Does Ordinality in Encoding Matter for XAI? A Case Study on Explainable IDS

Harry Chandra Tanuwidjaja[1[0000−0003−2668−559X]], Tao Ban[1[0000−0002−9616−3212]], Takeshi Takahashi[1[0000−0002−6477−7770]], Tsung-Nan Lin[2[0000−0001−5659−1194]], Boyi Lee[3[0009−0002−0426−505X]], and Muhamad Erza Aminanto[4[0000−0001−5614−2276]]

[1] National Institute of Information and Communications Technology, Tokyo, Japan
harryct@nict.go.jp, bantao@nict.go.jp, takeshi_takahashi@nict.go.jp
[2] National Taiwan University, Taipei, Taiwan
tsungnan@ntu.edu.tw
[3] National Applied Research Laboratories, Taipei, Taiwan
boyi@narlabs.org.tw
[4] Monash University Indonesia
erza.aminanto@monash.edu

Abstract. In this paper, we propose Fusion SHAP, a novel algorithm designed to overcome the limitations of SHAP in handling one-hot encoded features. While SHAP provides individual explanations for each feature, it struggles with generating meaningful insights for categorical variables represented through one-hot encoding. Recent publications have attempted to address this problem by using label encoding, which theoretically may introduce issues related to ordinality. Fusion SHAP addresses this by offering both individual explanations for each one-hot encoded feature and a merged explanation that considers the collective impact of these features. We present the mathematical proof for the correctness of our algorithm and demonstrate its effectiveness through experiments on a network intrusion detection system (IDS) dataset. Our results compare the explanations generated by Fusion SHAP with those of the original SHAP algorithm under one-hot encoding and label encoding. Specifically, we evaluate the explanations for seven different types of cyberattacks, including DDoS, Backdoor, Injection, Password, Ransomware, Scanning, and XSS. Finally, we analyze the correlation between feature importance and attack type, showcasing the advantages of Fusion SHAP in enhancing the interpretability of IDS models.

Keywords: XAI · Fusion SHAP · IDS · one-hot encoding · feature importance.

1 Introduction

Intrusion Detection Systems (IDS) play a pivotal role in safeguarding computer networks against unauthorized access and malicious activities. However, the complexity of these systems, often powered by advanced machine learning models,

poses a significant challenge in terms of interpretability and transparency. This paper explores the application of Explainable Artificial Intelligence (XAI) techniques to enhance the interpretability of IDS, specifically focusing on the use of Fusion SHAP for model interpretability. By integrating XAI with IDS, we aim to provide a deeper understanding of how these models make decisions, thereby increasing trustworthiness and enabling more informed decision-making by security professionals. Our experimental results demonstrate the effectiveness of Fusion SHAP in identifying the most influential features and explaining individual predictions, contributing to improved model transparency and security in network defense.

In the context of SHAP, which is a method to explain the output of machine learning models, one-hot encoding plays a crucial role when dealing with categorical data [7]. When categorical data is present, many machine learning models cannot directly handle these variables since they are non-numeric. One-hot encoding transforms each categorical variable into a new categorical variable with a binary value to represent the presence or absence of each category. Due to the challenges in directly applying the SHAP library to one-hot encoded features, where the explanation of the original categorical feature can be fragmented across multiple binary features, many studies opt for label encoding despite its limitations [3], [9], [6], [2]. Label encoding assigns a unique integer to each category, maintaining low dimensionality but implying an ordinal relationship that may not actually exist, potentially misleading the model about the nature of the categorical data.

The contributions of our paper are:

- **We propose Fusion SHAP**, an algorithm to address the **limitation of SHAP** in handling the **one-hot encoded features**.
- The **proposed algorithm** provides the **individual explanation** of each one-hot encoded feature and also the **merged explanation** of all those one-hot encoded features.
- We show the explanation result of **our proposed method and the original SHAP** when using one-hot encoding,
- Then, we also **compare** the explanation result of **our proposed method (Fusion SHAP and one-hot encoding)** with **state-of-the-art method (Original SHAP and label encoding)** [9, 4, 10, 5].
- We presented the explanation result for **seven types of attack** on IDS dataset: **DDoS, Backdoor, Injection, Password, Ransomware, Scanning, and XSS.**
- Finally, we analyze the **correlation** between the **feature importance and the type of attack.**

The remainder of this paper is organized as follows. Section 2 outlines the methodology used in this study, detailing the proposed approach, data preprocessing steps, and algorithms applied. Section 3 presents the experiment results and analysis, where we evaluate the performance of the proposed methods, providing insights through comparative analyses. Finally, Section 4 concludes the paper with a summary of key findings and discussions on potential future work.

2 Methodology

2.1 Dataset Details

The TON_IoT dataset [8], [1] is a comprehensive benchmark dataset, tailored for cybersecurity research, particularly within the domain of Internet of Things (IoT) networks. We use the network traffic data from the dataset, which represents the communication between IoT devices and the surrounding network infrastructure. This data offers insights into the behavior of IoT devices, both under normal operations and during malicious events. It has 46 features with more than 400,000 instances that are categorized into seven types of attacks: backdoor, DDoS, injection, password, ransomware, scanning, and XSS.

2.2 Experiment Environment

In this paper, we performed a classification to detect whether an instance is an attack or benign. Our primary classifier for this task is the random forest algorithm, which was selected due to its robustness and ability to handle complex datasets efficiently. We trained the model and achieved 99.02% accuracy. We prioritize explainability over model performance; therefore, we opt for a simpler model rather than a more complex one.

Preprocessing Prior to model training, we performed several data preprocessing steps to clean and optimize the dataset:

- Feature Removal: Certain features such as IP addresses, source IP, source port, and destination IP were excluded, as they are not relevant for classification and could introduce noise into the model.
- Data Cleaning: Missing or inconsistent data points were handled by employing standard data cleaning techniques.
- Normalization: To ensure that the data was appropriately scaled for the random forest model, we applied normalization to continuous features.

Encoding of Categorical Features We explored two different encoding strategies for the categorical features:

- Label Encoding: Categorical values were converted into numerical values representing the class labels. For this scenario, after training the Random Forest model, we used the original SHAP algorithm to generate explanations for the classification results.
- One-Hot Encoding: In this scenario, categorical features were transformed into a series of binary columns. After training the model, we used our proposed explanation method, Fusion SHAP, which extends the functionality of the original SHAP by merging explanations for one-hot encoded features. This method enables us to capture both individual and aggregated feature importance.

Explanation Methods

- For the Label Encoding scenario, we utilized the original SHAP algorithm to explain the classification decisions made by the Random Forest model.
- For the One-Hot Encoding scenario, we applied Fusion SHAP, a novel explanation method developed in this study. Fusion SHAP improves on the limitations of the original SHAP algorithm by providing both individual feature explanations and combined feature insights, specifically for one-hot encoded data.

