technique used primarily in machine learning and data mining to evaluate the dependence between categorical features and a target variable. The goal is to select the most relevant features, reducing dimensionality and improving the model's performance by eliminating irrelevant or redundant information.

II.METHOD

Stacking ensemble is a technique in machine learning that focuses on integrating various algorithms to enhance their overall performance. This method integrates various machine learning algorithms as foundational learners, enhancing overall generalization. The meta learner identifies the optimal approach for managing the predictions generated by the base learners. The foundational models are developed utilizing the initial training dataset. This study employs decision tree, support vector machine (SVM), and naïve bayes as base learners, with logistic regression (LR) serving as the meta learner for classification based on the predictions of the base learners, as illustrated in Figure 1.

The objective of the base learners is to produce the initial prediction and create a new dataset derived from the existing dataset. The objective of the meta learner is to derive the final prediction by utilizing the outputs generated by the base learners. The pseudocode for the stacked ensemble utilizing K-cross validation is outlined in Algorithm 1. The performance of the stacked ensemble algorithm is comparable to that of its constituent algorithms. The Decision Tree method takes into account the underlying structure present in the training data. The training samples are categorized into multiple groups, and each sample undergoes verification following the calculation of the most prevalent separation variables, utilizing specific metrics like information gain value, Gini index, and entropy metric. Furthermore, methods based on Decision Trees demonstrate a minimum of three distinct advantages. The Decision Tree model

is straightforward and relatively simple to implement. Secondly, the Decision Tree-based learning model places minimal emphasis on the sample preparation process, as it can be deemed unnecessary and time-consuming to a certain degree. The primary explanation for this phenomenon is that the Decision Tree-based model considers sample attributes as equally important learning bases, enabling it to manage relatively large data scales effectively. The verification process for assessing the effectiveness and robustness of the Decision Tree model can be conducted with ease [15].

SVM is a supervised classification algorithm commonly employed to address challenges across diverse domains, such as intrusion detection. The primary benefit of SVM lies in its capacity for generalization, particularly when dealing with high-dimensional datasets. Furthermore, SVM demonstrates the capability to manage high-dimensional datasets while maintaining low computational demands.[16]. The stacked ensemble method is recognized for its significant computational demands if not handled with expertise.

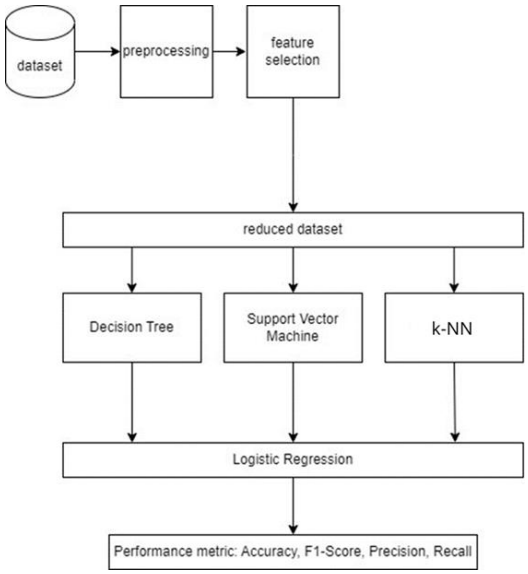

Figure 1 Proposed model.

Figure 1 show the step used in this study. The figure represents a comprehensive machine learning pipeline that begins with a dataset containing raw input data. This data first undergoes preprocessing to prepare it for analysis, which may involve tasks such as data cleaning, normalization, and encoding. Following preprocessing, feature selection is applied to identify and retain the most relevant features, thereby reducing the dimensionality of the dataset and potentially improving the efficiency and accuracy of subsequent models. The resulting reduced dataset is then used to train three different classification models: Decision Tree, Support Vector Machine (SVM), and k-Nearest Neighbors (k-NN). These models operate in parallel, each independently analyzing the refined data. The outputs of these models are then integrated into a Logistic Regression model, suggesting a stacking ensemble approach where Logistic Regression acts as a meta-classifier to combine the strengths of the individual models. Finally, the performance of the overall system is evaluated using standard metrics such as Accuracy, F1-Score, Precision, and Recall, providing a well-rounded assessment of the model's effectiveness in making accurate predictions.

