Optimal Packet Padding for IoT Traffic Obfuscation

Críston P. de Souza[*], **Antonio J. Pinheiro**[†], **Jeandro M. Bezerra**[*‡], and **Michele Nogueira**[‡]
[*] Universidade Federal do Ceará (UFC), Quixadá
[†] Centro de Estudos Avançados do Recife (CESAR)
[‡] Departamento de Ciência da Computação, Universidade Federal de Minas Gerais (UFMG)
Email: {jeandro, criston}@ufc.br, ajp@cesar.org.br, michele@dcc.ufmg.br

GLOBECOM 2024 - 2024 IEEE Global Communications Conference | 979-8-3503-5125-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/GLOBECOM52923.2024.10901754

***Abstract*—The Internet of Things (IoT) brings numerous benefits for society. However, it also allows the leakage of sensitive information (e.g., habits, behaviors, and health-related data), posing privacy risks. Solutions have tried to obfuscate network traffic and address this challenge. They typically involve padding and fragmentation techniques, that come with trade-offs. This paper presents an optimal packet padding solution to define the most efficient number of bytes to insert into packets and achieve obfuscation. The approach is based on dynamic programming and relies on a discrete method, which exhibits the time complexity is $\Theta(mn^2)$ and the memory complexity is $\Theta(mn)$. As the number of packets increases, memory consumption increases. Then, it introduces an approximate solution leveraging probability distributions to address scalability. Performance evaluation follows a trace-driven approach and employs metrics as accuracy, F1-score, and overhead. The proposal has obfuscated the identification of IoT devices by up to 50%, reducing overhead by up to 4 times compared to the traditional state-of-the-art methods.**

***Index Terms*—Obfuscation, padding, dynamic programming**

I. Introduction

The Internet of Things (IoT) paradigm integrates billions of devices performing sensing, actuation, and network communication. A passive adversary identifies IoT devices and their events through traffic-based analysis [1]. Traffic features such as packet length, the interval between packets, flow direction, and transmission rate represent critical concerns as they lead to data leaks and user behavior. For instance, applying Machine Learning (ML) to packet length allows identifying if a user made a voice command, sleep disorder events or extramarital activities [2]. Adversaries infer user information analyzing network traffic patterns, even over encrypted data [2], [3].

Preventing sensitive data from being leaked and potentially employed to make inferences about user private lives [2] is paramount. Data obfuscation mechanisms protect against traffic-based attacks in IoT by approaches as packet padding [4]–[7] and dummy traffic [1]. Padding consists in adding redundant *bytes* to packets transmitted over the network, adding a cost (*overhead*). The dummy traffic approach inserts fake traffic into the network to obfuscate the real traffic [1]. Both approaches cause overhead in the IoT network, that can cause delays or overloads, deteriorating user experience.

Hence, proposing traffic obfuscation solutions to balance privacy and performance is challenging. There are two dilemmas. On the one hand, the obfuscation must be secure. It must protect user privacy because network traffic closely relates to IoT users' private habits. On the other hand, the obfuscation must be efficient. It must cost low network bandwidth and delay since IoT devices are usually resource-constraint and delay-sensitive [7].

This paper presents an optimal packet padding solution for IoT traffic obfuscation. The solution is a dynamic programming approach to find optimized values for the number of bytes inserted into packets. Initially, subproblems are defined with the decision policy to determine the optimal size, *i.e.,* to minimize the padding size. Each subproblem is defined as the first $m' \leq m$ obfuscated packet lengths and the first $n' \leq n$ traffic packet lengths, where n is the number of distinct traffic packet lengths and m is a parameter defining the number of obfuscated packet lengths. As the optimal solution of subproblems are stored in main memory, the dynamic programming provides a solution with much less computational effort. The main contributions of this paper are: i) an optimal packet padding solution based on dynamic programming to obfuscate IoT traffic; ii) the theoretical analysis showing that the time complexity is $\Theta(mn^2)$ and the memory complexity is $\Theta(mn)$; iii) an approximate solution through empirical distributions for cases of increasing variation in packet lengths (that is, grouping packet lengths to reduce the n).

