New Perspectives on Data Exfiltration Detection for Advanced Persistent Threats Based on Ensemble Deep Learning Tree

Xiaojuan Cai[1] and Hiroshi Koide[2]
[1]*Department of Information Science and Technology, Information Science and Electrical Engineering, Kyushu University, Fukuoka, Japan*
[2]*Section of Cyber Security for Information Systems, Research Institute for Information Technology, Kyushu University, Fukuoka, Japan*

Keywords: Data Exfiltration, Command and Control Channel, Transfer Size Limitation, Advanced Persistent Threat, Deep Learning, Ensemble Tree, Extreme Gradient Boosting, Internet Traffic.

Abstract: Data exfiltration of Advanced Persistent Threats (APTs) is a critical concern for high-value entities such as governments, large enterprises, and critical infrastructures, as attackers deploy increasingly sophisticated and stealthy tactics. Although extensive research has focused on methods to detect and halt APTs at the onset of an attack (e.g., examining data exfiltration over Domain Name System tunnels), there has been a lack of attention towards detecting sensitive data exfiltration once an APT has gained a foothold in the victim system. To address this gap, this paper analyzes data exfiltration detection from two new perspectives: exfiltration over a command-and-control channel and limitations on exfiltration transfer size, assuming that APT attackers have established a presence in the victim system. We introduce two detection mechanisms (Transfer Lifetime Volatility & Transfer Speed Volatility) and propose an ensemble deep learning tree model, EDeepXGB, based on eXtreme Gradient Boosting, to analyze data exfiltration from these perspectives. By comparing our approach with eight deep learning models (including four deep neural networks and four convolutional neural networks) and four traditional machine learning models (Naive Bayes, Quadratic Discriminant Analysis, Random Forest, and AdaBoost), our approach demonstrates competitive performance on the latest public real-world dataset (Unraveled−2023), with Precision of 91.89%, Recall of 93.19%, and F1-Score of 92.49%.

1 INTRODUCTION

Advanced Persistent Threats (APTs) are able to establish a long-term presence in a target system that allows them to gather as much data as they can while remaining undetected by using sophisticated tools and zero-day vulnerabilities (Charan et al., 2021).

All APTs have a basic procedure during the attack, generally referred to as the life cycle, which is shown in Figure 1. In order to facilitate the *data exfiltration* stage, the attacker must maintain active communications with the victim system through the command and control (C2) channel, a communication channel through which the attacker receives exfiltrated data and sends commands (Edgar and Manz, 2017). The target information will be detected, retrieved, and transferred to attackers through C2 channels. This is the ultimate goal of APT attackers and is the last line of defense in the information defense war. Therefore, it is significant to focus on data exfiltration detection in APT attacks. (Irshad et al., 2021; King et al., 2021).

During the process of data exfiltration, attackers will gather and transform target data through outbound traffic. Therefore, data exfiltration is not completely invisible. For all APT attacks, it is a crucial step for attackers to establish C2 channels in order to send commands to the victim system and receive the exfiltrated data from the victim network (King et al., 2021).

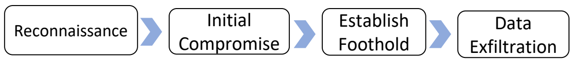


Figure 1: APT typical main stages.

There are plenty of works concentrating on how to identify and stop attackers the moment they enter the system and begin attacking (Stojanović et al., 2020; Chen et al., 2014; Alminshid and Omar, 2020). And the incorporation of machine learning has resulted in an even greater rate of attack detection accuracy

Cai, X. and Koide, H.
New Perspectives on Data Exfiltration Detection for Advanced Persistent Threats Based on Ensemble Deep Learning Tree.
DOI: 10.5220/0012181200003584
In *Proceedings of the 19th International Conference on Web Information Systems and Technologies (WEBIST 2023)*, pages 276-285
ISBN: 978-989-758-672-9; ISSN: 2184-3252

(Ghafir et al., 2018; Mamun and Shi, 2021; Lal et al., 2022; Abdullayeva, 2021). Meanwhile, some works have noticed the importance of detecting data exfiltration based on Domain Name System (DNS) (Lal et al., 2022; Mengqi et al., 2022; Alenezi and Ludwig, 2021; Zebin et al., 2022). However, the topic of how to secure sensitive data in an APT attack after the APT attacker has established a foothold on victim systems has received relatively little attention.

This void inspired us to investigate the following two questions: ***Q1.*** How to detect data exfiltration of APT attacks within normal traffic? ***Q2.*** How to detect sensitive data exfiltration of APT attacks if the leaked information is split into extremely small chunk sizes in different victim systems?

To answer the two questions above, we are facing the following two challenges in this paper:

Challenge 1: To identify the malicious sensitive data transfer while conventional Internet traffic is occurring. In order to prevent being detected, data exfiltration traffic will mimic legitimate user traffic as closely as possible. Hence, it is a main challenge to identify data exfiltration in the network output stream of APT attacks versus general Internet traffic activity.

Challenge 2: To detect APT attacks if the exposed information is transferred under small sizes (e.g., 1 MB) from the servers of several victims. The difficulty is that the features of the exfiltrated data transfer can easily be masked by legitimate file transfers of normal users to avoid detection by traffic monitors, since the larger the exfiltrated data transfer size, the more likely it is to trigger a monitor alert.