Feature selection is a crucial step in machine learning and data mining, aimed at reducing the dimensionality of data by selecting the most relevant features for model building. One of the widely used methods for feature selection in classification problems is the Chi-Square (χ^2) test. This statistical test assesses the independence between categorical features and the target variable. It helps in identifying features that have a strong relationship with the class labels, allowing models to focus on significant variables and improving performance.

The Chi-square test is a hypothesis testing method that measures the discrepancy between the observed and expected frequencies of categorical variables. The null hypothesis for the test is that the feature is independent of the target class. If the null hypothesis is rejected (i.e., the feature is dependent on the target variable), the feature is considered important for classification. Mathematically, the Chi-square statistic is computed as:

$$\chi^2 = \sum\{(\mathrm{O_i} - \mathrm{E_i})^2\}/\{\mathrm{E_i}\} \qquad (1)$$

where:

- O_i is the observed frequency.
- E_i is the expected frequency.

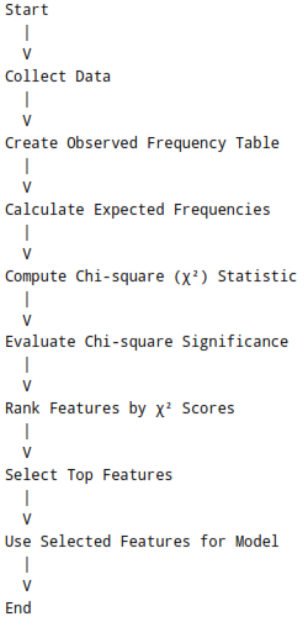

Figure 2. Chi-square step

The Figure 2 show chi-square step used in this study. The Chi-square (χ^2) feature selection flowchart begins with data collection, where the relevant dataset is gathered. Next, an observed frequency table is created to record the actual occurrences of each feature relative to different classes. Expected frequencies are then calculated based on the assumption of independence between features and classes. Following this, the Chi-square (χ^2) statistic is computed to quantify the difference between observed and expected frequencies. The significance of these Chi-square values is evaluated to determine the strength of association between each feature and the target variable. Features are subsequently ranked based on their Chi-square scores, and the top-ranking features, deemed most informative, are selected. These selected features are then utilized in building and improving predictive models, concluding the Chi-square feature selection process.

Accuracy metrics are essential for evaluating the performance of a model. Conversely, its application is limited to scenarios involving evenly distributed data. The calculation involves determining the ratio of accurately predicted events to the overall test sample size. The equation can be mathematically represented in the following manner:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

where:

TP = True Positive
TN = True Negative
FP = False Positive
FN = False Negative

The subsequent assessment involves the precision parameter, which refers to the percentage of predictions deemed accurate that indeed prove to be correct. The ratio of accurately identified positive samples (TP) in relation to the incorrectly classified positive samples (which are still labeled as positive). The equation can be mathematically represented in the following manner:

$$Precission = \frac{TP}{TP+FP} \quad (3)$$

Next is recall, which refers to the proportion of positive samples that have been accurately categorized and identified. The equation is as follows:

$$Recall = \frac{TP}{FN+TP} \quad (4)$$

The F1 score, which typically ranges from 1.0 to 0.0, serves as a metric to assess the harmonic mean of precision and recall. The F-1 score improves with higher levels of accuracy and precision. The equation is as follows:

$$F1\ Score = 2 * \frac{Precission*Recall}{Precission+Recall} \quad (5)$$

III.RESULT AND DISCUSSION

This study aims to apply the stacking ensemble method to detect anomalies in the intrusion detection system. The dataset used in this study is Edge Industrial Internet of Things Dataset [17] that can be accessed publicly from IEEE Dataport website. Compared with the older dataset like N-BaIoT and Bot-IoT, this dataset more comprehensive and newer. The dataset consists of 61 attributes and 1 class. Attributes in network intrusion detection was shown in Table 1.

Table 1 Attributes in network intrusion detection dataset.