Performance evaluation follows a trace-driven approach with the application of ML to identify IoT devices. Evaluations compare the solution to the packet padding strategy from [4] in terms of accuracy and overhead. The proposal has obfuscated the identification of IoT devices by up to 50%, reducing overhead by up to 4 times compared to Pinheiro *et al.*, and up to ≈ 6 times compared to traditional state-of-the-art methods.

This paper proceeds as follows. Section II discusses related works. Section III describes the packet padding proposal. Section IV details the experiments. Section V presents the results, and Section VI concludes the paper.

II. Related Works

Recent studies [1], [8] have reported identifying events and IoT devices that violate user privacy. In a nutshell, these works follow approaches such as packet padding and false traffic injection (dummy traffic) to prevent privacy leakage in IoT environments. This section highlights the main approaches.

Traffic obfuscation by padding consists of adding extra bytes to packets to mask their original sizes. This approach intends to eliminate patterns identified in traffic. It follows

This research is supported by São Paulo Research Foundation (FAPESP), grants 2018/23098-0 and 2023/13902-4.

different strategies to adjust the packet length, such as linear, mice/elephants, MTU [9], and random [10]. Pinheiro *et al.* [4] proposed a padding solution based on four levels that depend on network monitoring conditions. The levels, empirically defined as 100, 500, 700 and 900, denote the minimum length of the packets belonging to the respective level after padding. Level 100 has better performance because it generates less overhead. When the network is idle, it is more appropriate to apply level 900, which improves privacy. The work [7] presented a solution, which defines a fixed value to obfuscation. All packets sent by IoT devices are the same size and have fixed values for the interval between them. This approach makes it difficult for adversaries to gain information about the user's actions and device identification. However, the authors describe a weighty overhead. In this context, another obfuscation method called MFO (Multipath Fixed-value Obfuscation) permits sending smaller packets with larger packet intervals, lowering the bandwidth and delay overhead. MFO has a better trade-off between security and efficiency compared to the work of [4].

Chaddad *et al.* [11] proposed a system that uses optimal models to modify the packet length to obfuscate traffic. The mechanism can alter the probability distribution of packet length generated by applications installed on mobile devices. For each packet, the mechanism identifies which packet size distribution of other applications would produce the smallest packet size when used to confuse the attacker. This approach requires prior knowledge of the probability distribution of the applications for which the packet length will change over time due to multiple factors, such as software updates and changes in user behaviors. In this paper, the solution is not based on probability distributions of mobile device applications. The focus is on determining new packet sizes that minimize padding, considering the packet size distribution of all traffic. In addition to the padding strategy, the solution of [11] implements the fragmentation technique. The disadvantage of fragmentation is that it increases the number of packets processed by intermediate devices, such as switches and routers, which can degrade communication performance.

Santos *et al.* [1] developed a network obfuscation method, called MITRA, which considers dummy traffic insertion to mitigate attacks based on packet inspection on the network. MITRA analyzes the network traffic and dynamically generates four different levels of dummy traffic. Results show that MITRA achieves low and medium overhead in obfuscation performance. In 2024, Li *et al.* [8] proposed the HomeSentinel, an intelligent mechanism to protect smart homes against traffic-based attacks. The solution distinguishes IoT traffic from raw traffic without user operations and generates dummy IoT traffic considering adversarial spatial-temporal patterns. Then, the authors heuristically merge dummy IoT traffic with real IoT traffic in expected spatial-temporal patterns to contain traffic-based attacks. The authors did not specify which padding level was used to compare performance with works in the literature. For instance, the proposal by Pinheiro *et al.* [4] has four levels to implement padding.

III. Determining the obfuscated packet lengths

The proposal involves modifying the number of bytes inserted into packets to obfuscate IoT network traffic. The aim is to optimize the packet padding tradeoff between privacy and computing time, *i.e.*, increase users' privacy and minimize overhead. An assumption lies in a set of a finite number of distinct values for packet lengths, so Property 1 provides the expected padding size for a given list of packet lengths.

Property 1. *Let $L = x_1 < \cdots < x_n = U$ be the n distinct traffic packet lengths with positive probability, and let $p(x)$ be the probability that a packet has length x. Let Z be a random variable representing the padding size for this packet length distribution when packets have lengths $L = y_0 \leq y_1 \leq y_2 \leq \cdots \leq y_m = U$. Then, the expected padding size is given by*

$$E[Z] = \sum_{i=1}^{m} \sum_{y_{i-1} < x_j \leq y_i} p(x_j) \cdot (y_i - x_j). \tag{1}$$

Lemma 2 states that a list of obfuscated packet lengths minimizes the expected padding size and uses only values in the set of traffic packet lengths, thus reducing the search space to a finite set.