2.3 Fusion SHAP Algorithm

Algorithm 1 Fusion SHAP Algorithm

Step 1: Extracts the feature names of the one-hot-encoded columns and saves it to an array
`ohename` ← `one-hot encoded feature names`
Step 2: Create SHAP Explanation for One-Hot-Encoded Columns
`new_object` ← `shap.Explanation(shap_values.values[:, ohename])`
Step 3: Construct New Feature Data
`new_data` ← `(new_object.data * (sum(ohename)))`
Step 4: Updating SHAP Value
if shap_values.display_data == true **then**
    `newdd` ← `shap_values.display_data[:, ∼ ohename]`
    `newdd_data = concatenate(newdd, ohename[new_data])`
else
    `newdd_data` ← `shap_values.data[:, ∼ ohename]`
end if
Step 5: Combine SHAP Values
new_sv ← new_object.values.sum
svvalues ← concatenate((shap_values.values[:, ohename], new_sv))
Step 6: Construct New SHAP Explanation Object
sv ← shap.Explanation(svvalues,
    other SHAP metadata including the sv.data, newdd_data, and updated feature names)
Step 7: Return Final SHAP Object
if return_original == true **then**
    return sv, new_object
else
    return sv
end if

Our proposed algorithm, the Fusion SHAP, addresses the weakness of SHAP, which generates individual explanations for one hot encoded feature. The algorithm allows combining those one-hot features into a single categorical feature, making the SHAP explanations more interpretable by reducing the number of features and maintaining the total SHAP value contribution. Our algorithm provides two outputs: the subset explanation object and the aggregated explanation

object. The subset explanation object contains the individual explanation for the one-hot encoded features. On the other hand, the aggregated explanation object contains the explanation for the whole features that cover the non-categorical features and the aggregated one-hot encoded features.

Here is the detail of our algorithm:

1. **One-Hot Encoding Aggregation**: The algorithm assumes that the input `shap_values` represents the Shapley values for a one-hot-encoded categorical feature. The indices corresponding to the one-hot-encoded variables are indicated by the `ohename`. The Fusion SHAP algorithm aggregates these Shapley values to reflect the contribution of the combined categorical feature.

2. **Defining the Characteristic Function** v: Let's define the characteristic function $v(S)$ as the model's contribution for a subset $S \subseteq N$ of attributes. When we combine the features $\{1, \ldots, k\}$, we consider their joint contribution:

$$\Phi_k(v) = \sum_{S \subseteq N \setminus \{1,\ldots,k\}} \frac{|S|!(n - |S| - k)!k!}{n!} \left[v(S \cup \{1, \ldots, k\}) - v(S) \right].$$

3. **Contribution Summation**: In the Fusion SHAP algorithm, the summed Shapley values of the one-hot-encoded features ($\{1, \ldots, k\}$) represent the total effect of the categorical feature. We calculate the total Shapley value of the combined feature by summing individual contributions. Mathematically, this can be viewed as:

$$\Phi_k(v) = \sum_{i=1}^{k} \Phi_i(v),$$

where $\Phi_i(v)$ represents the Shapley value of each individual one-hot-encoded binary attribute. This summation is justified because the Shapley value is additive for disjoint feature sets.

4. **Restructuring the Data and Values**: We create a new feature representing the combined categorical attribute. This involves:

- Generating a new feature value by combining the one-hot-encoded attributes into a single integer.
- Concatenating these new feature values into the existing dataset.
- Updating the Shapley values by appending the aggregated sum to the existing values.

5. **Additivity and Permutations**: The Shapley value's additivity property ensures that the sum of the individual Shapley values across all features equals the sum of Shapley values after combining the one-hot-encoded features.

The total contribution before and after combining the features remains consistent due to this additivity:

$$\Phi_N(v) = \sum_{i=1}^{n} \Phi_i(v) = \Phi_{N \setminus \{1,\ldots,k\}}(v) + \Phi_k(v).$$

6. **Correctness of Shapley Value Calculation**: Since the sum of the individual Shapley values of the one-hot-encoded features ($\Phi_k(v)$) directly represents

their joint contribution to the model, the algorithm preserves the Shapley values' distribution across the feature set.

7. **Final Proof Statement**: By summing the individual Shapley values for the one-hot-encoded features and creating a new combined feature, the `combine_one_hot` function effectively captures the overall contribution of the categorical feature $\{1, \ldots, k\}$. The function respects the Shapley value formula:

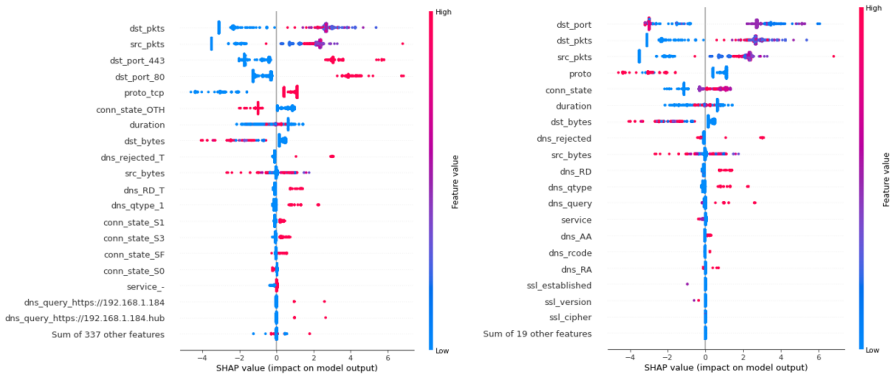
$$\Phi_k(v) = \sum_{S \subseteq N \setminus \{1,\ldots,k\}} \frac{|S|!(n - |S| - k)!k!}{n!} \left[v(S \cup \{1, \ldots, k\}) - v(S) \right],$$

by ensuring that the new combined feature's Shapley value correctly reflects the sum of the one-hot-encoded components' contributions. Thus, the correctness of the algorithm is guaranteed through the properties of additivity and distribution inherent to the Shapley values.

3 Experiment Result and Analysis

In this section, we will compare the explanation results of Fusion SHAP and the original SHAP to demonstrate the improvements offered by our algorithm. Following that, we will analyze the explanation results for each type of attack using two types of encoding: one-hot encoding and label encoding. We will compare the explanation results for these two encoding types to assess whether the ordinality issue matters significantly in this case study.

3.1 Fusion SHAP vs Original SHAP for OHE



(a) Beeswarm Plot of Original SHAP (b) Beeswarm Plot of Fusion SHAP

Fig. 1. Comparison of explanation result between original SHAP and Our Approach

Fig. 1 compares the explanations generated by Fusion SHAP and the original SHAP when using one-hot encoding. Original SHAP provides individual explanations for each newly generated OHE feature, which can obscure the overall explanation of the original features. By applying our algorithm, we merge the explanations of these OHE features, resulting in a clearer overall explanation. We also observe that the feature importance ranking changes before and after the merging. For instance, the destination port becomes the top-ranking feature after merging all the individual SHAP values for each port. This demonstrates that Fusion SHAP offers improved explanations, aiding operators in more effectively identifying important features compared to the original SHAP when one-hot encoding is used.