Number	Attribute	Type
1	**frame.time**	continuous
2	**ip.src_host**	categorical
3	**ip.dst_host**	categorical
4	**arp.dst.proto_ipv4**	categorical
5	**arp.opcode**	categorical
6	**arp.hw.size**	continuous
7	**arp.src.proto_ipv4**	categorical
8	**icmp.checksum**	categorical
9	**icmp.seq_le**	continuous
10	**icmp.transmit_timestamp**	continuous
11	**icmp.unused**	categorical
12	**http.file_data**	continuous
13	**http.content_length**	continuous
14	**http.request.uri.query**	categorical
15	**http.request.method**	categorical
16	**http.referer**	categorical
17	**http.request.full_uri**	categorical
18	**http.request.version**	categorical
19	**http.response**	categorical
20	**http.tls_port**	categorical
21	**tcp.ack**	categorical
22	**tcp.ack_raw**	categorical
23	**tcp.checksum**	categorical
24	**tcp.connection.fin**	categorical
25	**tcp.connection.rst**	categorical
26	**tcp.connection.syn**	categorical
27	**tcp.connection.synack**	categorical
28	**tcp.dstport**	categorical
29	**tcp.flags**	categorical
30	**tcp.flags.ack**	categorical
31	**tcp.len**	continuous
32	**tcp.options**	categorical
33	**tcp.payload**	continuous
34	**tcp.seq**	categorical
35	**tcp.srcport**	categorical
36	**udp.port**	categorical

Number	Attribute	Type
37	**udp.stream**	categorical
38	**udp.time_delta**	categorical
39	**dns.qry.name**	categorical
40	**dns.qry.name.len**	categorical
41	**dns.qry.qu**	categorical
42	**dns.qry.type**	categorical
43	**dns.retransmission**	categorical
44	**dns.retransmit_request**	categorical
45	**dns.retransmit_request_in**	categorical
46	**mqtt.conack.flags**	categorical
47	**mqtt.conflag.cleansess**	categorical
48	**mqtt.conflags**	categorical
49	**mqtt.hdrflags**	categorical
50	**mqtt.len**	continuous
51	**mqtt.msg_decoded_as**	categorical
52	**mqtt.msg**	categorical
53	**mqtt.msgtype**	categorical
54	**mqtt.proto_len**	continuous
55	**mqtt.protoname**	categorical
56	**mqtt.topic**	categorical
57	**mqtt.topic_len**	categorical
58	**mqtt.ver**	categorical
59	**mbtcp.len**	continuous
60	**mbtcp.trans_id**	categorical
61	**mbtcp.unit_id**	categorical
62	**Attack_label**	categorical

The machine learning used in this study is the decision tree, Support Vector Machine (SVM), K-Nearest Neighbors (KNN) and stacking ensemble method. The selection of the ensemble stacking method is based on the fact that this method is able to improve machine performance. The stacking ensemble method consists of two stages, namely the base learner and the meta learner. The base learner uses the decision tree, SVM and KNN algorithms, while the meta learner uses logistic regression. The stacking ensemble method shown in Figure 2.

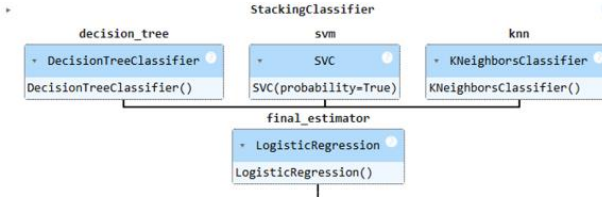

Figure 3 Staking ensemble method.

The classification began with preprocessing followed by applying single classifier. The dataset divides into two parts, namely training and testing data. Training data consist of 80% dataset and testing data consist of 20% dataset. The performance of machine learning in classifying anomalous events in the intrusion detection system before applying feature selection is shown in Table 2.

Table 2 Machine learning performance before feature selection using chi square.

Parameter	Algorithm			
	SVM	Decision Tree	KNN	Stacking Ensemble
Accuracy	97.34 %	97.78%	99.22 %	99.59%
Precision	0.97	0.97	0.99	1,00
Recall	0.97	0.98	0.99	1,00
F1 Score	0.97	0.98	0.99	1,00
Computation Time	1.94 s	0.028 s	0.008 s	118 s

Feature selection is applied to the dataset to increase the speed of computation time without experiencing a significant decrease in accuracy. The feature selection used in this study is a statistical-based feature selection, namely chi square. To perform feature selection, chi square will calculate the correlation between attributes and classes, in this case consisting of 61 attributes and 2 class. Based on the results of the calculation of attribute correlation to class using the chi square formula, 9 attributes were obtained that had a correlation to the class above or equal 0.5. The greater the correlation value, the more important the attribute is in determining the class. The nine attributes are frame.time, tcp.options, tcp.ack.raw, udp.stream, icmp.checksum, icmp.seq_le, tcp.ack, tcp.checksum, and tcp.payload. Furthermore, the dataset that has been applied to feature selection is applied to machine learning. The performance of machine learning after applying feature selection is shown in Table 3.