Lemma 2. *There is a solution $L \leq y_1^* \leq y_2^* \leq \cdots \leq y_m^* = U$ for the obfuscated packet lengths that minimize $E[Z]$ and satisfy $y_i^* \in \{x_1, x_2, \ldots, x_n\}$ for all $i = 1, 2, \ldots, m$.*

Proof. Suppose by contradiction that no optimal solution satisfies $y_i^* \in \{x_1, x_2, \ldots, x_n\}$ for all $i = 1, 2, \ldots, m$. Then, if $L \leq y_1^* \leq y_2^* \leq \cdots \leq y_m^* = U$ is an optimal solution, we have that there is a $k \in \{1, 2, \ldots, m-1\}$ such that $y_k^* \notin \{x_1, x_2, \ldots, x_n\}$. Let j be the index that satisfies $x_j < y_k^* < x_{j+1}$. Thus, we can reduce the expected padding size by an amount $p(x_j) \cdot (y_k^* - x_j) > 0$ by making the kth obfuscated packet length equal to x_j (instead of y_k^*). Therefore, the solution is not optimal (a contradiction). □

By the recurrence provided in Property 3, $d(m, n)$ is the smallest average padding size that can be obtained using m obfuscated packet lengths.

Lemma 3. *Let $d(m', j)$, for $1 \leq j \leq n$ and $m' \geq 1$, be the smallest average padding size for packets of length less than or equal to x_j, assuming that an obfuscated packet length has been fixed to x_j and we can still choose $m' - 1$ obfuscated packet lengths smaller than x_j. Then,*

$$d(m', j) = \begin{cases} 0, & j = 1 \\ c(0, j), & j > 1 \wedge m' = 1 \\ \min_{i=1}^{j-1}\{c(i, j) & \\ \qquad + d(m' - 1, i)\}, & \textit{otherwise} \end{cases} \tag{2}$$

where,

$$c(i, j) = \sum_{k=i+1}^{j-1} p(x_k) \cdot (x_j - x_k) \tag{3}$$

is the average padding size for packets larger than x_i and smaller than x_j, assuming that the only obfuscated packet length in the range $(x_i, x_j]$ has value x_j.

Proof. Let's consider only optimal solutions where the obfuscated packet lengths are traffic packet lengths (Lemma 2). The recurrence checks every possible value x_i for $y_{m'-1}$, assuming that $y_{m'}$ was fixed to x_j. In this case, as $c(i, j)$ is the average padding size for packets larger than x_i and smaller than x_j, the property arises from the optimal substructure of $E[Z]$. In other words, if there is an optimal solution O with $y_{m'-1} = x_i$, then the obfuscated packet lengths in O smaller than x_j form an optimal solution to the problem of minimizing the average padding of the packets of length at most x_i. □

Based on Lemma 3, Algorithm 1 provides an optimal solution for the average padding minimization. The algorithm receives as input a data structure containing the values of $c(i, j)$, defined in Equation (3). Note that $c(i, j)$ can be precomputed for all possible values of i and j in time $\Theta(n^2)$ using the recurrence of Equation (4).

$$c(i,j) = \begin{cases} 0, & j - i \leq 1 \\ c(i+1, j) & \\ \quad + p(x_{i+1}) \cdot (x_j - x_{i+1}), & \text{otherwise} \end{cases} \tag{4}$$

Theorem 4. *Algorithm 1 provides a list of m obfuscated packet lengths that minimize the average padding size, using $\Theta(mn^2)$ time and $\Theta(mn)$ memory.*

Proof. Algorithm 1 uses dynamic programming [12] to efficiently compute the recurrence values defined in Equation (2). While calculating these values, the algorithm stores (at line 11) the decision $pred$ that was made to obtain the value of each element of the dynamic programming table d, thus allowing the recovery of an optimal solution in lines 12–15. The instruction executed most times is the test on line 9, totaling $\sum_{m'=2}^{m} \sum_{j=2}^{n} \sum_{i=1}^{j-1} 1 \in \Theta(mn^2)$ tests. The algorithm allocates the array y with m elements and two matrices, d and $pred$, both with mn elements. □

