

SMOTE tree-based autoencoder multi-stage detection for man-in-the-middle in SCADA

Freska Rolansa^{1,2}, Jazi Eko Istiyanto¹, Afiahayati¹, Aufaclav Zatu Kusuma Frisky¹

¹Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences, Universitas Gadjah Mada, Yogyakarta, Indonesia

²Department of Electrical Engineering, Pontianak State Polytechnic, Pontianak, Indonesia

Article Info

Article history:

Received May 29, 2024

Revised Oct 22, 2024

Accepted Oct 30, 2024

Keywords:

Anomaly detection

Autoencoder

Multi-class classification

Multi-stage

SCADA

SMOTE

Tree classification

ABSTRACT

Security incidents targeting supervisory control and data acquisition (SCADA) infrastructure are increasing, which can lead to disasters such as pipeline fires or even lost of lives. Man-in-the-middle (MITM) attacks represent a significant threat to the security and reliability of SCADA. Detecting MITM attacks on the Modbus SCADA networks is the objective of this work. In addition, this work introduces SMOTE tree-based autoencoder multi-stage detection (STAM) using the Electra dataset. This work proposes a four-stage approach involving data preprocessing, data balancing, an autoencoder, and tree classification for anomaly detection and multi-class classification. In terms of attack identification, the proposed model performs with highest precision, detection rate/recall, and F1 score. In particular, the model achieves an F1 score of 100% for anomaly detection and an F1 score of 99.37% for multi-class classification, which is preeminence to other models. Moreover, the enhanced performance of multi-class classification with STAM on minority attack classes (replay and read) has shown similar characteristics in features and a reduced number of misclassifications in these classes.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Jazi Eko Istiyanto

Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences

Universitas Gadjah Mada

Yogyakarta, Indonesia

Email: jazi@ugm.ac.id

1. INTRODUCTION

A network comprising of various components responsible for supervising and controlling industrial processes is referred to as supervisory control and data acquisition (SCADA). Using modern technology such as computers, electrical, mechanical systems, and networking devices, SCADA is used in critical infrastructure to monitor physical processes [1]. It encompasses a range of heterogeneous components, including remote terminal units (RTUs), master terminal units (MTUs), human machine interface (HMI), historian, programmable logic controllers (PLCs), sensors, and actuators. The diversity of devices employed by SCADA renders system security maintenance challenging [2]. Incidents like pipeline fires, production process shutdowns, and nuclear reactor outages resulting from SCADA malfunctions underscore its increasingly pivotal role in critical infrastructure operations [3].

Healthcare, energy sector, nuclear reactors, agriculture, transportation, civil, chemical engineering, water plants, and research have widely adopted SCADA [3]. Compared to other sectors, the energy sector is the most targeted for SCADA cyberattacks [4]. Stuxnet is a worm that was discovered in 2010 that targets PLC, which are used in power plants and gas pipelines. The computer worm Stuxnet is a malicious program.

It has the ability to destroy itself in centrifuges at an Iranian uranium enrichment facility [5]. In the United States in 2021, ransomware cyberattacks targeted networked devices managing oil pipeline systems. The suspension of all pipeline operations led to a pandemic in the oil supply. The restoration of the systems required the payment of a ransom of 4.4 million US dollars [6].

Internet-connected SCADA exhibits numerous vulnerabilities, rendering it an increasingly attractive target for cyberattacks [3]. Vulnerability attracts attackers to disrupt SCADA because of the danger it can even cost lives [7], [8]. Network vulnerabilities must also be considered, as they can have negative impacts on businesses and user populations, particularly if the attack targets critical infrastructure used by many, like power system [9]. SCADA handles sensitive information, making the compromise or manipulation of such data a threat to system integrity and user privacy. The three most dangerous threat vectors in SCADA are ransomware, extortion, or other financially motivated crimes, followed by nation-state cyber-attacks, and finally devices and things added to the network [4]. Modbus and DNP3 are widely used protocols in the industry, but they possess security vulnerabilities and risks. This vulnerability is further compounded by legacy control elements like RTU or PLC [1]. Data communication in the Modbus protocol adheres to the structure of the protocol data unit (PDU) with function codes exchanged between the client and server [10].

Incidents caused by attackers can result in physical damage and even casualties. This study is designed for cybersecurity, adhering to the IEC 61850 protocol, primarily deployed in substations. In order to effectively defend against various attacks, the protocol specifications, physical knowledge, and logical behaviour has been employed to construct the intrusion detection system (IDS) [11]. Other works also examined three types of attacks in ICS, namely reconnaissance, false data injection, and replay attacks on the Modbus and S7 protocols [7]. Additionally, attack exploitation on the testbed utilized the Modbus/TCP with decision tree (DT) model, encompasses replay attack, MITM, denial of service (DoS), and reconnaissance [12]. Furthermore, a virtual testbed and documentation has been developed to investigate weaknesses in the Modbus protocol and DoS attacks [13].

Man-in-the-middle (MITM) attacks represent the most significant threat to SCADA networks and can have an impact on network reliability and security, especially in SCADA networks employing the Modbus protocol, owing to the protocol's inherent security limitations [1]. In MITM, the attacker poses as an authentic user between the ends of the communication. These attacks can disrupt Modbus communication protocols, permitting a malicious to pose as a controller and transmit damaging signals to field devices [14]. For instance, MITM attacks on smart grids [9]. MITM extorts victims by using a ransomware pattern. It forges messages from real criminals in order to put more pressure on their managers to make restitution. Furthermore, perpetrator modified the bitcoin address linked to the extortion payment and changed the email message [15]. MITM attack leads to unauthorized control, modifications, or injections prior to the packet reaching its intended destination, thereby disrupting industrial operations.

Some previous study has focused on the protection of SCADA systems. For instance, the detection of adversarial examples by identifying inconsistencies between manifold evaluations and the IDS model inference [16]. Using a filter-based approach [17] and one class support vector machines (OCSVM) [18] can effectively detect cyberattacks in industrial control system (ICS). In addition, the network traffic was classified using neural networks (NN) and decision tree (DT) within the constructed simulation environment. Diverse machine learning (ML) classification algorithms were employed and evaluated to detect Modbus-related threats [19]. Furthermore, a convolutional neural network (CNN) architecture for SCADA networks [20], [21] has been shown to improved the effectiveness of the detection.

