REFERENCE-BASED AI DECISION SUPPORT FOR CYBERSECURITY

SEMINAR REPORT

SUBMITTED

TO

AWH ENGINEERING COLLEGE

KUTTIKKATTOOR, KOZHIKODE - 8

IN PARTIAL FULFILMENT
OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE

OF

Master Of Computer Applications

BY

ARCHANA.T

AWH ENGINEERING COLLEGE
CALICUT.KERALA.INDIA

DEPARTMENT OF COMPUTER APPLICATIONS
AWH ENGINEERING COLLEGE KUTTIKKATTOOR,
KOZHIKODE
FEBRUARY 2024

DEPARTMENT OF COMPUTER APPLICATIONS

.2001.

AWH ENGINEERING COLLEGE
CALICUT.KERALA.INDIA

AWH ENGINEERING COLLEGE

KOZHIKODE

CERTIFICATE

This is to certify that this Seminar entitled ***"REFERENCE BASED AI DECISION SUPPORT FOR CYBERSECURITY"*** *submitted herewith is an authentic record of the Seminar work done by* ***ARCHANA.T (AWH22MCA-2011)*** *under our guidance in partial fulfillment of the requirements for the award of* ***Master of Computer Applications*** *from APJ Abdul Kalam Technological University during the academic year 2024.*

Mrs. Sruti Sudevan
Assistant Professor
Dept. of Computer Applications
Head of the Department

Ms. Prajina K
Assistant Professor
Dept. of Computer Applications
Project Guide

ACKNOWLEDGEMENT

I express my sincere gratitude to our beloved principal **Dr. Sabeena M V** for providing me an opportunity with the required facilities for doing this project. I express my hearty thanks to **Mrs. Sruti Sudevan**, Head of the department of Computer Applications and Assistant Professor for her guidance. I am thankful to all other staff of the MCA department for their encouragement, timely guidance, valuable suggestions and inspiring ideas given throughout this project. I am grateful to my friends for the way they have cooperated, expected me to achieve success and have always stirred my ambition to do the best. Above all, I am grateful to the almighty, who has showered His blessings on me throughout my life and throughout the project.

ARCHANA. T

ABSTRACT

Due to its ability to effectively process and utilize big data, Artificial Intelligence (AI) technology has been a field of active research and development since its early stages. Thanks to these research efforts, AI technology has evolved and has become applicable to various fields. However, in pursuit of such performance improvements, AI technology has adopted a more complex output logic, which has resulted in the decreased interpretability of its output. In other words, although it has demonstrated excellent performance, AI technology has acquired a black-box nature that makes it difficult to identify the mechanism behind its output. This characteristic of AI technology has become an obstacle to its adoption in fields that have a high risk of false positives. To address these shortcomings and make AI effective even in fields with a high risk of false positives, eXplainable Artificial Intelligence (XAI) technology is being develop.

CONTENTS

1. INTRODUCTION

In the cyber environment, massive amounts of data are generated daily. Artificial Intelligence (AI) technologies can effectively manage this vast data to support efficient operations in the cyber environment. Due to ongoing research efforts, there has been notable progress in the field of AI. However, as AI achieves higher performance, it becomes increasingly complex, which results in the low interpretability of AI outputs. This black-box nature of AI technology makes AI challenging to apply in fields like cybersecurity, where the risk of false positives is significant. To address this issue, researchers have been working on eXplainable Artificial Intelligence (XAI) technology, with the intention to enhance the utility of AI by providing interpretations of AI predictions.

Most previous research has focused on understanding how models function in terms of feature importance to interpret AI results. However, this approach fails to provide clear interpretations in fields where interpretability is crucial, such as security. Therefore, this proposes a framework that offers interpretations of AI results, even in unsupervised environments that are suitable for security scenarios. Additionally, this have improved the logic of calculation Reference and have enhanced the function and performance compared with previous research. It provides additional information that supports interpretation, such as P-Values and References, to offer more effective decision support to security analysts and to ultimately reduce false alarms and enhance model performance. Overall, this approach aims to improve the model’s performance by providing clear interpretations that are suitable for security tasks, thereby contributing to more effective decision-making by security analysts.

2. MOTIVATION

Previous well-known XAI technologies, such as Shapley Additive exPlanation(SHAP) and Class Activation Map(CAM), have primarily provided visual interpretations based on feature importance. These research approaches aim to explain how each feature contributes to AI's decision making by calculating the importance of each feature when AI makes specific decisions.

Such previous research mainly focused on providing interpretations through an understanding of AI's operational logic and was developed to target supervised learning models to compute feature importance for each label. However, visual interpretations based on feature importance in cybersecurity may not always provide clear explanations. This is because cybersecurity relies on attack detection based on the differences between the original feature values in normal scenarios and those in attack scenarios. Therefore, in cybersecurity, interpreting attack decisions based on differences in terms of feature values in the actual data, rather than relying solely on feature importance, can offer clearer interpretability.

The cybersecurity environment needs greater resources to label all data, and most of the collected data consists of normal data, which leads to class imbalance issues. Additionally, supervised learning relies on pre-labeled data for detection while achieving high discrimination accuracy, which makes it challenging to respond to unknown threats and attacks that have yet to be detected. Unsupervised learning models can effectively operate in cybersecurity environments, which do not require label information during AI training and the result-generation processes.

Therefore, we were motivated to generate interpretations for unsupervised learning models that operate in such scenarios. Based on Han's DeepAID, which provides clear interpretations through calculation a Reference that possesses a normal label while being most similar to malignant data and provide clear interpretations through feature value comparison with that Reference, we aim to offer clear interpretations for cybersecurity. Furthermore, we generate additional interpretive metrics such as P-values and the nearest real data based on the generated References. We aim to enhance interpretability, reduce false alarms, and improve model performance through AI decision support.

3. CONTRIBUTION

1) Providing clear and suitability for the security field interpretations

Most previous research relied on feature importance-based interpretations. However, in the security field, where detecting anomalies is done using differences in actual data values, existing techniques often need help to provide clear interpretations. Therefore, we generate References to explain anomalous signs and offer clear interpretations through Feature Value Comparison.

2) Supporting effective false alarm reduction

In the security domain, detecting anomalies early is crucial to protecting resources and services and to preparing for potential attacks. Various studies have focused on early detection and reducing false alarms to alleviate the operational burden that arises from these events. In our framework, we create metrics, such as References and P-Values, to effectively detect false alarms during AI operations, thus supporting false alarm reduction and improving the efficiency of analysts' Anomaly Detection tasks.

3) Enhancing reference generation performance

This improves the existing Reference generation logic to maximize its significance. This enhancement resulted in improved success rates for Reference generation, reduced time required for Reference generation, and better performance.

4. RELATED WORKS

4.1 Feature Importance-Based XAI

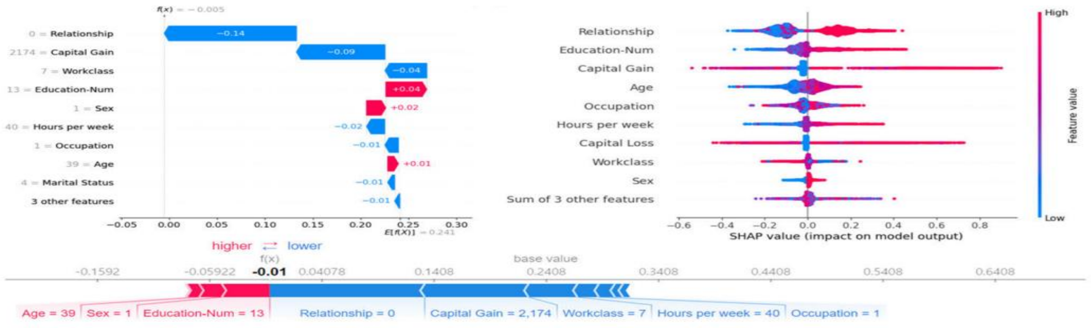


Figure 4.1.1- Example of Shapley Additive exPlanation(SHAP)

Figure 4.1.1 illustrates examples of interpretation based on a representative XAI research, Shapley Additive exPlanation(SHAP). This figure shows that previous XAI techniques were primarily applied to supervised learning models. This calculate Feature Importance values based on label information to visually explain the model's decisions.

Figure 4.1.1 represents the Global interpretation and Local interpretation obtained when SHAP is applied to a supervised learning-based model that predicts tabular data. Local interpretation information allows analysts to identify the factors that contribute to individual AI decisions. Global interpretation information consolidates these local interpretations and provides a visual understanding of how each feature influences the overall AI decision.

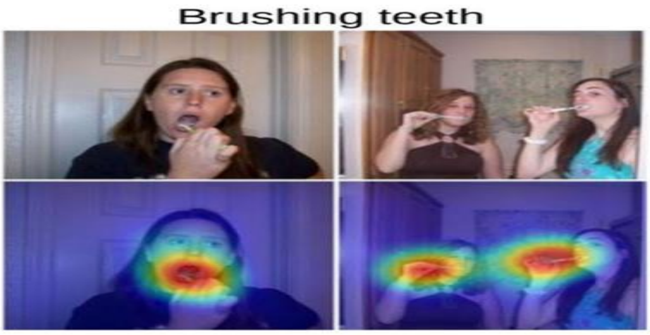


Figure 4.1.2- Example of Class Activation Map(CAM)

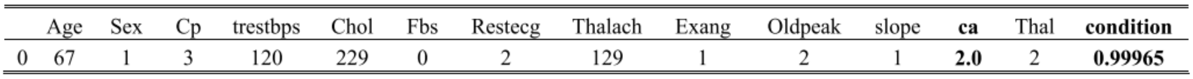
Figure 4.1.2 presents the interpretations obtained when a CAM is applied to a Convolutional Neural Network (CNN) model for image classification. CAM visually indicates which parts of an image the image classification model considered when it predicted the class. As shown in this figure, regions of the image that are related to the predicted class exhibit high activation values. Based on this interpretive information, analysts can determine which features influence the current AI's results and understand the AI model's operational process. This helps to compensate for the black-box nature of traditional AI technology and increases trust in AI applications.