Hence, in this paper, assuming the APT attacker has established a foothold in the victim system successfully, by analyzing the Internet traffic, our main purpose is to detect data exfiltration of APT from the following two perspectives using an ensemble deep learning tree based on eXtreme Gradient Boosting (EDeepXGB):

- Exfiltration over C2 channel
- Exfiltration transfer size limitation

Contributions

Our main contributions in this paper can be listed as follows:

- Assuming APT attackers has established a foothold in the victim system, we first focus attention on the data exfiltration detection from the perspective of exfiltration over APT C2 channels and exfiltrated data size transfer limitation.
- In order to detect data exfiltration of APT attack, we summarized the Transfer Lifetime Volatility (PTL) and the Transfer Speed Volatility (PTS) as detection mechanisms from the perspective of Exfiltration over C2 Channel and Exfiltration Transfer Size Limitation.
- A ensemble deep learning tree based on eXtreme Gradient Boosting (XGB), called EDeepXGB, is implemented. Like the existing ConvXGB system (Thongsuwan et al., 2021), we implement the XGB to the dense layer of deep learning models to detect data exfiltration of APT accurately and rapidly.
- To exclude chance and randomness, four promising Deep Neural Network (DNN) models and four Convolution Neural Networks (CNN) are trained using the newest dataset of Unraveled−2023 (Sowmya et al., 2023). Meanwhile, an optimal EDeepXGB model is determined after a couple of evaluation experiments.
- The performance of our proposed method is verified using the newest public dataset, which shows that our EDeepXGB successfully promotes the performance of detection Precision, Recall Score and the F1-Score, compared to baselines (e.g., Naive Bayes, Quadratic Discriminant Analysis (QDA), Random Forest and AdaBoost).

The rest of the paper is structured as follows. In Section 2, a brief introduction of related previous works is drafted. The detection mechanisms we summarized are introduced in Section 3. Section 4 describes the structure of our proposed method. The details of our experiments using the newest public real-world dataset are shown in Section 5. Additionally, the observation and the evaluation are depicted in this Section. Lastly, The conclusion and the future work are addressed in Section 6.

2 RELATED WORKS

A number of techniques are used to detect data exfiltration of APT attacks. (Zou et al., 2020) proposed a ranking list of APT tactics in the framework of APT tactics recognition through synthesizing analysis and correlation of data from various sources. (Veena and Brahmananda, 2022) built a destination host filter unit and a blacklist of host destinations to analyze the outbound connections which go to the same destination from a huge amount of traffic. Moreover, in some works, machine learning models have been implemented in APT threats detection (Sabir et al., 2022; Zimba et al., 2020; Moghaddam and Zincir-Heywood, 2020). With semi-supervised learning models, the work of (Zimba et al., 2020) proposed a framework to score the suspicious APT activities using an SNN-based clustering algorithm. (Ghafir et al., 2018) mines relationships between activities and flag sequences of suspicious events as they occur for operating system diagnosis using Bayesian network. And (Mamun and Shi, 2021) proposed a heterogeneous task tree base deep learning method to detect malicious traces of APT. With the advantage of Auto-Encoder based deep learning approach that is able to identify complex relationships between features, (Abdullayeva, 2021) uses Auto-Encoder neural network with a softmax regression layer against APT attacks.

The detection rates of these works were relatively high and attention was given to all stages (including *Reconnaissance*, *Initial Compromise*, *Estanblish Foothold* and *Data Exfiltration*) of APT detection.

By analyzing data exfiltration on the outbound traffic of APT threats from a different angle, in the work of (D'Agostino and Kul, 2021), the authors focus on the situation after being attacked by APT threats. They compared the post-APT attack reports with sensitive database information to discover which sensitive data has been stolen. Although, their experiments are based on the fact that only APT attack activities occurred in the network traffic when being attacked.

Besides, for data exfiltration of APT, there are a lot of works (Mengqi et al., 2022; Alenezi and Ludwig, 2021; Zebin et al., 2022) focusing on the Domain Name System over HTTPS (DoH) to prevent APT attackers from storing information from the domain names and the corresponding IP addresses (zone file). In order to prevent data exfiltration from DNS Text messages, those works classify and detect DoH tunnelings by analyzing DNS queries and responses. However, in text classification, encrypted DNS queries can hardly be detected.

Consequently, it can be noted that all these previous works did not discuss the data exfiltration of APT attacks when data is transferred over C2 channels within transfer size limitation.

There are various strategies to establish and maintain C2 channels in APT attacks. For instance, APT1 (Mandiant, 2014) uses domain names to imitate the usual naming of online advertising services or websites (e.g., *yahoodaily.com*) to set up C2 servers. Moreover, in the attack of Duqu (Eric et al., 2012), the attacker uses intermediary servers as proxies to improve the availability and stealth of C2 channels. On the other hand, according to the report of (NSA et al., 2021), there are plenty of attacks that will split exfiltrated files into small chunks to avoid being detected, for example, APT28 (NSA et al., 2021) can split files under 1MB, and Kevin (Aseel Kayal, 2021) can exfiltrate data in blocks of 27 characters to the C2 server.

Hence, it is meaningful and necessary to investigate data exfiltration of APT from the perspective of Exfiltration over C2 channel and Exfiltration transfer size limitation.

3 DETECTION MECHANISMS

The gap from previous works leads us to the following two questions:

Q1. How to detect data exfiltration of APT attacks within normal traffic?

Q2. How to detect sensitive data exfiltration of APT attacks if the leaked information is split into extremely small sizes in different victim systems?

In this paper, to answer these two questions, we aim to investigate data exfiltration of APT from the perspective of Exfiltration over C2 channel and Exfiltration transfer size limitation.

Generally, HTTP/HTTPS ports 80 and 443 are typically employed for establishing C2 channels, since in well-secured corporate environments or governmental organizations, only these ports are permitted for outgoing connections. And the communication transmitted through the HTTP/HTTPS port can be identified as legitimate HTTP protocol or binary communication (Mengqi et al., 2022; Veena and Brahmananda, 2022). Hence, to detect the exfiltrated data, basic features (e.g., the destination IP/MAC address, the source IP/MAC address, the Transport layer port and the Transport layer protocol) are required.

