Cybersecurity Analysis Through Shapley Values for a Network Traffic Dataset of Android Malware

Daria Nikolaeva, Alessandro Buratto, and Leonardo Badia
Dept. of Information Engineering (DEI), University of Padova, Italy
{ daria.nikolaeva@studenti. , alessandro.buratto.1@phd. , leonardo.badia@} unipd.it

***Abstract*—We explore the use of machine learning, specifically Random Forest classifiers, combined with SHapley Additive exPlanations values, to detect Android malware. We leverage diverse datasets, including the Android Genome Project and Drebin, to distinguish between benign and malicious applications. Emphasizing feature importance through SHAP analysis, we aim to enhance model interpretability and effectiveness in cybersecurity. This approach not only improves threat detection accuracy, but also contributes to the broader field of explainable AI in cybersecurity. The paper is structured to cover theoretical foundations, methodology, results, and future directions in this evolving area of study. Also, based on practical findings, we highlight the importance of the data source and transmission patterns as a way to identify malware.**

***Index Terms*—Machine learning, Random forest classifier, Shapley value, Android Malware, Cybersecurity.**

I. Introduction

Over the last few years, the Android operating system (OS) has gained widespread and pervasive diffusion, not only in mobile phones and tablets, but also for heavily heterogeneous Internet of things (IoT) devices, thanks to its versatility and adaptability [1]. However, its success also implies a heavy vulnerability to cyberattacks, which may be connected to several factors.

First of all, since Android systems are designed for a vast audience, security misconfigurations are quite common, as pointed out by many studies [2], [3]. This can be due to several possible flaws in the chain from developers to manufacturers and end users, which may make Android platforms susceptible to cyberattacks that simply leverage the inexperience of security countermeasures over a large number of users. Especially, Android's permission system that implies each piece of software asking for authorization to access personal information, can be exploited by malware developers [4].

Combined with a general lack of user awareness about cybersecurity, and the extensive operating time that Android-based devices are commonly operating without being directly interacted with, this might allow hackers to extract sensitive information without detection [5], which may further lead to data exfiltration, theft, and spoofing [6], [7].

This study harnesses machine learning (ML) to fortify defenses against cyber threats and successfully detect malware applications, particularly focusing on Android's application and network layers [8]. Leveraging the vast data generated by Android's open-source ecosystem, supervised ML techniques as a random forest classifier are employed to discern between benign and malicious applications [9]. The core of the analysis is the application of Shapley values [10], [11], derived from cooperative game theory and quantified through the SHapley Additive exPlanations (SHAP) toolbox [12], to evaluate the contribution of individual data points. This approach is essential in understanding the significance of each feature in the dataset, thereby enabling more accurate threat detection and analysis [13], [14].

We use a dataset [15] encompassing a wide array of Android applications, both benign and malicious. It includes various open data available to the research community, from the Android Genome Project (MalGenome) [16], which contains 1260 applications categorized into 49 families, and Drebin [17], which offers 5560 samples across 179 families. Furthermore, it comprises a significant number of applications from AndroZoo and samples from VirusShare and DroidCollector, enriching the analysis with web traffic captures in virtual environments. This collection allows for a comprehensive examination of Android malware, with the goal of developing ML models for effective threat detection [18].

The concept of Shapley value, dating back to [19], is guided by the desire of identifying the most relevant features in a decision-making process. Recent applications in ML context suggest its usage for the identification of the most significant features that contribute to the classification of data points [20], [21], in our case as malicious or non-malicious. This method not only enhances the understanding of feature influence on malware detection, but also guides the optimization of ML models for more effective performance. By applying SHAP, the average marginal contribution of each feature is calculated across all possible combinations [22]. Thus, we can provide a meaningful evaluation of data points in this cybersecurity-focused ML application [23]–[25].

Furthermore, the approach extends beyond mere feature importance. It offers insights into identifying outliers, corrupted data, and guides the acquisition of future data to refine the predictor's performance. This not only bolsters the capabilities of cybersecurity measures, but also contributes to the broader discourse on data evaluation in ML [26].