Table 3 Machine learning performance after feature selection using chi square

Parameter	Algorithm			
	SVM	Decision Tree	KNN	Stacking Ensemble
Accuracy	94.34%	96.76%	99.11%	99.35%
Precision	0.93	0.97	0.99	0.99
Recall	0.95	0.96	0.99	0.99
F1 Score	0.94	0.96	0.99	0.99
Computation Time	1.62 s	0.018	0.025	

Based on the research results shown in table 1 and table 2, it shows that the application of the

stacking ensemble method can improve the performance of single machine learning, both decision trees, SVM and KNN. Based on Tables 1 and 2, it is obtained that stacking ensemble learning has the best performance in classifying anomalous events in the intrusion detection system. However, the time required for the stacking ensemble method to perform training is the largest compared to single machine learning. However, Table 2 shows that the application of feature selection using the chi square statistical method can increase the speed of computation time in stacking ensemble learning by 14%.

The study [1] introduces the creation of an intelligent intrusion detection system (IDS) utilizing deep learning, specifically aimed at tackling the escalating security issues in Internet of Things (IoT) networks. As IoT devices grow more networked, they simultaneously become more susceptible to cyber threats, which traditional security systems, particularly those dependent on rule- or signature-based detection, find challenging to manage, especially in resource-limited contexts. The authors offer an Integrated Intrusion Detection (IID) system that functions independently of particular network protocols and does not necessitate prior knowledge of network traffic signatures or patterns. The proposed Intrusion Detection System (IDS) consists of three primary phases: a network connection phase for interpreting communication protocols, an anomaly detection phase utilizing a deep neural network (DNN) model to classify traffic based on features such as transmission rates, IP addresses, and packet ratios, and a mitigation phase to address identified attacks. The DNN is trained on both benign and malicious traffic and is perpetually enhanced through a feedback loop. The system was assessed through simulated networks and actual IoT testbeds (e.g., Raspberry Pi and TI sensor tags) and juxtaposed with established methodologies such as inverse weight clustering. Experimental findings indicate enhanced precision, recall, and F1-scores in the detection of diverse assaults, such as blackhole, sinkhole, DDoS, wormhole, and opportunistic service attacks. The results validate the practicality and efficacy of deep-learning-based Intrusion Detection Systems for the Internet of Things, even on low-power devices, and indicate the need for future research in identifying more sophisticated assaults and improving adaptability to new threats like zero-day exploits.

This study outlines a versatile machine learning pipeline encompassing data preparation, feature selection, and classification through conventional techniques, including Decision Tree, Support Vector Machine (SVM), and k-Nearest Neighbors (k-NN). The outputs of these classifiers are subsequently integrated by Logistic Regression, and the entire performance is assessed using standard metrics such as accuracy, precision, recall, and F1-score. This methodology is modular and suitable for various classification issues in structured datasets, however it lacks domain-specific expertise. Conversely, the study [1] concentrates only on cybersecurity within IoT networks. It presents an intelligent Intrusion Detection System (IDS) built using a deep neural network (DNN) trained on custom features extracted from network traffic. The system functions autonomously from network protocols, adjusts to emerging risks via a feedback mechanism, and is tailored for implementation on resource-constrained IoT devices. The flowchart-based approach prioritizes simplicity and traditional methodologies, whereas the research article presents a more sophisticated, scalable, and domain-specific solution designed for real-time anomaly identification in the IoT context. The primary distinction resides in the complexity and adaptability of the models—while the flowchart depicts a conventional static model, the article illustrates a dynamic, deep-learning architecture adept at addressing growing security concerns in contemporary IoT systems.

