جامعة نايف العربية للعلوم الأمنية
NAIF ARAB UNIVERSITY FOR SECURITY SCIENCES
تأسست ١٩٧٨ Est. 1978

Naif Arab University for Security Sciences
Journal of Information Security and Cybercrimes Research
مجلة بحوث أمن المعلومات والجرائم السيبرانية
https://journals.nauss.edu.sa/index.php/JISCR

JISCR

A Model-agnostic XAI Approach for Developing Low-cost IoT Intrusion Detection Dataset

CrossMark

Enoch Opanin Gyamfi[1,2,*], Zhiguang Qin[1], Daniel Adu-Gyamfi[2], Juliana Mantebea Danso[1], Judith Ayekai Browne[4], Dominic Kwasi Adom[2], Francis Effirim Botchey[1,3], Nelson Opoku-Mensah[1,5]

[1]School of Information and Software Engineering (SISE), University of Electronic Science and Technology of China, Sichuan Province, P.R. China.

[2]Department of Cyber Security and Computer Engineering Technology (DCSCET), School of Computing and Information Sciences (SCIS), C.K. Tedam University of Technology and Applied Sciences (CKT-UTAS), Navrongo, Ghana.

[3]Department of Computer Science, Koforidua Technical University, Koforidua, Ghana.

[4]School of Computer Science and Engineering (SCSE), University of Electronic Science and Technology of China (UESTC), Sichuan Province, P.R. China.

[5]St. Monica's College of Education, Mampong. Ashanti Region, Ghana.

Received 08 Oct. 2023; Accepted 10 Dec 2023; Available Online 21 Dec. 2023.

Abstract

This study tackles the significant challenge of generating low-cost intrusion detection datasets for Internet of Things (IoT) camera devices, particularly for financially limited organizations. Traditional datasets often depend on costly cameras, posing accessibility issues. Addressing this, a new dataset was developed, tailored for low-cost IoT devices, focusing on essential features. The research employed an Entry/Exit IoT Network at CKT-UTAS, Navrongo, a Ghanaian University, showcasing a feasible model for similar organizations. The study gathered location and other vital features from low-cost cameras and a standard dataset. Using the XGBoost machine learning algorithm, the effectiveness of this approach for cybersecurity enhancement was demonstrated. The implementation included a model-agnostic eXplainable AI (XAI) technique, employing Shapley Additive Explanations (SHAP) values to interpret the XGBoost model's predictions. This highlighted the significance of cost-effective features like Flow Duration, Total Forward Packets, and Total Length Forward Packet, in addition to location data. These features were crucial for intrusion detection using the new IoT dataset. Training a deep-learning model with only these features maintained comparable accuracy to using the full dataset, validating the practicality and efficiency of the approach in real-world scenarios.

I. Introduction

Intrusion detection datasets for IoT camera devices have become increasingly prevalent, but their creation often relies on expensive and high-end camera devices. This poses a challenge for financially constrained environments, such as African communities and organizations in remote regions. The proposed approach leverages on model-agnostic eXplainable AI (XAI) techniques to create a robust dataset for intrusion detection,

Keywords: Cybersecurity, Intrusion Detection Dataset, IoT, Model Agnostic XAI, SHAP, XGBoost, Cybercrime, Ghanaian University.

Production and hosting by NAUSS

* Corresponding Author: Enoch Opanin Gyamfi
Email: :enochopaningyamf@outlook.com

doi: 10.26735/LPAO2070

albeit, considering low-cost features of the dataset.

The escalating complexities of cyberattacks, particularly intrusion attempts, present a growing challenge in terms of handling and responding to these threats. Managing and responding to them is becoming increasingly difficult [1]. As indicated in the work of [2], traditional algorithms that rely on rule-based, statistics-based, and signature-based security policies, are commonly used for intrusion detection. It is important to mention that all these approaches depend on datasets to achieve their effectiveness [3]. Some latest state-of-the-art intrusion detection datasets include ISCX_2012 [4] ADFA-LD/-WD [5], CIC-IDS2017 [6], CSE-CIC-IDS-2018 [7], IEC 60870-5-104-IDD [8] and CICIoT2023 [9].

Nevertheless, with the proliferation of data transmitted over the Internet and the emergence of new computing paradigms like the Internet of Things (IoT) and Artificial Intelligence (AI), has led to challenges in generating features of these datasets. Not only is the process time-consuming, but also requires the use of sophisticated and expensive devices to collect feature values [3]. Data obtained from affordable devices [10] present inherent constraints for AI-based and IoT-based cybersecurity systems. Affordable devices may have less powerful electronic components or limited data collection capabilities, which can lead to lower-quality or less comprehensive data for cybersecurity analysis. Consequently, the accuracy of any system trained on such data may be limited or compromised. To overcome these limitations, the IoTID20 dataset was developed with a focus on utilizing inexpensive and readily available IoT devices for data gathering [11]. This dataset serves as a solution to enhance the accessibility and affordability of data for AI-driven cybersecurity applications, all while maintaining high levels of accuracy.

Moreover, considering the earlier discussed drawbacks in intrusion detection research, another significant challenge is the black-box nature of algorithms. This aspect requires more attention and consideration when integrating these models into the field of cybersecurity [12]. An essential factor to consider is the creation of datasets. Because algorithms and models operate with these black-box characteristics, the predictions and decisions they produce often lack transparency and rationale, posing challenges for individuals, particularly users and expert-developers, in understanding the underlying processes [13]. Consequently, even expert-developed cyber defense systems may lack the necessary components to effectively counteract threats, rendering these defensive systems susceptible to potential data breaches [14]. Additionally, regular users find them challenging to provide clear and straightforward explanations when an attack occurs. To address these limitations in utilizing such algorithms for cybersecurity, eXplainable AI (XAI) has emerged as a solution to mitigate the black-box issue associated with these algorithms. XAI enables users and experts to understand the logical explanations and core data evidence behind the outcomes produced by these algorithms, enhancing interpretability [15]. Siganos et. al [25] also introduced an AI-powered IDS with explainability functions for the IoT. They proposed IDS that relies on machine learning and deep learning methods, using XAI to explain decision-making

Likewise, in this paper, an eXplainable AI strategy categorised as model-agnostic was adopted. Specifically, SHAP was used to interpret the prediction capability of the machine learning algorithm, XGBoost, on the IoTID20 dataset [11] that was modified on a low-cost budget. Through this, important features of these datasets can be noted. Organizations, researchers and other stakeholders interested in intrusion detection but are on low budgets can be confidently advised to mimic the procedures and devices used to collect such data at a low cost.

As described in [21-23], XAI techniques can be organized based on multiple categories with the possibility of some techniques fitting into more than one category due to overlapping characteristics. To enhance clarity, it would be more appropriate to classify XAI techniques under either '*Model-Specific or Model-Agnostic*' categorization perspective. This categorization perspective provides a more comprehensive understanding of the characteristics of an adopted XAI technique.

XAI techniques can be categorized based on the types of models they are applicable to, which are either model-specific or model-agnostic.