However, to prevent being detected, attackers would mimic legitimate user communication as closely as possible. It means that: APT attackers would encode the exfiltrated data into normal communications using the same transport layer ports and the same transport protocol with fake domain names generators (e.g., Domain Generation Algorithm). Moreover, some APT attackers limit the exfiltrated data transfer size to an extremely small chunk size to prevent detection by the traffic monitor (NSA et al., 2021; Aseel Kayal, 2021).

Thus, as well as the basic features (ports, protocol, IP address, etc.) mentioned above, we summarized two detection mechanisms to detect data exfiltration over C2 channel and to detect data exfiltration with exfiltration transfer size limitation.

3.1 Transfer Lifetime Volatility

The value of transfer lifetime (which we called PTL_i) focuses on the one-way lifetime of packets transfer at one traffic connection between a normal user host and the destination (the suspect C2 server or the suspect proxy) in a period of time [t_i^{first}, t_i^{last}].

Exfiltration over C2 Channel

While exfiltrating the target data from the host server, although in order to prevent being detected, the attacker would try to keep the number of outflow packets per time unit as close to the average as possible, the ambition makes the attacker try to complete the exfiltration transfer as fast as possible at the same time (Charan et al., 2021; Sabir et al., 2021). It means that for the victim system, more packets are transmitted out than in over a period of time.

Hence, as shown as Equation 1, with the number of the transferred packets between the host to the destination N_i over time unit [t_i^{first}, t_i^{last}], the transfer lifetime volatility PTL_i^I from the destination to the host will be smaller than the transfer lifetime volatility PTL_i^O from the host to the destination in the i_{th} communication channel ($PTL_i^I \leq PTL_i^O$).

$$PTL_i = \frac{N_i}{t_i^{last} - t_i^{first}} \tag{1}$$

Exfiltration Transfer Size Limitation

For the same size of sensitive data, with the limitation of data transfer packets, the attacker needs to maintain a relatively longer period of alive C2 channel to keep exfiltrated data transferring compared to not limiting transfer size over C2 channels. Meanwhile, much longer time would be spent when the APT attacker trying to transmit sensitive data with small chunk size than normal users.

Hence, in the i_{th} communication connection, with the number of the transferred packets $N_i(\hat{t})$ between the host to the destination in a period of time $[t_{first}, \hat{t}]$, the transfer lifetime volatility PTL_i^I from the destination to the host can be calculated as Equation 2. The transfer lifetime volatility PTL_i^I from the destination to the host will be greater than the transfer lifetime volatility PTL_i^O which from the host to the destination in the i_{th} communication channel ($PTL_i^I \geq PTL_i^O$).

$$PTL_i = \sum_{\hat{t}=t_i^{first}}^{t_i^{last}} \frac{N_i(\hat{t})}{\hat{t} - t_i^{first}} \tag{2}$$

3.2 Transfer Speed Volatility

The value of transfer speed (called PTS_i) focuses on the one-way packet transfer speed at one traffic connection between the normal user host (the victim's system) and the destination (the suspect C2 server or the proxy).

Exfiltration over C2 Channel

To prevent triggering the traffic monitor's alarm, attackers need to control the transfer speed. However, the outbound transfer speed (from the host server to the destination) will be greater than the inbound transfer speed (from the destination to the host server), because of the ongoing transfer of exfiltrated data.

As Equation 3 shows, $Byte_i$ means the total transfer size of packets between the host server and the destination in one communication channel, while $byte_{ij}$ means the transfer size of payload of the packet j between the host server and the destination in i_{th} communication channel. With the Equation 4, the transfer speed volatility PTS_i^I from the destination to the host will be smaller than the transfer lifetime volatility PTS_i^O which from the host to the destination in the i_{th} communication channel ($PTS_i^I \leq PTS_i^O$).

$$Byte_i = \sum_{j}^{k} byte_{ij} \tag{3}$$

(where k represents the k_{th} packet of i_{th} communication channel.)

$$PTS_i = \frac{Byte_i}{t_i^{last} - t_i^{first}} \tag{4}$$

Exfiltration Transfer Size Limitation

To prevent triggering the traffic monitor's alarm, attackers need to extend the transfer time when setting a limitation of data transfer packets. That means, the total number of the transferred packets between the host to the destination, and the lifetime ($t_i^{last} - t_i^{first}$) between the host to the destination will increase while the average transfer bytes $AveByte_i$ (shown as Equation 5) from the host to the destination will be decreased in the i_{th} communication connection.

Thus, according to the Equation 6, the transfer speed volatility PTS_i^I from the destination to the host will be greater than the transfer lifetime volatility PTS_i^O which from the host to the destination in the i_{th} communication channel ($PTS_i^I \geq PTS_i^O$).

$$AveByte_i = \frac{\sum_j^k byte_j}{N_i} \tag{5}$$

(where k represents the k_{th} packet of i_{th} communication channel.)

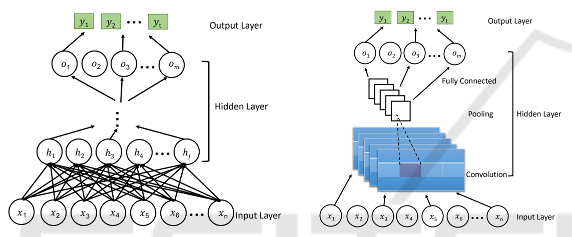
$$PTS_i = \frac{AveByte_i}{t_i^{last} - t_i^{first}} \tag{6}$$

4 PROPOSED METHODOLOGY

In this paper, we aim to detect the data exfiltration of APT attacks from two perspectives (*Exfiltration over C2 channel & Exfiltration transfer size limitation*) under the assumption that the APT attacker has established a foothold in the victim system. Hence, we proposed an ensemble deep learning tree based on eXtreme Gradient Boosting (EDeepXGB) for data exfiltration detection analysis.