However, most previous XAI techniques were developed primarily around supervised learning, and they utilized model training to explain the AI's operational process. As a result, these XAI techniques differ in their learning mechanisms and have limited applicability to the unsupervised environments that are suitable for real security scenarios. Furthermore, in security environments, where attacks must be detected based on differences between normal scenarios and attack scenarios, providing interpretations through the feature value comparison of actual data is often more likely to offer clearer interpretability than dealing solely with Feature Importance.

4.2 Example Technique-Based XAI

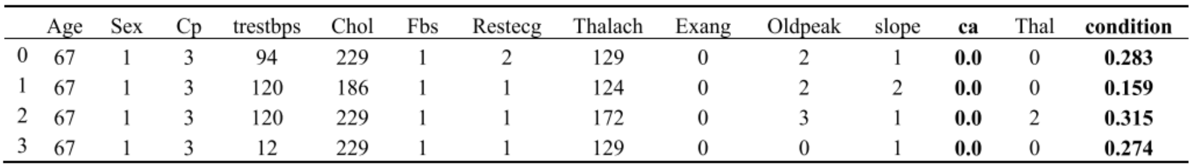
Dwivedi et al. classified existing XAI techniques into four major categories based on their solutions and main ideas. In this chapter, we focus on one category: Example-Based Techniques.

Examples of XAI techniques that are Example-Based Techniques include Anchors introduced by Marco, Kernel Shap, and Contrastive Explanation Method. The key idea behind these techniques is to explain the model's decisions using specific examples.



	Age	Sex	Cp	trestbps	Chol	Fbs	Restecg	Thalach	Exang	Oldpeak	slope	**ca**	Thal	**condition**
0	67	1	3	120	229	0	2	129	1	2	1	**2.0**	2	**0.99965**

Table 4.2.1- Input of counterfactuals method: A Specific instance in which the patient has heart disease



	Age	Sex	Cp	trestbps	Chol	Fbs	Restecg	Thalach	Exang	Oldpeak	slope	**ca**	Thal	**condition**
0	67	1	3	94	229	1	2	129	0	2	1	**0.0**	0	**0.283**
1	67	1	3	120	186	1	1	124	0	2	2	**0.0**	0	**0.159**
2	67	1	3	120	229	1	1	172	0	3	1	**0.0**	2	**0.315**
3	67	1	3	12	229	1	1	129	0	0	1	**0.0**	0	**0.274**

Table 4.2.2- Different outputs of counterfactual explainer

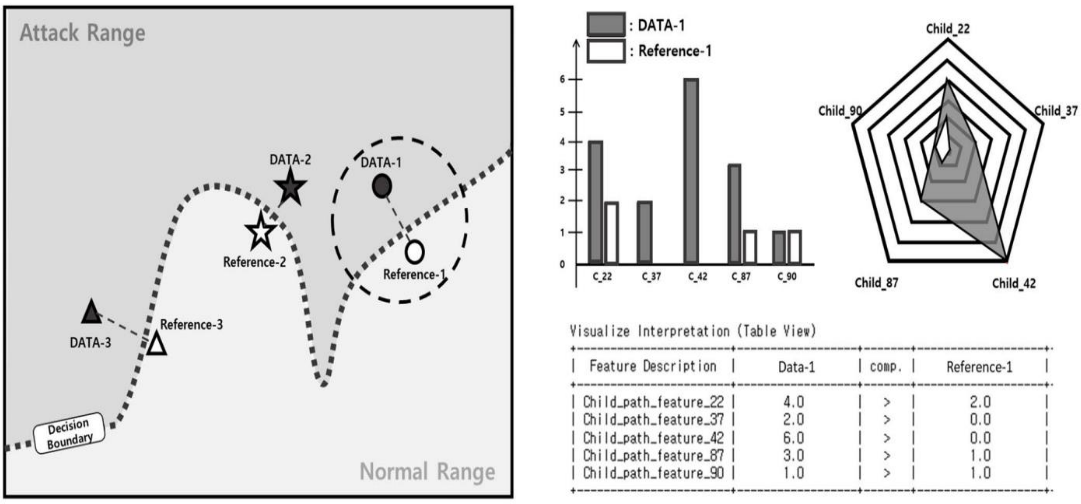
The results of such example technique-based XAI are presented in Tables 4.2.1 and 4.2.2, The results shown here are interpretations that are obtained when Counterfactual explanation, one of the Example-Based XAI Techniques mentioned by Dwivedi, is applied. The Counterfactual technique is a method where the prediction results of a model are presented differently by modifying the feature values of the input data. This experiment's dataset consists of 13 features and is known as the Heart Disease dataset. In Tables 4.2.1 and 4.2.2, the list of features is the same for both, except for the ‘‘condition’’ item, which represents the model's predicted probability for the occurrence of heart disease. In this experiment, Dwivedi et al. kept unmodifiable features like Age, Sex, and Cp fixed while modifying the rest of the features to perform Counterfactual analysis. As the ‘‘condition’’ value was very high in the input data for Counterfactual Explainer, the results produced according to the Counterfactual algorithm showed output samples with lower ‘‘condition’’ values, as shown in Table 4.2.2. These output samples can provide insights into the model's predictions. All four samples with lower ‘‘condition’’ values in Table 4.2.2 have a common feature value of ‘‘ca’’ equal to 0. Therefore, we can conclude that having a ‘‘ca’’ feature value of 0 results in a lower probability of having heart disease. In other words, the high ‘‘condition’’ value in the input data is because the ‘‘ca’’ feature value was set high. We can understand the factors that contribute to the model's predictions based on feature values through such interpretations.

In this way, Example Technique-Based XAI gives the analyst insights into how each feature influences the model's predictions and the reasons behind the model's decisions based on feature values. Therefore, in security environments where the detection of anomalies between normal and attack data is crucial, such as in technique-based XAI, which provides interpretations by comparing the features of target data identified as anomalies and example data identified as normal, offers better interpretability than do the interpretation methods based on Feature Importance.

4.3 DeepAID

The DeepAID technique proposed by Han follows a approach similar to that of Example Technique-Based XAI. In DeepAID, two losses are generated for data that are classified as anomalies, and the data values are updated using an optimizer while these losses are minimized.

This process creates what is known as a ''Reference,'' which is an example with the closest features to the target data but having a normal label. DeepAID generates interpretations through this Reference. Reference Value are typically created near the decision boundary between normal data and anomalies. Through a Feature Comparison between the Reference and Target data, DeepAID explains to the analyst why the Target data are classified as an anomalies.



```
Visualize Interpretation (Table View)
+------------------------+-------------------+-------+-------------------+
| Feature Description    |      Data-1       | comp. |    Reference-1    |
+------------------------+-------------------+-------+-------------------+
| Child_path_feature_22  |        4.0        |   >   |        2.0        |
| Child_path_feature_37  |        2.0        |   >   |        0.0        |
| Child_path_feature_42  |        6.0        |   >   |        0.0        |
| Child_path_feature_87  |        3.0        |   >   |        1.0        |
| Child_path_feature_90  |        1.0        |   >   |        1.0        |
+------------------------+-------------------+-------+-------------------+
```

Figure 4.3.1- Reference-based XAI techniques for effective interpretability in security.

The tendencies of the generated References and the provided explanations are similar to those in Figure 4.3.1. First, on the left side of Figure 4.3.1, References 1 to 3 are generated for DATA 1 to 3, which fall within the Attack Range. These References are created close to the normal range while being very close to the Target data. Therefore, Reference 1 to 3 are all created near the decision boundary, meaning their values are within the normal range but have highly anomalous feature values. This provides an interpretation stating that the Target data are classified as anomalous because it has more anomalous values than the most anomalous Reference Value, even within the Normal range.

Similarly, Feature Value Comparison is performed between Data-1, classified as an Anomaly, and Reference-1, resulting in an interpretation as shown on the right. By interpreting this, we can identify the five features that have contributed the most to Data-1's Anomaly classification. Data-1's feature values are the same as in the Data-1 column, and Reference-1's values for these features are the same as in the Reference-1 column. As mentioned earlier, the generated References fall within the Normal range but have highly anomalous feature values. As a result of the Feature Value Comparison between Reference-1 and Data-1, Data-1 is classified as having highly anomalous feature values for all features. Therefore, Data-1 is interpreted as an Anomaly.

This approach allows for the generation of interpretations for unsupervised learning models and provides interpretations for Anomaly Detection through Feature Value Comparison. DeepAID's Reference generation technique is expected to provide clearer interpretations in the security field. Therefore, in this paper, we use and improve previously researched Reference generation mechanisms to create more effective Reference generation. We propose a framework that performs AI decision support that is suitable for the security field by creating metrics based on the generated Reference and analyzing these metrics comprehensively.

5. PROPOSED METHOD

Reference-based AI decision support for cybersecurity propose a framework to support effective AI Decision Support in security.