The proposed algorithm has an execution time that grows only linearly with the number of obfuscated packet lengths m but grows with the square of the number of distinct traffic packet lengths n. Thus, if n is high enough to make the application of the algorithm unfeasible, we can obtain an approximate solution by pre-processing the input to provide an empirical distribution of traffic packet lengths. In this case, the execution time will grow with the square of the number of bins in the empirical distribution. Therefore, as the users choose this number of bins, they control the tradeoff between execution time and increased privacy.

Algorithm 1: Discrete Optimal Padding

input : m: number of obfuscated packet lengths. n: number of traffic packet lengths. $x_1, \ldots, x_n$: traffic packet lengths. $c(i, j)$ as defined in Property 3.
output: List y of obfuscated packet lengths.

```
// dynamic programming matrix
let d[1..m, 1..n] be a matrix of floats
// predecessors matrix
let pred[1..m, 1..n] be a matrix of integers
// fill the matrices d and pred
for m' ← 1 to m do d[m', 1] ← 0
for j ← 2 to n do d[1, j] ← c(0, j)
for m' ← 2 to m do
    for j ← 2 to n do
        d[m', j] ← ∞
        for i ← 1 to j − 1 do
            if c(i, j) + d[m' − 1, i] < d[m', j] then
                d[m', j] ← c(i, j) + d[m' − 1, i]
                pred[m', j] ← i
// compute solution y from pred
let y[1..m] be a list of floats
y[m] ← U
for m' ← m − 1 downto 1 do
    y[m'] ← pred[m' + 1, y[m' + 1]]
return y
```

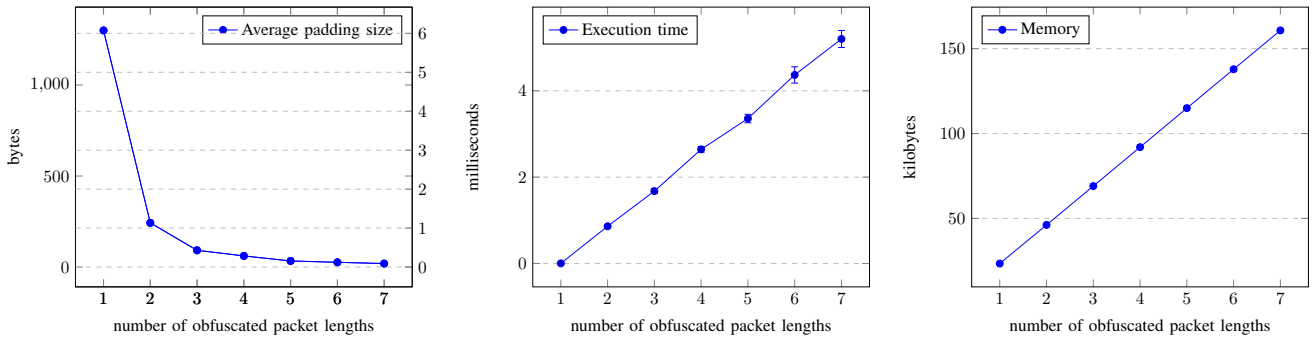
IV. Experiments

Experiments have been executed in a notebook with 8 Core i7 with 2.8 GHz and 12 MB of cache, 32 GB of RAM, and Ubuntu 22.04 operating system. Experiments employ existing datasets comprising 20 packet capture (PCAP) files, each corresponding to 24 hours of traffic capture generated by 21 IoT devices from [13]. Network traffic discloses many metadata, but only few are employed to minimize the computational cost. Metadata packet length has been chosen because it is more appropriate for device characterization [2].

A. Adversary Models

Experiments evaluate different aspects of privacy improvement and the impact on network communication performance. The two adversary scenarios are as follows: (i) an external observer who monitors activities within an IoT domain. The external adversary captures traffic generated by the devices of its victims outside the domain after the router, wherein the padding mechanism is implemented [4]; (ii) an internal observer (the provider of the padding solution) can use its condition to surveil its users.