4.1 DL Models

In this paper, to exclude chance and randomness, we prepared four Deep Neural Networks ($\{DNN_{1-4}\}$) and four Convolutional Neural Networks ($\{CNN_{1-4}\}$) as the base Deep Learning (DL) models for our EDeepXGB.


y_1 y_2 ⋯ y_t
Output Layer
o_1 o_2 o_3 ⋯ o_m
Hidden Layer
h_1 h_2 h_3 h_4 ⋯ h_j
x_1 x_2 x_3 x_4 x_5 x_6 ⋯ x_n
Input Layer
Fully Connected
Pooling
Convolution
Hidden Layer

(a) *DNN*. (b) *CNN*.
Figure 2: Model Structure of Deep Learning Networks.

4.1.1 DNN

Figure (a) of Figure 2 shows the structure of the Deep Neural Network (DNN). As Equation 7 shows, with n input nodes, the output result y_j of the node j is able to be calculated by the input X_i, synaptic weights (W_{ji}) between two neural layers and the bias (θ_j) of hidden layers or the output layer.

$$y_j = f(\sum_{i=1}^{n} W_{ji}X_i + \theta_j) \tag{7}$$

4.1.2 CNN

Figure (b) of Figure 2 shows the structure of the Convolutional Neural Network (CNN). While there is some $N*N$ square neuron layer which is followed by the convolutional layer (as shown in Equation 9), if an $n*n$ filter ω is used, at unit l, the convolutional layer output y^l_{ij} will be of size $(N-n+1)*(N-n+1)$. x^l_{ij} is the pre-nonlinearity input in the layer, which means the contributions (weighted by the filter components) from the previous layer cells. As Equation 10 shows, Y^l_{ij} is the output of Max-Pooling, which is to reduce the spatial dimensions of the convolutional layer output y^l_{ij} (e.g., width and height).

Convolutional Layers:

$$x^l_{ij} = \sum_{a=0}^{n-1}\sum_{b=0}^{n-1} \omega_{ab} y^{l-1}_{(i+a)(j+b)} \tag{8}$$

$$y^l_{ij} = f(\sum W x^l_{ij} + \theta) \tag{9}$$

Max-Pooling:

$$Y^l_{ij} = \max(y^l_{ij}) \tag{10}$$

4.2 EDeepXGB

In our EDeepXGB, using the prepared eight DL models as deep learning feature extractors, the prediction results will be obtained by using the trees of XGB to predict those extracted features. The node distribution of each EDeepXGB model is shown in Table 1.

Table 1: Distribution of EDeepXGB.

Models	**Layer Structure**
$EDeepXGB-DNN_1$	(In.-32)[1]-XGB
$EDeepXGB-DNN_2$	(In.-128-64)-XGB
$EDeepXGB-DNN_3$	(In.-64-32)-XGB
$EDeepXGB-DNN_4$	(In.-128-256-64)-XGB
$EDeepXGB-CNN_1$	(In.-64-30)-XGB
$EDeepXGB-CNN_2$	(In.-64-64-256)-XGB
$EDeepXGB-CNN_3$	(In.-64-128-30)-XGB
$EDeepXGB-CNN_4$	(In.-64-128-128-30)-XGB

Input-Dense layer distribution of DL models.

4.2.1 Prediction Drive Force

In this paper, all of our experiments are multi-class classification problems. Therefore, for base DL models, as shown in Figure 2, to predict t classes in the case of (N, n) input x, outputs of hidden layers $(o_1, o_2, ..., o_m)$ need a prediction drive force (usually using an activation function as the prediction drive force) to do class prediction $(y_1, y_2, ..., y_t)$.

Generally, the SOFTMAX function σ (shown as Equation 11) is an activation function for an output layer of neural networks in multi-class classification problems. It normalizes a numerical vector $\vec{o}$ to a vector of probability distributions with individual probabilities summing to 1.

$$\sigma(\vec{o})_i = \frac{e^{o_i}}{\sum_{j=1}^{K} e^{o_j}}, \tag{11}$$

(where $\vec{o}$ means the input vector, and K represents the number of classes in the multi-class classifier.)

However, works of (Thongsuwan et al., 2021; Mamun and Shi, 2021) inspired us to use a tree structure in the feature classification layer to overcome the overfitting problem. Thus, in this paper, we use XGB as a prediction drive force of our DL models to predict data exfiltration of APT attacks more accurately.

As a supervised learning model, the prediction of XGB is given as function (12) (Chen and Guestrin, 2016). For each input data y_i, XGB will assign a prediction numerical score. Hence, the classification problem of each input y_i becomes a '*yes*' and '*no*' problem (Xianrui and Joan, 2020). Hence, the output layer of our EDeepXGB is implemented as Figure 3.

$$\hat{y}_i = \sum_{j=1}^{n} f_j(x_i) \tag{12}$$

(where n is the number of trees, f is the function space of the set of possible classification trees)

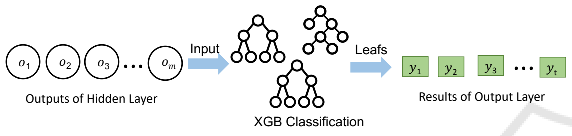


Figure 3: Class Prediction of EDeepXGB using XGB as a Prediction Drive Force.

5 EXPERIMENT

To demonstrate that the data exfiltration of APT attacks can be detected from perspectives of *Exfiltration over C2 channel* and *Exfiltration transfer size limitation*, we integrated our summarized detection mechanisms (Transfer Lifetime Volatility: PTL, Transfer Speed Volatility: PTS) into the EDeepXGB, using the selected samples of Unraveled−2023 dataset (Sowmya et al., 2023). Additionally, the entire Unraveled−2023 real-world dataset was used to evaluate the performance of our EDeepXGB. The overall performance of EDeepXGB will be discussed in comparison with four baseline models (e.g., Naive Bayes, Quadratic Discriminant Analysis (QDA), Random Forest and AdaBoost) (Kostas, 2018).