The remainder of this paper is organized as follows: Section II reviews the theoretical underpinnings of cooperative games, the Shapley value, and related work in ML and cybersecurity. Section III outlines the methodology, dataset details, and application of the Shapley value in malware recognition. Section IV presents the findings, emphasizing the effectiveness of approach applied. Finally, Section V concludes the paper and discusses future research directions.

II. Related Work

In the realm of machine learning for cybersecurity threat recognition, an array of studies [21], [23] has underscored the efficacy of forest classifiers, notably Random Forest and XGBoost. These classifiers have garnered significant attention due to their robustness and remarkable capacity to handle voluminous and intricate datasets. This extends across diverse domains, encompassing areas such as cybersecurity, transportation, and healthcare [20], [27], [28]. The inherent capability of these techniques to scrutinize a multitude of features and evaluate their relative importance in predicting outcomes renders particularly suitable for this task.

The integration of Shapley values, rooted in the domain of game theory [13], [29], has emerged as a valid tool for augmenting the interpretability of machine learning models, particularly within the SHAP framework. Shapley values introduce a systematic methodology for quantifying the individual contributions of features to a model's predictions, shedding light on the model's decision-making process. This can be especially beneficial in the context of large datasets [21], [30], where comprehending and enhancing the model represents a serious challenge, but on the other hand can significantly simplify further data collection and understanding.

Within the domain of cybersecurity, the synergy between machine learning models and Shapley values can assist in gaining a better understanding of cyber attacks at multiple levels, analyzing features from physical, access, network, and application layers [4], [6], [31]. Indeed, differently from other classification problems, the presence of malicious cyber threats reflects a semantic intention of the attacker to harm the system functionalities. The incorporation of ensemble methods such as boosting and bagging, coupled with the use of Shapley values, has yielded results in mitigating misclassifications in malware detection [25]. Beyond the realm of bolstering accuracy in threat detection, these methodologies have contributed to the elucidation of the pivotal features that exert the most pronounced influence on model predictions [23].

The adoption of such multifaceted approaches underscores the burgeoning importance of explainable artificial intelligence (AI) in sensitive domains, where the comprehension of the underlying rationale behind predictions holds tantamount significance to the predictions themselves. These methods, supported by forest classifiers, Shapley values, and the SHAP framework, constitute an arsenal for advancing the frontiers of interpretable AI in the landscape of cybersecurity and related fields.

III. Theoretical Model

To analyze feature importance in malware classification, we present a cooperative game wherein to define and calculate the Shapley value [19].

A. Cooperative Game

We define a cooperative game where players correspond to model features, and the payoff function mirrors predictions [22]. We individuate a finite player set as $\mathcal{N} = \{1, \dots, n\}$, with each player representing a model feature. We introduce coalitions $\mathcal{S}$, defined as non-empty subsets of $\mathcal{N}$ and define their magnitude $c = |S|$. Note that the union of all coalitions forms the entire set of players $\mathcal{N}$. For each coalition $\mathcal{S}$, a set $v(\mathcal{S}) \subset \mathbb{R}^c$ is specified, containing c-dimensional payoff vectors, serving as the characteristic function. The pair $(\mathcal{N}, v)$ constitutes a cooperative game [32].

The characteristic function, $v(\mathcal{S})$, represents the coalition's gain. It adheres to $v(\emptyset) = 0$, signifying that an empty coalition yields no payoff. When a non-empty coalition, $\mathcal{S} \neq \emptyset$, forms with player $j \in \mathcal{S}$, we examine whether player j can rightfully claim a share of $v(\mathcal{S})$ due to enhancing the coalition's payoff, quantified using the Shapley value [11], whenever

$$v(\mathcal{S} \setminus \{j\}) < v(\mathcal{S}) \ . \tag{1}$$

B. Shapley Values

Shapley value, originally a concept from game theory, has found application in ML as part of the SHapley Additive exPlanations (SHAP) framework [12]. In this context, it is employed to interpret model predictions. The analogy drawn here is that the game is replaced by the model itself, and the players in the game are the features of the model. The primary objective of SHAP is to provide explanations regarding the contribution of each individual feature towards the model's prediction.

