IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 2 June 2023, accepted 9 June 2023, date of publication 14 June 2023, date of current version 21 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3286313

RESEARCH ARTICLE

SHAP Interpretations of Tree and Neural Network DNS Classifiers for Analyzing DGA Family Characteristics

NIKOS KOSTOPOULOS[1], DIMITRIS KALOGERAS[2], DIMITRIS PANTAZATOS[1], MARIA GRAMMATIKOU[1], AND VASILIS MAGLARIS[1]

[1]School of Electrical and Computer Engineering, National Technical University of Athens, 15780 Athens, Greece
[2]Institute of Communication and Computer Systems, National Technical University of Athens, 15780 Athens, Greece

Corresponding author: Nikos Kostopoulos (nkostopoulos@netmode.ntua.gr)

This work was supported in part by the European Commission Horizon 2020 GN4-3 project under Grant 856726, in part by the Special Account for Research Funding of the National Technical University of Athens, in part by the FlExible assembLy manufacturIng with human-robot Collaboration and digital twin modEls (FELICE) under Grant 101017151, and in part by the Security Protection Tools for Networked Medical Devices (SEPTON) H2020 Projects under Grant 101094901.

ABSTRACT Domain Generation Algorithms (DGA's) have been employed by botnet orchestrators for controlling infected hosts (bots), while evading detection by performing multiple DNS requests, mostly for non-existing domain names. With blacklists ineffective, modern DGA filtering methods rely on Machine Learning (ML). Emerging needs for higher intrusion detection accuracy lead to complex, non-interpretable black-box classifiers, thus requiring eXplainable Artificial Intelligence (XAI) techniques. In this paper, we utilize SHapley Additive exPlanation (SHAP) to derive model-agnostic, post-hoc interpretations on DGA name classifiers. This method is applied to binary supervised tree-based classifiers (e.g. eXtreme Gradient Boosting - XGBoost) and deep neural networks (Multi-Layer Perceptron - MLP) to assess domain name feature importance. SHAP visualization tools (summary, dependence, force plots) are used to rank features, investigate their effect on model decisions and determine their interactions. Specific interpretations are detailed for identifying names belonging to common DGA families pertaining to arithmetic, wordlist, hash and permutation based schemes. Learning and interpretations are based on up-to-date datasets, such as Tranco for benign and DGArchive for malicious names. Domain name features are extracted from dataset instances, thus limiting time-consuming and privacy-invasive database operations on historical data. Our experimental results demonstrate that SHAP enables explanations of XGBoost (the most accurate tree-based model) and MLP classifiers and indicates the characteristics of specific DGA schemes, commonly employed in attacks. In conclusion, we envision that XAI methods will expedite ML deployment in networking environments where justifications for black-box models are required.

INDEX TERMS Cybersecurity, domain generation algorithms (DGA's), domain name system (DNS), explainable artificial intelligence (XAI), machine learning, shapley additive explanation (SHAP).

I. INTRODUCTION

Machine Learning (ML) algorithms have been widely employed within the cybersecurity domain for effectively filtering massive amounts of data and classifying malignant traffic. Such algorithms have been commonly used in the field of botnet traffic detection and for classifying names originating from Domain Generation Algorithms (DGA's) [1]. Tree-based ML classifiers and deep neural networks are utilized to differentiate between legitimate and malicious Domain Name System (DNS) names with promising accuracy results.

Development of DGA name classifiers has been motivated by the desire for ML models of higher performance.

The associate editor coordinating the review of this manuscript and approving it for publication was Sotirios Goudos.

Therefore, simple and intrinsically explainable ML classifiers have been replaced by complex, black-box models that are not interpretable. Thus, developers are incapable of understanding their models to debug them and assert their intended operation, while users cannot receive justifications on model decisions made on their data. Finally, regulators are unable to ensure that models deployed within critical infrastructures comply with General Data Protection Regulation (GDPR) [2] or equivalent legislations.