3.2 DDoS

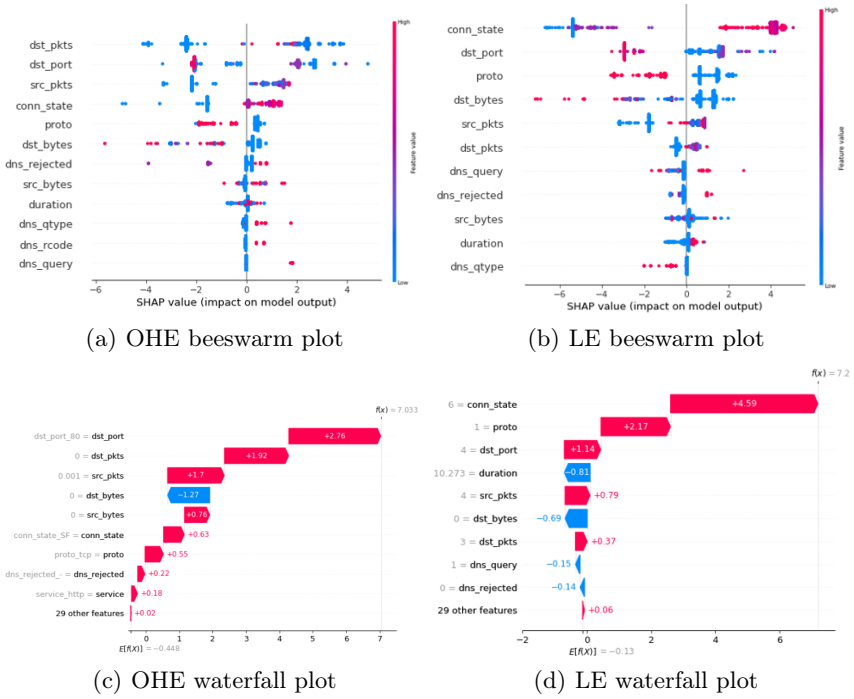
Table 1. Defined rule for DDoS attack

Attack Type	**Algorithm**	**Feature Importance Result**		**Analysis Summary**
DDoS	Fusion SHAP with OHE	Destination packet	large	Provide better insight for packet-related features
		Destination port	80,443	
		Source packet	varies	
		Connection state	S3,SHR	
	Original SHAP with LE	Connection state	varies	Aligning insight for protocol and traffic-related features
		Destination port	varies	
		Protocol	TCP	
		Destination bytes	varies	

Destination port aspect During a DDoS attack, there is a high volume of traffic to a specific port. We can see in Fig.3(a) that ports 80 and 443 are being targeted. Our algorithm also provides the individual explanation for each categorical feature, so that we can have a more detailed analysis.

Connection state aspect Fig. 3(b) shows that if the connection state is S3, SHR, or SF; there is a high chance that the DDoS attack is currently occurring.

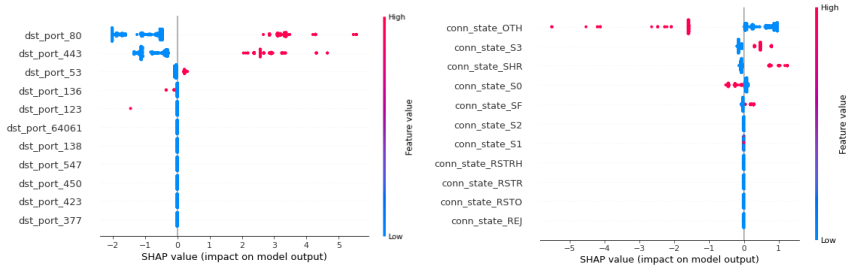
- **S3 (SYN Sent, Established, and FIN Received)**: During some types of DDoS attacks, such as a SYN flood, the attacker sends a large number of SYN packets to overwhelm the target. The presence of many S3 states in a short time period could indicate a flood of connection attempts that are initiated and closed rapidly, which can disrupt the server's normal operations.
- **SHR (SYN Sent, SYN-ACK Received, Host Reset)**: A high number of SHR states can occur during certain DDoS attacks, particularly those that exploit the TCP three-way handshake. Attackers may initiate connections and then abruptly reset them, flooding the target server with half-open or reset connections.



(a) OHE beeswarm plot (b) LE beeswarm plot

(c) OHE waterfall plot (d) LE waterfall plot

Fig. 2. Comparison of Explanation Result for DDoS Attack



(a) Destination port beeswarm plot (b) Connection state beeswarm plot

Fig. 3. Individual Explanation for OHE features in DDoS attacks

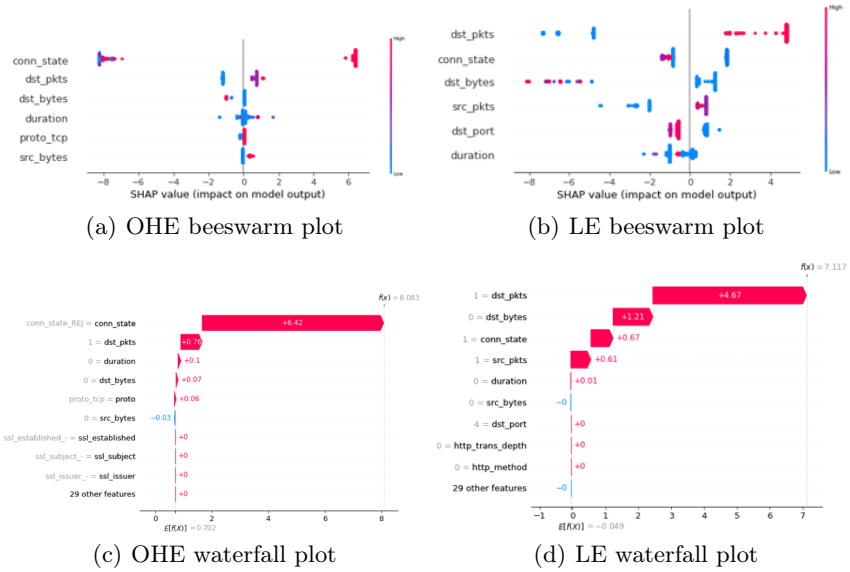
Comparison between OHE and LE Fig. 2(a) and Fig. 2(b) shows the beeswarm plots of OHE and LE, respectively. There is a similarity among the top features, although their order is mixed. The differences in the beeswarm plots are reflected in the decision-making process, as shown by the waterfall plots (Fig.2(c) and Fig. 2(d)). We used the same sample and generated waterfall plots for both the OHE and LE scenarios. OHE tends to assign more weight to packet-related features, while LE emphasizes the connection state. Theoretically, packet-related features are more important for high-bandwidth, volumetric DDoS attacks where the goal is to flood the network with massive traffic. Packet-related features give an immediate view of traffic volume and anomalies. On the other hand, connection state-related features are more crucial for application-layer and protocol-based attacks that exploit connection mechanisms, like SYN floods or HTTP floods. Connection states reveal how the server's connection-handling mechanisms are being overwhelmed. In summary, Connection state is important in both methods, though in Fusion SHAP, it comes lower in importance compared to original SHAP. Destination port appears in both methods as a significant feature, but the order differs slightly. Destination packet and source packet are unique to Fusion SHAP, while protocol and destination bytes are emphasized by the original SHAP.