This study leverages the SHAP framework to elucidate the functioning of a tree-based model within the domain of cybersystems. SHAP, being a local feature attribution method, attributes contribution scores to each feature based on a single input sample or trial input data, typically represented as x. In simpler terms, SHAP aims to explain a prediction, denoted as $f(x)$, concerning a specific input vector x.

To connect the concept of the characteristic function in cooperative games to the model prediction, $v(\mathcal{S})$ is considered as analogous to $f(x)$. In cooperative games, $v(\mathcal{S})$ represents the payoff of a coalition, while in the SHAP framework within ML, $f(x)$ assesses the predictive value of a combination of features in an input sample x, translating the value assessment from game theory to feature importance in ML models.

To determine the Shapley value for a feature j, all possible combinations of the n features, excluding j, are considered. The model f is evaluated both with and without feature j (i.e., $f(\mathcal{S} \cup \{j\})$ and $f(\mathcal{S})$), and the difference in predictions for x quantifies feature j's marginal contribution. The input data x comprises $c+1$ features in $f(\mathcal{S} \cup \{j\})$ and $\mathcal{S}$ features in $f(\mathcal{S})$. This process is repeated for every possible combination of features.

The Shapley value for feature j is calculated as the average of these marginal contributions over all permutations, using the formula

$$\phi_j(f) = \sum_{\mathcal{S} \subseteq \mathcal{N} \setminus \{j\}} \Bigg[\frac{c!(n-c-1)!}{n!} \cdot \Big(f(\mathcal{S} \cup \{j\})(x_{\mathcal{S} \cup \{j\}}) - f(\mathcal{S})(x_{\mathcal{S}}) \Big) \Bigg]. \tag{2}$$

Here, $\mathcal{N}$ is the total feature set, $\mathcal{S}$ a subset excluding j, n the total number of features, and c the size of $\mathcal{S}$. The formula computes the average marginal contribution of j in all combinations of characteristics, considering differences in the predictions of the model with and without j. The factor preceding the brackets accounts for the combinatorial possibilities of forming $\mathcal{S}$ [33].

IV. Machine Learning Pipeline

We used a specific Kaggle dataset [15], "Network Traffic Android Malware." The dataset consists of features from both the application and network layers of Android systems. It emphasizes the use of data from the Android Genome Project and Drebin, as well as samples from AndroZoo, VirusShare, and DroidCollector [8].

A. Dataset Overview

The dataset presented in the Table I consists of the following features:

- **tcp pks (TCP Packets)**: Number of TCP packets transmitted in a session.
- **port tcp (Distinct TCP Ports)**: Count of distinct TCP port numbers in communication.
- **external ips**: Number of external IP addresses contacted.
- **volume bytes**: Total data volume transmitted, in bytes.
- **udp pks (UDP Packets)**: Number of UDP packets transmitted.
- **tcp urg pk (TCP URG Packets)**: Count of TCP packets with URG flag set.
- **source pks (Source App Packets)**: Packets sent from the source application.
- **remote pks (Remote App Packets)**: Packets received by the remote application.
- **source bytes**: Data volume sent from the source application in bytes.
- **remote bytes**: Data volume received by the remote application in bytes.
- **source pks 1 (Source App Packets - Alternate Count)**: Alternate count of packets from the source application.
- **dns query**: Number of DNS queries made during the session.
- **type**: Session classified as 'malicious' or 'benign', indicating traffic nature.

In summary, the dataset, as shown in Table I, encompasses a range of pertinent features essential for the analysis. These features include key network and communication metrics, which collectively constitute the foundation upon which analytical investigations and subsequent modeling are constructed.

B. Data Preprocessing

The preparation of the dataset for the analysis encompassed a series of preprocessing steps designed to ensure data quality and suitability for subsequent analytical procedures. These preprocessing steps were diligently executed to maintain the integrity and relevance of the dataset.

As a first step, any columns containing missing values, represented as NaN, were systematically eliminated from the dataset. This is done to mitigate the problem of incomplete or unreliable data, even though we remark that possible extensions of this approach include data completion approaches [34].