There are two types of IDS: signatures (SIDS) and anomalies (AIDS). An extensive methodology comparison between AIDS and SIDS is carried out [22]. The IDS was developed using machine learning approaches [23]–[28], Deep learning [29], [30]. Combining several ML [31]–[33] such as random forest (RF), boosting with extreme gradient (Xgboost) and adaptive (AdaBoost), has proven to be able to detect ransomware and other malicious software [34]. Furthermore, the integration of DT and AdaBoost increases accuracy in detecting fraud [35]. An alternative approach is to implement a dimensional reduction strategy, which enhances the accuracy [25]. In addition, hybrid deep learning techniques [36], including principal component analysis (PCA), spatial clustering using density with noise, particle swarm optimization (PSO), and autoencoder (AE), have been demonstrated to achieve near-perfect accuracy in the development of IDS [37].

Attack types are changing quickly. This makes the public datasets used to train ML models out of date and ineffective against new types of attacks. A further study specifically detects anomalies in ICS by analyzing network packets using the Modbus protocol with the latest Electra dataset. There are two methods to use the ML approach: supervised and unsupervised. Supervised techniques include RF, SVM, and NN. Unsupervised learning techniques include the isolation forest (IF) and the OCSVM. Based on the results, the RF demonstrated the highest precision, while the SVM achieved the highest recall and F1 scores [7]. Another study proposes to identify anomalies in ICS using a combined DNN and generative adversarial network (GAN) model. As a result, the recall metric was 0.98 [38]. Binary class classification is applied to anomaly

detection in several studies, but multi-class classification is only used in research [39]. Furthermore, there are still many detection errors [39], especially for minorities.

Nevertheless, there is a gap in the existing research on multi-class classification, which is only conducted by research on analyzing network packets using the Modbus protocol with the Electra dataset. Furthermore, the number of minority classification errors remains high. Therefore, this research proposes SMOTE Tree-based autoencoder multi-stage detection for man-in-the-middle in SCADA. Our proposed model has four main stages: preprocessing, balancing, autoencoder, and tree classification, which requires sequential execution to detect anomalies and classify multi-classes with preeminence. A tree classification model was developed using optimized hyperparameters and SMOTE-based techniques to handle unbalanced data, specifically to improve the detection and classification of minority attack classes. In addition, by including an autoencoder architecture for the adjustment of the variation in the data prior to the reduction of the dimensionality.

2. METHOD

SCADA systems are used to control large and complex facilities with industrial control processes. The factory comprises SCADA endpoints, which are sensors and actuators. The proposed detection model (STAM) is used to detect attacks during Modbus TCP communication between client and server. A detection model is then developed using the Electra public dataset, which represents the real world of industrial control in SCADA. The proposed model needs to be run in a sequential manner, with each stage follows the previous one. The stages of the proposed model is shown in Figure 1.

The preprocessing stage imports the Electra dataset, and removes redundant data. Then the category data is converted to numerical data using both one hot encoding (OHE) and label encoding. The next step is to normalize the data using standard scaler normalization. The data is then balanced using the synthetic minority oversampling technique (SMOTE). The autoencoder is then used to adjust variation and reduce the dimension of the data. The Electra dataset consists of the training set and the testing set. These needs to be split into 80% training set and 20% testing set. In the training set, five classifiers (SVM, KNN, LR, RF, and DT) are evaluated, and the best is selected. A tree model is made using the DT classifier with hyperparameter optimization. The testing set is carried out by evaluating the model and measuring the performance of anomaly detection and multiclass classification.

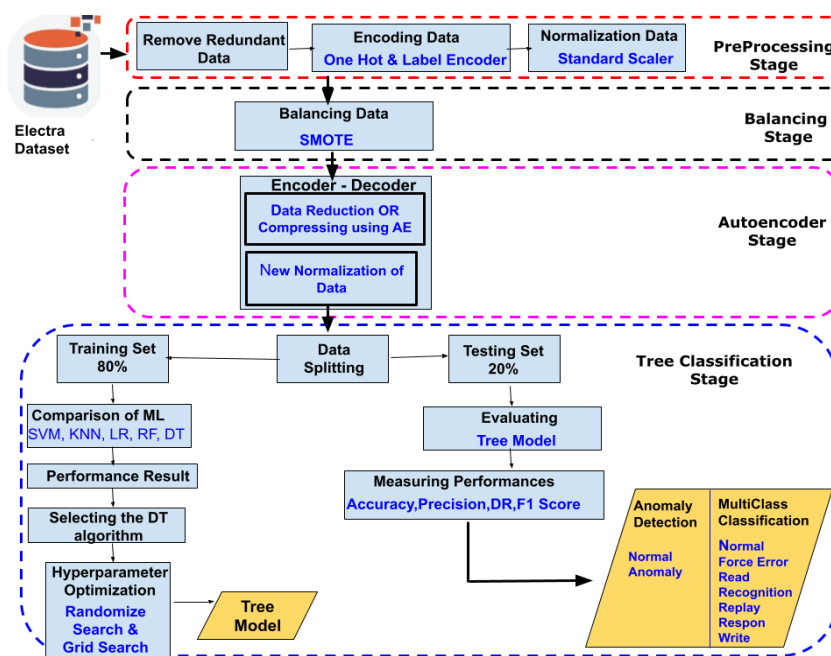


Figure 1. The proposed detection models

2.1. Testbed and dataset

The liquid handling system (LHS) is an ICS testbed applied to the beverage industry and ensures high-quality products are safe for consumption. LHS utilizing PLC controller using CPX-E-CEC-M1 type.