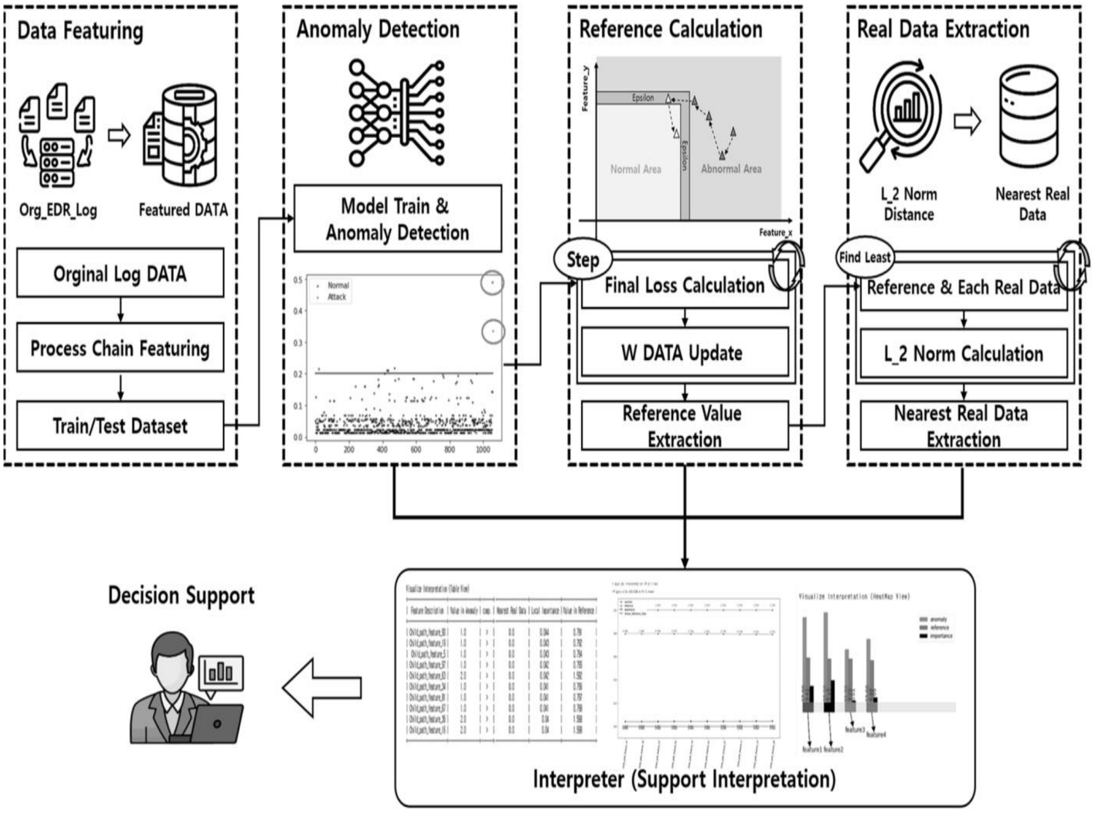


Figure 5.1-Proposed framework: Providing clear interpretability for anomaly data.

The proposed Framework provides clear interpretation and AI Decision Support for an Autoencoder model that performs Anomaly Detection based on raw log data collected in Endpoint Detection and Response (EDR). The proposed framework can provide interpretations for unsupervised learning models and enhance the Reference generation algorithm proposed by Han and others to Provide clear Interpretation in the security field. It confirms through validation work that the proposed improved Reference generation logic, specifically the calculation of Optimum Reference, can enhance functional and performance metrics. Based on this Optimum Reference, it generates various metrics such as Nearest Real Data, P-Value, and others to enhance the clarity of Interpretation.

The proposed framework analyzes these metrics comprehensively and provides AI Decision Support for each AI prediction made by AI. This allows analysts to choose whether to cite the final judgment made by AI. In false alarms caused by incorrect AI predictions, analysts can contribute to false alarm reduction and enhance AI’s performance by not citing AI’s prediction.

5.1. Calculation of Optimum Reference

Improve the Reference generation algorithm proposed by Han and generate the Optimum Reference. To create data that align with the purpose of the Reference the DeepAID method is used.

By using DeepAID, to create data that align with the purpose of the Reference, two types of losses, Loss1 and Loss2, are calculated for the updating data W at each step, starting from the Target Data. Then, an optimization technique is used to update the feature values of the data W to reduce the computed losses. The updated data w are then used as the Reference.

Here the two losses mentioned earlier, loss1 and loss2, and examine the meaning of each loss. And also explore how data that align with the Reference is generated through this process. The first loss, Loss1, is typically calculated based on the Mean Squared Error (MSE), a commonly used Anomaly detection metric for Anomaly Detection mechanisms. The formula for calculating MSE is given in Equation (1).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(W_i - \check{W}_i)^2 \qquad (1)$$

In Equation (1),

- n represents the number of features
- W represents the original values
- W’ represents the predicted reconstruction values.

Typically, in autoencoder-based Anomaly Detection tasks, W’ represents the predicted reconstruction values for the original data W. Therefore, a high error, or MSE value, between the original data W and the predicted reconstruction values W’ indicates that the model finds it difficult to reconstruct that data, showing it to be an Anomaly.

This MSE can be used as an Anomaly score for Anomaly detection, as shown in Equation (2), where loss1 is calculated based on the previously mentioned Anomaly Detection criterion, the threshold MSE.

$$loss1 = Relu(MSE\,(W) - thres * (1 - eps\_rate)) \qquad (2)$$

- The data W identified in Equation (2) represent the intermediate values that start from the original target data and are eventually generated as the Reference through updating.
- ‘‘Thres’’ in Equation (2) is the MSE value that serves as a threshold used by the trained model for Anomaly Detection.

If the updating W data are considered anomalous; the MSE(W) value exceeds Thres, causing loss1 to have a positive value. When the Optimizer updates the Feature Value to minimize loss1, the MSE(W) value decreases. If W data move into the normal range through updates, the MSE(W) value becomes smaller than Thres. In this case, the result of the equation inside the Rectified Linear Unit (ReLU) activation function becomes negative, and negative values result in loss1 being set to 0 by the Rectified Linear Unit (ReLU) activation function.

Consequently, loss1 decreases when the updating data W moves into the normal range, reaching its minimum when it reaches the normal range.

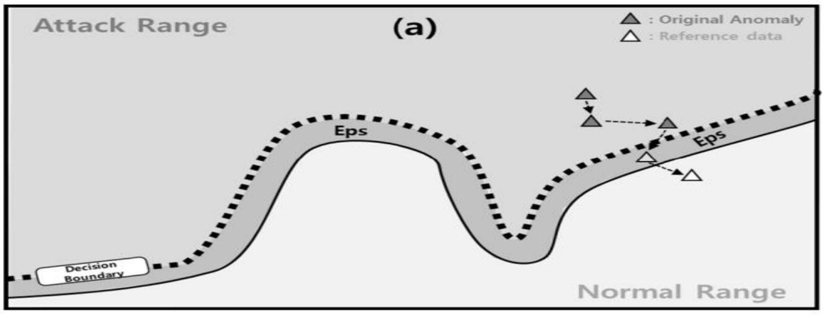


Figure 5.1.1- Trends in W data updates through loss-based optimization (a): loss1

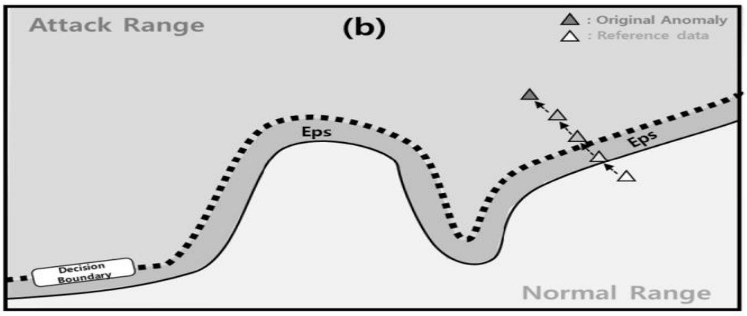


Figure 5.1.2- Trends in W data updates through loss-based optimization (b): loss2

In other words, when updating the data based on loss1, the data W move into the normal range, as shown in Figure 5.1.1. As mentioned earlier, Equation (3) represents the formula for calculating the second loss, loss2.

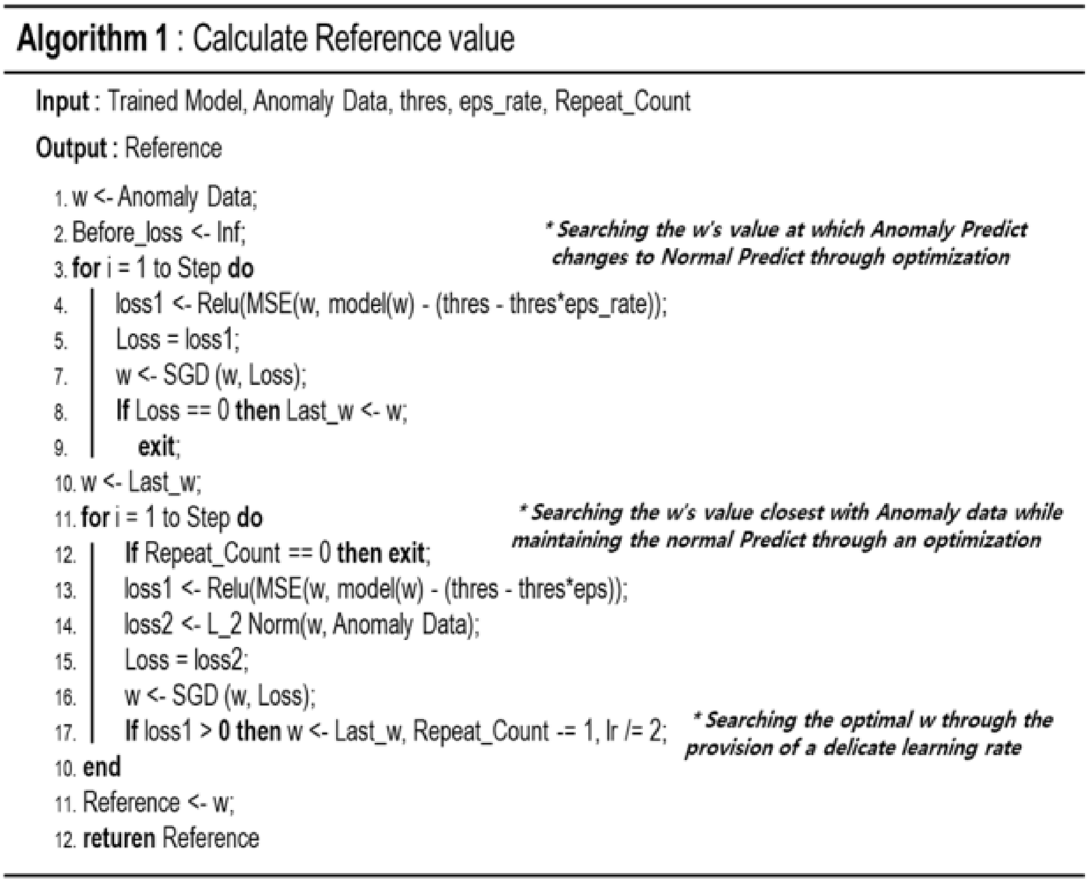
$$loss2 = \sqrt{\sum_{i=1}^{n} (W_i - Origianl\ DATA_i)^2} \qquad (3)$$

Loss2 as calculated according to Equation (3) represents the Euclidean distance between the data W being updated and the original data. Therefore, as the updating data W moves further away from the original data, the value of loss2 increases. When the optimizer updates the feature values of W in the direction of reducing loss2, the W data move towards the original data, as shown in Figure 5.1.2.