3.3 Backdoor

Table 2. Defined rule for backdoor attack

Attack Type	Algorithm	Feature Importance	Result	Analysis Summary
Backdoor	Fusion SHAP with OHE	Connection state	REJ	Provide a more nuanced explanation, due to its inclusion of duration and protocol.
		Destination packet	large	
		Destination byte	varies	
		Duration	varies	
	Original SHAP with LE	Destination packet	large	Emphasis on source packets, Helpful if the attack involves abnormal outbound traffic.
		Connection state	REJ	
		Destination byte	varies	
		Source packet size	large	

Connection state aspect As shown in Fig. 5, connection state REJ emerged as the decisive feature. A REJ state indicates that a connection attempt was rejected, usually because the target device refused to accept the connection request. Correlating the behavior of backdoor attack and REJ state, we can analyze that before deploying a backdoor, attackers often perform reconnaissance to gather information about the target network. During this phase, attackers might attempt connections to various services or ports to identify vulnerabilities. If these connection attempts are made to closed or restricted ports, the resulting connection state could be REJ. A high number of rejected connections to various ports



(a) OHE beeswarm plot (b) LE beeswarm plot

(c) OHE waterfall plot (d) LE waterfall plot

Fig. 4. Comparison of Explanation Result for Backdoor Attack

could indicate scanning or probing activities often associated with the initial stages of a backdoor attack.

Network traffic aspect Backdoor attacks can manifest as either frequent low-volume communication or sudden high-volume transfers, affecting both the destination bytes and destination packets metrics. Due to these characteristics, the explanation results show that this feature cannot stand alone and requires other features in the decision-making process.

Comparison between OHE and LE Fig. 4(a) and Fig. 4(b) present the beeswarm plots for OHE and LE, respectively. While the top features are similar in both plots, their order varies. Fusion SHAP ranks connection state as the

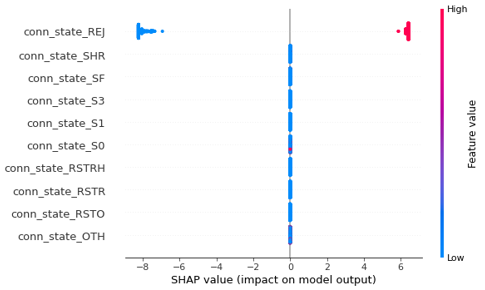
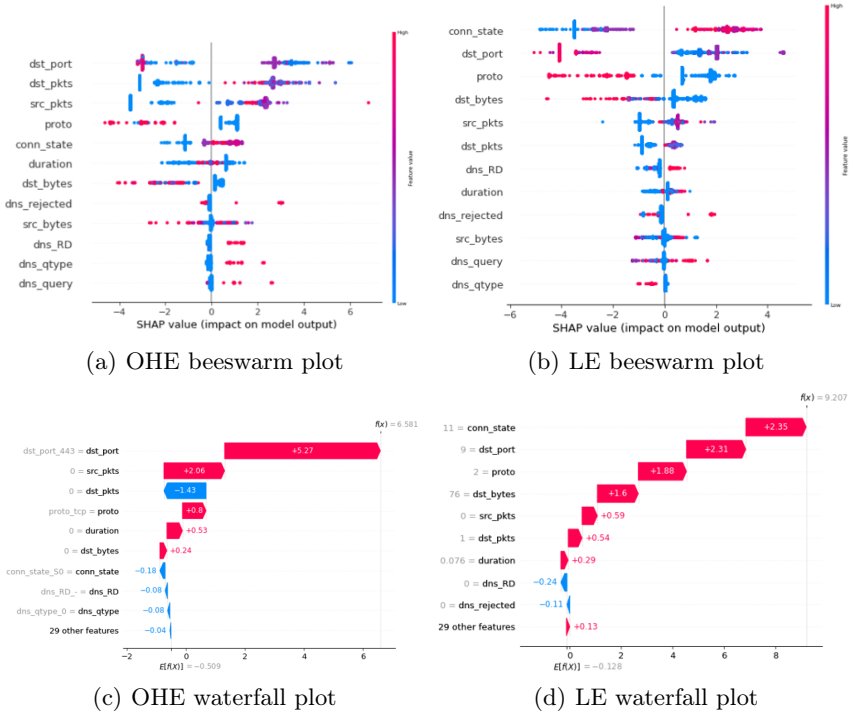


Fig. 5. Individual Explanation result for OHE Feature in Backdoor Attack

most important feature, while Original SHAP ranks it second. Connection state is a critical indicator for backdoor attacks, as these attacks often manipulate or maintain specific connection states. Fig. 4(c) and Fig. 4(d) show the waterfall plots of OHE and LE, respectively. We analyze the same instance detected as a backdoor attack in both encoding scenarios. For OHE, the connection state is the dominant factor in determining that the instance is an attack. In contrast, for LE, the decisive factors include the accumulation of destination packets, destination bytes, connection state, and source packets. This is consistent with the characteristics shown in the beeswarm plot. Fusion SHAP seems to provide a more nuanced explanation, particularly by highlighting connection state, packet traffic, and duration. Fusion SHAP offers better feature importance results due to its inclusion of duration. However, depending on the specific behavior of the backdoor attack, both methods have their merits.

3.4 Injection



(a) OHE beeswarm plot (b) LE beeswarm plot

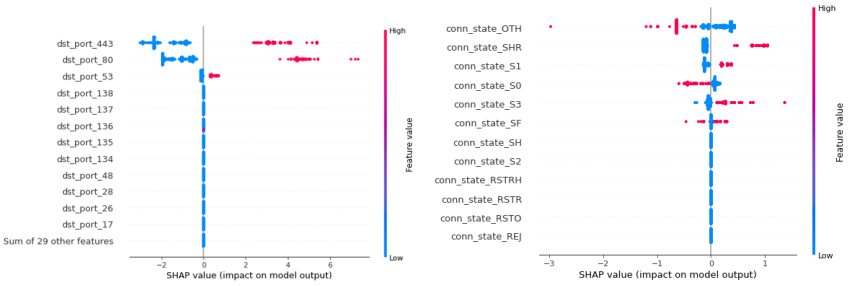
(c) OHE waterfall plot (d) LE waterfall plot

Fig. 6. Comparison of Explanation Result for Injection Attack

Network traffic aspect Table 3 shows that the injection attack is indicated by high destination packets and high source packets. Theoretically, if a web server

Table 3. Defined rule for Injection attack

Attack Type	Algorithm	Feature Importance Result		Analysis Summary
Injection	Fusion SHAP with OHE	Destination port	443,80	Provide more granular insight for packet-related features
		Destination packet	large	
		Source packet	large	
		Protocol	TCP	
	Original SHAP with LE	Connection state	varies	Focusing more on connection metadata
		Destination port	varies	
		Protocol	TCP	
		Destination bytes	varies	



(a) Destination port beeswarm plot (b) Connection state beeswarm plot

Fig. 7. Individual Explanation for OHE features

suddenly starts receiving a large number of packets directed at it within a short time frame, this could implicate an ongoing injection attack. Conversely, an injection attack that probes for vulnerabilities might involve sending many small packets to a destination in a short amount of time, leading to an unusually high number of sessions with small byte counts. An unexplained spike in destination bytes might indicate a large payload being sent in an attempt to exploit a vulnerability. If a destination consistently receives a high volume of bytes, it might suggest automated attacks or data exfiltration efforts.