This system has 3 axis X, Y, and Z movements using a stepper motor drive. It uses a conveyor that functions to receive empty bottles and caps and to deliver fully filled bottles. The working system is to fill the liquid into the bottle and then close it to be sent to the robot assembly. PLC is connected to sensors (proximity switch) and actuators (toothed belt, stepper motor, servo drive, mini slide unit, rotary gripper module, parallel gripper, pressure vacuum generator (PVGA), pipette head). Control programming on LHS using CODESYS V3.5 with communication using the Modbus/TCP protocol. The physical testbed is shown in Figure 2.

The dataset used in this research is Electra, which is a recent, realistic, and customized dataset for training machine learning-based IDS models based on network traffic data. It is generated from network traffic in electrical traction substations operating under attack and normal conditions. The Electra dataset is constructed from SCADA and PLC system devices, and it is controlled using the Modbus and S7comm protocols, mirroring real-world scenarios [40]. The 10 attributes of this dataset are categorized into one label, namely MITM unaltered, recognition, read, write, response, force attack, and normal. A full description of the Electra dataset can be found in Table 1.

An attacker must perform a reconnaissance attack using the "function code recognition attack" to obtain information about the target and attack the PLC. False data injection attacks attempt to gain control of control devices in an ICS using control protocols to transmit modified data. These attacks are classified based on the modified data. Spoofed packets attempt 'Read' or 'Write' on the PLC's memory address. 'Response modification attack' or 'force error' via forged slave device packets. 'Command modification attack' through manipulated master device packets. Packets delivered by slave or master devices may have their reception rate altered by 'replay attacks'. Within the Electra dataset, there are 16.289.277 records of network traffic in the Modbus protocol, which encompass data variations consisting of 15.444.940 data records under Normal conditions (normal, MITM unaltered) and 844.337 data records under attack conditions (recognition, read, write, response, force attack and replay attack).



Figure 2. The liquid handling station and servers

Table 1. Overview of the Electra

No	Feature	Description	Data type
1	Time	Time traffic network	string
2	Smac	Originating Mac address	string
3	dmac	Target Mac address	string
4	sip	Originating IP address	string
5	dip	Target IP address	string
6	request	whether or not the request	string
7	fc	Function Code in Modbus	integer
8	error	Displays whether an error	boolean
9	madd	Memory address read/write operations	integer
10	data	Displays data sent or received	integer
11	Label	Class for type attack or Normal	string

2.2. Preprocessing stage

The preprocessing steps applied to the research include eliminating redundant data, encoding categorical data, and normalizing data. The industrial control system dataset, numerous redundant data packages were identified due to repeated executions in multiple control processes. This elimination of redundant data is achieved by disregarding the time feature to identify identical data. For identical data, only the initial data is retained, and the rest is considered redundant and must be removed. There are numerous duplicate data records throughout the dataset.

The second step involves encoding. Some fields have been modified to perform categorical data conversion using both OHE and Label Encoding. OHE is employed for categorical data that lacks a sequential relationship, such as the smac, dmac, sip and dip. Additionally, OHE converts categorical data to integers, with values ranging from 0 to 1, utilizing a fixed number of dimensions. On the other hand, categorical data, which exhibits little or no sequential relationship, is encoded using Label Encoding. For instance, this applies to attributes such as fc, madd, and data.

Normalization is applied to ensure that all remaining features in the dataset fall within the same range as the last step of the first phase. The standard scaler normalization method is employed for this purpose. Rescales the distribution of values so that the mean of the observed values is 0 and the standard deviation is 1, thus reducing the differences in the features. This process is applied to training and test data during the development of the classification model. The standard scaler normalization is given in (1).

$$X = \frac{X_i - X_{\text{mean}}}{X_{\text{std}}} \tag{1}$$

Where,

X - normalized data, X_i - input value, X_{mean} - feature mean, and X_{std} - feature standard deviation.

2.3. Balancing stage

The second phase focuses on creating balanced data. The Electra is an unbalanced data set that shows a small number of attacks compared to the large number of normal classes. Oversampling the minority class is one approach to dealing with unbalanced datasets. The simplest approach is to duplicate examples in the minority class. The SMOTE technique [41] is applied in this work. Let m be the oversampling rate, meaning that each minority sample will be oversampled m times, n being the total number of minority samples. When X_i is a minority sample, $1 \leq i \leq n$ These are the steps that SMOTE will take in order to create m new samples based on X_i .

- First step: apply the (K-Nearest Neighbour) KNN to X_i (belonging to the minority sample) to find the set R_i of k minority samples closest to X_i .
- Second step: a minority sample X_j is arbitrarily selected from R_i , and a new synthesized sample X_{new} is generated based on (2).

$$X_{\text{new}} = X_i + w (X_j - X_i) \tag{2}$$

where w is a value between 0 and 1 that can be chosen at random.

- Third step: if the number of new samples synthesised based on X_i is less than the oversampling rate, proceed to step 2.

In imbalanced classification tasks, the minority class is usually the most important. By synthesizing new examples from the minority class, the SMOTE technique is used to increase the number of examples from the minority. This allows the model to outperform the majority class in predicting the class or probability of the minority class. Figure 3 shows a comparison of the number of attack class distributions on the Electra dataset before and after applying the SMOTE method. Figure 3(a) shows the imbalanced class distribution before SMOTE, and Figure 3(b) the balanced class distribution after SMOTE.