However, these two losses tend to be inversely proportional to each other. As the intermediate W data start from the original Anomaly data and move towards the normal category, the value of loss1 decreases. This implies that the original feature values classified as Anomaly have become similar to the normal, and as loss1 decreases, loss2 tends to increase. Therefore, even if W data is moved toward the normal category using loss1, the W data may return to the Attack Range if there are no limitations due to loss2 updates.

Hence, previous researchers, including Han et al., combined loss1 and loss2 in the Reference generation method and effectively utilized both losses by adjusting the influence of each loss using hyperparameters. Combining inversely proportional losses using the previous method significantly influences the hyperparameters and diminishes the meaning of the calculated losses. Therefore, we propose a new Reference generation method that divides the process into two major steps, to effectively utilize each loss and reduce the influence of hyperparameters.

The Reference generation algorithm



Algorithm 1 : Calculate Reference value

Input : Trained Model, Anomaly Data, thres, eps_rate, Repeat_Count

Output : Reference

```
w <- Anomaly Data;
Before_loss <- Inf;                    * Searching the w's value at which Anomaly Predict
for i = 1 to Step do                     changes to Normal Predict through optimization
    loss1 <- Relu(MSE(w, model(w) - (thres - thres*eps_rate));
    Loss = loss1;
    w <- SGD (w, Loss);
    If Loss == 0 then Last_w <- w;
        exit;
w <- Last_w;
for i = 1 to Step do                   * Searching the w's value closest with Anomaly data while
    If Repeat_Count == 0 then exit;      maintaining the normal Predict through an optimization
    loss1 <- Relu(MSE(w, model(w) - (thres - thres*eps));
    loss2 <- L_2 Norm(w, Anomaly Data);
    Loss = loss2;
    w <- SGD (w, Loss);
    If loss1 > 0 then w <- Last_w, Repeat_Count -= 1, lr /= 2;   * Searching the optimal w through the
end                                                                provision of a delicate learning rate
Reference <- w;
returen Reference
```

Figure 5.1.3- Proposed algorithm of optimum reference calculation.

The Reference generation algorithm is described by Equation (4).

$$wdata = \begin{cases} (Step\ 1)\ w_t = SGD\,(w_{t-1}, loss1_{t-1}) \\ \quad \boldsymbol{if}\ loss1_t \leqslant 0 \\ \quad \boldsymbol{then\ Stop}\ Step1\ and\ \boldsymbol{goto}\ Step2 \\ (Step\ 2)\ w_t = SGD\,(w_{t-1}, loss2_{t-1}) \\ \quad \boldsymbol{if}\ loss_1 > 0 \\ \quad \boldsymbol{then}\ w_t = w_{t-1}\ and\ Optim \\ \quad Lr_t = Optim\ Lr_{t-1}/2 \end{cases} \qquad (4)$$

The proposed Reference generation process consists of two main steps: Step 1 and Step 2. In each step, only one type of loss is used for optimizer-based updates, which allows us to utilize both loss1 and loss2 effectively. Additionally, this aims to reduce the influence of hyperparameters by eliminating the trade-off parameter that existed in the previous approach. Furthermore, in Step 2, we apply a dynamic learning rate based on the update trend to facilitate effective Reference generation through fine-grained exploration. The details of the update process at each step are as follows:

(**Step 1**)—Calculate loss1 for the initial w data, which start from the original Anomaly data, and perform updates based on an optimizer. Repeat Step 1 until the loss1 of w data reaches 0, meaning it moves to the normal category. After this point is reached, conclude Step 1 and proceed to Step 2.

(**Step 2**)—If in Step 1, w data have reached the normal category, then in Step 2, update based on loss2 to bring w data closer to the original anomalous data while still belonging to the normal category. During these update iterations, if significant fluctuations cause w data to move back to the anomalous category, revert w data to the state it was in just before the previous update and halve the learning rate to restart the fine-grained exploration.

The advantages that can be obtained through the proposed Reference Generation Logic are as follows.

1) More effective utilization of the roles of each loss with contrasting characteristics

In the previous approach, the losses were combined into one. However, combining loss1 and loss2, which have contrasting tendencies, into a single loss and performing optimizer-based optimization may dilute the significance of the resulting loss. Therefore, divide the Reference Generation into two major steps. In Step 1, we use loss1 exclusively; in Step 2, use loss2 exclusively. This preserves the numerical values of the generated losses and the roles of each loss, which leads to more effective optimization.

2) Generating data that are more suitable for reference by providing a dynamic learning rate

In the previous approach, the termination point for Reference Generation was set as either when the final loss decreased by a certain amount during the optimization process or when the degree of updating, as seen in Figure 5.1.4, was high enough for the w data to revert to the attack category. In the latter case, if the learning rate is reduced to decrease amount of Feature Value updates. it allows for more precise exploration, enabling the generation of a Reference that is closer to the attack data while maintaining the normal label, compared to the previous approach.

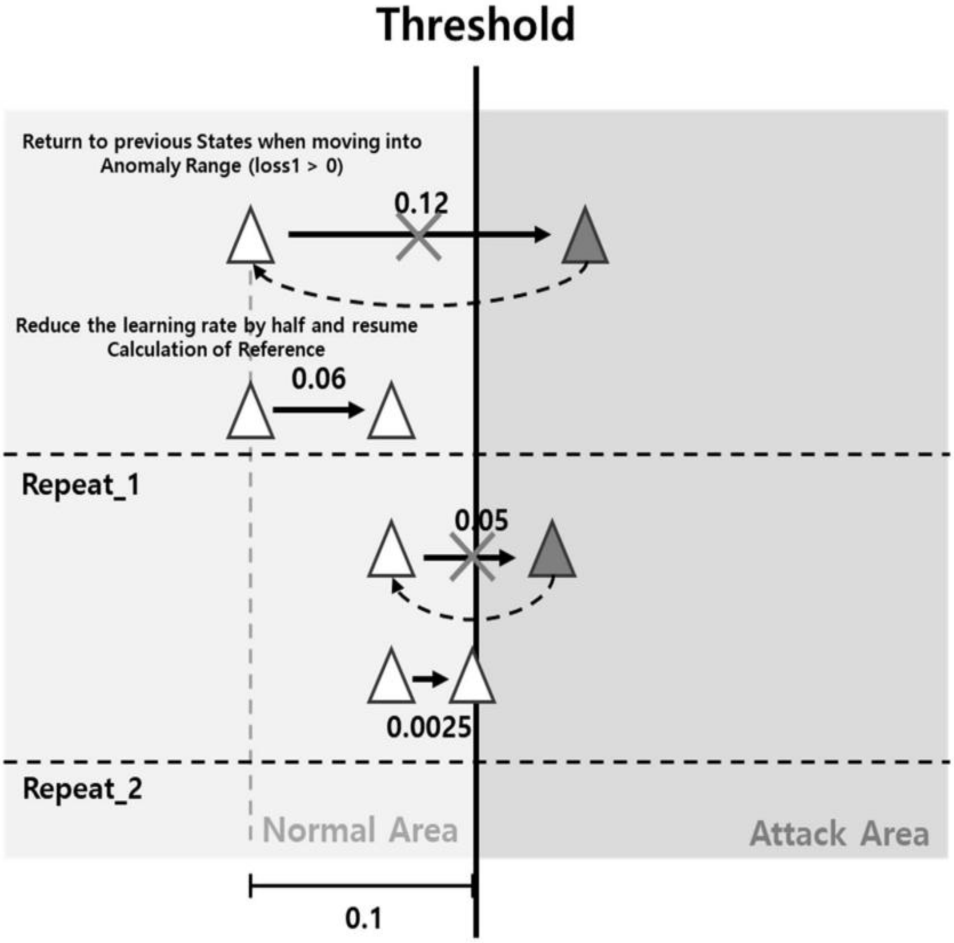

Threshold
Return to previous States when moving into Anomaly Range (loss1 > 0)
0.12
Reduce the learning rate by half and resume Calculation of Reference
0.06
Repeat_1
0.05
0.0025
Repeat_2
Normal Area
Attack Area
0.1

Figure 5.1.4- Trends in w data updates through step

Therefore, in the step where updates are performed based on loss2 to get closer to the original anomaly data, as shown in Figure 5.1.4, if the distance to the attack is 0.1, yet the update amount is higher, around 0.12, indicating that the data is about to enter the attack category. In the conventional method, such a case would have immediately terminated the search. However, in the proposed method, if it is anticipated that the data is about to enter the attack category, it reverts the data to its previous state, halves the learning rate responsible for the update magnitude, reducing the update amount to half of the original at 0.06, enabling a more precise exploration.

5.2 Nearest Real Data

Previous methods aimed to create References for Target Anomaly Data based on thresholds used as Anomaly Detection criteria in unsupervised learning environments. The goal was to provide explanations for Anomaly Detection results even in situations where labels are not available. However, the feature values of the final generated Reference are merely estimates that have been iteratively updated to reduce loss by the optimizer. Therefore, although the generated Reference may fall within the normal category according to the model’s judgment, it could contain feature values that may not exist in the original data format.