Model-specific XAI techniques are tailored to a single model or a specific group of models. For instance, the Graph Neural Network (GNN) explainer [16] provides interpretable explanations for predictions made by GNN-based models on graph-related machine learning problems, which is beyond the scope of this study. This categorization of XAI technique is outside the scope of this study. In contrast, model-agnostic XAI techniques are designed to be compatible with any machine learning model in theory. This category of techniques is intentionally developed to work seamlessly with diverse number of machine learning models. The term “agnostic” signifies that these XAI techniques do not discriminate based on the specific type or architecture of the machine learning model in use.

These model-agnostic XAI techniques operate primarily by analyzing the inputs and outputs of a given machine learning model. They are designed to extract insights and explanations without needing to access the internal details of the model, such as its weight values or structural information. In other words, model-agnostic XAI techniques do not require knowledge of how the model was trained or its internal parameters; they focus solely on the inputs and outputs of the model. A widely used example is the SHAP tools [17], which was chosen as the model-agnostic explanation tool for this study. Siganos et. al [25] used SHapley Additive exPlanations (SHAP) method is to explain decisions made by deep learning models.

In the current highly competitive and dynamic world, contemporary organizations need to operate efficiently and affordably to ensure their success. Security strategies are important in the success of contemporary organizations, necessitating measures to safeguard data integral to their business operations. Financially capable organizations often invest in the latest and more advanced IoT devices and security systems, even those at higher costs. Unfortunately, financially constrained organizations encounter challenges in adapting to such advanced security measures, limiting their competitiveness in data protection on IoT networks. AI-based Intrusion Detection Systems (IDSs) is a viable approach for organizations to secure their data on IoT networks. However, organizations in financially constrained environments have not effectively adopted IDSs, primarily due to the substantial expenses associated with their implementation. AI-based IDSs involves the use of expensive and sophisticated devices to generate datasets for training the AI modules. As datasets form the fuel for any AI-based system, the costs associated with their creation significantly contribute to the overall expenses of implementing AI-based IDSs. Consequently, there is a critical need for research aimed at reducing the costs related to dataset creation for training AI-based IDSs. Addressing this aspect is crucial in enabling financially constrained organizations to embrace advanced security technologies, enhancing their ability to compete effectively in safeguarding their data and operations.

The problem has to do with the availability of robust intrusion detection dataset features generated from low-cost IoT devices. These features need also to be comparably standard to datasets generated with expensive high-end IoT devices. The main objective of this research is to address this issue by developing an intrusion detection dataset tailored to the needs of a financially challenged environment. To achieve this main research objective, standard dataset was selected, wherein, the IoT devices used for collecting its features are low-cost devices and are the ones commonly used in people’s daily routines. This aligns with the research objective of addressing the challenges associated with financially constrained environments. By using data generated from affordable IoT devices, the study demonstrates that effective intrusion detection is still achievable without the need for expensive infrastructure or resources. A network of IoT camera devices was set up to automatically capture location features for monitoring intrusion detection. Compatibility test was conducted between features of the selected dataset and that of the automatically captured location features. Compatible features of the selected dataset were appended to location features to create a new IoT dataset. Finally, model-agnostic XAI method is employed on the XGBoost algorithm to provide insights into which features of the new IoT dataset are most influential in making predictions of intrusion detection.

The subsequent sections of the paper are organized as follow: Section II discusses the methodology, Section III covers the experiments and results, and finally, Section IV is the conclusion of the paper.

II. Methodology

The methodology begins by selecting a standardized dataset, IoTID20, which utilizes affordable IoT camera devices. Following pre-processing, an IoT Network is set up within a university campus using low-cost camera devices to automatically capture two important location features: the locations of the camera devices initiating and receiving packets on the network. A Shapira-Wilk test is employed to identify features from the IoTID20 dataset that are compatible with these location features. Features from the IoTID20 dataset demonstrating compatibility with the location features are then appended. After this, the dataset's feature count is reduced to create a new IoT intrusion detection dataset that includes only features captured by low-cost IoT devices. For intrusion prediction, an XGBoost regression model is implemented on the new dataset, with its parameters optimized through a grid search algorithm. A model-agnostic Explainable Artificial Intelligence (XAI) technique calculates SHAP values to interpret predictions made by the XGBoost algorithm. The analysis of SHAP values on the XGBoost model's predictions aids in identifying the contributions of globally significant dataset features to the overall predictive outcomes. Fig. 1 provides an overview of the methodology's processing steps, each of which is subsequently elaborated upon in detail.

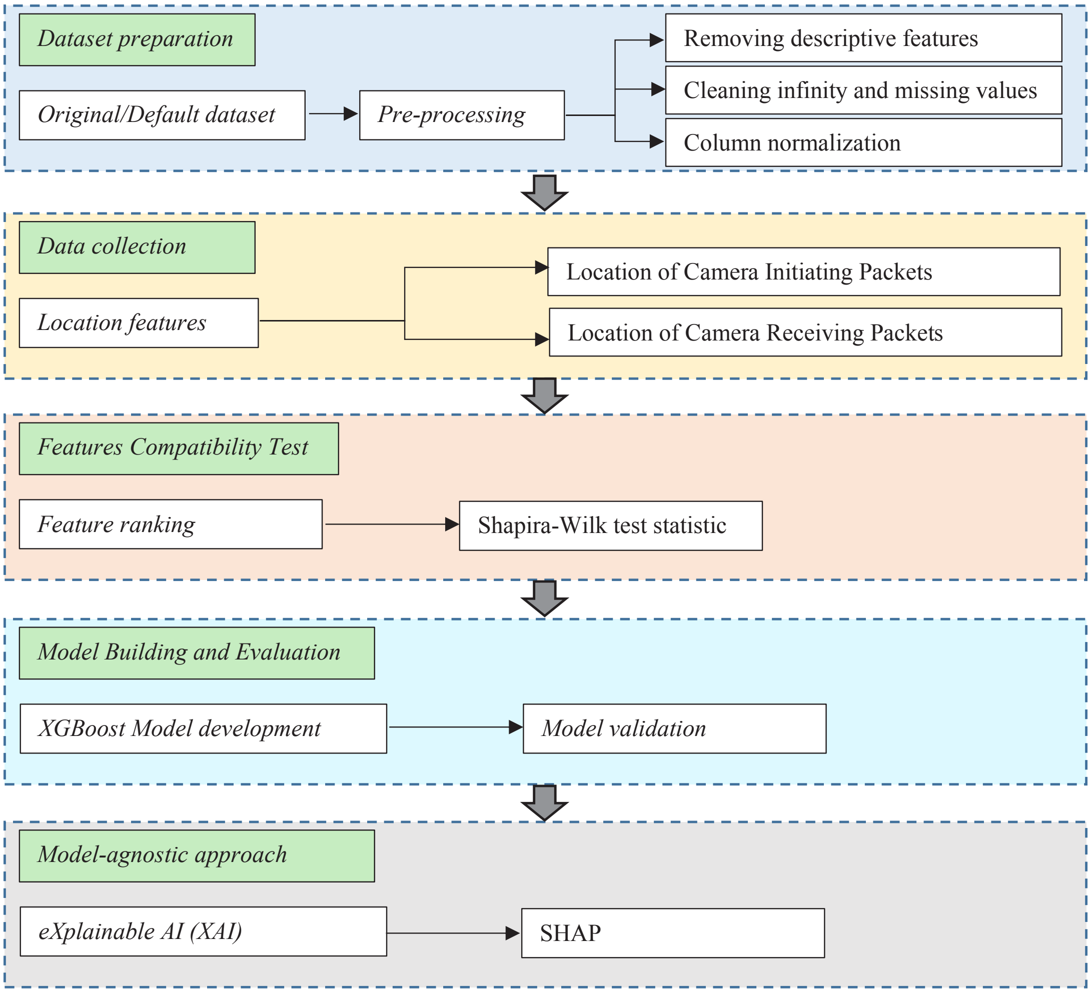

Dataset preparation
Original/Default dataset
Pre-processing
Removing descriptive features
Cleaning infinity and missing values
Column normalization
Data collection
Location features
Location of Camera Initiating Packets
Location of Camera Receiving Packets
Features Compatibility Test
Feature ranking
Shapira-Wilk test statistic
Model Building and Evaluation
XGBoost Model development
Model validation
Model-agnostic approach
eXplainable AI (XAI)
SHAP

Fig. 1. Workflow of the proposed approach.