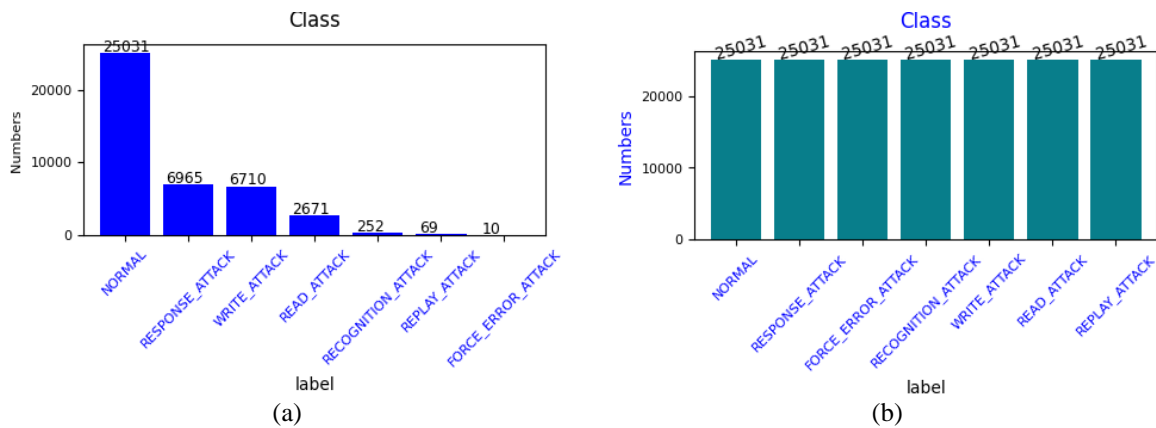


Figure 3. The number of attack class distributions (a) before SMOTE and (b) after SMOTE

2.4. Autoencoder

An autoencoder is a type of neural network architecture that consists of an encoder and a decoder to encode input data to essential features. The original input is rebuilt from the compressed representation. The encoder derives features from raw data, and the decoder rebuilds the data using these features. The extracted features allow the decoder to reconstruct the data. The AE architecture consists of input, latent, and output layers connected between neurons.

$$y = \alpha(n \cdot x^T + b) \quad (3)$$

Where y is vector output, x is vector input, b is a bias value, n is the vector of neuron connection weights, α is activation function, and x^T is the transpose of the input vector x . The structure of AE is shown in Figure 4.

The autoencoder stage is performed by adjusting the variation of the data before dimension reduction. Variation is crucial to the classification process. Table 2 depict the parameters in anomaly detection and multi-class classification.

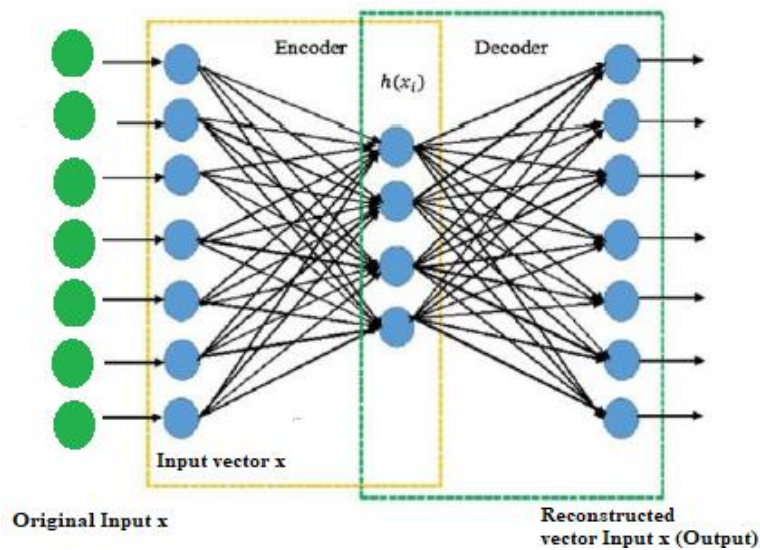


Figure 4. The structure of autoencoders

Table 2. Model parameters for autoencoder

Model architectures		Anomaly	Multi-Class
Encoder	Dense 1	64	128
	Dense 2	32	64
Decoder	Dense 1	32	64
	Dense 2	64	128
Epoch		20	50
Batch size		50	100
The optimizer		Adadelata	Adadelata
Activation function of each layer		ReLU	ReLU
Activation function of output layer		SoftMax	SoftMax

2.5. Tree classification

The training process starts in the third phase which is done by splitting the data into training and testing. Specifically, 80% of the data is used to train, and the remaining 20% is used to test, randomly assigned. In this study, 20% of records from the benign class were randomly selected for testing, while the remaining 80% were utilized for training. Additionally, to maintain dataset balance in terms of attack classes, 20% of the samples from each attack class are set aside for testing, and the remaining 80% are employed for training. Subsequently, all the 20% segments are consolidated to construct the test set, and the same procedure is applied to the training set. The selection of classification algorithms is based on the results of experimental work on several classifiers, namely DT, KNN, SVM, LR, and RF. The results of multi-class classification experiments show that RF and DT algorithms have the highest accuracy results. On the basis of

the results of the experimental tests on the Electra dataset, it was found that the RF algorithm has the disadvantage of showing more detection faults than the DT. In addition, RF algorithms have difficulty interpreting data, which is more ambiguous for the classification process. Therefore, in this work, DT algorithm with hyperparameter optimization is adopted for models. Parameter tuning is conducted on the DT anomaly detection and multi-class classification. Random and grid searches are combined and used to select the best hyperparameter values. In Table 3, the values selected for classification are marked with asterisks and in bold. Based on experiments comparing the results of the best classifiers, we select the parameters of the grid search. The process of multi-class classification for dataset is described in Algorithm 1.

Table 3. Hyperparameter tuning

Hyperparameters	Randomize search	Grid search
Maximum depth of tree	None, 2, 4, 6, 8* , 10	3, 5, 7, 8* , 10
Minimum number of samples to a split	2* , 5, 10	2* , 4, 5, 7
Minimum number of samples to be at a leaf node	1, 2, 4*	1, 2, 3* , 4

Algorithm 1. The pseudocode of tree detection with hyperparameter optimization

```

Input X:   time, smac, dmac, sip, dip, request, fc,error, madd, data
Output O: Normal, Recognition attack, Read attack, Write attack, Responses attack,
          Force error Attack.
Function TreeDetection(Sample D, Input X, Output O, Hyperparameters H):
If stopping_condition(D, X) is true then
    Leaf = createNode()
    leafLabel = classify(D, O)
    Return Leaf
Root = createNode()
Root.test_condition = findBestSplit(D, X, H)
Z = {z | z is a potential outcome of Root.test_condition}
For each value z in Z:
    Subclass = {d | Root.test_condition(d) = z and d is in D}
    Child = TreeDetection(Subclass, X, O, H)
    Add Child as a child of Root and label the edge
    {Root → Child} as z
Return Root

```

2.6. Evaluation metrics

The quality of a machine learning model or algorithm is determined by a parameter called the evaluation matrix. Since Electra dataset used has imbalanced data, precision, recall/detection rate (DR), and F1 score were selected as metrics in the performance evaluation. These metrics are defined in (4), (5) and (6). The accuracy metric was not adopted in the model evaluation.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (4)$$

$$\text{Recall/DR} = \frac{TP}{TP+FN} \quad (5)$$

$$\text{F1 Score} = \frac{2 * (\text{Precision} * \text{Recall})}{\text{Precision} + \text{Recall}} \quad (6)$$

Where TP is true positive, TN is true negative, FP is false positive, and FN is false negative.