Destination port and protocol aspect Injection attacks often target specific applications or services that run on particular protocols. For example, SQL Injection typically occurs over protocols like HTTP (port 80) or HTTPS (port 443), targeting web applications that interact with a database. This theory aligns with our experimental results, as shown in Fig. 7(a); most of the detected injection attacks occur on port 443 and port 80.

Connection state aspect Injection attacks often result in unusual connection patterns. For example, a successful SQL injection might cause a database connection to remain open for longer than normal as the attacker retrieves data. Conversely, repeated unsuccessful attempts may result in many short-lived connections. Some injection attacks may involve incomplete sessions, where a connection is initiated but never properly terminated. This could happen if an at-

tacker is probing for vulnerabilities without completing a full connection handshake or is exploiting a service in a way that leaves the session in a half-open state or equivalent. Our explanation results in Fig. 7(b) show that some abnormal connections like SHR, S1, S3, and SF indicate an injection attack.

Comparison between OHE and LE Then, we correlate these beeswarm plots to the results from the waterfall plots shown in Fig. 6(c) and Fig. 6(d). The results of OHE place more emphasis on the destination port, source packets, and destination packets, with the destination port being the dominant factor. In contrast, the results of LE indicate that connection state, destination port, protocol, and destination bytes are the key factors in detecting an attack, each with almost equal weight.

3.5 Password

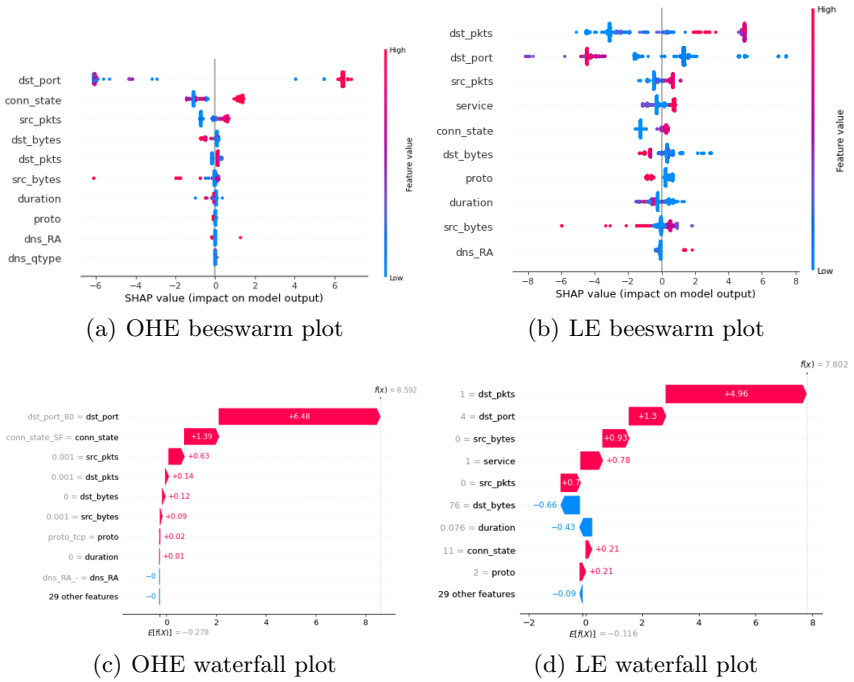
Table 4. Defined rule for password attack

Attack Type	Algorithm	Feature Importance Result		Analysis Summary
Password	Fusion SHAP with OHE	Destination port	80,21	The top features directly tied to behaviors seen in password-based attacks
		Connection state	SF	
		Source packet	large	
		Destination byte	varies	
	Original SHAP with LE	Destination packet	large	Pointing to a higher-level indicator but is less specific about password-based attacks' behavior
		Destination port	varies	
		Source packet	varies	
		Service	varies	

Destination port aspect Password attacks often target specific ports associated with services that require authentication. The explanation results in Fig. 9(a) showed that port 80 (HTTP) is the most frequently targeted port. For the other port, the impact on the model output is low, meaning that the password attacks in the dataset are distributed across a wide range of ports.

Connection state aspect In relation to connection state, password attacks often involve repeated attempts to connect to a service to guess or brute-force credentials. As shown in Fig. 9(b), connection state SF has the most significant impact on the decision-making process compared to other connection states. SF indicates that the connection is established and completed successfully. A password attack that tries numerous login attempts will generate many short-lived, normal-looking connections that complete the TCP handshake, perform the authentication attempt, and then close properly, hence resulting in a large number of connections marked with the SF state.

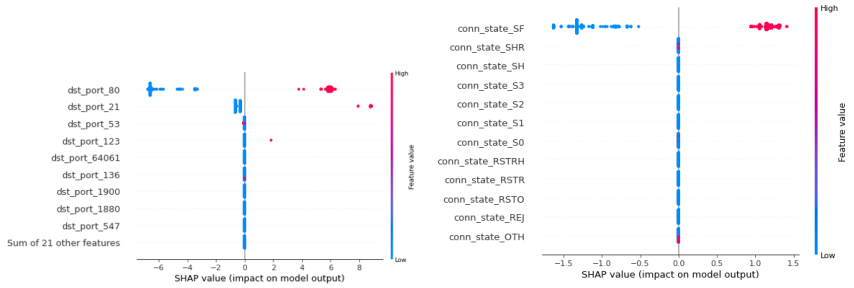
Network traffic aspect From the OHE beeswarm plot (Fig. 8(a)), we can identify that there is a positive correlation with destination packets since these attacks typically involve sending a large number of packets. It also has a negative



(a) OHE beeswarm plot (b) LE beeswarm plot

(c) OHE waterfall plot (d) LE waterfall plot

Fig. 8. Comparison of Explanation Result for Password Attack



(a) Destination port beeswarm plot (b) Connection state beeswarm plot

Fig. 9. Individual Explanation for OHE features

or low correlation with dst_bytes since password attacks are characterized by sending many small packets rather than large data transfers. We also note that there is a variable connection with source bytes and source packets because it depends on how the target system responds to repeated login attempts.