Furthermore, columns characterized by the presence of a singular variable, thus offering no meaningful variance or discriminatory power, were judiciously omitted from the dataset. This curation process aimed to streamline the dataset by retaining only those features that held substantive value for subsequent analyses.

Subsequently, the data underwent a normalization process employing the `MinMaxScaler` from `sklearn` module. This normalization procedure standardized the feature values, ensuring that all variables resided within the bounded interval of [0, 1]. This normalization avoids issues related to varying scales among features and facilitates the meaningful comparison of features within the ensuing analytical framework.

TABLE I
DATASET SAMPLE

tcp pks	port tcp	external ips	volume bytes	udp pks	tcp urg pk	source pks	remote pks	source bytes	remote bytes	source pks 1	dns query	type
12	12	4	888	0	0	14	2	395	1046	14	2	malicious
0	0	0	0	0	0	2	2	545	155	2	2	malicious
19	0	1	1993	0	0	21	18	5159	2155	21	2	benign
318	0	5	21709	0	0	324	336	458241	22154	324	6	benign
6	0	1	1308	0	0	7	7	1947	1383	7	1	benign

TABLE II
DATASET NORMALIZED

tcp pks	port tcp	external ips	volume bytes	udp pks	tcp urg pk	source pks	remote pks	source bytes	remote bytes	source pks 1	dns query	type
0.000323	0.005538	0.093023	0.000210	0.0	0.0	0.000350	0.000044	0.000006	0.000231	0.000350	0.002191	1
0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.000027	0.000044	0.000008	0.000020	0.000027	0.002191	1
0.000512	0.000000	0.023256	0.000472	0.0	0.0	0.000538	0.000392	0.000076	0.000493	0.000538	0.002191	0
0.008562	0.000000	0.116279	0.005136	0.0	0.0	0.008695	0.007316	0.006716	0.005224	0.008695	0.006572	0
0.000162	0.000000	0.023256	0.000309	0.0	0.0	0.000162	0.000152	0.000029	0.000311	0.000162	0.001095	0

In the context of supervised learning, labels were assigned to the data points, with a classification schema wherein ’1’ denoted malicious instances and ’0’ signified benign instances. This labeling procedure is preliminary to the subsequent training and evaluation of ML models.

Finally, the dataset was partitioned into distinct training and test sets, observing an 80-20% ratio, respectively. This partitioning strategy was instrumental in enabling model training on a substantial portion of the data while reserving an independent subset for rigorous model evaluation. Such a segregation of data facilitates the assessment of model performance, ensuring that the model's predictive capabilities are rigorously scrutinized against unseen data instances.

In summary, these preprocessing steps collectively show the robustness and reliability of the dataset, setting the stage for rigorous analytical investigations and the subsequent development and evaluation of ML models. The final dataset is presented in Table II.

C. *Classifier*

In this paper we employ a Random Forest Classifier from the `sklearn` library for cybersecurity threat detection. We configure the classifier with 20 trees (`n_estimators=20`) and a maximum depth of 50 (`max_depth=50`). We fixed this values after an extensive grid-search in the hyperparameters space to balance comprehensiveness and overfitting prevention. We further fixed a random state (`random_state=45`) to ensure result reproducibility.

After training the model on the training set util convergence, the model's predictive performance was assessed on a separate test set. The classifier achieved a noteworthy accuracy of 89.42%, underscoring its efficiency in accurately classifying cybersecurity threats.

To interpret the RandomForestClassifier's predictions, SHAP analysis was conducted. Feature importance was initially evaluated, visualized in a bar chart to depict each feature's relative importance. SHAP values for the test set were computed, offering insights into individual predictions. This analysis was extended with `shap.summary_plot`, providing an overview of feature contributions. The most significant features were further explored using `shap.dependence_plot`, illustrating the relationship between feature values and the model's output.

V. RESULTS AND DISCUSSION

The analysis of SHAP values in this study reveals the critical importance of specific features such as “Source App Bytes” and “Remote App Packet” in the context of malware detection. These features, prominently presented in Fig. 1 and Fig. 2, demonstrate a significant impact on identifying malware, aligning with typical malware behaviors involving anomalous data transmission and reception patterns.