5.1 Dataset

In order to delve into the characteristic of data exfiltration, the newest public real-world dataset (Unraveled−2023) are utilized with two reasons that ***(i)***: traffic flows are captured from nineteen user hosts and there are eight attackers set in Unraveled−2023 dataset; ***(ii)***: the questions we investigated can be reflected in scenarios of the APT groups (APT28, DragonFly, Bronze Butler and Sandworm Team) excused in this dataset, especially APT28 which split files into small chunk sizes and exfiltrates them through C2 channels (NSA et al., 2021).

As shown in Table 2, there are 7522 examples of data exfiltration in the Unraveled−2023 dataset. Among the Unraveled−2023 dataset, there is an attack implementation timeline where the data exfiltration of the APT occurs in weeks 5 and 6. Therefore, we only selected 7374 samples (including Internet traffic flows with suspected data exfiltration under the assumption that APT attackers have established a foothold in the victim system) from these two weeks (week 5 & 6), shown as Figure 4.

Table 2: Experiment Datasets.

Dataset	APT Life-Cycle	Training	Testing
Unraveled-2023 (Week 5&6)	NormalTraffic	1454574	969837
	LateralMovement	16435	10806
	EstablishFoothold	16249	10862
	DataExfiltration	4499	3023
	Coverup	229	133

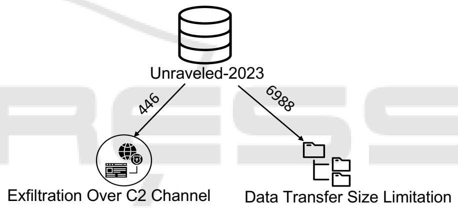


Figure 4: Exfiltration Data Distribution of Experiment Data.

5.2 Evaluation of Detection Mechanisms

Before detecting data exfiltration by APT from the perspective of *Exfiltration over C2 channel* ($DExfil-C2$) and *Exfiltration transfer size limitation* ($DExfil-Size$) with PTL and PTS, it is necessary to investigate whether the detection mechanism we summarized is able to detect data exfiltration or not. For this reason, we selected samples from the Unraveled−2023 dataset into a training set (Establish Foothold: 16278, $DExfil-C2$: 4125, $DExfil-Size$: 265) and a testing set (Establish-Foothold: 10662, $DExfil-C2$: 2787, $DExfil-Size$: 181). Under the assumption that the APT attacker has established a foothold in the victim system, evaluation experiments based on EDeepXGB are implemented.

The classification reports of experiments' results are shown as Figure 5 and Figure 6 ($E-foothold$ represents Establish Foothold).

It can be noticed that with eight different DL models (DNN_{1-4}, CNN_{1-4}), our detection mechanisms

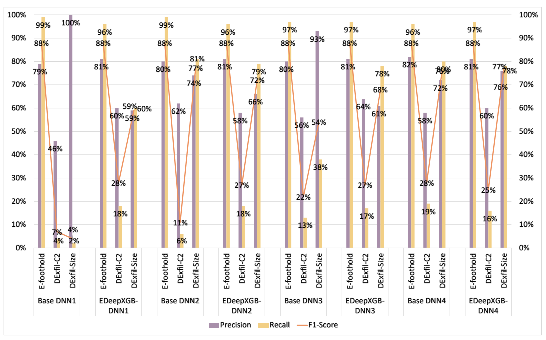


Figure 5: Classification Report Based on DNN_{1-4}.

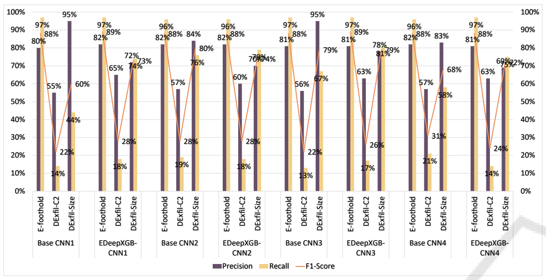


Figure 6: Classification Report Based on CNN_{1-4}.

can successfully help to detect data exfiltration from the perspective of *Exfiltration over C2 channel* and *Exfiltration transfer size limitation*. Moreover, the overall performance (Precision, Recall, and F1-Score) in detecting $DExfil-C2$ and $DExfil-Size$ can be improved by EDeepXGB compared to the performance of their base DL models (DNN_{1-4}, CNN_{1-4}).

In addition, we can see that the F1-Score of $E-foothold$ is almost keeping on 88% both base DL models ($BaseDNN_{1-4}$, $BaseCNN_{1-4}$) and EDeepXGB models ($EDeepXGB-DNN_{1-4}$, $EDeepXGB-CNN_{1-4}$). It means that there is nearly no negative affection on the detection of the other APT steps (except for the step of Data Exfiltration) using our detection mechanisms.

Therefore, it can be said that the detection mechanism we summarized is able to detect data exfiltration.

5.3 Evaluation of Overall Performance

After confirming the feasibility of detecting data exfiltration of APT attacks from two perspectives (*Exfiltration over C2 channel* ($DExfil-C2$) and *Exfiltration transfer size limitation* ($DExfil-Size$)) based on EDeepXGB, we executed a couple of experiments on the whole Unraveled−2023 testing dataset (which be processed by (Sowmya et al., 2023)) to ensure the performance of our proposed method. In addition, to evaluate the overall performance of our EDeepXGB, we also implement four basic machine learning models (Naive Bayes, QDA, Random Forest, and AdaBoost) as a baseline for comparison.