The procedure described simulates a scenario in which an observer trains classifiers with IoT data and uses them to identify their victim's devices [4]. What differentiates an external observer from an internal observer in practice is the training of ML models with obfuscated data. The internal applies stratified tenfold cross-validation.



(a) Optimal average padding (b) Average execution time (c) Memory usage

Fig. 1: Optimal average padding, execution time and memory usage of Algorithm 1 varying the number of obfuscated packet lengths. The error bars represent two standard deviations.

B. Privacy-Preserving Evaluation

The traffic-based attack uses machine learning algorithms to violate the user's privacy [1], [2]. Observers make inferences based on data from their victims' devices. Therefore, ML algorithms evaluate traffic obfuscation mechanisms through packet padding. ML algorithms measure the improvement in privacy produced by the proposed solution. The performance of these algorithms in identifying IoT devices was measured using packet length. The aim is to reduce this ML model's accuracy with the proposed optimal padding solution. Generally, these models inspect the packet payload or employ dozens of traffic attributes, making classifying IoT devices one computationally expensive process [2]. The devices are identified by applying the ML mechanism from three packet length statistics: mean, standard deviation, and the number of bytes [2]. These statistics are from encrypted traffic grouped into one-second windows. Finally, the classification mechanism has been applied to these statistics. Four classifiers were implemented: k-Nearest Neighbors (k-NN), Random Forest (RF), Decision Tree (DT), and Support Vector Machine (SVM). The classifiers belong to different categories – including nearest neighbors, trees, ensemble, and vector machines – to clarify the proposed solution performance with several algorithms. The classifiers were trained offline.

The padding strategies are expected to eliminate patterns that can be identified in the traffic. The impact of padding on the tradeoff between privacy and communication overhead must be considered. The proposal is compared to the solution of Pinheiro *et al.* [4] and well-known padding strategies.

V. Results

This section presents the results for evaluation metrics as execution time, memory usage and byte overhead when applying the obfuscation process. Finally, it shows a discussion about the classifiers' accuracy.

A. Time and Space Performance Analysis

Table I and Fig. 1 show results for running Algorithm 1 having as input the previously described datasets. In Table I, the average padding size is about 6 times the average traffic packet length when there is only 1 obfuscated packet length, and the byte overhead drops to about the traffic packet length for $m = 2$, and to less than half the traffic packet length for $m = 3$. This ratio reaches about 6% for $m = 10$ obfuscated packet lengths. There is a small padding overhead, even using a few obfuscated packet lengths. Note that the greater the number of obfuscated packet lengths, the more difficult it is to identify devices on the network through traffic analyses.

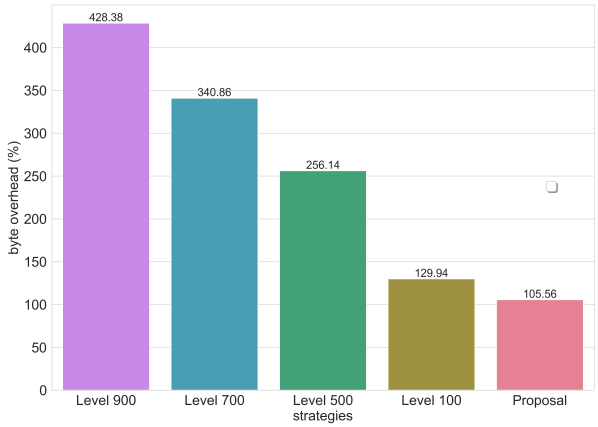
Pinheiro *et al.* [2] showed that the mean packet length of most devices is less than 200 bytes. Therefore, when $m = 1$, the amount of padding added to packets is more significant than other values of m. From Fig.(1a), as the value of m increases, the required padding decreases, reducing the overhead generated by the solution. The y-axis in Fig. (1a) has two scales: bytes (left axis) and average padding size divided by the average traffic packet length (right axis). From figures (1b) and (1c), the execution time and memory consumption of Algorithm 1 increase linearly with the number of obfuscated packet lengths, making the algorithm viable even for a much larger number of obfuscated traffic. The average execution time was extracted from 1000 executions. Theoretical analysis showed that memory consumption is also linear with the number of different packet lengths. Execution time grows with the square of the number of distinct obfuscated packets, which may require approximate solutions through empirical distributions, as discussed in Section III.