A. Selecting the IoTID20 Dataset

The IoTID20 dataset [11], which contains data pertaining to attacks on common smart home devices, is used. As previously highlighted, the key benefits of choosing this dataset are that its collected features match modern IoT network trends and use data from affordable devices such as cameras (e.g. the SKT NGU and the EZVIZ Wi-FI cameras), Wi-Fi routers, laptops, tablets, and smartphones, among others. This dataset has a total of 86 network features and 3 label features. The 3 labels of the features are the binary, categorical, and the sub-categorical features. The original dataset contains 625,784 data instances.

B. Pre-processing for Feature Selection from IoT-ID20 Dataset

To get 620,673 data instances and 61 features, three processing techniques were performed as follow. First 1 to 9 features were intentionally removed because they were mere descriptive features that identify devices on the network e.g. *Flow ID, Source and Destination IPs, Source and Destination Ports, Protocols, Timestamps and Flow Duration. Following* this, data cleaning was conducted to address infinity and missing values, performed column normalization on feature values, and lastly, conducted feature correlation analysis. Features with a correlation coefficient of 0.70 or higher were removed from the dataset.

C. The Location Dataset Feature

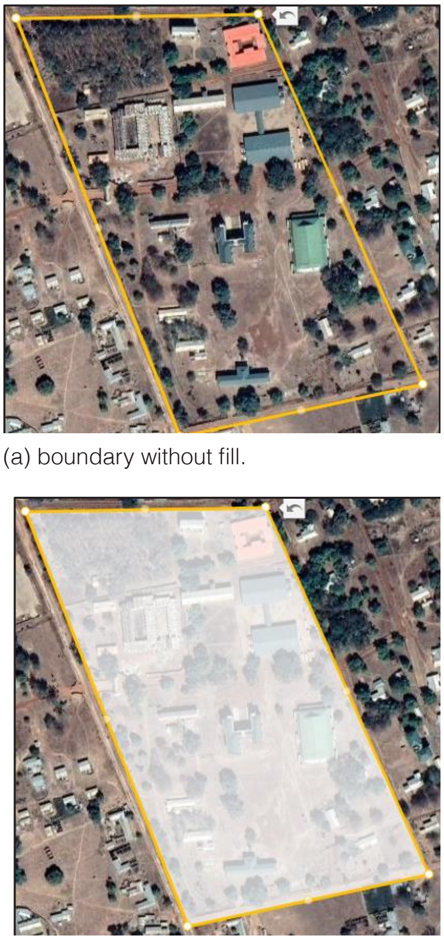
The retained 61 features were appended to the automatically collected two (2) location features which were termed as 'Location for Initiating Packets' and 'Location for Receiving Packets'. The set of features is denoted as $F=\{x_n\}_{i=1}^{n}$, with the x_1 to x_{n-2} being the retained 61 features of IoTID20 dataset, x_{n-1} and x_n being the two automatically collected location data features and n therefore being n=61+2, i.e. the total number of dataset features after appending.

D. Data Collection for the Location Features

The study hypothesizes that the location of an IoT camera device is also an important feature that could be used to monitor attacks for intrusion detection in an IoT network. Any packet needs to be initiated and received by the IoT camera within this location. The IPVM Design Calculator (Version 3.1) was used to design geographical plan for our IoT camera devices that are to be set up to monitor entry and exit of the university campus. The location of the CKT-UTAS University was searched, navigated, and set to the position coordinates 10.866 and –1.078 (in decimal degrees format). See Fig. 2.

After locations are set, a simulation was made where 7 cameras were added to the Campus Entry/Exit IoT network and placed at vantage entrance points of the university. Each camera with corner of coverage cone is also shown in Fig. 3.

Fig. 4 shows a preview of the viewing angle and area of one camera.



(a) boundary without fill.

(b) boundary with fill (translucent white).

Fig. 2. Location setup for the University campus.

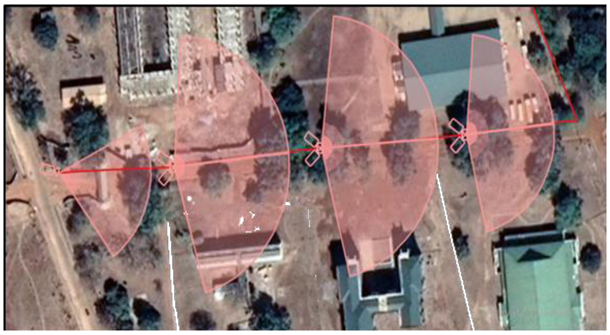


Fig. 3. Camera positions simulated on the IoT network of CKT-UTAS.

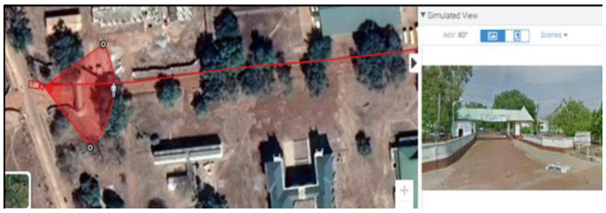


Fig. 4. Previewing the CCTV camera device placed exactly at the entrance to show the camera view of CKT-UTAS entrance.

E. Feature Ranking for Compatibility between IoTID20 and Location Features

Then, feature ranking was performed to extract from the retained 61 features of the IoTID20 dataset, those that are very compatible to the automatically collected two location features. Feature Ranking in increasing order is computed with a Shapira-Wilk test statistic. Initial data is ranked from the feature set:

$$F = \{x_1, x_2, x_3, \ldots x_{63}\}, n = 63 \tag{1}$$

and then equation (2) is calculated as:

$$b_{n,n-1} = \quad a_1(x_{n,n-1} - x_1) + a_2(x_{n,n-1} - x_2), + \cdots +, a_n(x_{n,n-1} - x_{n-2}) \tag{2}$$

where a_1,a_2,… a_n are the coefficients from Table A.6 of [19]. Test statistic computes the equation (1) with Calc W= $\frac{b_n^2+b_{n-1}^2}{(n-1)s^2}$, where s is the standard deviation of the feature set and (2) test statistic with a critical value Tab W from Table A.7 in the Appendix section of [19]. These statistics were then compared. If a feature's Calc W is greater than Tab W (i.e. Calc W>Tab W), it indicates a regular distribution of occurrences concerning the location features, ranking it as highly compatible with the dataset's features.