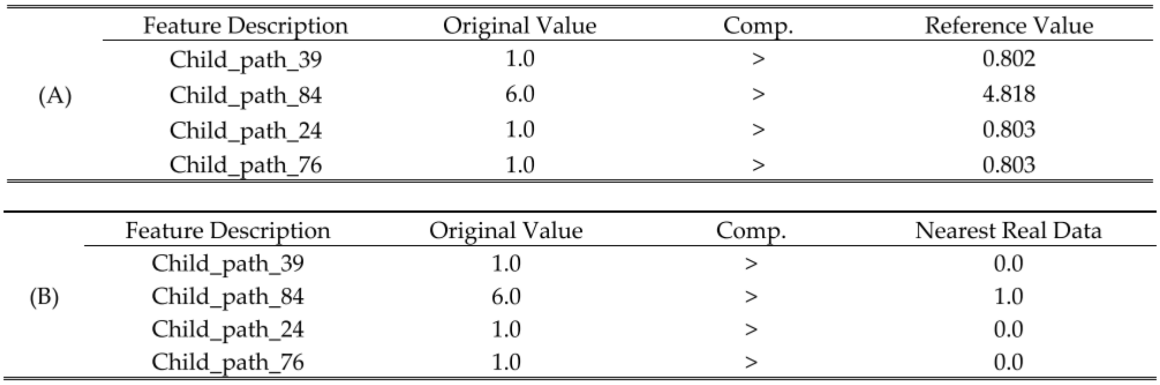
If the generated Reference in this approach has feature values that do not correspond to actual data format, it may reduce the clarity of providing interpretation through feature value comparisons. Therefore, we seek to generate actual data that could replace the generated Reference, i.e., Nearest Real Data. Using real, existing data as a basis, it intend to provide clearer interpretations.,

$$Distance_x = \sqrt{\sum_{i=0}^{n} x_i - Reference_i}, \quad \{x \mid x \in Normal\}$$

$$Nearest\ Real\ Data = x\{x/x \in Normal, MIN(\llbracket(Distance)\rrbracket\_x)\} \quad (5)$$

The Nearest Real Data generation method proposed in this paper is depicted in Equation (5). As can be deduced from this equation, we utilize a K-Nearest Neighbor(KNN) method with K=1 using the data that the model considers as normal to generate the Nearest Real Data. This method identifies the data point in the real dataset that is most similar to the Reference generated for normal data according to the model’s judgment. Therefore, the resulting Nearest Real Data maintains the meaning of the original Reference while containing feature values that can occur in the real world.

A more precise interpretation based on actual values is achieved by replacing the Reference with Nearest Real Data during the feature value comparison step. Analysts ultimately obtain a much clearer Feature Comparison-based interpretation through this method than through the conventional approach.



	Feature Description	Original Value	Comp.	Reference Value
(A)	Child_path_39	1.0	>	0.802
	Child_path_84	6.0	>	4.818
	Child_path_24	1.0	>	0.803
	Child_path_76	1.0	>	0.803

	Feature Description	Original Value	Comp.	Nearest Real Data
(B)	Child_path_39	1.0	>	0.0
	Child_path_84	6.0	>	1.0
	Child_path_24	1.0	>	0.0
	Child_path_76	1.0	>	0.0

Table 5.2.1- Examples of feature value comparison-based interpretation.

Interpretation through feature value comparison in Table 5.2.1(a), the ‘‘Feature Description’’ lists the key features that significantly impact the Anomaly determination of the Target Data. ‘‘Original Value’’ represents the Feature Value of the Target Data, whereas ‘‘Reference Value’’ represents the Feature Value of the Reference. According to the model’s judgment, the generated Reference is the closest value within the normal category to the Target Anomaly, and this makes it the most anomalous value within the normal category. Therefore, Feature Comparison makes it evident that the Feature Values of the Target Data are more anomalous than those of the Reference, which leads to the clear interpretation that the Target Data are identified as an Anomaly.

However, it is important to note that these datasets are preprocessed based on a 2-gram hash mapping-based natural language processing, and this results in all features having integer values. Despite this, when values optimizing to reduce the loss based on the optimizer, it can still be observed that the resulting Reference contains fractional values. As a result, the generated Reference contains fractional values for features that cannot occur in real-world scenarios, which may hinder the interpretability of Comparison-based Interpretation.

On the other hand, as shown in Table 5.2.1(b), Nearest Real Data are based on real-world data that are the most similar to the Reference, and therefore, the Feature Values are represented as integers. Through Feature Comparison based on these actual values, the interpretation of the cause of Anomaly for the Target Data becomes clearer. Utilizing the

Nearest Real Data allows analysts to obtain a more precise Feature Comparison-based Interpretation.

5.3 P-Value-based Improvement of Interpretation

Here utilizes the distance between Nearest Real Data and original data as one of the AI Decision Support metrics and a Feature Comparison between data points. The computed distance values are expected to increase as the Anomaly level of the Target, which is the Original Data, increases, indicating a greater distance from the Nearest Real Data (considered normal). Conversely, distance values are expected to decrease as the Anomaly level decreases. Therefore, these distance values allow for comparing the Anomaly levels of each data.

However, it is important to note that the computed distance values for each pair of original data and Nearest Real Data may have inconsistent meanings and can be subject to change based on the data value ranges. For example, if two groups have average distances of 1000 and 10, respectively, and a data is measured with a distance of 100 in each group, it could be considered a normal situation in the first group but a highly anomalous situation in the second group.

In summary, while distance metrics provide valuable information for comparing Anomaly levels, their interpretation may vary depending on the context and distribution of data values, which should be considered when making decisions based on these metrics.

$$
\begin{aligned}
CDF\left(x\right) &= \int_{-\infty}^{x} p\left(x\right) dt, \quad p\left(x\right) = \frac{d}{dx} CDF\left(x\right) \\
CDF\left(Distance\right) &= P\left(X \leqslant Distance\right) \\
&= \sum\nolimits_{x \leqslant Distance} p\left(x\right) \\
P - Value\left(Distance\right) &= 1 - CDF\left(Distance\right)
\end{aligned} \tag{6}
$$

Therefore, to provide consistent metrics across all data types, we applied the Cumulative Distribution Function (CDF) to the Target Data, as demonstrated in experiments [19], [20], to represent the percentiles of the values within the data. Furthermore, we adjusted the metrics to match the existing Anomaly Detection environment, which detects anomalies based on the top n% of values, by calculating P-Values as shown in Equation (6).

We then employed these P-Values as AI Decision Support metrics. We applied these P-Values to the MSE and the distance to identify Anomalies.

For True Positives, the MSE and distance tend to be higher than for False Positives, and this results in lower P-Values. Conversely, for False Positives, the MSE and distance tend to be lower than for True Positives, and this results in higher P-Values. We utilize these metrics to provide AI Decision Support and, through this approach, we propose an AI Decision Support framework that is suitable for the security field. This framework helps achieve False Alarm Reduction and enhance AI's performance in security applications.

6. EXPERIMENTAL RESULTS

1) Data Description and Data Featuring

Here preprocess the log data collected in the EDR environment and perform Anomaly Detection using unsupervised learning-based Autoencoders. The Original Log is in the format in which logs are recorded when a new process is executed.

Each log contains important information for identifying processes, such as a unique identifier field for the newly created process, parent process information, process name, process execution path. The recorded information comes in various formats, and fields like floating-point or integer types can be used as they are without the need for additional preprocessing.

However, for string fields, appropriate embedding techniques must be applied to represent the meaning of the string as numerical values for the model to work effectively. Furthermore, when log data are explored recursively, it is possible to extract the sequence of processes involved, which we refer to as the Process chain. In the case of normal processes, one would typically observe regular Process chains that occur during routine operations. However, a series of attack Process chains may appear related to malicious processes, as executing malicious scripts often involves a sequence of processes that would not occur in regular situations.

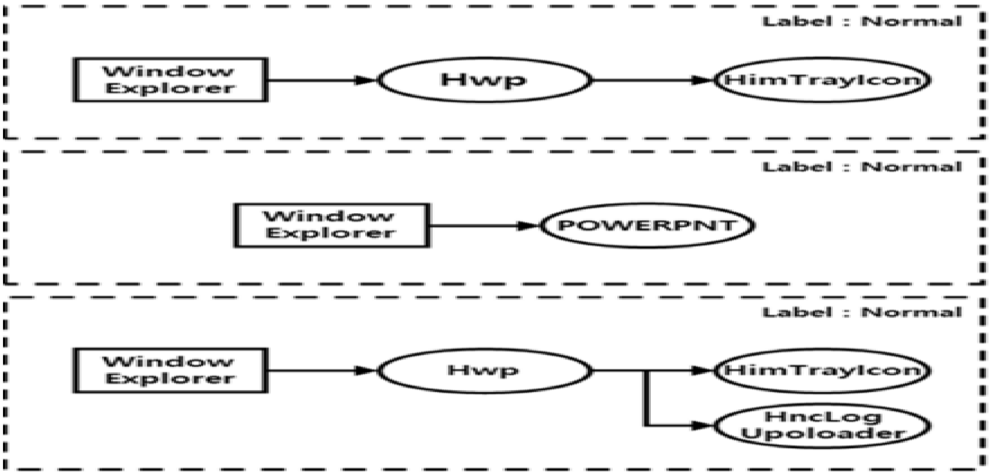


Figure 6.1.- Proposed algorithm of optimum reference calculation-Normal phase

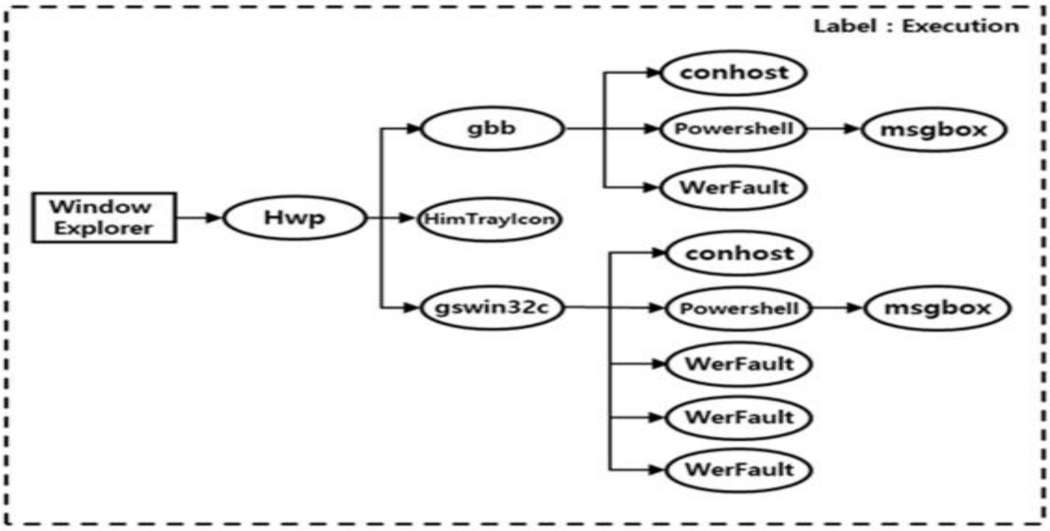


Figure 6.2- Proposed algorithm of optimum reference calculation-Attack phase

As a result, the Process chains observed in normal situations may be quite different from those observed during attacks, as illustrated in Figure 6.1, Figure 6.2. During attacks, one may notice the involvement of processes like Powershell.exe and gswin32c.exe, which are not typically seen in normal situations but are linked to the execution of malicious scripts.