“Source App Bytes”, indicating the volume of outgoing data from an application, emerges as a key indicator in distinguishing between benign and malicious software. This observation is aligned with common characteristics of malware, which often engage in atypical data transmission behaviors. Similarly, “Remote App Packets”, signifying incoming data to an application, stands out as another significant predictor. This highlights the criticality of monitoring data traffic in both directions for effective malware detection.

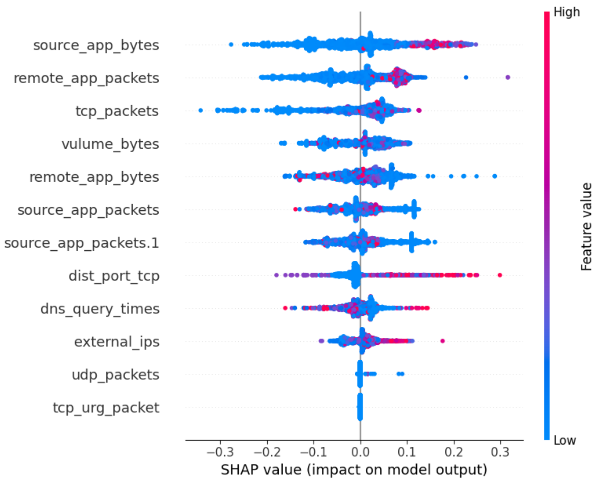

Fig. 1. SHAP values. Impact on model output.

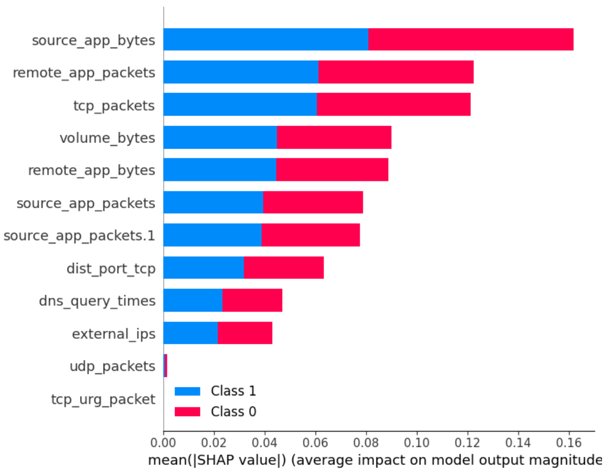

Fig. 2. Average impact on model output.

The application of Random Forest classifiers enables to successfully interpret complex data patterns, as they consistently identify “Source App Bytes” and “Remote App Packets” as significant features in various models, emphasizing their recurring role in differentiating between benign and malicious software.

An interesting aspect of this study is the comparison between SHAP values summary plot in Fig. 2 and global feature importance plot in Fig. 3. The two methodologies diverge in their approach to evaluating feature significance. Global feature importance may not fully capture the complex interplay between features, whereas SHAP values provide a more detailed and nuanced understanding of each feature’s influence on the model’s predictions, both individually and in combination with other features. This distinction often leads to different interpretations of feature importance between these two methods.

To sum up, the analysis enhances the understanding of key features in malware detection, highlighting the importance of features that characterize unique communication patterns associated with malicious behavior. As a result, it represents a contribution towards the development of more effective and robust classification models, emphasizing the utility of SHAP values in providing a comprehensive understanding of model predictions for cybersecurity.

VI. Conclusions

We assessed the use of Shapley values as a support to the use of Random Forest classifiers for the detection of Android malware [8]. This research can provide insights into the influence of individual features on the model’s predictions and contribute to the broader understanding of feature importance in complex datasets.

Key findings include the important roles of features pertaining packet number and sizes as primary indicators in malware detection, implying the relevance of data transmission patterns in identifying malicious activities. The application of SHAP values refined the model’s interpretability, ensuring a comprehensive understanding of how each feature impacts the classification process [14], [21].