TABLE I: Ratio between optimal padding size and average packet length, varying the number of obfuscated packets.

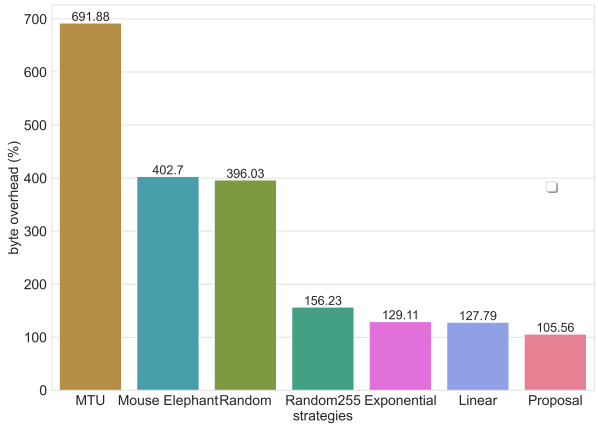
# obfus. packets	avg. padding / avg. packet length
1	607.5 %
2	113.4 %
3	42.9 %
4	28.6 %
5	15.6 %
6	12.1 %
7	8.9 %
8	7.8 %
9	6.8 %
10	5.9 %

B. Byte Overhead Analysis

The increase in the number of bytes transmitted in the network impacts the performance. The byte overhead negatively influences the communication performance. Therefore, minimizing the number of bytes in packets is essential to reduce the communication response time in IoT devices. Fig. (2a) shows the byte overhead generated by solutions of [4] that increases the number of bytes inserted into obfuscation according to the level and well-known approaches Fig. (2b).



(a) Proposal and Pinheiro *et al.* [4]



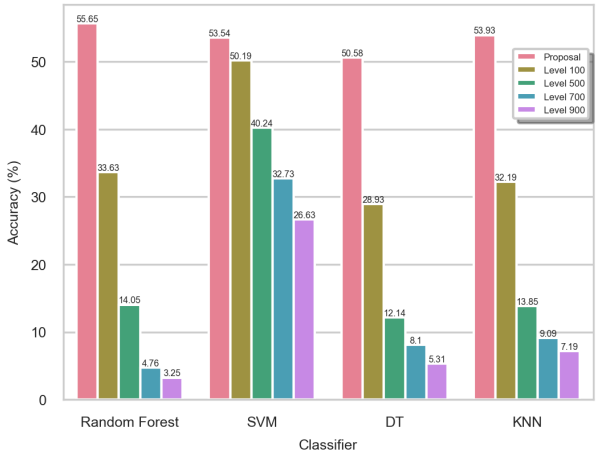
(b) Proposal and well-known approaches

Fig. 2: Comparing padding strategies.

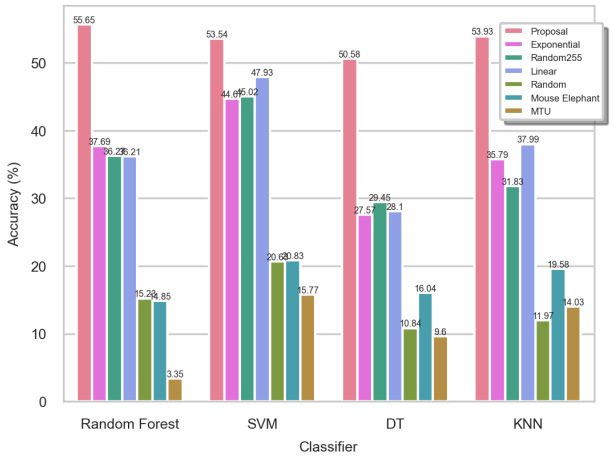
Fig. (2a) shows that the proposal reduces ≈ 23% about Level 100 and ≈ 322% to Level 900. Compared to well-known solutions in Fig. 2b, the proposal reduces ≈ 586% compared to MTU and ≈ 22% compared to Linear. The Linear, Exponential and Random255 solutions generate low overhead because they increase the packet length slightly, generating a low gain in privacy. Furthermore, these solutions are static; *i.e.,* that is, they continually modify the packet length in the same way, making it impossible to react to traffic fluctuations on the network.