Furthermore, in EDR environments, rapid response is crucial, so the log preprocessing steps must be lightweight. Considering this, this paper proposes a lightweight feature extraction technique that focuses on identifying process chains using the Original Log. It extracts only three key pieces of information: parent/child process identifiers, process names, and process image paths. The featuring targets two main aspects:

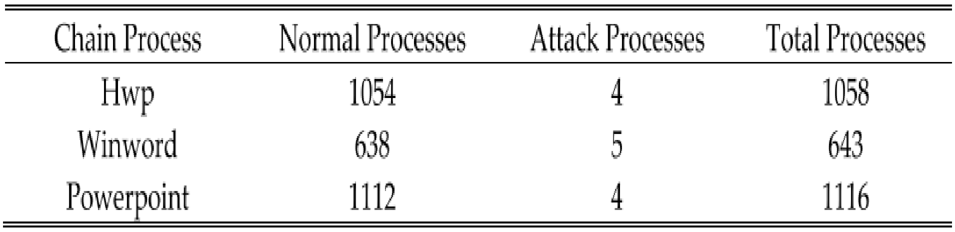
- **Parent_Process Featuring**

The parent process is the central process in the Process chain, and it connects various processes. By focusing on the paths of these parent processes, the paper embeds information about whether the Process chain was executed on an anomalous path. The path is tokenized at directory branches to feature path strings, and a 2-gram is applied. The resulting values are then hashed mapping(mod 20).

- **Merged_Child_Process Featuring**

The names of child processes that can be obtained through the Process chain are concatenated and tokenized at the character level using a 2-gram. These tokens are then hashed (mod 100). This approach detects anomalies when an unusual process is connected in the Process chain, resulting in a feature count increase or higher mapping values than for normal situations. Through the proposed featuring techniques, the paper incorporates Anomaly information from attacks that are present in the original logs as features. In the end, and feature results in 120 feature embeddings on a Process chain basis.

In this the Anomaly Detection is performed using Autoencoder models. The model we used consists of three encoder layers and three decoder layers. Each Encoder compresses the original features into representations of 75%, 50%, and 25% of the dimensions. The Decoder then reconstructs these compressed representations.

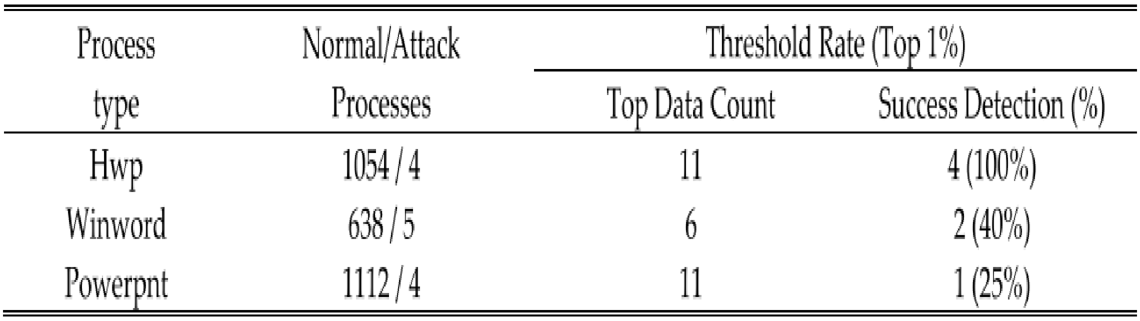


Chain Process	Normal Processes	Attack Processes	Total Processes
Hwp	1054	4	1058
Winword	638	5	643
Powerpoint	1112	4	1116

Table 6.1- Dataset description.

The dataset used is the same as that obtained through the previously mentioned Process Chain Feature technique, as described in Table 6.1.

Based on this dataset, three Autoencoder models were constructed to perform Anomaly Detection for three types of Process Chains. Anomaly determination was based on the top 1% of data having the highest MSE values.



Process	Normal/Attack	Threshold Rate (Top 1%)	
type	Processes	Top Data Count	Success Detection (%)
Hwp	1054 / 4	11	4 (100%)
Winword	638 / 5	6	2 (40%)
Powerpnt	1112 / 4	11	1 (25%)

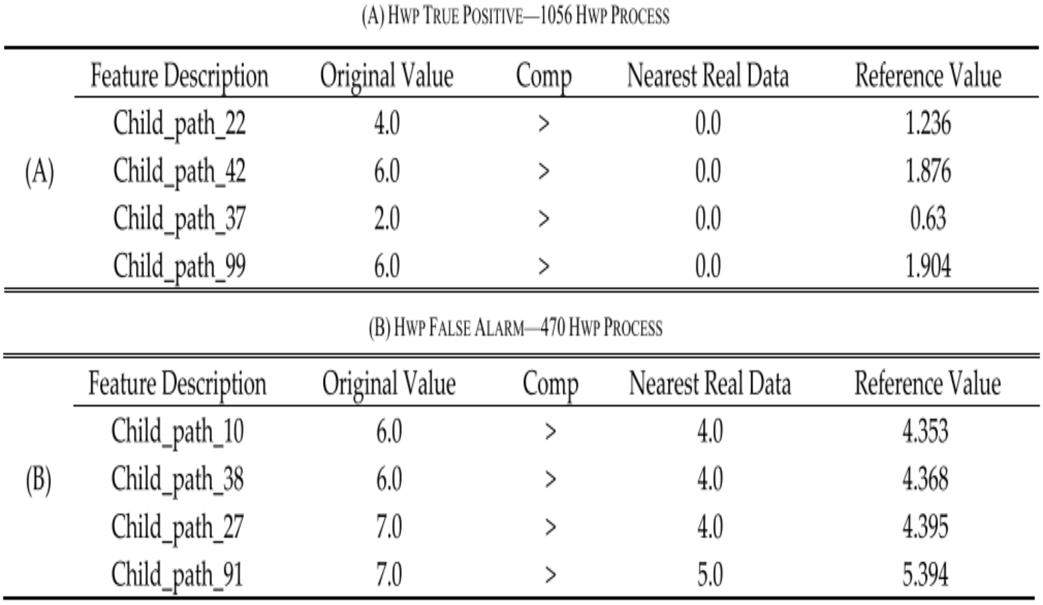
Table 6.2- Anomaly detection result.

The Anomaly Detection results on the Hwp dataset show that all attacks were detected with 100% accuracy among the top 1% of data. Unsupervised learning models like these have the advantage of being able to detect unknown threats in label-less environments.

However, they are also subject to significant performance variation based on specified thresholds and can produce numerous false alarms. Therefore, for effective unsupervised Anomaly Detection, it is essential to provide decision support for the detected anomalies. In cases of false alarms, the results of AI judgment should be carefully considered to reduce false alarms, while in cases of true positives, clear interpretations of the detections should be provided. This approach aims to support the effective utilization of unsupervised Anomaly Detection models in security.

2) Providing Effective Interpretation Through Feature Value Comparison

Aims to support interpretation in an unsupervised learning environment using References. The target data consist of two processes, the 1056th Hwp Process (true positive) and the 470th Hwp Process (false positive), which have the highest Anomaly score, MSE in different scenarios. By applying the framework proposed in this paper to these two processes and interpreting the Nearest Real Data generated, we obtain the results.



(A) Hwp True Positive—1056 Hwp Process

	Feature Description	Original Value	Comp	Nearest Real Data	Reference Value
(A)	Child_path_22	4.0	>	0.0	1.236
	Child_path_42	6.0	>	0.0	1.876
	Child_path_37	2.0	>	0.0	0.63
	Child_path_99	6.0	>	0.0	1.904