F. Extreme Gradient Boosting (XGBoost).

Extreme Gradient Boosting (XGBoost) evolved as an improved version of the Gradient Boosting Decision Tree (GBDT) algorithm [20]. When dealing with a dataset, denoted as $D = \{x_i, y_i\}$, where D *consists of* n *examples and* m features, a tree ensemble model incorporates K additive functions, denoted as $f_k \in \mathcal{F}$ to predict the output values $\hat{y}_i$ as illustrated in equation (3):

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i) \tag{3}$$

In this equation, each f_k represents an individual decision tree within the ensemble, and they work collectively to predict the output values based on the input features from the dataset D. The objective of XGBoost is to iteratively enhance the performance of these additive functions (trees) to yield accurate predictions for the given dataset.

To reduce errors within the ensemble trees, the objective function of XGBoost is shown in equation (4):

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} \left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t) \tag{4}$$

where the penalizing term Ω is computed using equation (5) as follows:

$$\Omega(f_t) = \gamma T + \frac{1}{2}\gamma \sum_{i=1}^{T} w_i^2 \tag{5}$$

The loss objective function can be expanded as shown in equation (6):

$$\hat{\mathcal{L}}^{(t)} = \sum_{i=1}^{n} \left(g_i, f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)\right) + \left(\gamma T + \frac{1}{2}\gamma \sum_{j=1}^{T} w_j^2\right) \tag{6}$$

where an optimal weight of each leaf j, and the corresponding optimal error/loss value $\hat{\mathcal{L}}^{(t)}$ that measure the quality of a tree structure q as is finally computed using equation (7) and (8) as follows:

$$w_i^* = -\frac{\sum_{i \epsilon I_j} g_i}{\sum_{i \epsilon I_j} h_i + \lambda} \tag{7}$$

$$\hat{\mathcal{L}}^{(t)}(q) = \sum_{j=1}^{T} \frac{\left(\sum_{i \epsilon I_j} g_i\right)^2}{\sum_{i \epsilon I_j} h_i + \lambda} + \gamma T \tag{8}$$

G. SHapley Additive exPlanations (SHAP)

SHAP [17] explains the output of machine learning models. They are calculated using the game theory concept called Shapley values. With the values, the average marginal contribution of each feature to the model's prediction can be measured [18].

A key reason for choosing SHAP for this research is TreeSHAP, designed for efficient Shapley value estimation in tree models [17] like XGBoost. SHAP provides a structured framework to explain predictions, enhancing model understanding. SHAP explains model's predictions using equation (9):

$$g(z') = \phi_0 + \sum_{j=1}^{M} \phi_j z_j' \tag{9}$$

where $\mathbf{g}$ is the explanation function for XGBoost model's prediction, z' is a coalition vector in $\{0,1\}^M$; $\mathbf{M}$ aggregates all data features as the maximum coalition size; $\phi_j \in \mathbb{R}$ are estimated Shapley values that denote j feature attribution. They specify each feature's contribution to the prediction.

To calculate the Shapley values ϕ, the formula simplifies to equation 10:

$$g(x') = \phi_0 + \sum_{j=1}^{M} \phi_j \tag{10}$$

In their paper [17], SHAP outlines the following three properties of ϕ and its related expressions (as shown from equations (11) to (15)):

1. Local Accuracy (Efficiency Property)

$$\hat{f}(x) = g(x') = \phi_0 + \sum_{j=1}^{M} \phi_j\, x_j' \tag{11}$$

Equation (11) can be expanded to equation (12) as:

$$\hat{f}(x) = \phi_0 + \sum_{j=1}^{M} \phi_j\, x_j' = E_X\left(\hat{f}(x)\right) + \sum_{j=1}^{M} \phi_j \tag{12}$$

If you define $\phi_0 = E_X\left(\hat{f}(x)\right)$ and $\forall x_j' = 1$,

2. Missingness Property:

$$x_j^i = 0 \Rightarrow \phi_j = 0 \tag{13}$$

The Missingness property of SHAP ensures fairness in assigning Shapley values to features in machine learning models, especially when some features are missing. This property states that when a feature is missing (meaning its value is unknown or undefined), it should be assigned a Shapley value of 0.

3. Consistency

The Consistency property of SHAP is an important attribute that ensures the fairness and reliability of feature attributions in machine learning models. To understand this property, let us first express the Consistency property as shown in equation (14).

$$\hat{f}_x'(z') - \hat{f}_x'(z_j') \geq \hat{f}_x(z') - \hat{f}_x(z_j') \tag{14}$$

The Consistency property essentially states that for any two models, $\hat{f}_x(z')$ and $\hat{f}_x(z_j')$, if the change in the prediction, expressed as $\hat{f}_x'(z') - \hat{f}_x'(z_j')$, is greater than or equal to the change in prediction for the original model $\hat{f}_x(z') - \hat{f}_x(z_j')$ for all possible input configurations $z' \epsilon \{0,1\}^M$, then Equation (15) holds true.

$$\phi_j(\hat{f}', x) \geq \phi_j(\hat{f}_x, x) \tag{15}$$

Equation (15) shows that, given these conditions, the Shapley value (ϕ_j) for a specific feature j in the modified model $(\hat{f}')$ is greater than or equal to the Shapley value for the same feature in the original model (f). Consistency property ensures that if a machine learning model is changed in a way that increases or maintains the impact of a particular feature (regardless of what happens to other features), then the Shapley value attributed to that feature also increases or stays the same.

III. Experiments and Results

The 'shap' Python Package was used. The package provides a set of tools and functions to compute and interpret SHAP values for different machine learning models. The 'shap' package is designed to be compatible with the widely used Python machine learning library, ‘scikit-learn’, which means that SHAP could easily be applied to explain predictions made by tree-based models created using ‘scikit-learn’. The XGBoost algorithm was used as a tree-based model with ‘scikit-learn’. The ‘shap’ package was in conjunction with ‘scikit-learn’ package’s tree boosting framework, the XGBoost.

A. Compatibility test between the IoTID20 Dataset and the Location Features

We show results from the Feature Ranking with Shapira-Wilk test statistic in Fig. 5. The first features labelled 9 – 20 of the 61 features ranked high in compatibility with the location features, with a value greater than 0.50.

B. XGBoost regression

1) Model development

The input data features were divided into 80-20 training and testing subsets. Five-folds cross validation was applied to train and evaluate the model. The XGBoost parameters were optimized using a simple grid search algorithm [21] to select the optimal parameters in Table I.

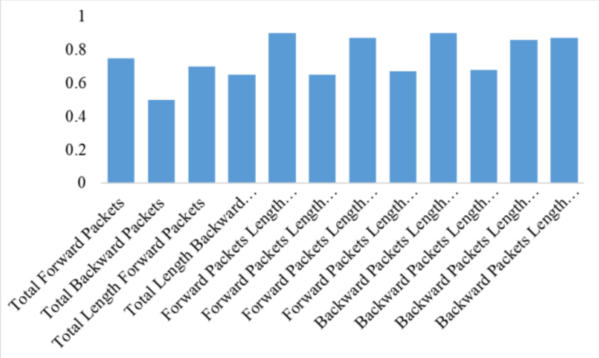


Fig. 5. Feature Ranking with Shapiro-Wilk test.