Comparison between OHE and LE Fig. 8(a) and Fig. 8(b) present the beeswarm plots for OHE and LE, respectively. To detect a password attack, OHE primarily relies on the destination port, along with some influence from

the connection state and source packets. In contrast, LE depends on destination packets and the destination port, with a smaller contribution from source packets. From the waterfall plot shown in Fig. 8(c) and Fig. 8(d), the destination port has the greatest weight for OHE, while destination packets hold the most significant weight for LE in identifying a password attack. In summary, Fusion SHAP's result, focusing on destination port and connection state, seems more aligned with the behavior expected in password attacks, while Original SHAP's destination packet and service might be broader but less specific.

3.6 Ransomware

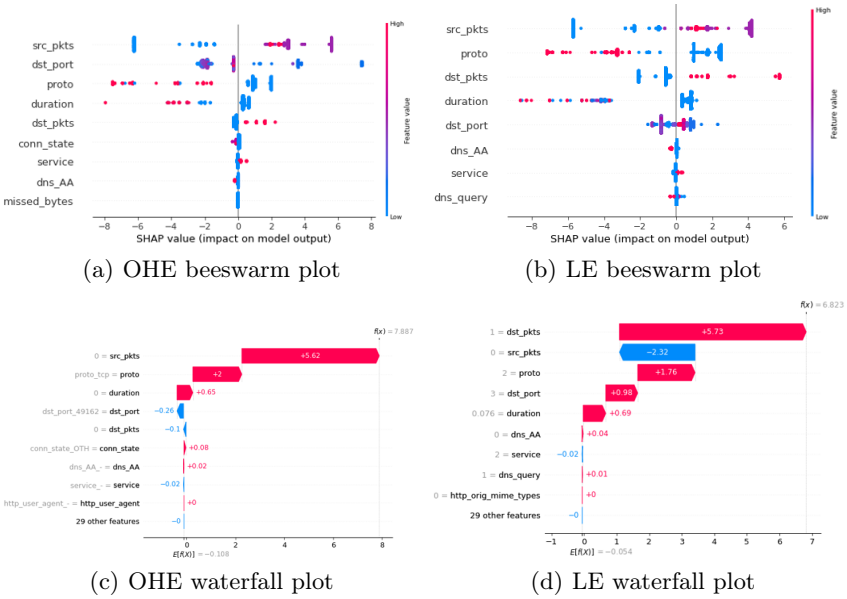
Table 5. Defined rule for ransomware attack

Attack Type	Algorithm	Feature Importance Result		Analysis Summary
Ransomware	Fusion SHAP with OHE	Source packet	80,21	Focus more on the source of the attack and the nature of communication, which in line with ransomware behavior
		Destination port	SF	
		Protocol	large	
		Duration	varies	
	Original SHAP with LE	Destination packet	large	Emphasize destination-related behavior, which could be critical when ransomware is encrypting files on targeted systems
		Destination port	varies	
		Source packet	varies	
		Service	varies	

Destination port aspect Ransomware attacks may utilize uncommon or unexpected ports to avoid detection or blend in with normal traffic. For example, if an IoT device rarely uses a specific port under normal operation but suddenly increases traffic to that port, it might indicate a ransomware attack or other malicious activity. While ransomware can technically use any port, some attacks may target common ports like 445 (SMB) for spreading or exploiting vulnerabilities. Fig. 11(a) shows that port 445 has a higher impact compared to other ports, validating the theoretical behavior of ransomware that tends to attack SMB ports.

Network traffic aspect During a ransomware attack, the number of packets sent may increase significantly. Ransomware often communicates with external servers to receive instructions or encryption keys, which can result in an unusual number of packets being sent and received. Unusual or unexpected traffic patterns, such as a high number of packets to a single destination or rapid bursts of packets, also could indicate ransomware activity. Fig. 10(a) and Fig. 10(b) show that high source packets and destination packets indicate ransomware attacks.

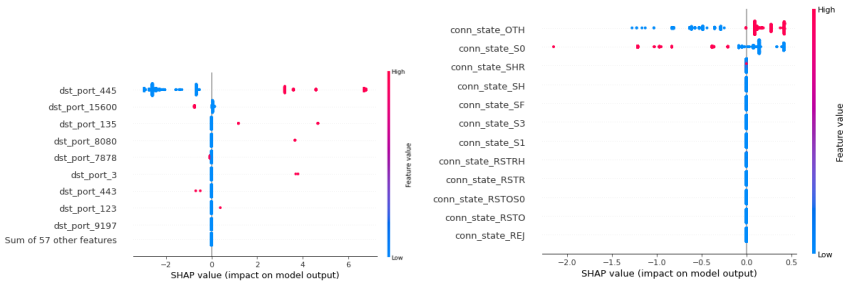
Connection state aspect Our experiment result, as shown in Fig. 11(b) indicated that connection states OTH and S0 have a high impact on detecting ransomware attacks. The OTH state is positively correlated with the attack, while the S0 state is negatively correlated with the attack. Connections marked



(a) OHE beeswarm plot (b) LE beeswarm plot

(c) OHE waterfall plot (d) LE waterfall plot

Fig. 10. Comparison of Explanation Result for Ransomware Attack



(a) Destination port beeswarm plot (b) Connection state beeswarm plot

Fig. 11. Individual Explanation for OHE features

as OTH are often indicative of unusual or unexpected behavior, which could be a sign of probing, scanning, or attempts to establish connections in non-standard ways. On the other hand, the S0 state indicates that a SYN packet (used to initiate a TCP connection) was sent, but no reply was received from the destination. This often signifies an unsuccessful attempt to establish a connection. Since the connection attempt failed, the ransomware attack could not be executed, resulting in a benign condition.

Comparison between OHE and LE Fig. 10(a) and Fig. 10(b) present the beeswarm plots for OHE and LE, respectively. The top five features from both encoding methods are similar, although their order varies. The waterfall plot is

shown in Fig. 10(c) and Fig. 10(d). OHE assigns more weight to source packets and protocol, whereas LE places greater emphasis on destination packets, source packets, and protocol. In summary, Fusion SHAP provides a more comprehensive view of the attack process, focusing on how ransomware communicates (source packets and protocol) and how long the malicious session lasts. It gives insight into early detection and understanding the behavior of the ransomware during transmission. on the other hand, the original SHAP focuses on destination system impact and the services under attack, which is critical for understanding how ransomware affects the target system after gaining entry. Since ransomware detection often involves both communication and system impact, Fusion SHAP may provide better insight for early identification, while Original SHAP might help in understanding how to mitigate the damage after an attack begins.