(B) Hwp False Alarm—470 Hwp Process

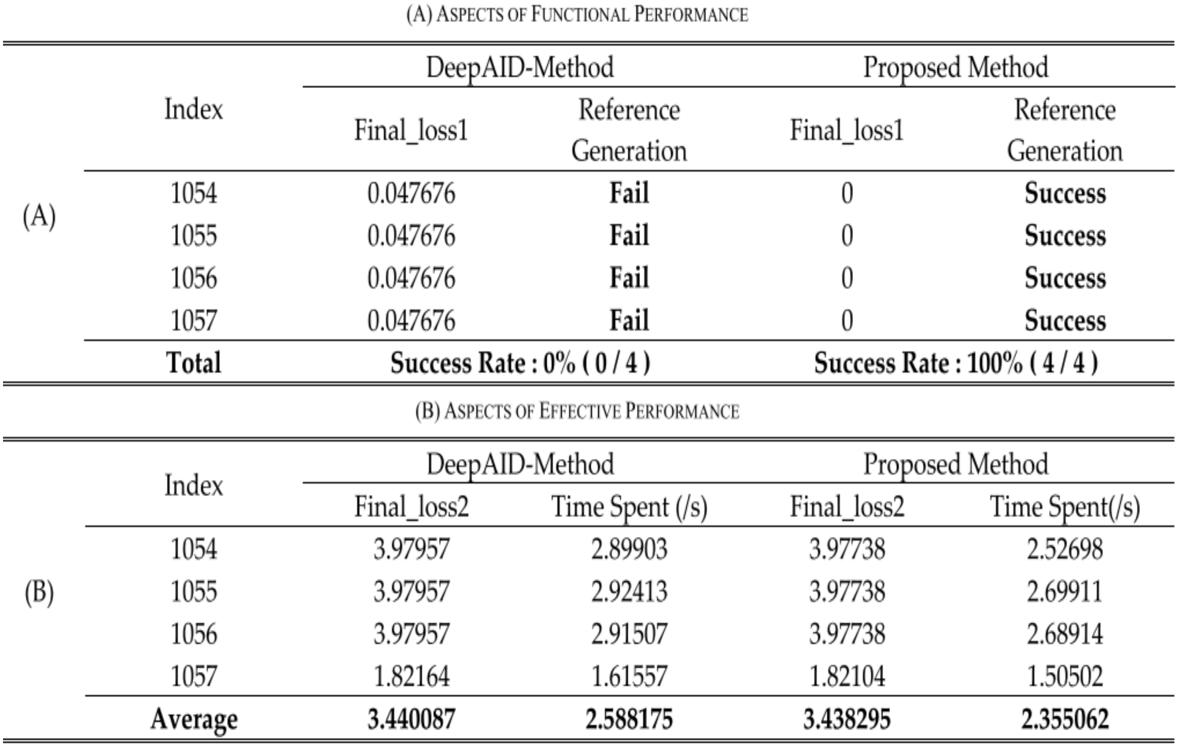
	Feature Description	Original Value	Comp	Nearest Real Data	Reference Value
(B)	Child_path_10	6.0	>	4.0	4.353
	Child_path_38	6.0	>	4.0	4.368
	Child_path_27	7.0	>	4.0	4.395
	Child_path_91	7.0	>	5.0	5.394

Table 6.3- Example of providing interpretation through reference

For the true positive data (A) in Table 6.3, we provide an interpretation that it is considered an Anomaly due to the features that are only present in attacks, such as Powershell.exe, which does not appear in normal states. When we compare the Feature Value Comparison results obtained from the true positive data with the Feature Value Comparison results in Table 6.3, for the false positive data (B), we can see that the false positive data show smaller differences.

This is because, in the case of true positives, attacks occur, and this introduces unusual features into the feature values. Therefore, the Nearest Real Data show a significant difference in the case of true positives and almost no difference in the case of false positives. These results confirm that the interpretations generated through Feature Value Comparison show larger differences as the Anomaly score increases, depending on the actual severity of the attack.

Furthermore, a comparison of the proposed Reference generation logic with the Reference generation techniques of prior research, such as that of Han.



(A) ASPECTS OF FUNCTIONAL PERFORMANCE

	Index	DeepAID-Method		Proposed Method	
		Final_loss1	Reference Generation	Final_loss1	Reference Generation
(A)	1054	0.047676	**Fail**	0	**Success**
	1055	0.047676	**Fail**	0	**Success**
	1056	0.047676	**Fail**	0	**Success**
	1057	0.047676	**Fail**	0	**Success**
	Total	**Success Rate : 0% (0 / 4)**		**Success Rate : 100% (4 / 4)**	

(B) ASPECTS OF EFFECTIVE PERFORMANCE

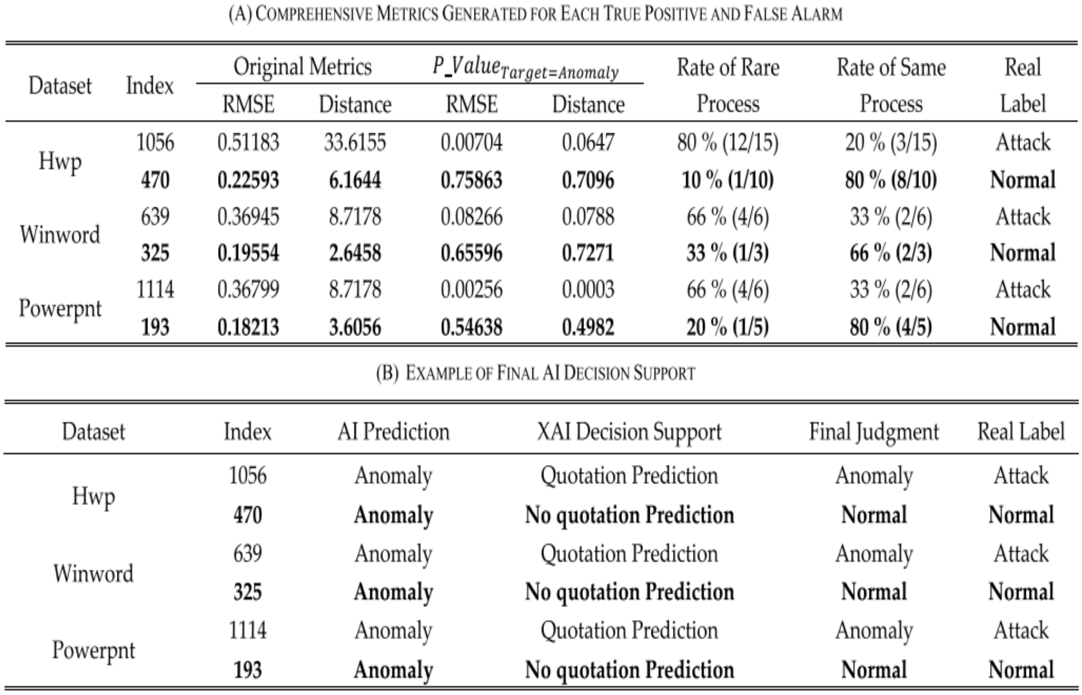
	Index	DeepAID-Method		Proposed Method	
		Final_loss2	Time Spent (/s)	Final_loss2	Time Spent(/s)
(B)	1054	3.97957	2.89903	3.97738	2.52698
	1055	3.97957	2.92413	3.97738	2.69911
	1056	3.97957	2.91507	3.97738	2.68914
	1057	1.82164	1.61557	1.82104	1.50502
	Average	**3.440087**	**2.588175**	**3.438295**	**2.355062**

Table 6.4- Performance comparison of reference generation.

Table 6.4 shows the results for generating References for the Hwp dataset. From a functional perspective, prior research fails to create References for four attacks as, despite performing detection until the last step, the final loss1 remains less than 0, and this prevents the w data from moving into the normal category.

In contrast, when the proposed method is used, then confirm that all four attacks successfully generate References, as the final loss1 is less than 0. Moreover, from a performance perspective, the fact that the final loss2 obtained through the proposed method shows a similar value to that of the previous method indicates that the security proposal logic is functioning correctly. Additionally, it is observed that the time required for Reference generation is further reduced.

3) Total Anomaly Decision Support



(A) COMPREHENSIVE METRICS GENERATED FOR EACH TRUE POSITIVE AND FALSE ALARM

Dataset	Index	Original Metrics		$P\_Value_{Target=Anomaly}$		Rate of Rare	Rate of Same	Real
		RMSE	Distance	RMSE	Distance	Process	Process	Label
Hwp	1056	0.51183	33.6155	0.00704	0.0647	80 % (12/15)	20 % (3/15)	Attack
	470	**0.22593**	**6.1644**	**0.75863**	**0.7096**	**10 % (1/10)**	**80 % (8/10)**	**Normal**
Winword	639	0.36945	8.7178	0.08266	0.0788	66 % (4/6)	33 % (2/6)	Attack
	325	**0.19554**	**2.6458**	**0.65596**	**0.7271**	**33 % (1/3)**	**66 % (2/3)**	**Normal**
Powerpnt	1114	0.36799	8.7178	0.00256	0.0003	66 % (4/6)	33 % (2/6)	Attack
	193	**0.18213**	**3.6056**	**0.54638**	**0.4982**	**20 % (1/5)**	**80 % (4/5)**	**Normal**

(B) EXAMPLE OF FINAL AI DECISION SUPPORT

Dataset	Index	AI Prediction	XAI Decision Support	Final Judgment	Real Label
Hwp	1056	Anomaly	Quotation Prediction	Anomaly	Attack
	470	**Anomaly**	**No quotation Prediction**	**Normal**	**Normal**
Winword	639	Anomaly	Quotation Prediction	Anomaly	Attack
	325	**Anomaly**	**No quotation Prediction**	**Normal**	**Normal**
Powerpnt	1114	Anomaly	Quotation Prediction	Anomaly	Attack
	193	**Anomaly**	**No quotation Prediction**	**Normal**	**Normal**

Table 6.5- Performance comparison of reference generation.

The comprehensive results are summarized in Table 6.5 and provide decision-support results for the unsupervised learning model. For Reference generation, one data point with the highest Anomaly Score from both the true and false positives was selected for each of the three dataset, resulting in six results, as specified. Table 6.5-(A) records the items and their corresponding results, which were created to support Decision Support. The meanings of each item are as follows: RMSE, which was P-Values were calculated based on the model's judgments to provide a clear interpretation of these values.

Additionally, to emphasize the clarity of the Anomaly causes, processes that appeared less frequently in each Process Type were counted, and the least frequent 2% of processes were designated as Rare Processes. This allowed the calculation of the Rare Process linkage ratio. Lastly, the similarity between the Process Chain generated from the Nearest Real Data and the original Process Chain was calculated to represent the similarity to normal processes intuitively.