2) Model validation

For validating the chosen model, the developed model is used to verify the performance of the model on the independent test set from the dataset feature as compared to one other popular regression model, i.e. the multiple regression model. The R-squared score was utilized for this verification and found that the XGBoost regression model outperformed the multiple regression model in both training and testing, with the average R-squared values being above 80%. In addition, the XGBoost model resulted in more consistent values and smaller MSE, RMSE and MAE values, compared to the multiple regression model (see Table II).

C. SHAP Results

We use TreeSHAP estimation method to explain individual predictions, since XGBoost algorithm creates a sequential ensemble of tree models. This helps in extracting knowledge from the IoTID20 dataset using the SHAP method. The results will be in different domains interpreting XGBoost model using the SHAP method, as shown in Fig. 6.

D. SHAP Feature Importance

The SHAP feature importance plot, shown in Fig. 7, provides insights into which features are most influential in making predictions using the XGBoost algorithm for intrusion detection. It helps in identifying which aspects of the input data have the greatest impact on the model's decision-making process. Fig. 7 reveals that three features—Flow Duration, Total Forward Packets, and Total Length Forward Packet—stand out as the most globally important features. This means that these three features play a significant role in the model's ability to detect intrusions across the entire dataset.

TABLE I
Average Validation Metrics

Regression Model	training set				test set			
	MSE	**RMSE**	**MAE**	**R-squared**	**MSE**	**RMSE**	**MAE**	**R-squared**
XGBoost	2.087	1.559	1.105	0.868	2.057	2.079	1.085	0.962
Multiple	4.033	3.837	1.501	0.571	4.389	3.903	2.516	1.684

TABLE II
CHOSEN XGBOOST PARAMETERS AFTER SIMPLE GRID SEARCH

Parameters	Values	Selected Optimal value
Learning rate	0.1 ,0.01	0.01
Max tree depth	65 ,50 ,47 ,30 ,12	50
Min feature weights	12 ,8 ,6 ,4 ,1	12
Fraction of random samples for each tree	0.7 ,0.5	0.5
Subsample ratio of columns when constructing each tree	0.7 ,0.5	0.5
Number of trees to fit	1000 ,500 ,250 ,100	1000

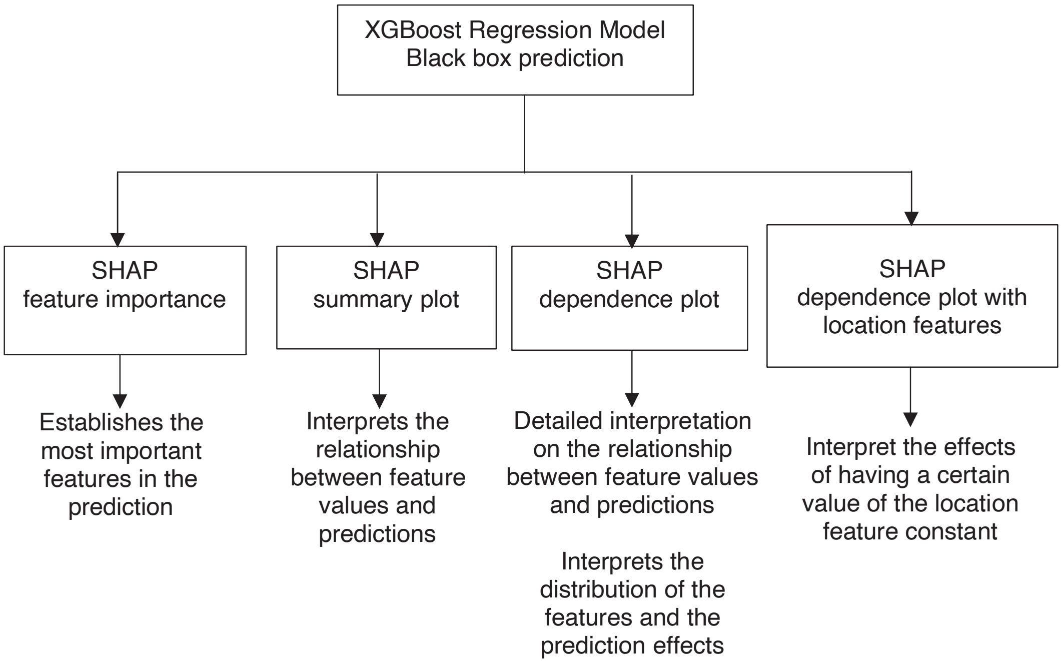

XGBoost Regression Model
Black box prediction
SHAP
feature importance
SHAP
summary plot
SHAP
dependence plot
SHAP
dependence plot with
location features
Establishes the most important features in the prediction
Interprets the relationship between feature values and predictions
Detailed interpretation on the relationship between feature values and predictions
Interprets the distribution of the features and the prediction effects
Interpret the effects of having a certain value of the location feature constant

Fig. 6. Pipeline for interpreting XGBoost model using the SHAP method.

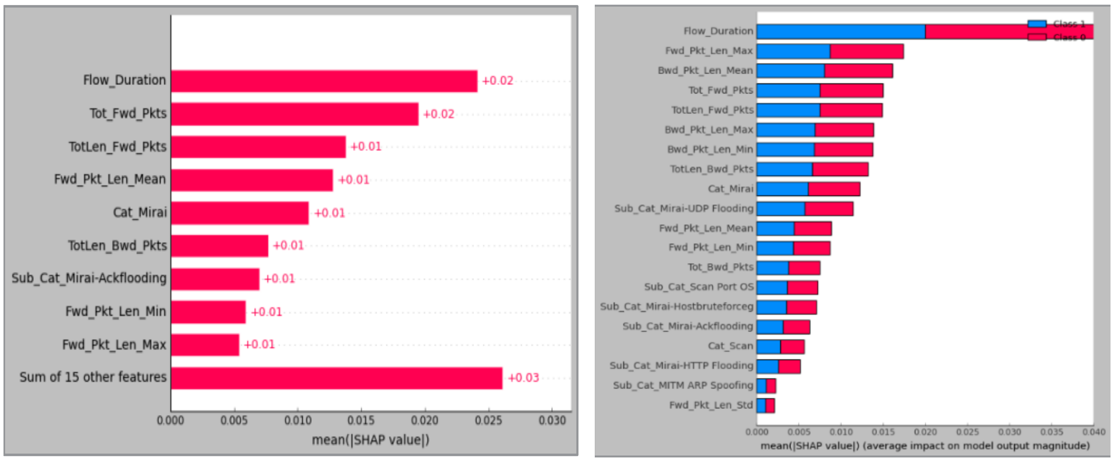


Fig. 7. SHAP Feature Importance.

1) SHAP Summary Plot

Focusing on the Flow Duration feature of Fig. 8, it becomes evident that high values of this feature are associated with intrusion detection. In other words, when network flows have long durations, it may be indicative of intrusion events. Additionally, it is noteworthy that data instances with a high total number of forward packets (Tot_Fwd_Pkts) values are also considered as an important feature for distinguishing intrusion events.

2) SHAP Dependence Plot

SHAP dependence plot in Fig. 9 confirms that a feature like Total Forward Packets have SHAP values of nearly -1.74 for the intrusion detection are extremely negative.

3) SHAP Force Plot

The force plot offers a visual representation of the contribution of individual feature to the XGBoost prediction. The values ranging from -0.0257 to 0.0716 represent the magnitude of the entire feature contribution to the final XGBoost prediction.