3. RESULTS AND DISCUSSION

This section includes the performance and comparison of the proposed models, a detailed analysis and discussion.

3.1. Performance evaluation

Communication using Modbus protocol is used by PLC to control pressure (output) and valve (PGVA setting) parameters with single register writing applying function code (06). In addition, function code (04) is applied to read input registers on the actpress parameter for the actual output pressure of PGVA and the actpress tank to know the actual pressure of the PGVA tank. PLC is connected to TP-LINK switch via Ethernet with IP address 192.168.0.102/24. In addition, there are SCADA and IDS systems that use Dell Power Edge R250 servers with IP address ranges 192.168.0.100/24 and 192.168.0.99/24. Cybersecurity

attacks are mostly carried out on SCADA. Attacker PC is used as an attacker to perform MITM attacks with IP Range 192.168.0.103/24. IDS is connected to the switch to capture all network activities that occur in the test bed. Furthermore, TP link routers are used to connect the testbed to an open network or the internet. Finally, the attacker's device can connect through the switch via Ethernet cable or router via the internet.

The model achieves perfect anomaly detection with 100% precision, recall/DR and f1 score. Additionally, the multi-class classification of MITM with the model detection framework achieves an f1 score of 99%. The anomaly detection and multi-class classification report is displayed in Table 4.

Table 4. The performance of proposed model

Binary Class	Precision	DR	F1 Score	Supp	Multi-Classes	Precision	DR	F1 Score	Supp
Normal	1.00	1.00	1.00	4900	Normal	1.00	0.98	0.99	4937
					Force Error Attack	1.00	1.00	1.00	5066
					Read Attack	1.00	0.98	0.99	4835
					Recognition Attack	1.00	1.00	1.00	5049
Anomaly	1.00	1.00	1.00	4864	Replay Attack	0.98	1.00	0.99	4973
					Response Attack	0.98	1.00	0.99	4987
					Write Attack	1.00	1.00	1.00	4978
					Accuracy		1.00	9764	Accuracy

For multiple classes, including normal, force error, read, recognition, replay, response, and write. Without using any external training data, the results show that the proposed approach can provide superior classification results for MITM multi-class classification. Figure 5 demonstrates the confusion matrix (CM) results of the proposed STAM model. Figure 5(a) anomaly detection's CM and Figure 5(b) multi-class classification's CM.

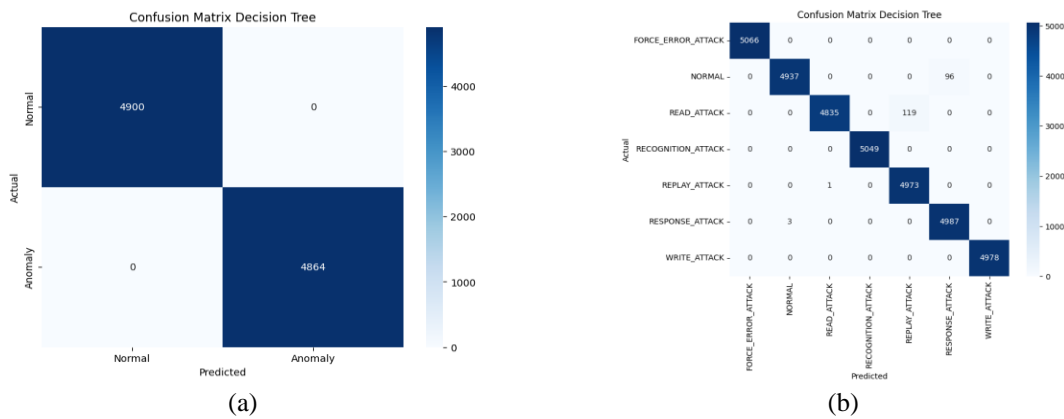


Figure 5. CM proposed models (a) anomaly detection and (b) multi-class classification

3.2. Comparative analysis

Comparative analysis was conducted on anomaly detection and multi-class classification of the proposed model with other models. Table 5 compares the evaluation results of different models, including RF, SVM, NN, OCSVM, IF, GAN + DNN, DAE + SMOTE+T-Link +XGboost and the proposed model.

Table 5. Comparison methods for anomaly detection

Method	Precision	DR	F1 Score
RF[7]	98,77	98,71	98,74
SVM[7]	97,56	100	98,76
NN[7]	96,92	100	98,43
OCSVM[7]	98,62	98,56	98,59
IF[7]	87,39	100	93,27
GAN + DNN[38]	-	98	-
DAE + SMOTE +			
T-Link+ XGBoost [39]	100	100	100
Proposed model	100	100	100

The Electra dataset is used for this comparison, and the anomaly detection results are evaluated using precision, DR, and F1 scores. It can be seen that the performance of the proposed model is equal to model DAE + SMOTE+T-Link + XGboost and higher than the other methods. The proposed anomaly detection model achieves precision, detection rate, and an F1 score of 100%.

Table 6 shows the comparison of the evaluation results between DAE+SMOTE+T-Link+XGBoost and the proposed model, as well as the evaluation of the multi-class classification result using precision, DR, and F1 scores. To the best of our knowledge, there is only one study [39] that has reported an anomalous multi-class classification.