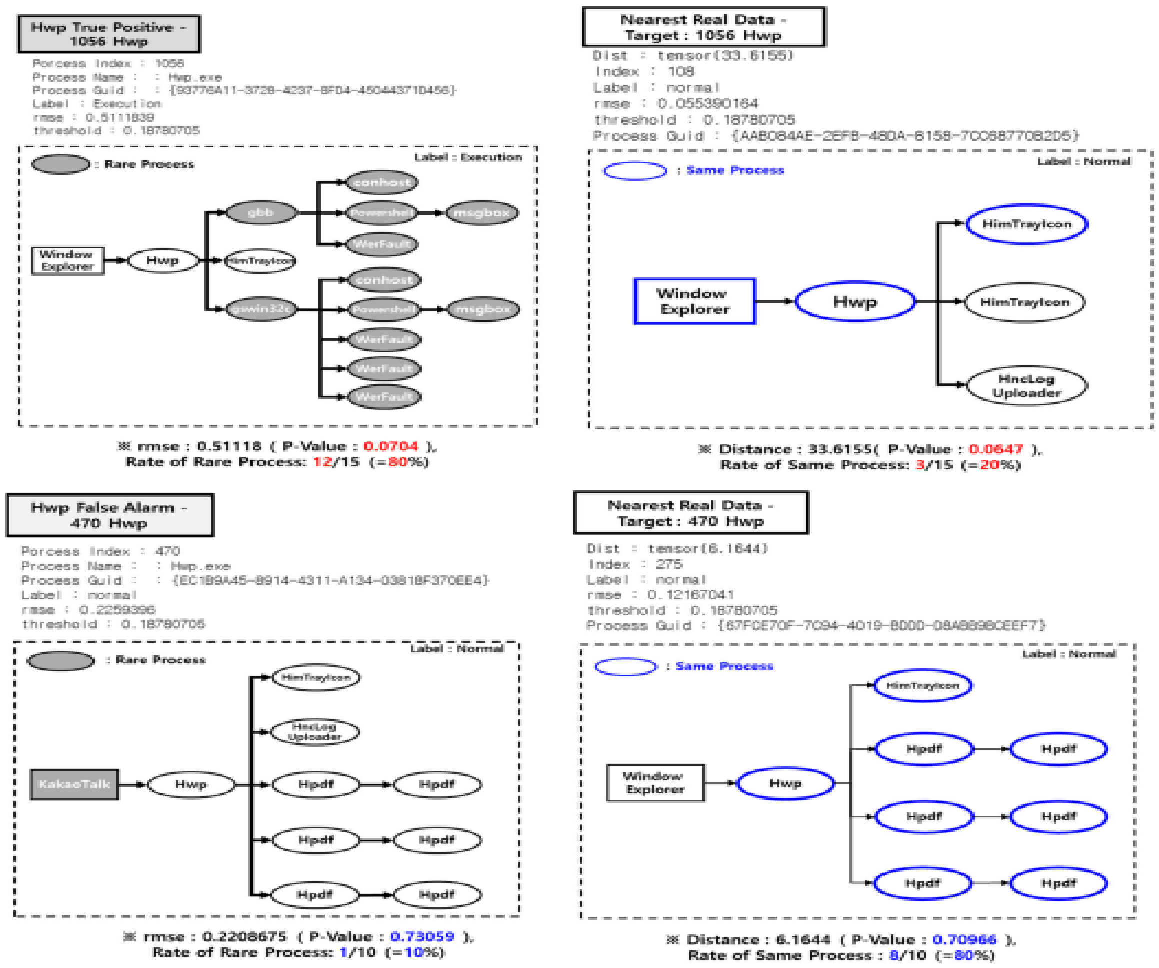
In Table 6.5-(A), Hwp 1056 and 470 represent true and false positives, respectively. The P-Values for their RMSE are 0.00704 and 0.75863. This means that the probability of having a stronger Anomaly tendency than 1056 Hwp is approximately 0.7%, and the probability of having a stronger tendency than 470 Hwp is 75.8%.

Therefore, it is evident that 1056 Hwp exhibits a significantly stronger Anomaly tendency than does 470 Hwp. This distinction is also identified through the Distancebased P-values, where 1056 Hwp has a P-value of 0.0647, whereas 470 Hwp has a P-value of 0.7096. Distance, as previously mentioned, represents the L_2 Norm value between the original data and the Nearest Real Data derived from it.

Therefore, a higher ‘‘Distance’’ indicates that the data is significantly distant from normal patterns, signifying a higher likelihood of it being an Anomaly. These results emphasize that 1056 Hwp is significantly further from normal processes and exhibits a notably higher level of Anomaly within the Anomaly group.

Additionally, the ratio of Rare Process associations is based on the frequency of occurrences, which means that all processes associated with attacks are categorized as Rare Processes. As a result, in processes where actual attacks occur, a high ratio of Rare Process associations is observed.

In contrast, the Rare Process association ratio is lower in false positive processes where no attacks occur. Finally, the degree of match between the original process and the Nearest Real Data process is also indicative. In the case of true positives, process chains that are significantly different from normal ones result in a lower match rate. However, process chains that are relatively similar to normal ones lead to a higher match rate for false positives.



```
Porcess Index : 1056
Process Name :  : Hwp.exe
Process Guid :  : {93776A11-3728-4237-8FD4-450443710456}
Label : Execution
rmse : 0.5111839
threshold : 0.18780705
```

※ rmse : 0.51118 (P-Value : 0.0704),
Rate of Rare Process: 12/15 (=80%)

```
Dist : tensor(33.6155)
Index : 108
Label : normal
rmse : 0.055390164
threshold : 0.18780705
Process Guid : {AAB084AE-2EFB-480A-8158-7C0687708205}
```

※ Distance : 33.6155(P-Value : 0.0647),
Rate of Same Process: 3/15 (=20%)

```
Porcess Index : 470
Process Name :  : Hwp.exe
Process Guid :  : {EC189A45-8914-4311-A134-03818F370EE4}
Label : normal
rmse : 0.2259396
threshold : 0.18780705
```

※ rmse : 0.2208675 (P-Value : 0.73059),
Rate of Rare Process: 1/10 (=10%)

```
Dist : tensor(6.1644)
Index : 275
Label : normal
rmse : 0.12167041
threshold : 0.18780705
Process Guid : {67FCE70F-7094-4019-B000-08A8898CEEF7}
```

※ Distance : 6.1644 (P-Value : 0.70966),
Rate of Same Process : 8/10 (=80%)

Figure 6.3- Result of topology visualztion for representatvie two samples of Hwp dataset (Up: Hwp true Positive – 1056 Hwp, Down: Hwp false alarm – 470 Hwp)

The results of visualizing the anomaly tendencies in the mentioned topology are shown in Figure 6.5. As identified in the figure, representative true positive of 1056 Hwp exhibits a distinct difference in topology compared to the most similar normal data. In contrast, the false positive detection of 470 Hwp shows a very similar topology when compared to the most similar normal data. Thus, true positive and false positive data show clear differences in topology.

Based on Table 6.5-(A), comprehensive analysis of the information, AI's Anomaly determination is referenced for the processes identified as anomalies, specifically 1056 Hwp, which is a clear true positive. Conversely, AI's determination is not cited for 470 Hwp, which is suspected to be a false positive. Consequently, AI decision support is provided. As illustrated in Table 6.5-(A), it can be observed that the differing interpretation trends between false positives and true positives are not limited to Hwp processes but also extend to processes of other types. Therefore, approaches like Table 6.5-(B) are employed to assist analysts in making decisions and enhance AI performance through false alarm reduction by leveraging the interpretation information derived from Table 6.5-(A).

7. CONCLUSION

The XAI technique based on Feature Value Comparison, as found that existing Feature Importance based XAI methods needed to provide clear interpretations in the security field. Therefore, proposed an AI Decision Support Framework that is suitable for the security environment using this technique. Demonstrated that this approach could provide interpretations, even in unsupervised learning environments, that are relevant to security and offer clearer interpretability in the security domain by providing a Comparison Interpretation based on actual Feature Values.

Furthermore, achieved performance and functional improvements relative to prior research through an enhanced generation logic. Here generated various metrics based on the produced Reference for AI Decision Support. Subsequently, when we comprehensively compared these metrics and observed significant differences between the true positive and false positive data in existing AI models. This allowed us to provide an interpretation and AI Decision Support and ultimately contribute to False Alarm Reduction and the enhancement of AI model performance.

8. FUTURE SCOPE

In this manner, the identified limitations in the application of conventional XAI techniques in security, and addressed them by developing a Feature Comparison-based XAI technique to provide clear interpretations. This aimed to support the effective utilization of unsupervised learning-based Anomaly Detection models in the security domain. In the future plan to capitalize on the advantages of this Framework to support Interpretation in unsupervised learning while transitioning to newly developed unsupervised learning models as central models. And will analyze the results produced, identify further areas for improvement, and enhance the AI Decision Support Framework to suit the security environment.

9. BIBLIOGRAPHY

Websites

[1] *https://www.balbix.com/insights/artificial-intelligence-in-cybersecurity/*

[2] *https://www.checkpoint.com/cyber-hub/cyber-security/what-is-ai-cyber-security/*

[3] *https://www.sophos.com/en-us/cybersecurity-explained/ai-in-cybersecurity/*

[4] *https://www.malwarebytes.com/cybersecurity/basics/risks-of-ai-in-cyber-security/*

[5] *https://www.techmagic.co/blog/ai-in-cybersecurity/*

Journal Papers

[1] *S. M. Lundberg and S. Lee, ‘‘A unified approach to interpreting model predictions,’’ in Proc. Adv. Neural Inf. Process. Syst., vol. 30, 2017.*

[2] *B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, ‘‘Learning deep features for discriminative localization,’’ in Proc. IEEE Conf. Com□put. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 2921–2929.*

[3] *J. Meira, R. Andrade, I. Praça, J. Carneiro, V. Bolón-Canedo, A. Alonso-Betanzos, and G. Marreiros, ‘‘Performance evaluation of unsupervised techniques in cyber-attack anomaly detection,’’ J. Ambient Intell. Humanized Comput., vol. 11, no. 11, pp. 4477–4489, Nov. 2020.*

[4] *C. Wheelus, E. Bou-Harb, and X. Zhu, ‘‘Tackling class imbalance in cyber security datasets,’’ in Proc. IEEE Int. Conf. Inf. Reuse Integr. (IRI), Jul. 2018, pp. 229–232.*

[5] *X. Xia, X. Pan, N. Li, X. He, L. Ma, X. Zhang, and N. Ding, ‘‘GAN-based anomaly detection: A review,’’ Neurocomputing, vol. 493, pp. 497–535, Jul. 2022*