According to [17] Random Forest (RF), Support Vector Machine (SVM), k-Nearest Neighbours (k-NN) and Deep Neural Network (DNN) obtained highest accuracy which achieved 99,99% for binary classification. The feature selection method used wrapped method Random Forest. The study addresses increased security challenges arising from IoT networks' high vulnerability to web attacks, mainly due to the extensive variety and large number of devices compared to traditional computer networks. The authors present a system named EDL-WADS, which comprises four principal modules: feature learning, which converts URL requests into anomaly vectors utilizing methods such as CBOW and TF-IDF; deep learning models including Multi-Resolution Network (MRN), Long Short-Term Memory (LSTM), and Convolutional Neural Network (CNN) for feature extraction; a comprehensive decision module that integrates outputs from the three models through an ensemble classifier based on multilayer perceptron; and a fine-tuning and updating module designed to continuously mitigate emerging web threats. Evaluations conducted with the publicly available CSIC 2010 dataset and real-world data demonstrate high performance of the proposed system in detecting SQL injection and Cross-site scripting (XSS) attacks, showing low false-positive and false-negative rates. Additional research is recommended to enhance detection capabilities for various attack types and to optimize the effectiveness of the CNN model. The results of this study are slightly below which achieved 99,59% even though it uses a different feature selection method, namely the filter method using chi square.

IV.CONCLUSION

The application of the stacking ensemble method is able to improve the performance of single machine learning in the classification of anomalous events even though the computation time required is quite large compared to the computation time of single machine learning. The application of statistical-based feature selection using chi square is able to increase the computation speed of the stacking ensemble method. Although the deep-learning-based intrusion detection system proposed in saome reserach offers a more robust and adaptive solution specifically designed for the dynamic and resource-constrained landscape of IoT networks, this research is more suited for general-purpose applications with structured datasets and stable environments. Future research should focus on enhancing the current intrusion detection framework by expanding its ability to detect a broader spectrum of IoT-specific attacks, including device ID spoofing, Sybil attacks, and RPL-based routing misbehaviors. To improve scalability and privacy, developing a distributed or federated learning-based IDS would allow multiple devices to collaboratively train detection models without exchanging sensitive data.

ACKNOWLEDGMENT

This research was funded by Politeknik Negeri Jember through the research grant program under contract number 782/PL17.4/PG/2024. We would like to express our deepest gratitude to the Director of Politeknik Negeri Jember for the support provided. Our sincere thanks also go to the Center for Research and Community Service (P3M) for their facilitation and guidance throughout the research process. We highly appreciate the dedication and collaboration of the entire research team who contributed to the successful completion of this study.

REFERENCE

[1] C. A. de Souza, C. B. Westphall, and R. B. Machado, “Two-step ensemble approach for intrusion detection and identification in IoT and fog computing environments,” Comput. Electr. Eng., vol. 98, no. January, 2022, doi: 10.1016/j.compeleceng.2022.107694.

[2] H. Tyagi and R. Kumar, “Attack and anomaly detection in IoT networks using supervised machine learning approaches,” Rev. d’Intelligence Artif., vol. 35, no. 1, pp. 11–21, 2021, doi: 10.18280/ria.350102.

[3] G. Thamilarasu and S. Chawla, “Towards deep-learning-driven intrusion detection for the internet of things,” Sensors (Switzerland), vol. 19, no. 9, 2019, doi: 10.3390/s19091977.

[4] Reinsel D., “How You Contribute to Today’s Growing DataSphere and Its Enterprise Impact,” IDC Blog. Accessed: Mar. 22, 2024. [Online]. Available: https://blogs.idc.com/2019/11/04/how-you-contribute-to-todays-growing-datasphere-and-its-enterprise-impact/

[5] Y. Alotaibi and M. Ilyas, “Ensemble-Learning Framework for Intrusion Detection to Enhance Internet of Things’ Devices Security,” Sensors, vol. 23, no. 12, 2023, doi: 10.3390/s23125568.

[6] J. J. Hephzipah, R. R. Vallem, M. S. Sheela, and G. Dhanalakshmi, “An efficient cyber security system based on flow-based anomaly detection using Artificial neural network,” Mesopotamian J. Cyber Secur., vol. 2023, pp. 48–56, 2023, doi: 10.58496/mjcs/2023/009.

[7] C. Luo, Z. Tan, G. Min, J. Gan, W. Shi, and Z. Tian, “A Novel Web Attack Detection System for Internet of Things via Ensemble Classification,” IEEE Trans. Ind. Informatics, vol. 17, no. 8, pp. 5810–5818, 2021, doi: 10.1109/TII.2020.3038761.