The results indicated that the proposed model exhibited preeminence performance compared to the alternative models, achieving improvements of up to 99,37% in DR, and 99.37% in F1-score. In addition, the performance of the model in performing a multi-class classification on unbalanced data has also been evaluated using precision recall curves. Figure 6 shows the curve with nearly perfect classification results.

In Modbus traffic, some anomalies due to read attacks are incorrectly classified as replay attacks. Anomalies from read and replay attacks display similar network traffic patterns, as outlined in the report [7]. The read attack and replay attack classes represent a minority of the data in the imbalance dataset. The proposed model showed improved classification performance for handling imbalances against replay attacks, with only 1 misclassification compared to the other model [39], which had 9 misclassifications. Furthermore, STAM model only mis predicted 119 read attacks compared to 6858 [39]. Figure 7 illustrates the model comparison of the results of misclassification against minority classes, namely the read attack and replay attack.

Table 6. Comparison methods for multi-class classification

Method	Precision	DR	F1- Score
DAE + SMOTE + T-Link+ XGBoost [39]	99,99	97,67	98,50
Proposed Model	99,38	99,37	99,37

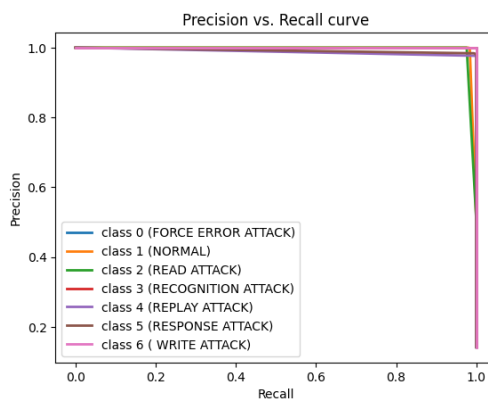


Figure 6. Precision recall curve for multi-class classification

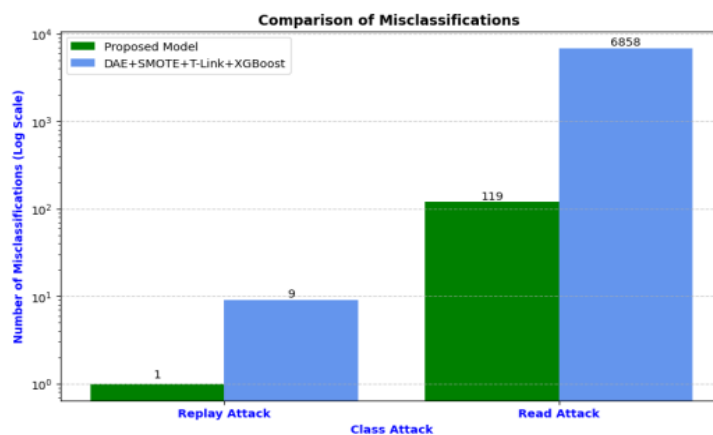


Figure 7. Comparison of misclassification for minority attacks

In future works, the effectiveness of detection will be developed with different types of attacks, such as botnet, DDoS, and zero-day without using the real environment of SCADA. Model development can also use other industrial communication protocols such as ethercat, profinet, and others that have different characteristics or features. Moreover, the potential to compare the performance of the model with other detection methods related to the development of future attack types is potentially possible.

4. CONCLUSION

The performance of the proposed model shows a nearly perfect classification. The proposed anomaly detection model demonstrates an optimal level of precision, detection rate, and F1 score, with a value of 100%. In the multi-class classification, the proposed model has the highest detection rate and F1 scores compared to other methods. Furthermore, the multi-class classification performance using STAM is better in the minority attack classes (read and replay attack), which have fewer misclassifications even though the attributes or features have similar patterns.

ACKNOWLEDGEMENTS

The research has been funded by Universitas Gadjah Mada via the RTA scheme number 5075/UN1.P.II/Dit-Lit/PT.01.01/2023. The authors are grateful for the financial support.