The plot has two force bars, one pink and one blue. The pink bar is labelled "higher" and the blue bar is labelled "lower". These bars represent the positive and negative contributions of the features towards the prediction. The length of each bar indicates the magnitude of the feature's effect. In Fig. 10, it appears that certain features ("such as 'Pkt_len_Min' and 'Sub_Cat_Mirai-Ackflooding') are pushing the XGBoost model's output higher (pink bar) when combined with the collected location features, while others (such as 'Bwd_Pkt_Len_Mean' and 'Sub_cat_Scan-Port') are pushing the output lower (blue bar).

E. Comparison with dataset created using high-end cameras

The effectiveness of the newly created dataset built from low-cost features is compared to that of an original dataset lacking such specifications. The newly created dataset is created through a methodology that retains approximately 65% of the original dataset's features, focusing solely on low-cost features. Importantly, experiment is conducted to verify if this reduction in features streamlined to emphasize low-cost features, and does not compromise the accuracy of the machine learning model and still preserves the essential details of the dataset.

The experiment is executed on a desktop PC equipped with an AMD Ryzen 7 3700X CPU with a Base Clock of 3.60 GHz, a 32 GB of RAM, and an RTX 3060 GPU with 12GB GDDR6 VRAM. The software utilized comprises open-source libraries including Python, PyTorch, and scikit-learn.

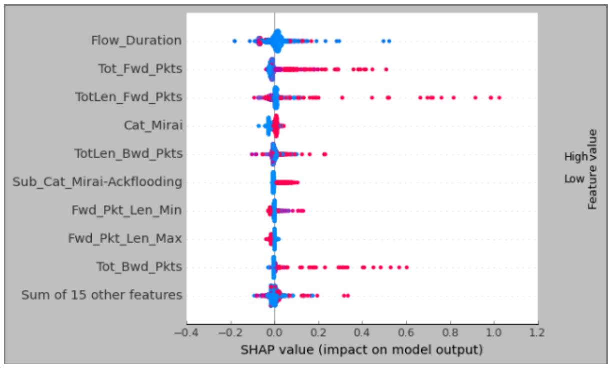


Fig. 8. Summary plots.

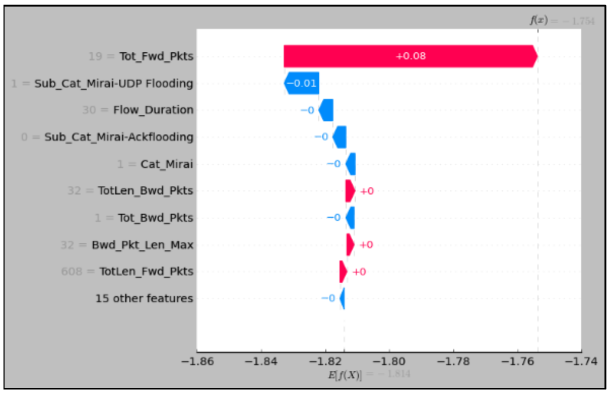


Fig. 9. SHAP Dependence Plot.

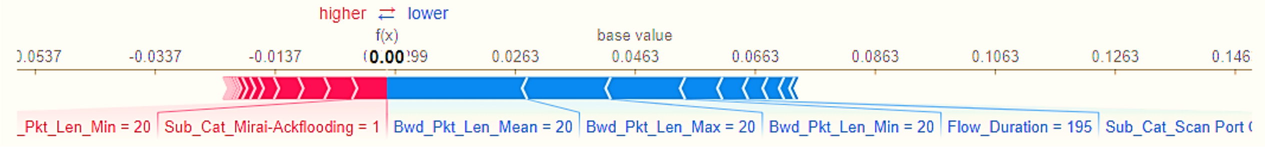


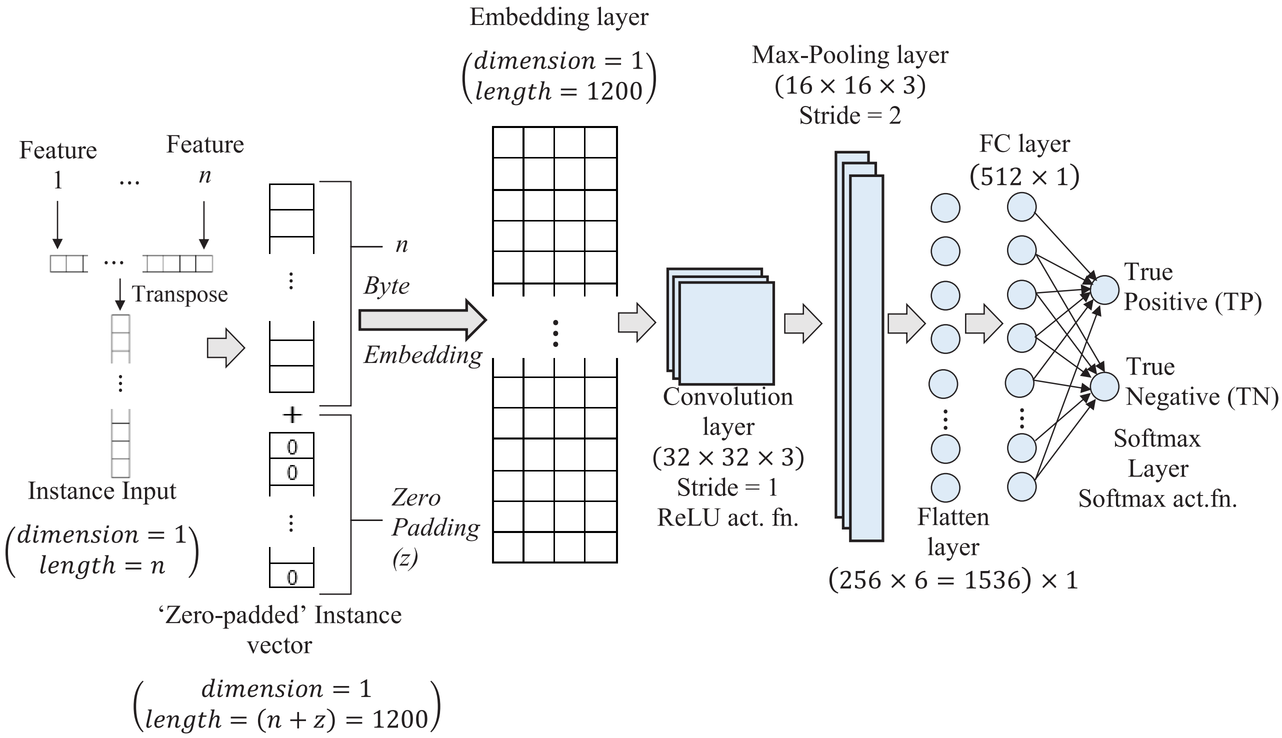
Fig. 10. SHAP Force Plot.

The procedural steps outlined in Fig. 1 for the proposed approach are entirely implemented in Python, encompassing tasks such as Dataset preparation, Data collection, Features Compatibility Test, Model Building and Evaluation and Model-agnostic approach. These tasks are in green-colored rectangular boxes in Fig. 1. While the primary objective of the proposed method is to generate a new dataset from existing ones, where the new dataset selectively incorporates features compatible with location features obtained from low-cost devices, it is imperative to validate that the machine learning model's performance on the new dataset remains unaffected.