[8] S. H. Haji and S. Y. Ameen, “Attack and Anomaly Detection in IoT Networks using Machine Learning Techniques: A Review,” Asian J. Res. Comput. Sci., vol. 9, no. 2, pp. 30–46, 2021, doi: 10.9734/ajrcos/2021/v9i230218.

[9] K. A. P. da Costa, J. P. Papa, C. O. Lisboa, R. Munoz, and V. H. C. de Albuquerque, “Internet of Things: A survey on machine learning-based intrusion detection approaches,” Comput. Networks, vol. 151, pp. 147–157, 2019, doi: 10.1016/j.comnet.2019.01.023.

[10] F. Hussain, R. Hussain, S. A. Hassan, and E. Hossain, “Machine Learning in IoT Security: Current Solutions and Future Challenges,” IEEE Commun. Surv. Tutorials, vol. 22, no. 3, pp. 1686–1721, 2020, doi: 10.1109/COMST.2020.2986444.

[11] M. Hasan, M. M. Islam, M. I. I. Zarif, and M. M. A. Hashem, “Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches,” Internet of Things (Netherlands), vol. 7, p. 100059, 2019, doi: 10.1016/j.iot.2019.100059.

[12] X. Liu, Y. Liu, A. Liu, and L. T. Yang, “Defending ON-OFF attacks using light probing messages in smart sensors for industrial communication systems,” IEEE Trans. Ind. Informatics, vol. 14, no. 9, pp. 3801–3811, 2018, doi: 10.1109/TII.2018.2836150.

[13] A. Khraisat and A. Alazab, “A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges,” Cybersecurity, vol. 4, no. 1, 2021, doi: 10.1186/s42400-021-00077-7.

[14] D. Rani, N. S. Gill, P. Gulia, and J. M. Chatterjee, “An Ensemble-Based Multiclass Classifier for Intrusion Detection Using Internet of Things,” Comput. Intell. Neurosci., vol. 2022, 2022, doi: 10.1155/2022/1668676.

[15] Y. Liu and S. Yang, “Application of Decision Tree-Based Classification Algorithm on Content Marketing,” J. Math., vol. 2022, 2022, doi: 10.1155/2022/6469054.

[16] H. Chen, S. Hu, R. Hua, and X. Zhao, “Improved naive Bayes classification algorithm for traffic risk management,” EURASIP J. Adv. Signal Process., vol. 2021, no. 1, 2021, doi: 10.1186/s13634-021-00742-6.

[17] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras and H. Janicke, "Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning," in IEEE

Access, vol. 10, pp. 40281-40306, 2022, doi: 10.1109/ACCESS.2022.3165809.

AHMAD FAHRIYANNUR ROSYADY was born in jember, East Java Indonesia, in 1992. He Received the Bachelor Degree from Bina Nusantara University Jakarta, in 2017 in Informatic Engineering and the Master Degree From Institut Teknologi Sepuluh Nopember, Surabaya Indonesia, in Technology . Management of Information System His Research Interest Include Information Technology, Internet Of things, Machine Learning, Artificial Intelegent.

BEKTI MARYUNI SUSANTO, was born in Yogyakarta Province, Indonesia, in 1984. He received the Bachelor degree from the Yogyakarta State University, Indonesia in 2010 in Electrical Engineering Education and the Master degree from the STMIK Nusa Mandiri Jakarta, Indonesia, in 2012, in Computer Science. His research interests include cloud computing, internet of things, and machine learning. He can be contacted at email: bekti@polije.ac.id.

AGUS HARIYANTO was born in Jember, East Java Province, Indonesia, in 1978. He received the Bachelor degree from the Institut Teknologi Sepuluh November, Surabaya, Indonesia in 2003 in Informatic Engineering the Master degree from the Institut Teknologi Sepuluh November, Surabaya, Indonesia, in 2011. His research interests include computer network, internet of things and machine learning.

MUKHAMMAD ANGGA GUMILANG holds a Bachelor in Informatics Engineering Education from State University of Malang, Master of Engineering (M. Eng.) from Gadjah Mada University. He is currently lecturing in Information Techology department of State Polytechnic Jember. His research areas in artificiall intelligence, social media analytics and human-computer interaction. He can be contacted at email: angga.gumilang@polije.ac.id