REFERENCES

- [1] M. Conti, D. Donadel, and F. Turrin, "A Survey on Industrial Control System Testbeds and Datasets for Security Research," *IEEE Commun. Surv. Tutorials*, vol. 23, no. 4, pp. 2248–2294, 2021, doi: 10.1109/COMST.2021.3094360.
- [2] A. B. Ajmal, M. Alam, A. A. Khaliq, S. Khan, Z. Qadir, and M. A. P. Mahmud, "Last Line of Defense: Reliability through Inducing Cyber Threat Hunting with Deception in SCADA Networks," *IEEE Access*, vol. 9, pp. 126789–126800, 2021, doi: 10.1109/ACCESS.2021.3111420.
- [3] G. Yadav and K. Paul, "Architecture and security of SCADA systems: A review," *Int. J. Crit. Infrastruct. Prot.*, vol. 34, p. 100433, 2021, doi: 10.1016/j.ijcip.2021.100433.
- [4] M. Bristow, "A SANS 2021 Survey: OT/ICS Cybersecurity," no. August, pp. 1–23, 2021, [Online]. Available: www.cisa.gov/critical-infrastructure-sectors.
- [5] W. J. Broad, J. Markoff, and D. E. Sanger, "Israeli Test on Worm Called Crucial in Iran Nuclear Delay," <https://www.nytimes.com/2011/01/16/world/middleeast/16stuxnet.html> (accessed Feb. 19, 2024).
- [6] C. Eaton and D. Volz, "U.S. Pipeline cyberattack forces closure," 2021. <https://www.wsj.com/articles/colonial-pipeline-ceo-tells-why-he-paid-hackers-a-4-4-million-ransom-11621435636> (accessed Dec. 20, 2024).
- [7] Á. L. P. Gómez *et al.*, "On the Generation of Anomaly Detection Datasets in Industrial Control Systems," *IEEE Access*, vol. 7, pp. 177460–177473, 2019, doi: 10.1109/ACCESS.2019.2958284.
- [8] R. Leszczyna, "Review of cybersecurity assessment methods: Applicability perspective," *Comput. Secur.*, vol. 108, 2021, doi: 10.1016/j.cose.2021.102376.
- [9] P. Wlazlo *et al.*, "Man-in-the-middle attacks and defence in a power system cyber-physical testbed," *IET Cyber-Physical Systems: Theory and Applications*, vol. 6, no. 3, pp. 164–177, 2021, doi: 10.1049/cps2.12014.
- [10] V. Ranade, "A laboratory for cyber-attack generation and testing in Industrial Control Systems: Design and Simulation," 2021. [Online]. Available: <https://resolver.tudelft.nl/uuid:ad554d68-4503-4544-b51b-e48379fc7216>.
- [11] Y. Yang, H.-Q. Xu, L. Gao, Y.-B. Yuan, K. McLughlin, and S. Sezer, "Multidimensional Intrusion Detection System for IEC 61850-Based SCADA Networks," *IEEE Trans. Power Deliv.*, vol. 32, no. 2, pp. 1068–1078, 2017.
- [12] M. Bashendy, S. Eltanbouly, A. Tantawy, and A. Erradi, "Design and Implementation of Cyber-Physical Attacks on Modbus/TCP Protocol," *World Congress on Industrial Control Systems Security (WCICSS-2020)*, no. December, pp. 38–45, 2021, doi: 10.20533/wcicss.2020.0005.
- [13] A. Rahman, G. Mustafa, A. Q. Khan, M. Abid, and M. H. Durad, "Launch of denial of service attacks on the modbus/TCP protocol and development of its protection mechanisms," *Int. J. Crit. Infrastruct. Prot.*, vol. 39, no. September, p. 100568, 2022, doi: 10.1016/j.ijcip.2022.100568.
- [14] D. Pliatsios, P. Sarigiannidis, T. Lagkas, and A. G. Sarigiannidis, "A Survey on SCADA Systems: Secure Protocols, Incidents, Threats and Tactics," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1942–1976, 2020, doi: 10.1109/COMST.2020.2987688.
- [15] P. Duclin, "Ransomware tales: The MitM attack that really had a Man in the Middle," 2023. <https://news.sophos.com/en-us/2023/05/24/ransomware-theses-the-mitm-attack-that-really-had-a-man-in-the-middle/> (accessed Feb. 21, 2024).
- [16] N. Wang, Y. Chen, Y. Hu, W. Lou, and Y. T. Hou, "MANDA: On adversarial example detection for network intrusion detection system," *Proc. - IEEE INFOCOM*, vol. 2021-May, pp. 1–10, 2021, doi: 10.1109/INFOCOM42981.2021.9488874.
- [17] F. SICARD, É. ZAMAI, and J. M. FLAUS, "An approach based on behavioral models and critical states distance notion for improving cybersecurity of industrial control systems," *Reliab. Eng. Syst. Saf.*, vol. 188, no. March 2018, pp. 584–603, 2019, doi: 10.1016/j.ress.2019.03.020.
- [18] M. Wan, W. Shang, and P. Zeng, "Double Behavior Characteristics for One-Class Classification Anomaly Detection in Networked Control Systems," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 12, pp. 3011–3023, 2017, doi: 10.1109/TIFS.2017.2730581.
- [19] S. D. Anton, S. Kanoor, D. Fraunholz, and H. D. Schotten, "Evaluation of machine learning-based anomaly detection algorithms on an industrial modbus/TCP data set," *ACM Int. Conf. Proceeding Ser.*, 2018, doi: 10.1145/3230833.3232818.
- [20] H. Yang, L. Cheng, and M. C. Chuah, "Deep-Learning-Based Network Intrusion Detection for SCADA Systems," *2019 IEEE Conf. Commun. Netw. Secur. CNS 2019*, 2019, doi: 10.1109/CNS.2019.8802785.