5.3.1 Results of EDeepXGB

With the detection mechanisms (PTL, PTS) we summarized in Section 3, by testing in the whole testing dataset of Unraveled−2023, Figure 7 shows the True Negative (TN) results, True Positive (TP) results, False Negative (FN) results and False Positive (FP) results of data exfiltration detection.

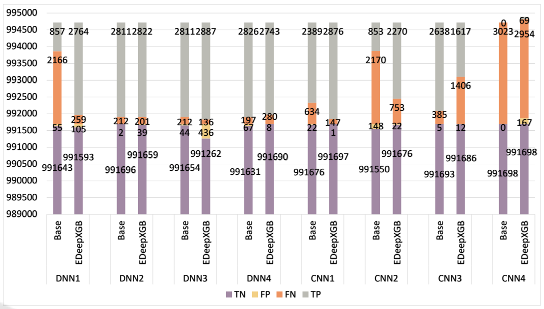


Figure 7: Confusion Matrix of Data Exfiltration Detection.

It can be observed that, with the exception of $EDeepXGB-DNN_4$ and $EDeepXGB-CNN_3$, the other EDeepXGB models predicted data exfiltration more accurately than their corresponding base DL models. In particular, $EDeepXGB-DNN_1$ contributed to an increase in the prediction sample by 1907. This implies that $EDeepXGB-DNN_4$ and $EDeepXGB-CNN_3$ did not perform well in predicting data exfiltration samples for the Unraveled−2023 dataset with the current tree structure. The underlying reason for this could be an overfitting issue, or that the models $EDeepXGB-DNN_4$ and $EDeepXGB-CNN_3$ are not effectively learning from the unbalanced training data provided.

Moreover, although the base CNN_4 is not sensitive to our test data exfiltration samples (TP-(*Base* CNN_4) = 0), it becomes capable of detecting data exfiltration after being incorporated with our proposed method (TP-($EDeepXGB-CNN_4$) = 69).

On the other hand, with the whole dataset of Unraveled−2023, the performance results of EDeepXGB based on DL models (DNN_{1-4}, CNN_{1-4}) are shown in Table 3 and Table 4. It can be observed that the data exfiltration detection ability of $EDeepXGB-CNN_4$ is weakest compared to all other seven ensemble deep learning models, with a Precision of 29%, Recall of 2%, F1-Score of 4%. This deficiency may attributed to a severe overfitting problem in $EDeepXGB-CNN_4$, causing its neuronal structure to become overly tailored to the provided training data.

Table 3: Classification Report of EDeepXGB Based *DNN*.

	Precision	Recall	F1-Score
$EDeepXGB - DNN_1$			
Benign	1.00	1.00	1.00
Data Exfiltration	0.96	0.91	0.94
Establish Foothold	0.98	1.00	0.99
Lateral Movement	0.93	0.96	0.95
Cover up	0.74	0.19	0.30
$EDeepXGB - DNN_2$			
Benign	1.00	1.00	1.00
Data Exfiltration	0.99	0.93	0.96
Establish Foothold	0.98	1.00	0.99
Lateral Movement	0.96	0.99	0.98
Cover up	0.53	0.42	0.47
$EDeepXGB - DNN_3$			
Benign	1.00	1.00	1.00
Data Exfiltration	0.87	0.96	0.91
Establish Foothold	0.99	0.96	0.98
Lateral Movement	0.96	0.99	0.98
Cover up	0.78	0.75	0.76
$EDeepXGB - DNN_4$			
Benign	1.00	1.00	1.00
Data Exfiltration	1.00	0.91	0.95
Establish Foothold	0.98	1.00	0.99
Lateral Movement	0.95	0.99	0.97
Cover up	0.67	0.90	0.77

Table 4: Classification Report of EDeepXGB Based *CNN*.

	Precision	Recall	F1-Score
$EDeepXGB - CNN_1$			
Benign	1.00	1.00	1.00
Data Exfiltration	1.00	0.95	0.97
Establish Foothold	0.98	1.00	0.99
Lateral Movement	0.99	0.96	0.98
Cover up	0.28	0.32	0.30
$EDeepXGB - CNN_2$			
Benign	1.00	1.00	1.00
Data Exfiltration	0.99	0.75	0.85
Establish Foothold	0.97	1.00	0.98
Lateral Movement	0.98	0.96	0.97
Cover up	0.45	0.21	0.29
$EDeepXGB - CNN_3$			
Benign	1.00	1.00	1.00
Data Exfiltration	0.99	0.53	0.70
Establish Foothold	0.97	1.00	0.99
Lateral Movement	1.00	0.96	0.98
Cover up	0.05	0.46	0.09
$EDeepXGB - CNN_4$			
Benign	0.99	1.00	1.00
Data Exfiltration	0.29	0.02	0.04
Establish Foothold	0.76	1.00	0.86
Lateral Movement	0.23	0.01	0.01
Cover up	0.00	0.03	0.00

5.3.2 Results of Baseline Models

In order to confirm the overall performance of our proposed method, we implemented Naive Bayes, QDA, Random Forest, and AdaBoost as our overall performance baseline models. The results of those baselines are shown in Figure 8. All four baseline models can detect APT attacks well (with 97.39% average accuracy). However, the detection ability (Precision, Recall and F1-Score) of baselines are all under 40%.

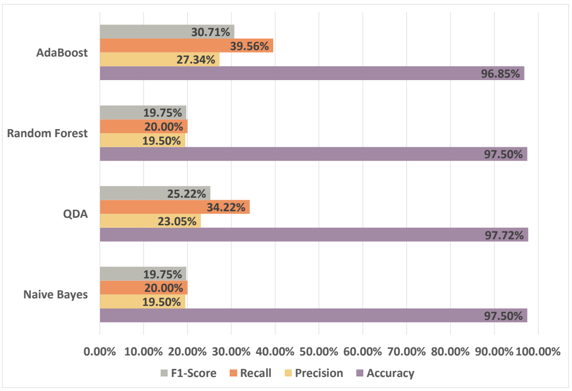


Figure 8: Overall Performance of Baseline Models.