3.7 Scanning

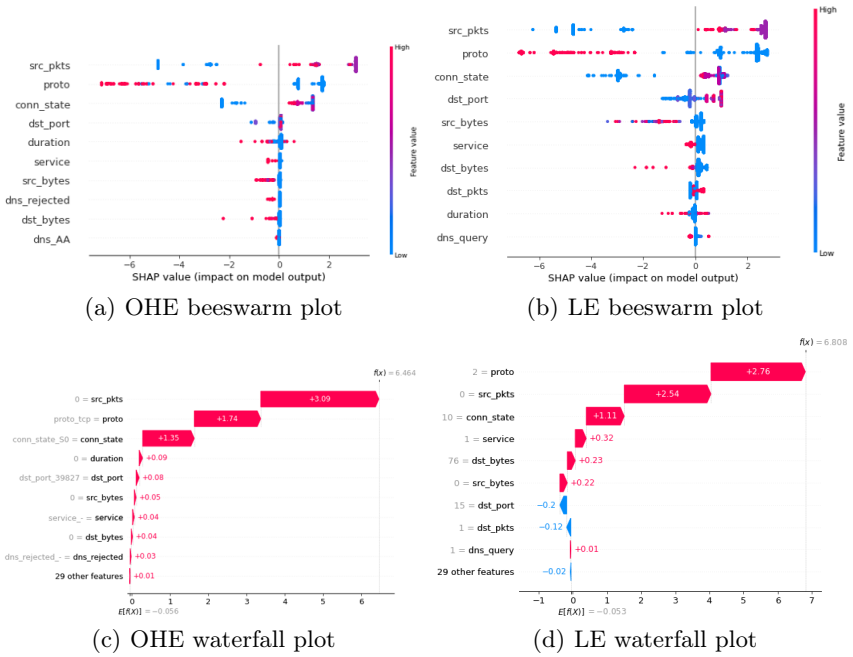
Table 6. Defined rule for scanning attack

Attack Type	**Algorithm**	**Feature Importance Result**		**Analysis Summary**
Scanning	Fusion SHAP with OHE	Source packet	large	Both approaches produce identical results in terms of feature selection and ranking for ransomware detection
		Protocol	TCP	
		Connection state	∼OTH	
		Destination port	varies	
	Original SHAP with LE	Source packet	large	
		Protocol	varies	
		Connection state	varies	
		Destination port	varies	

Protocol aspect When a scanning attack occurs, we might see an unusual frequency of certain protocols. For example, an increase in ICMP traffic could indicate a ping sweep, while a high volume of TCP traffic could suggest a port scan. Our explanation result showed that most of the scanning attacks targeted the TCP protocol.

Connection state aspect In scanning attacks, connection states tend to display patterns that are different from normal behavior. For example, numerous incomplete connections (SYN_SENT without corresponding established) or frequent RESET states might indicate scanning. A spike in specific connection states that typically occur during scanning, such as SYN_SENT or reset, could suggest malicious scanning activity. These states are often more frequent in scans compared to normal network traffic.

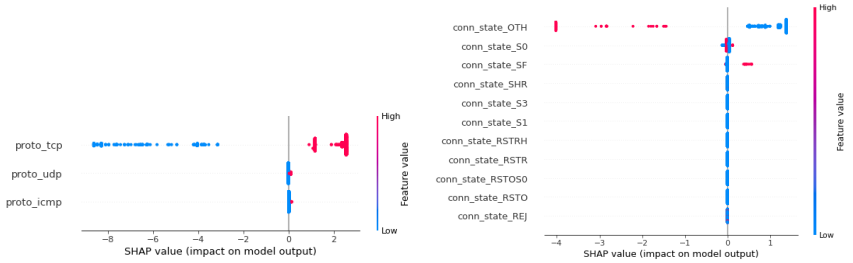
Scanning attacks are characterized by a high volume of very short connections. This is because the attacker is usually not interested in maintaining a long connection but rather in quickly probing multiple IP addresses or ports. For example, in a TCP SYN scan, the attacker sends a SYN packet to initiate



(a) OHE beeswarm plot (b) LE beeswarm plot

(c) OHE waterfall plot (d) LE waterfall plot

Fig. 12. Comparison of Explanation Result for Scanning Attack



(a) Protocol beeswarm Plot (b) Connection State Beeswarm Plot

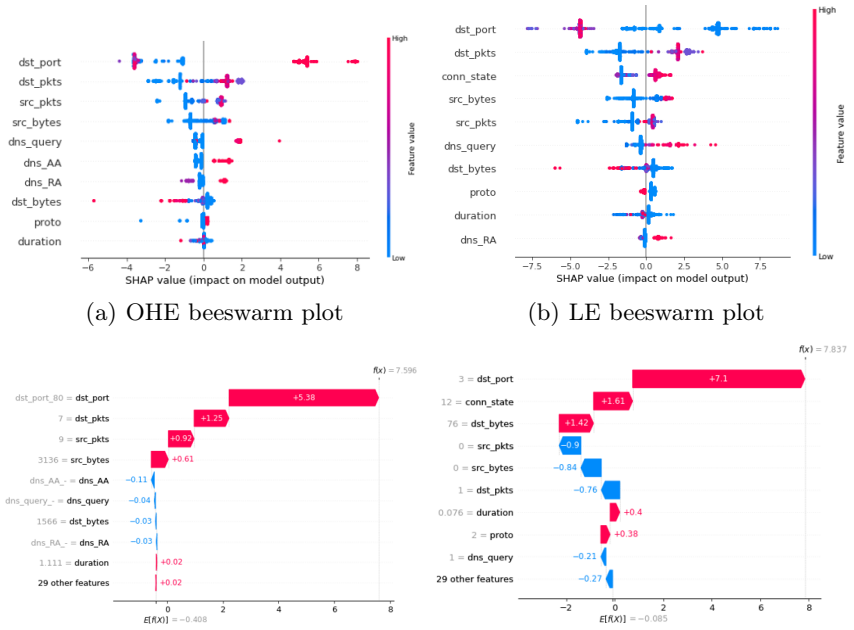
Fig. 13. Individual Explanation for OHE features

a connection but does not complete the handshake, leading to a very short-lived connection. Similarly, ICMP (ping) and UDP scans typically result in brief connections as they do not require a handshake and are often stateless. A ping request and its corresponding response are usually completed in milliseconds, resulting in a very short duration. Our explanation result validates this reasoning, showing that most of the Scanning attack is indicated by short duration.

Comparison between OHE and LE Fig. 12(a) and Fig. 12(b) present the beeswarm plots for OHE and LE, respectively. There is not much difference in the selected top features for both encoding methods. Even the top four features are

identical in their order. The waterfall plot is shown in Fig. 12(c) and Fig. 12(d). The waterfall plot reveals a slight difference in the decision-making process. OHE gives more consideration to source packets, followed by protocol, whereas LE prioritizes protocol, followed by source packets.

3.8 XSS



(a) OHE beeswarm plot (b) LE beeswarm plot

(c) OHE waterfall plot (d) LE waterfall plot

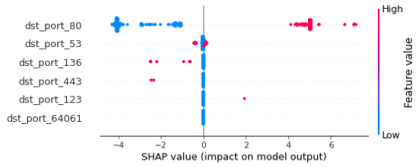
Fig. 14. Comparison of Explanation Result for XSS Attack

Destination port aspect XSS attacks usually target web applications, which commonly run on ports 80 (HTTP) and 443 (HTTPS). Most HTTP and HTTPS traffic is directed towards these ports, which are the standard for web services. If an IoT device or service is running on a non-standard port but serving web content, the destination port associated with the XSS attack would correspond to that specific port. The explanation result in Fig. 15(a) shows that most of the XSS attacks targeted port 80.