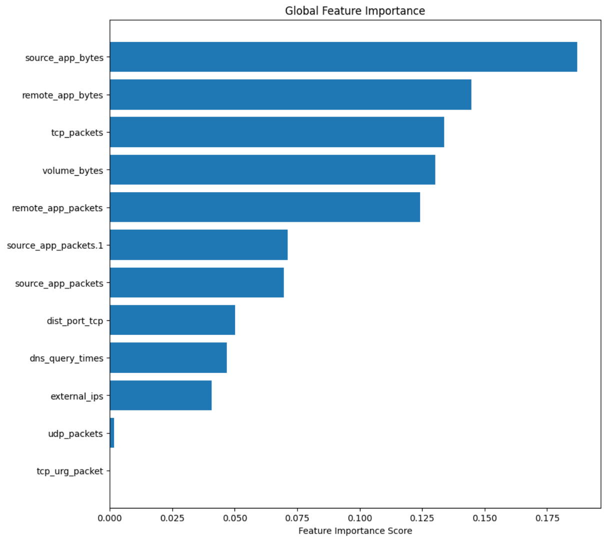

Fig. 3. Global feature importance for the dataset.

Moving forward, this research can open further avenues for the exploration in the field of explainable AI, particularly in cybersecurity [26]. Future work could involve expanding the dataset, experimenting with different ML models, and refining the SHAP analysis technique to improve the accuracy and interpretability of the model. The ultimate goal remains to develop robust, transparent, and efficient tools for cybersecurity threat detection, making significant steps forward in safeguarding digital infrastructures against malware threats [5].

References

[1] P. Saraswat, "An inclusive analysis of Google's Android operating system and its security," in *AIP Conference Proceedings*, vol. 2427, no. 1, 2023.

[2] D. Vecchiato, M. Vieira, and E. Martins, "The perils of Android security configuration," *Computer*, vol. 49, no. 06, pp. 15–21, Jun. 2016.

[3] A. R. Khunt and P. Prabu, "An empirical analysis of Android permission system based on user activities." *J. Comput. Sci.*, vol. 14, no. 3, pp. 324–333, 2018.

[4] A. Kazlouski, T. Marchioro, H. Manifavas, and E. P. Markatos, "I still see you! inferring fitness data from encrypted traffic of wearables," in *Proc. BIOSTEC*, 2021, pp. 369–376.

[5] K. Shaukat, S. Luo, V. Varadharajan, I. A. Hameed, and M. Xu, "A survey on machine learning techniques for cyber security in the last decade," *IEEE Access*, vol. 8, pp. 222 310–222 354, Dec. 2020.

[6] V. Bonagura, S. Panzieri, F. Pascucci, and L. Badia, "A game of age of incorrect information against an adversary injecting false data," in *Proc. IEEE CSR*, 2023, pp. 347–352.

[7] B. Sabir, F. Ullah, M. A. Babar, and R. Gaire, "Machine learning for detecting data exfiltration: A review," *ACM Comput. Surv. (CSUR)*, vol. 54, no. 3, pp. 1–47, 2021.

[8] C. C. Urcuqui López, J. S. Delgado Villarreal, A. F. Perez Belalcazar, A. Navarro Cadavid, and J. G. Diaz Cely, "Features to detect Android malware," in *Proc. IEEE COLCOM*, 2018.

[9] R. Hazra, M. Banerjee, and L. Badia, "Machine learning for breast cancer classification with ANN and decision tree," in *Proc. IEEE IEMCON*, 2020, pp. 0522–0527.

[10] D. Fryer, I. Strümke, and H. Nguyen, "Shapley values for feature selection: The good, the bad, and the axioms," *IEEE Access*, vol. 9, pp. 144 352–144 360, Oct. 2021.

[11] D. Scapin, G. Cisotto, E. Gindullina, and L. Badia, "Shapley value as an aid to biomedical machine learning: a heart disease dataset analysis," in *Proc. IEEE CCgrid*, 2022, pp. 933–939.

[12] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Proc. NIPS*, vol. 30, 2017.

[13] G. Quer, F. Librino, L. Canzian, L. Badia, and M. Zorzi, "Inter-network cooperation exploiting game theory and Bayesian networks," *IEEE Trans. Commun.*, vol. 61, no. 10, pp. 4310–4321, Oct. 2013.