5.4 Discussion

From the results we obtained, we can answer those two questions: ***Q1.*** How to detect data exfiltration of APT attacks within normal traffic? ***Q2.*** How to detect sensitive data exfiltration of APT attacks if the leaked information is split into extremely small sizes in different victim systems?

Since the APT attacker must keep the C2 channel alive to receive exfiltrated sensitive data and balance the transfer time and the size of the exfiltrated data transfer at the same time, those questions become to *detecting data exfiltration over C2 channel* and *detecting exfiltrated data being transfer size limited*. Hence, with the discussion of detection mechanisms (PTL, PTS) in Section 3 and the evaluation results of Section 5.2, it can be said that our proposed method can detect data exfiltration of APT successfully from the perspective of *Exfiltration over C2 channel* and *Exfiltration transfer size limitation*.

Meanwhile, the performance in Section 5.3.1 shows that in a larger sample space (the whole Unraveled−2023 dataset), our EDeepXGB is still strong in detecting data exfiltration of APT attacks.

Otherwise, for the evaluation of overall performance, by supporting the newest public real-world dataset (Unraveled−2023), the comparison results between EDeepXGB and their base DL models are shown in Table 3 and Table 4. Moreover, comparison results between EDeepXGB and our baseline models are all shown in Figure 9.

From these comparative results in Table 3 and Table 4, we can note that although not all of EDeepXGB's classification accuracies are better than that of their base DL models, the overall detection abilities (Precision, Recall, and F1-Score) of EDeepXGB are stronger than the base models. It means that our ensemble deep learning tree helps the model to predict data exfiltration more precisely. Meanwhile, compared to baseline models (shown in Figure 9), the overall performance of our EDeepXGB is significantly better than that of the baselines.

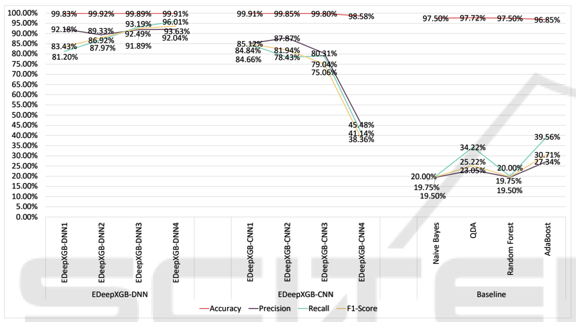


Figure 9: Comparison with Baseline Models.

As Figure 9 shows, in the Unraveled−2023 dataset, the overall performance of $EDeepXGB-DNN_4$ is optimal (Precision: 92.04%, Recall: 96.01%, F1-Score: 93.63%).

However, from the discussion of confusion matrices (Section 5.3.1), it is known that $EDeepXGB-DNN_4$ can not predict well for the data exfiltration samples in the current deep learning tree structure. Hence, by considering the ability and the robustness of detecting data exfiltration of APT attacks and the overall performance results, for the data exfiltration detection problems, the optimal EDeepXGB we implemented is $EDeepXGB-DNN_3$ (Accuracy: 99.98%, Precision: 91.89%, Recall: 93.19%, F1-Score: 92.49%).

6 CONCLUSIONS

To fill the gap of little attention being paid to detecting sensitive data exfiltration after an APT attack has established a foothold on victim systems, and to detect data exfiltration of APT attacks within normal traffic if the exposed information is converted into extremely small sizes in different victim systems, in this paper, we verified that data exfiltration detection problems of APT attack can be analyzed from two perspectives: *Exfiltration over C2 channel* and *Exfiltration transfer size limitation*.

Compared to base deep learning models (four different Deep Neural Networks and four Convolutional Neural Networks) and basic machine learning models (Naive Bayes, Quadratic Discriminant Analysis, Random Forest and AdaBoost), with the detection mechanisms we summarized (*Transfer Lifetime Volatility* and *Transfer Speed Volatility*), the ensemble deep learning tree based on eXtreme Gradient Boosting (EDeepXGB) we proposed can detect the data exfiltration of APT attacks in high overall performance. Meanwhile, with all $EDeepXGB$ based eight DL models, the optimal ensemble deep learning tree we implemented can obtain an overall performance of (Accuracy: 99.98%, Precision: 91.89%, Recall: 93.19%, F1-Score: 92.49%).

Since the data exfiltration detection samples are very small compared to other attack data in public datasets, our next step will be to expose our proposed method to a real-world APT attack environment to test the detection performance. Moreover, our proposed deep learning architecture requires more optimization experiments to find the best implementation.

ACKNOWLEDGEMENTS

This work is supported by JSPS Kakenhi Grant Number 21K11888 and Hitachi Systems, Ltd.

REFERENCES

Abdullayeva, F. J. (2021). Advanced persistent threat attack detection method in cloud computing based on autoencoder and softmax regression algorithm. *Array*, 10:100067.

Alenezi, R. and Ludwig, S. A. (2021). Classifying dns tunneling tools for malicious doh traffic. In *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–9.

Alminshid, K. A. and Omar, M. N. (2020). A framework of apt detection based on packets analysis and host destination. *Iraqi Journal of Science*, page 215–223.

Aseel Kayal, Mark Lechtik, P. R. (2021). Lyceum reborn: counterintelligence in the Middle East; VB2021 localhost — vblocalhost.com. https://vblocalhost.com/uploads/VB2021-Kayal-etal.pdf/. [Accessed 15-April-2023].