C. Privacy Preserving

The RF, SVM, DT and k-NN models achieve 96%, 94%, 96% and 94%, respectively, for accuracy, recall and F1-score metrics. These results indicate how much information an adversary can infer from IoT devices users. Figs. 3 and 4 present the average accuracy results for the external and internal adversarial scenarios, respectively. The proposal presents ≈ 53% accuracy in the external scenario. Even with this tradeoff: byte overhead optimization, the proposal presents an average accuracy metric higher than the state-of-the-art. For classification, this accuracy is low; therefore, the proposal achieves good performance. Compared with the Level 100 (3a), in RF, there is a difference of ≈ 22%; this percentage does not make data obfuscation impossible.



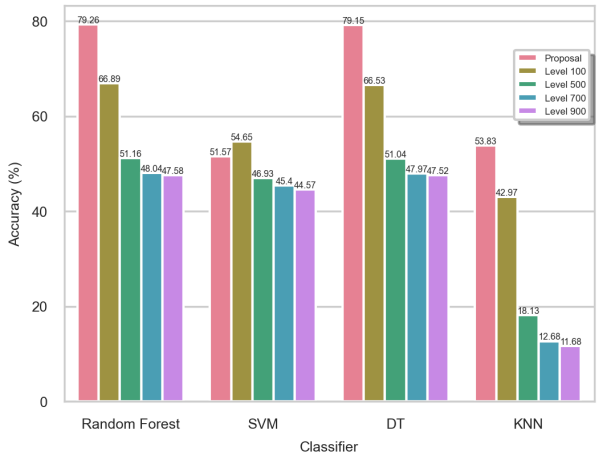
(a) Proposal and Pinheiro *et al.* [4]



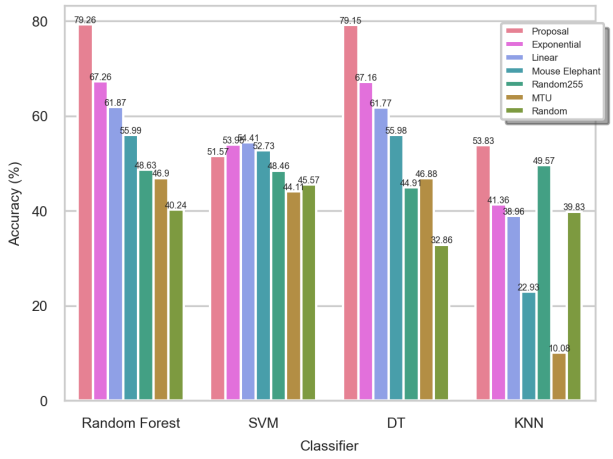
(b) Proposal and well-known approaches

Fig. 3: Comparing performance for the external observer.

Fig. 4 shows the average accuracy results for the scenario where the solution provider is the adversary. The proposal presented an average accuracy of ≈ 79% for RF and DT. The difference is ≈ 12% for Level 100. The performance reduction did not occur in SVM and KNN in the two comparisons in this scenario. This result is because models are trained through 10-*fold cross-validation*, which uses more data and increases training time. Fig. 5 presents the F1-score for the external observer scenario. The recall and F1-score results were the



(a) Proposal and Pinheiro *et al.* [4]



(b) Proposal and Well-known approaches

Fig. 4: Comparing performance for the internal observer.

same as those for accuracy due to the necessary application of the microaveraged approach, in which all instances of the data set have the same weight, handling unbalanced data.

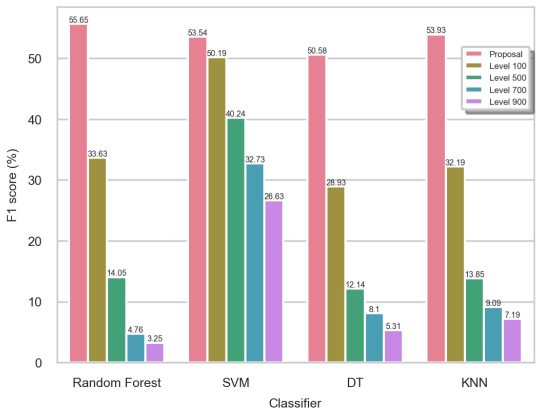


Fig. 5: Average F1-score for the external observer.