- [21] D. Al-safaar, "Hybrid AE-MLP: Hybrid Deep Learning Model Based on Autoencoder and Multilayer Perceptron Model for Intrusion Detection System," *International Journal of Intelligent Engineering & Systems*, vol. 16, no. 2, 2023, doi: 10.22266/ijies2023.0430.04.
- [22] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, 2019, doi: 10.1186/s42400-019-0038-7.
- [23] M. Riyadh and D. R. Alshibani, "Intrusion detection system based on machine learning techniques," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 2, pp. 953–961, 2021, doi: 10.11591/ijeecs.v23.i2.pp953-961.
- [24] S. A. Z. Mghames and A. A. Ibrahim, "Intrusion detection system for detecting distributed denial of service attacks using machine learning algorithms," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 32, no. 1, pp. 304–311, 2023, doi: 10.11591/ijeecs.v32.i1.pp304-311.
- [25] D. Manikandan and J. Dhilipan, "Machine learning approach for intrusion detection system using dimensionality reduction," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 34, no. 1, pp. 430–440, 2024, doi: 10.11591/ijeecs.v34.i1.pp430-440.
- [26] F. A. Vadhil, M. L. Salihi, and M. F. Nanne, "Machine learning-based intrusion detection system for detecting web attacks," *IAES International Journal of Artificial Intelligence*, vol. 13, no. 1, pp. 711–721, 2024, doi: 10.11591/ijai.v13.i1.pp711-721.
- [27] A. S. Jaradat, M. M. Barhoush, and R. B. Easa, "Network intrusion detection system: machine learning approach," *Indones. J. Electr. Eng. Comput. Sci*, vol. 25, no. 2, pp. 1151–1158, 2022, doi: 10.11591/ijeecs.v25.i2.pp1151-1158.
- [28] N. S. Gill and P. Gulia, "A review on machine learning based intrusion detection system for internet of things enabled environment," *International Journal of Electrical & Computer Engineering*, vol. 14, no. 2, pp. 1890–1898, 2025, doi: 10.11591/ijece.v14i2.pp1890-1898.
- [29] K. Farhana, M. Rahman, and T. Ahmed, "An intrusion detection system for packet and flow based networks using deep neural network approach," *International Journal of Electrical & Computer Engineering*, vol. 10, no. 5, pp. 5514–5525, 2020, doi: 10.11591/ijece.v10i5.pp5514-5525.
- [30] I. Idrissi, M. Boukabous, M. Azizi, O. Moussaoui, and H. El Fadili, "Toward a deep learning-based intrusion detection system for IoT against botnet attacks," *IAES International Journal of Artificial Intelligence*, vol. 10, no. 1, pp. 110–120, 2021, doi: 10.11591/ijai.v10.i1.pp110-120.
- [31] M. Amru, R. J. Kannan, and E. N. Ganesh, "Network intrusion detection system by applying ensemble model for smart home," *International Journal of Electrical & Computer Engineering*, vol. 14, no. 3, pp. 3485–3494, 2024, doi: 10.11591/ijece.v14i3.pp3485-3494.
- [32] J. Al Amien, H. A. Ghani, N. I. Saleh, E. Ismanto, and R. Gunawan, "Intrusion detection system for imbalance ratio class using weighted XGBoost classifier," *TELKOMNIKA Telecommunication, Computing, Electronics and Control*, vol. 21, no. 5, pp. 1102–1112, 2023, doi: 10.12928/TELKOMNIKA.v21i5.24735.
- [33] M. E. Magdy, A. M. Matter, S. Hussin, D. Hassan, and S. A. Elsaid, "Anomaly-based intrusion detection system based on feature selection and majority voting," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 30, no. 3, pp. 1699–1706, 2023, doi: 10.11591/ijeecs.v30.i3.pp1699-1706.
- [34] M. Kante, V. Sharma, and K. Gupta, "Mitigating ransomware attacks through cyber threat intelligence and machine learning," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 33, no. 3, pp. 1958–1965, 2024, doi: 10.11591/ijeecs.v33.i3.pp1958-1965.
- [35] S. M. Naf'an, I. Meiska, M. Kallista, I. P. D. Wibawa, and B. F. Aina, "Improving Decision Tree Accuracy through AdaBoost Ensemble with SMOTE Oversampling and ExtraTreeClassifier Feature Selection," *Int. Conf. Electr. Eng. Comput. Sci. Informatics*, no. September, pp. 557–564, 2023, doi: 10.1109/EECSI59885.2023.10295750.
- [36] R. Harwahyu, F. Henri, E. Ndolu, and M. V. Overbeek, "Three layer hybrid learning to improve intrusion detection system performance," *International Journal of Electrical & Computer Engineering*, vol. 14, no. 2, pp. 1691–1699, 2024, doi: 10.11591/ijece.v14i2.pp1691-1699.
- [37] K. Prabu and P. Sudhakar, "A hybrid deep learning approach for enhanced network intrusion detection," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 33, no. 3, pp. 1915–1923, 2024, doi: 10.11591/ijeecs.v33.i3.pp1915-1923.
- [38] B. Ning, S. Qiu, T. Zhao, and Y. Li, "Power IoT Attack Samples Generation and Detection Using Generative Adversarial Networks," in *2020 IEEE 4th Conference on Energy Internet and Energy System Integration*, pp. 3721–3724, doi: 10.1109/EI250167.2020.9346661.
- [39] J. R. Jiang and Y. T. Chen, "Industrial Control System Anomaly Detection and Classification Based on Network Traffic," *IEEE Access*, vol. 10, pp. 41874–41888, 2022, doi: 10.1109/ACCESS.2022.3167814.
- [40] "Electra dataset: Anomaly detection ICS dataset." <http://perception.inf.um.es/ICS-datasets/> (accessed Feb. 19, 2024).
- [41] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

BIOGRAPHIES OF AUTHORS






Freska Rolansa    is a researcher at the Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences, Universitas Gadjah Mada. He has 15 years of teaching experience in Informatics Engineering, Department of Electrical engineering, Pontianak State Polytechnic, Indonesia. He is currently Pursuing the Doctor in the field of security in industrial control systems and machines. He can be contacted via email: freskarolansa@polnep.ac.id.






Jazi Eko Istiyanto    is currently Full Professor in Electronics and Instrumentation at Universitas Gadjah Mada, Faculty of Mathematics and Natural Sciences, a position he held since 2010 before took office at BAPETEN (Indonesia Nuclear Energy Regulatory Agency) as the Chairman from February 2014 until October 2021. Before serving the Government of Indonesia, he held academic managerial positions as Head of the Computer Science and Electronics Department (2011-2014), and Head of the Physics Department (2007-2011) Universitas Gadjah Mada. He holds a Ph.D. (1995) in Electronic Systems Engineering, and an M.Sc. (1988) in Computer Science from University of Essex, Colchester, United Kingdom, and a B.Sc. (1986) in Nuclear Physics from Universitas Gadjah Mada, Yogyakarta, Indonesia. His research interests cover embedded systems and cyber-physical systems security. He is also a registered engineer (electronic engineering) in Indonesia and ASEAN countries. He can be contacted via email: jazi@ugm.ac.id.



Afiahayati    completed his Ph.D. from Department of Biosciences and Informatics/Faculty of Science and Technology, Keio University, Japan. She is an Associate Professor Department of Computer Science and Electronic, Faculty of Mathematics and Natural Sciences, Universitas Gadjah Mada, Yogyakarta, Indonesia. Her research interests include metagenomic assembler, genome assembly, comparative genomic, machine learning, probabilistic, bioinformatics, data mining, and artificial intelligence. She has authored more than 70 papers in national and international peer reviewed journals, and international conference. She can be contacted via email: afia@ugm.ac.id.



Aufaclav Zatu Kusuma Frisky    completed his Doctorate in Faculty of Informatics, Technische Universität Wien (TU-Wien), Austria. He is Assistant Professor Department of Computer Science and Electronic, Faculty of Mathematics and Natural Sciences, Universitas Gadjah Mada, Yogyakarta, Indonesia. His research interests include machine learning, smart robotic, multimedia IR (image, audio, video processing), computer vision, image processing, and 3D construction. He has authored more than 70 papers in national and international peer reviewed journals. He can be contacted via email: aufaclav@ugm.ac.id.