To achieve this, a basic deep-learning model is instantiated and trained twice. In the first instance, the model undergoes training with the IoTID20 dataset, utilizing all its original features. This model is referred to as the DEFAULT dataset model. Subsequently, in the second instance, the model is trained on the same dataset, but with features selected using our proposed methodology's workflow, wherein only a specific number of features are chosen to create the new dataset. This is termed as the NEW dataset model.

Fig. 11 presents the deep-learning model developed, which is built on the convolutional neural network (CNN) framework proposed in [24]. The model processes the packet of an intrusion instance into byte-level data, passing them through an embedding layer, a convolution layer, a max-pooling layer, a flatten layer, and finally, a fully-connected layer. By assessing the correlation between each byte in the packet, the model determines whether the intrusion instance classifies as a true positive or true negative. It is assumed that the maximum length of the instance is denoted by the number of features 'n'; if some features have missing values, resulting in an instance length less than the 'n' bytes, zero padded to add up to the length. The loss function employed is binary cross entropy. The ADAM optimizer is adopted. The non-linear activation function ReLU, is also used. The Softmax function is applied in the last step.

This experiment shows the effectiveness of the proposed approach in demonstrating the usefulness of low-cost features of the dataset, rather than aiming to enhance the classification performance of the CNN model for the intrusion



Embedding layer
$\begin{pmatrix} dimension = 1 \\ length = 1200 \end{pmatrix}$
Max-Pooling layer
$(16 \times 16 \times 3)$
Stride = 2
Feature 1
...
Feature n
FC layer
(512×1)
n
Byte
Embedding
Transpose
True Positive (TP)
True Negative (TN)
Softmax Layer
Softmax act.fn.
Convolution layer
$(32 \times 32 \times 3)$
Stride = 1
ReLU act. fn.
Flatten layer
$(256 \times 6 = 1536) \times 1$
+
0
0
0
Zero Padding (z)
Instance Input
$\begin{pmatrix} dimension = 1 \\ length = n \end{pmatrix}$
'Zero-padded' Instance vector
$\begin{pmatrix} dimension = 1 \\ length = (n+z) = 1200 \end{pmatrix}$

Fig. 11. Architecture of CNN model for experiment on comparing the newly created dataset with existing ones.

problem. Consequently, we employ the same CNN as base model with consistent hyper-parameter values for both the DEFAULT dataset and the NEW dataset models. The number of training epochs is configured at 32 for the DEFAULT dataset and 64 for the NEW dataset. The batch size is uniformly set to 128 for both dataset models. Each dataset is partitioned into training and testing sets to assess the impact of the proposed approach on the performance of a machine learning model. The same model, as illustrated in Fig. 11, undergoes training twice: once on the DEFAULT dataset and once on the NEW dataset. The outcomes are then juxtaposed based on accuracy, precision, recall, and F1-score, with the following definitions:

$$Accuracy=\frac{TP+TN}{TP+TN+FN+FP} \qquad (16)$$

$$Precision=\frac{TP}{TP+FP} \qquad (17)$$

$$Recall=\frac{TP}{TP+FN} \qquad (18)$$

$$F1\text{-}Score=2\times\frac{Precision\times Recall}{Precision+Recall} \qquad (19)$$

Here are the explanations for TP, TN, FP, and FN:

- True Positive (TP): An attack instance is correctly identified as an attack instance.
- True Negative (TN): A non-attack instance is correctly identified as a non-attack instance.
- False Positive (FP): A non-attack instance is incorrectly classified as an attack instance.
- False Negative (FN): An attack instance is incorrectly classified as a non-attack instance.

The F1-score takes into account both precision and recall, making it a comprehensive metric that effectively illustrates the overall performance of the deep learning model on the datasets.

We visualized the experimental results. Machine learning models trained on the different dataset are compared and the resulting data is visualized in Fig. 12.

It is interesting that values computed for the metrics for the DEFAULT dataset model is almost equal to that of the NEW dataset. Thus, the figure reveals that the machine learning model trained on the NEW dataset does not degrades in terms of precision, recall, and F1-score. For the DEFAULT dataset model, the Precision, Recall, Accuracy and F1-Score values are 0.85, 0.86, 0.91 and 0.85 respectively. For the NEW dataset model, the Precision, Recall, Accuracy and F1-Score values are 0.84, 0.85, 0.93 and 0.84, respectively. Comparing these metrics reveals that the NEW dataset maintains the performance of the machine learning model almost identical to the DEFAULT dataset.

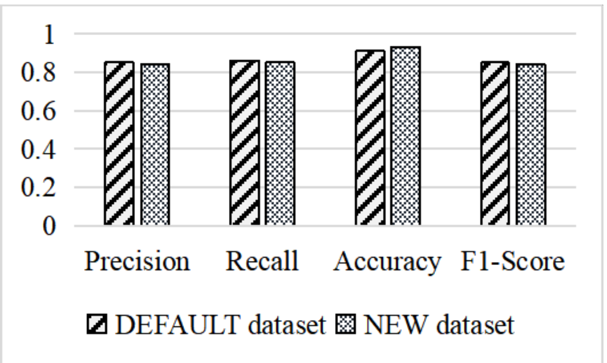


Fig. 12. Performance comparison of deep learning model on NEW dataset vs. DEFAULT dataset.

IV. Conclusion

This research addresses a significant challenge related to obtaining intrusion detection dataset features from cost-effective devices, with the goal of ensuring their comparability to those derived from high-end counterparts. The primary aim is to construct an intrusion detection dataset customized to meet the specific demands of financially constrained environments, without requiring costly infrastructure. The methodology starts by selecting the IoTID20 dataset, specifically designed to capture common IoT network characteristics, with a distinctive focus on using low-cost camera devices. Subsequently, an Entry/Exit IoT Network is simulated within a university campus using budget-friendly camera devices to automatically capture two essential location features: the locations of the initiating and receiving packets of the camera devices on the network. A Shapira-Wilk test statistic is executed to identify which features from the IoTID20 dataset is compatible with the two location features. The identified compatible location features are then appended to the existing features of the IoTID20 dataset. Through the compatibility test, features that received high rankings were found to be compatible for integration with the location dataset. This confirms the initial concept that the development of the IoTID20 dataset was intended for a budget-friendly process.

Following a pre-processing phase, the dataset feature count is reduced to create a new IoT intrusion detection dataset, streamlined in such a way that it includes only features captured by low-cost IoT devices. To offer a practical solution that uses the cost-effective features of this new dataset, an important aspect of the research involves implementing the XGBoost machine learning algorithm on this new dataset for intrusion prediction. The implemented XGBoost regression model with the selection of its parameters optimized using a simple grid search algorithm was found to predict better on the new low-budget IoT intrusion detection dataset than other popular multiple regression models.

A model-agnostic XAI approach was adopted in using SHAP values to interpret the predictions made by XGBoost algorithm. The computation of SHAP values on the XGBoost model's predictions shows the contributions of a substantial number of dataset features to the overall predictive outcomes. The research also found that the SHAP results highlight certain globally important, low-cost features within the IoTID20 dataset when the location features collected in this study were appended to them to create a new IoT dataset. The Flow Duration, Total Forward Packets, and Total Length Forward Packet are deemed important global features in the context of intrusion detection using the implemented XGBoost algorithm on the new low-cost IoT dataset.