[14] A. Ghorbani and J. Zou, "Data Shapley: Equitable valuation of data for machine learning," in *Proc. ICML*, 2019, pp. 2242–2251.

[15] "Network traffic Android malware," accessed Jan. 23, 2024. [Online]. Available: https://www.kaggle.com/datasets/xwolf12/network-traffic-android-malware

[16] "Android malware genome project," accessed Jan 21, 2024. [Online]. Available: http://www.malgenomeproject.org/

[17] D. Arp, M. Spreitzenbarth, M. Hübner, H. Gascon, and K. Rieck, "DREBIN: Effective and explainable detection of Android malware in your pocket." in *Proc. NDSS*, vol. 14, 2014, pp. 23–26.

[18] S. Kumar, B. P. Singh, and V. Kumar, "A semantic machine learning algorithm for cyber threat detection and monitoring security," in *Proc. IEEE ICAC3N*, 2021, pp. 1963–1967.

[19] L. S. Shapley, "A value for n-person games," *Contrib. Th. Games II, Ann. Math. Stud.*, vol. 28, 1953.

[20] A. V. Guglielmi and L. Badia, "Analysis of strategic security through game theory for mobile social networks," in *Proc. IEEE CAMAD*, 2017.

[21] R. Alenezi and S. A. Ludwig, "Explainability of cybersecurity threats data using SHAP," in *Proc. IEEE SSCI*, 2021.

[22] L. Merrick and A. Taly, "The explanation game: Explaining machine learning models using Shapley values," in *Proc. CD-MAKE*, 2020, pp. 17–38.

[23] R. Kumar and G. Subbiah, "Zero-day malware detection and effective malware analysis using Shapley ensemble boosting and bagging approach," *Sensors*, vol. 22, no. 7, p. 2798, 2022.

[24] A. Buratto, B. Yivli, and L. Badia, "Machine learning misclassification within status update optimization," *Proc. IEEE COMNETSAT*, 2023.

[25] M. A. Setitra, M. Fan, B. L. Y. Agbley, and Z. E. A. Bensalem, "Optimized MLP-CNN model to enhance detecting DDoS attacks in SDN environment," *Network*, vol. 3, no. 4, pp. 538–562, 2023.

[26] T. Marchioro, L. Giaretta, E. Markatos, and S. Girdzijauskas, "Federated naive Bayes under differential privacy," in *Proc. SECRYPT*, 2022, pp. 170–180.

[27] S. Soderi, D. Masti, and Y. Zacchia Lun, "Railway cyber-security in the era of interconnected systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 7, pp. 6764–6779, Jul. 2023.

[28] G. Cisotto, A. V. Guglielmi, L. Badia, and A. Zanella, "Classification of grasping tasks based on EEG-EMG coherence," in *Proc. IEEE Healthcom*, 2018.

[29] G. Gnecco, Y. Hadas, and M. Sanguineti, "Public transport transfers assessment via transferable utility games and Shapley value approximation," *Transportmetrica A: Transp. Sc.*, vol. 17, no. 4, pp. 540–565, 2021.

[30] M. Wang, K. Zheng, Y. Yang, and X. Wang, "An explainable machine learning framework for intrusion detection systems," *IEEE Access*, vol. 8, pp. 73 127–73 141, Apr. 2020.

[31] M. Scalabrin, V. Vadori, A. V. Guglielmi, and L. Badia, "A zero-sum jamming game with incomplete position information in wireless scenarios," in *Proc. European Wireless Conf.*, 2015.

[32] F. Shams and M. Luise, "Basics of coalitional games with applications to communications and networking," *EURASIP J. Wirel. Commun. Netw.*, vol. 2013, no. 1, pp. 1–20, 2013.

[33] H. Alsuradi, W. Park, and M. Eid, "Explainable classification of EEG data for an active touch task using Shapley values," in *Proc. HCII*, 2020, pp. 406–416.

[34] F. Biancalani, G. Gnecco, R. Metulini, and M. Riccaboni, "Prediction of annual CO_2 emissions at the country and sector levels, based on a matrix completion optimization problem," *Optimiz. Lett.*, 2023.