DNS query aspect In some cases, an attacker may use an XSS vulnerability to execute a script that forces the victim's browser to make DNS queries. If a sudden burst of DNS queries to unusual or previously unseen domains is detected, it could be indicative of an XSS attack where the attacker's payload is interacting with external domains. Fig. 15(b) shows that domains "testphp.vulnweb.com"

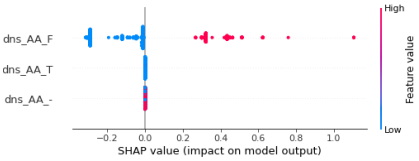
Table 7. Defined rule for XSS attack

Attack Type	Algorithm	Feature Importance Result		Analysis Summary
XSS	Fusion SHAP with OHE	Destination port	80	Provides more detailed insight into the role of DNS queries in XSS attacks, which captures critical aspect of how XSS attacks occurred
		Destination packet	large	
		Source packet	large	
		Source byte	large	
		DNS query	malicious websites	
	Original SHAP with LE	Destination port	varies	Emphasizes connection state, which is less important than DNS query in XSS attack
		Destination packet	large	
		Connection state	varies	
		Source byte	large	
		Source packet	large	

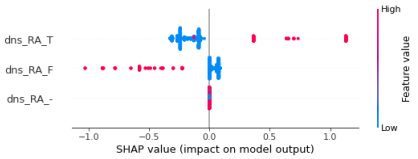


(a) Destination port beeswarm Plot

(b) DNS query beeswarm plot



(c) DNS_AA waterfall plot



(d) DNS_RA waterfall plot

Fig. 15. Individual Explanation for OHE features

and "mqtt-dashboard.com" are detected as malicious. These domains are vulnerable websites designed for XSS testing, simulating real-world vulnerabilities.

DNS activity aspect XSS attacks typically aim to execute malicious scripts in a victim's browser, but sometimes these scripts interact with external domains. In such cases, the victim's browser makes DNS queries to resolve the external domain, which may trigger DNS responses containing the dns_AA and dns_RA flags. From the explanation result in Fig. 15(c) and Fig. 15(d), we can see that an XSS attack is detected when the DNS server does not have direct knowledge about the domain. Dns_RA means that the DNS server can perform recursive lookups on behalf of the client. A recursive DNS query is where the DNS server takes on the task of resolving a domain name by querying other DNS servers if necessary. A recursive lookup might occur when the attacker's malicious script requests resources from domains not cached by the local DNS server.

Network traffic aspect When XSS attacks occur, anomalies in source packets, destination packets, source bytes, and destination bytes compared to normal traffic patterns can indicate malicious activity. For example, an unusually high number of small packets or an unexpected burst in byte count could signify attempts to inject scripts or test different payloads. A low bytes-to-packets ratio might indicate small payloads (typical in XSS attacks where scripts are sent), while a high ratio could suggest the transmission of larger data chunks in responses.

Comparison between OHE and LE Fig. 14(a) and Fig. 14(b) present the beeswarm plots for OHE and LE, respectively. The key difference we observe is that OHE does not consider the connection state to be an important feature, whereas LE does. This is further validated by the waterfall plot in Fig. 14(c) and Fig. 14(d), where LE assigns significant weight to the connection state, alongside network traffic features. In contrast, OHE relies solely on network traffic features for detection. In summary, While connection state (highlighted by original SHAP) is also important, especially in network-based attacks, the added emphasis on DNS by Fusion SHAP could provide a richer understanding for this specific type of web-related attack like XSS.

4 Conclusion

In this paper, we introduced Fusion SHAP to address the limitations of SHAP in handling one-hot encoded features. Our method successfully provides both individual and merged explanations, offering a more comprehensive view of the contribution of categorical features. We demonstrated that while label encoding has been used in recent efforts to tackle this issue, it introduces potential problems related to ordinality that Fusion SHAP avoids. Through a detailed analysis of seven types of cyberattacks in an IDS dataset, we showed that Fusion SHAP produces more accurate and interpretable explanations compared to the original SHAP algorithm. Additionally, our correlation analysis between feature importance and attack types further underscores the benefits of our approach in improving the interpretability of models that rely on one-hot encoded data. Overall, Fusion SHAP represents a significant advancement in the field of explainable AI, providing both theoretical rigor and practical effectiveness in real-world cybersecurity applications. For future work, we plan to improve the computational efficiency of Fusion SHAP for large-scale datasets and high-dimensional features, which is another potential direction, ensuring scalability without sacrificing interpretability.

References

1. Booij, T.M., Chiscop, I., Meeuwissen, E., Moustafa, N., Den Hartog, F.T.: Ton_iot: The role of heterogeneity and the need for standardization of features and attack types in iot network intrusion data sets. IEEE Internet of Things Journal **9**(1), 485–496 (2021)

2. Fosić, I., Žagar, D., Grgić, K., Križanović, V.: Anomaly detection in netflow network traffic using supervised machine learning algorithms. Journal of industrial information integration **33**, 100466 (2023)
3. Gaitan-Cardenas, M.C., Abdelsalam, M., Roy, K.: Explainable ai-based intrusion detection systems for cloud and iot. In: 2023 32nd International Conference on Computer Communications and Networks (ICCCN). pp. 1–7. IEEE (2023)
4. Hasan, M.K., Sulaiman, R., Islam, S., Rehman, A.U., et al.: An explainable ensemble deep learning approach for intrusion detection in industrial internet of things. IEEE Access (2023)
5. Keshk, M., Koroniotis, N., Pham, N., Moustafa, N., Turnbull, B., Zomaya, A.Y.: An explainable deep learning-enabled intrusion detection framework in iot networks. Information Sciences **639**, 119000 (2023)
6. Kocher, G., Kumar, G.: Performance analysis of machine learning classifiers for intrusion detection using unsw-nb15 dataset. Comput. Sci. Inf. Technol.(CS IT) **10**(20), 31–40 (2020)
7. Manai, E., Mejri, M., Fattahi, J.: Impact of feature encoding on malware classification explainability. In: 2023 15th International Conference on Electronics, Computers and Artificial Intelligence (ECAI). pp. 1–6. IEEE (2023)
8. Moustafa, N.: A new distributed architecture for evaluating ai-based security systems at the edge: Network ton_iot datasets. Sustainable Cities and Society **72**, 102994 (2021)
9. Sharma, B., Sharma, L., Lal, C., Roy, S.: Explainable artificial intelligence for intrusion detection in iot networks: A deep learning based approach. Expert Systems with Applications **238**, 121751 (2024)
10. Wang, M., Zheng, K., Yang, Y., Wang, X.: An explainable machine learning framework for intrusion detection systems. IEEE Access **8**, 73127–73141 (2020)