Charan, P. S., Anand, P. M., and Shukla, S. K. (2021). Dmapt: Study of data mining and machine learning techniques in advanced persistent threat attribution and detection. In Thomas, C., editor, *Data Mining*, chapter 5. IntechOpen, Rijeka.

Chen, P., Desmet, L., and Huygens, C. (2014). A study on advanced persistent threats. In De Decker, B. and Zúquete, A., editors, *Communications and Multimedia Security*, pages 63–72, Berlin, Heidelberg. Springer Berlin Heidelberg.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794, New York, NY, USA. Association for Computing Machinery.

D'Agostino, J. and Kul, G. (2021). Toward pinpointing data leakage from advanced persistent threats. In *2021 7th IEEE Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*, pages 157–162.

Edgar, T. W. and Manz, D. O. (2017). Chapter 11 - applied experimentation. In Edgar, T. W. and Manz, D. O., editors, *Research Methods for Cyber Security*, pages 271–297. Syngress.

Eric, C., Liam, O., and Nicolas, F. (2012). W32.duqu: The precursor to the next stuxnet. In *USENIX Workshop on Large-Scale Exploits and Emergent Threats*.

Ghafir, I., Hammoudeh, M., Prenosil, V., Han, L., Hegarty, R., Rabie, K., and Aparicio-Navarro, F. J. (2018). Detection of advanced persistent threat using machine-learning correlation analysis. *Future Generation Computer Systems*, 89:349–359.

Irshad, H., Ciocarlie, G., Gehani, A., Yegneswaran, V., Lee, K. H., Patel, J., Jha, S., Kwon, Y., Xu, D., and Zhang, X. (2021). Trace: Enterprise-wide provenance tracking for real-time apt detection. *IEEE Transactions on Information Forensics and Security*, 16:4363–4376.

King, J., Bendiab, G., Savage, N., and Shiaeles, S. (2021). Data exfiltration: Methods and detection countermeasures. In *2021 IEEE International Conference on Cyber Security and Resilience (CSR)*, pages 442–447.

Kostas, K. (2018). Anomaly Detection in Networks Using Machine Learning. Master's thesis, University of Essex, Colchester, UK.

Lal, A., Prasad, A., Kumar, A., and Kumar, S. (2022). Data exfiltration: Preventive and detective countermeasures. In *Proceedings of the International Conference on Innovative Computing & Communication (ICICC) 2022*.

Mamun, M. and Shi, K. (2021). Deeptaskapt: Insider apt detection using task-tree based deep learning. In *2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 693–700.

Mandiant (2014). APT1 — Exposing One of China's Cyber Espionage Units — mandiant.com. https://www.mandiant.com/resources/reports/apt1-exposing-one-chinas-cyber-espionage-units. [Accessed 18-April-2023].

Mengqi, Z., Yang, L., Guangxi, Y., Bo, L., and Weiping, W. (2022). Detecting dns over https based data exfiltration. *Computer Networks*, 209:108919.

Moghaddam, A. K. and Zincir-Heywood, N. (2020). Exploring data leakage in encrypted payload using supervised machine learning. *Proceedings of the 15th International Conference on Availability, Reliability and Security*.

NSA, CISA, FBI, and NCSC (2021). Russian gru conducting global brute force campaign to compromise enterprise and cloud environments. https://media.defense.gov/2021/Jul/01/2002753896/-1/-1/1/CSA_GRU_GLOBAL_BRUTE_FORCE_CAMPAIGN_UOO158036-21.PDF. [Accessed 15-April-2023].

Sabir, B., Ullah, F., Babar, M. A., and Gaire, R. (2021). Machine learning for detecting data exfiltration: A review. *ACM Comput. Surv.*, 54(3).

Sabir, B., Ullah, F., Babar, M. A., and Gaire, R. (2022). Machine Learning for Detecting Data Exfiltration. *ACM Computing Surveys*, 54(3):1–47.

Sowmya, M., Kritshekhar, J., Abdulhakim, S., Garima, A., Yuli, D., Ankur, C., and Dijiang, H. (2023). Unraveled — a semi-synthetic dataset for advanced persistent threats. *Computer Networks*, 227:109688.

Stojanović, B., Hofer-Schmitz, K., and Kleb, U. (2020). Apt datasets and attack modeling for automated detection methods: A review. *Computers & Security*, 92:101734.

Thongsuwan, S., Jaiyen, S., Padcharoen, A., and Agarwal, P. (2021). Convxgb: A new deep learning model for classification problems based on cnn and xgboost. *Nuclear Engineering and Technology*, 53(2):522–531.

Veena, R. C. and Brahmananda, S. H. (2022). A framework for apt detection based on host destination and packet—analysis. In Smys, S., Bestak, R., Palanisamy, R., and Kotuliak, I., editors, *Computer Networks and Inventive Communication Technologies*, pages 833–840, Singapore. Springer Singapore.

Xianrui, M. and Joan, F. (2020). Privacy-preserving xgboost inference. In *NeurIPS 2020 Workshop on Privacy Preserving Machine Learning*.

Zebin, T., Rezvy, S., and Luo, Y. (2022). An explainable ai-based intrusion detection system for dns over https (doh) attacks. *IEEE Transactions on Information Forensics and Security*, 17:2339–2349.

Zimba, A., Chen, H., Wang, Z., and Chishimba, M. (2020). Modeling and detection of the multi-stages of Advanced Persistent Threats attacks based on semi-supervised learning and complex networks characteristics. *Future Generation Computer Systems*, 106:501–517.

Zou, Q., Singhal, A., Sun, X., and Liu, P. (2020). Automatic recognition of advanced persistent threat tactics for enterprise security. In *Proceedings of the Sixth International Workshop on Security and Privacy Analytics*, IWSPA '20, page 43–52, New York, NY, USA. Association for Computing Machinery.