The Flow Duration feature represents the duration of a network flow, which could be significant in identifying patterns associated with normal or abnormal network behavior. The Total Forward Packets suggests that the number of forward packets in a network flow is a significant factor in determining whether an intrusion is occurring. It could indicate that certain patterns in packet transmission are indicative of security threats. Lastly, the Total Length Forward Packet implies that the total length of forward packets in a network flow plays a crucial role in intrusion detection. It could suggest that the size or content of transmitted data is a key consideration in identifying potential security issues.

In essence, these findings suggest that focusing on these specific aspects of network activity—flow duration, the number of forward packets, and the total length of forward packets—provides valuable insights for effectively detecting intrusions using the XGBoost algorithm on the new budget-friendly IoT dataset.

From the experimental results, it was found that the newly created dataset maintains the performance of a machine learning model while selecting only low-cost features of dataset of an intrusion detection instances. This means that selecting only the low-cost features of the original dataset using our proposed approach is sufficient for training a deep learning model for intrusion detection, and the financial burden on using expensive features of the datasets could be lessened

From the abovementioned, the study demonstrate the feasibility of building an effective intrusion detection dataset suited to financially constrained settings, ensuring that institutions in resource-limited areas (like CKT-UTAS, Navrongo, Ghana) can enhance their cybersecurity measures without the need for costly infrastructure. The findings presented in this paper can serve as a valuable reference for organizations seeking to improve their security posture without incurring substantial financial burdens.

Funding

This work was supported in part by the National Natural Science Foundation of China (NSFC)'s: Key Project Number for International Cooperation (国家自然科学基金国际合 作重点项目编号) 61520106007, Privacy and Security Critical Theories and Technologies Based on Data Lifecycle (基于 数据生命周期的隐私和安全关键理论与技术), Funding: 964000 (经费：96.4万), Start and Finish Date 2016.1.1 – 2020.12.31 (起止日期：2016.1.1-2020.12.31); Major Instrument Project Number 62027827, Development of Heart-Sound Cardio-Ultrasonic Multimodal Auxiliary Diagnostic Equipment for Fetal Heart.

Conflict of Interest

Authors declare that they have no conflict of interest.

References

[1] D. Ucci, L. Aniello, and R. Baldoni, "Survey of machine learning techniques for malware analysis," Computer Security, vol. 81, pp. 123-147, 2019. [Online]. Available: https://doi.org/10.1016/j.cose.2018.11.001

[2] D. Gumusbas, T. Yldrm, A. Genovese, and F. Scotti, "A comprehensive survey of databases and deep learning methods for cybersecurity and intrusion detection systems," IEEE Systems Journal, vol. 15, no. 2, pp. 1717–1731, 2021. [Online]. Available: https://doi.org/10.11090/JSYST-.2020.2992966.

[3] R. Donida L., A. Genovese, V. Piuri, F. Scotti, and S. Vishwakarma, "Computational intelligence in cloud computing," in Recent Advances in Intelligent Engineering, L. Kovács, T. Haidegger, and A. Szakál, Eds. Springer, 2020, pp. 111–127. [Online]. Available: https://doi.org/10.1007/978-3-030-14350-3_6.

[4] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," Computers & Security, vol. 31, no. 3, pp. 357-374, 2012.

[5] G. Creech, "Developing a high-accuracy cross-platform host-based intrusion detection system capable of reliably detecting zero-day attacks," Ph.D. dissertation, University of New South Wales (UNSW) Sydney, Australia, 2014.

[6] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP), 2018, pp. 108-116.

[7] P. Radoglou-Grammatikis et al., "IEC 60870-5-104 Intrusion Detection Dataset," IEEE Dataport, 2022. [Online]. Available: https://dx.doi.org/10.21227/fj7s-f281.

[8] E. C. P. Neto et al., "CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment," Sensors, vol. 23, 5941, 2023. [Online]. Available: https://doi.org/10.3390/s23135941.

[9] R. A. Nafea and M. A. Almaiah, "Cybersecurity threats in the cloud: A literature review," in Proceedings of the International Conference on Information Technology (ICIT), 2021, pp. 779-786. [Online]. Available: https://doi.org/10.1109/ICIT52682.2021.9491638.

[10] K. Hyunjae et al., "IoT Network Intrusion Dataset," 2023. [Online]. Available: http://dx.doi.org/10.21227/q70p-q449. Accessed April 1, 2023.

[11] J. Gerlings, A. Shollo, and I. Constantiou, "Reviewing the need for explainable artificial intelligence (xAI)," arXiv preprint arXiv:2012.01007, 2012.

[12] T. Perarasi et al., "Malicious vehicles identifying and trust management algorithm for enhancing security in 5G-VANET," in Proceedings of the 2nd International Conference on Inventive Research in Computer Applications (ICIRCA), 2020, pp. 269-275. [Online]. Available: https://doi.org/10.1109/ICIRCA48905.2020.9183184.

[13] G. Jaswal, V. Kanhangad, and R. Ramachandra, Eds., AI and Deep Learning in Biometric Security: Trends, Potential, and Challenges. CRC Press, 2021.

[14] C. Rudin, "Stop explaining black box machine learning models for high-stakes decisions and use interpretable models instead," arXiv preprint arXiv:1811.10154, 2018.

[15] R. Ying et al., "GNNExplainer: Generating explanations for graph neural networks," arXiv preprint arXiv:1903.03894, 2019.

[16] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in Proceedings of the Advances in Neural Information Processing Systems (Vol. 30), 2017, pp. 1-10.

[17] E. Winter, "The Shapley value," in Handbook of Game Theory with Economic Applications (Vol. 3), 2002, pp. 2025-2054.

[18] A. P. King and R. J. Eckersley, "Appendix A - Statistical Tables," in Statistics for Biomedical Engineers and Scientists: How to Visualize and Analyze Data, Eds. A. P. King and R. J. Eckersley, Academic Press, 2019.

[19] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785-794.

[20] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.

[21] S. Holm and L. Macedo, "The Accuracy and Faithfullness of AL-DLIME-Active Learning-Based Deterministic Local Interpretable Model-Agnostic Explanations: A Comparison with LIME and DLIME in Medicine," in World Conference on Explainable Artificial Intelligence, 2023, pp. 582-605. Cham: Springer Nature Switzerland.

[22] S. Ali et al., "Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence," Information Fusion, vol. 99, 101805, 2023.

[23] M. N. K. Sikder et al., "Model-agnostic scoring methods for artificial intelligence assurance," in 2022 IEEE 29th Annual Software Technology Conference (STC), 2022, pp. 9-18. [Online]. Available: https://doi.org/10.1109/STC51895.2022.961.

[24] W. Jang, H. Kim, H. Seo, M. Kim, and M. Yoon, "SELID: Selective Event Labeling for Intrusion Detection Datasets," Sensors, vol. 23, no. 13, pp. 6105, 2023.

[25] M. Siganos, P. Radoglou-Grammatikis, I. Kotsiuba, E. Markakis, I. Moscholios, S. Goudos, and P. Sarigiannidis, "Explainable AI-based Intrusion Detection in the Internet of Things," in Proceedings of the 18th International Conference on Availability, Reliability and Security, 2023, pp. 1-10.