VI. Conclusion

This paper presented an optimal discrete padding solution based on dynamic programming to obfuscate traffic to protect user privacy. The proposal's analytical deduction shows that it is feasible to generate an optimal solution through the instrument properties, lemmas and theorems. The proofs show a list of m size of obfuscated packet lengths that minimizes the average padding size, using $\Theta(mn^2)$ time, and the memory complexity is $\Theta(mn)$. Validation through performance evaluation follows a trace-driven approach and employs metrics such as accuracy, F1-score, and byte overhead. The optimization proposal reduced $\approx$ 322% for the highest level of padding and $\approx$ 23% for the lowest level about the relevant work from literature [4]. The proposal's performance was lower in the accuracy of the ML models, 53% in the external observer scenario. This metric's performance was lower, but this accuracy rate is within satisfactory performance. The most significant challenge in padding approaches is keeping the data hidden and protecting IoT user privacy without compromising the overhead. The optimization model proposed in this paper achieved this objective.

References

[1] B. V. Santos, A. Vergütz, R. T. Macedo, and M. Nogueira, "A dynamic method to protect user privacy against traffic-based attacks on smart home," *Ad Hoc Networks*, vol. 149, p. 103226, 2023.

[2] A. J. Pinheiro, J. de M. Bezerra, C. A. Burgardt, and D. R. Campelo, "Identifying iot devices and events based on packet length from encrypted traffic," *Computer Communications*, vol. 144, pp. 8–17, 8 2019.

[3] M. Skowron, A. Janicki, and W. Mazurczyk, "Traffic fingerprinting attacks on internet of things using machine learning," *IEEE Access*, vol. 8, pp. 20386–20400, 2020.

[4] A. J. Pinheiro, P. F. D. Araujo-Filho, J. D. M. Bezerra, and D. R. Campelo, "Adaptive packet padding approach for smart home networks: A tradeoff between privacy and performance," *IEEE Internet of Things Journal*, vol. 8, pp. 3930–3938, 3 2021.

[5] A. Shenoi, P. K. Vairam, K. Sabharwal, J. Li, and D. M. Divakaran, "iPET: Privacy enhancing traffic perturbations for secure IoT communications," *Proceedings on Privacy Enhancing Technologies*, vol. 2, pp. 206–220, 2023.

[6] S. Xiong, A. D. Sarwate, and N. B. Mandayam, "Network traffic shaping for enhancing privacy in IoT systems," *IEEE/ACM Transactions on Networking*, vol. 30, no. 3, pp. 1162–1177, 2022.

[7] G. He, X. Xiao, R. Chen, H. Zhu, Z. Zhang, and B. Xu, "Secure and efficient traffic obfuscation for smart home," in *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, pp. 6073–6078, 2022.

[8] B. Li, Y. Chen, L. Zhang, L. Wang, and Y. Cheng, "Homesentinel: Intelligent anti-fingerprinting for iot traffic in smart homes," *IEEE Transactions on Information Forensics and Security*, pp. 1–1, 2024.

[9] K. P. Dyer, S. E. Coull, T. Ristenpart, and T. Shrimpton, "Peek-a-boo, i still see you: Why efficient traffic analysis countermeasures fail," in *2012 IEEE Symp. on Secur. and Privacy*, (San Francisco, CA, USA), pp. 332–346, May 2012.

[10] S. Xiong, A. D. Sarwate, and N. B. Mandayam, "Defending against packet-size side-channel attacks in IoT networks," in *2018 IEEE Int. Conf. on Acoustics, Speech and Signal Process. (ICASSP)*, pp. 2027–2031, April 2018.

[11] L. Chaddad, A. Chehab, I. H. Elhajj, and A. Kayssi, "Optimal packet camouflage against traffic analysis," *ACM Transactions on Privacy and Security (TOPS)*, vol. 24, no. 3, pp. 1–23, 2021.

[12] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2022.

[13] A. Sivanathan, H. H. Gharakheili, F. Loi, A. Radford, C. Wijenayake, A. Vishwanath, and V. Sivaraman, "Classifying iot devices in smart environments using network traffic characteristics," *IEEE Transactions on Mobile Computing*, vol. 18, no. 8, pp. 1745–1759, 2018.