The aforementioned limitations led to investigations for eXplainable Artificial Intelligence (XAI) techniques [3] to provide interpretations (and possibly explanations) on ML model operation. As mentioned in [4], post-hoc and model-agnostic XAI algorithms are typically preferred. Post-hoc algorithms are applied to ML models after learning is completed; model-agnostic ones are independent of the selected ML models, e.g. tree classifiers and neural networks. Explanations may be (i) *global* detailing model behavior on entire sets of sample points and (ii) *local* reporting how models make classification decisions for specific inputs. A promising post-hoc and model-agnostic approach is SHapley Additive exPlanation (SHAP) [5], [6], which is capable of *global* and *local* explainability.

Our work leverages on XAI to analyze the operation of binary, supervised DGA name classifiers that distinguish between legitimate and malicious[1] names, thus detecting botnet traffic abusing DNS. We train and evaluate various tree-based classifiers (Random Forests - RF's, Gradient Boosting - GB, eXtreme Gradient Boosting - XGBoost, Adaptive Boosting - AdaBoost, Extremely Randomized Trees - ExtraTrees) and a deep neural network (Multi-Layer Perceptron - MLP). SHAP is subsequently employed to determine and compare the classification criteria of XGBoost [7], which was the most accurate tree model, and MLP deep neural network [8] in a post-hoc and model-agnostic manner. Our experimental analysis focuses on *global* and *local* model interpretations used to rank the impact of utilized features and indicate how their individual values contribute to classification decisions. Relying on multiple SHAP visualization tools (i.e. summary, dependence and force plots [3], [6]) we investigate how the developed models (i) differentiate between benign and malicious domain names and (ii) identify which features have the most significant contribution in classifications of names originating from well-known fundamental DGA generation schemes that produce malicious names [1]. Learning and interpretations are based on linguistic and statistical features, directly extracted from domain names included within up-to-date datasets of benign and malignant DNS names.

Our main contributions are summarized as follows:

- SHAP-based interpretations of DGA name classifiers based on deep neural networks (MLP's) and comparison of their decision-making criteria versus tree-based ML models (XGBoost).
- Identification of dominant features utilized for malicious domain name detection pertaining to specific DGA generation schemes (arithmetic, wordlist, hash and permutation based).
- Extraction of linguistic and statistical features leading to accurate and real-time classification of DGA names with no reliance on time-consuming and privacy sensitive external repository operations.
- Training and interpretations based on the most updated and inclusive dataset of DGA names, i.e. the *DGArchive* repository [1], [9] including 105 DGA families.
- Open-sourced implementation available from our *GitHub* repository [10].

The remainder of this paper is structured as follows: Section II provides brief background and summarizes related work; Section III provides a high-level overview of our methods used for interpreting DGA name classifiers; Section IV elaborates on implementation details pertaining to our approach; Section V includes our experimental results and interpretations of DGA name classifiers based on XGBoost and MLP. Finally, in Section VI we conclude our work and discuss future steps.

II. BACKGROUND AND RELATED WORK

This section provides brief background on concepts used in our paper (subsection II-A), outlines related research approaches (subsection II-B) and details our key contributions (subsection II-C).

A. BACKGROUND

In subsection II-A1 we describe the operation and characteristics of Domain Generation Algorithms (DGA's), whereas in subsection II-A2 we summarize the basics of the SHapley Additive exPlanation (SHAP) method.

1) DOMAIN GENERATION ALGORITHMS (DGA's)

DGA's are a common technique for establishing communication between hacked devices, i.e. bots, and their orchestrators, i.e. Command & Control (C&C) servers. Bots generate DNS requests based on a seeding technique that is known to C&C servers. A small number of domain names is registered and bots are expected to request their resolution. These names correspond to valid C&C IP addresses, thus bots are capable of locating them. Specifically, bots perform several DNS requests; although most of these requests involve invalid domain names (i.e. NXDOMAIN responses are returned), a limited number of them is successfully resolved to the C&C IP addresses. This typically large number of queried domain names combined with constant changes to the seed render domain name blacklists ineffective.

Therefore, ML algorithms have been suggested as an alternative solution to blacklisting. They leverage on previous knowledge and generalize to newly observed domain names

[1]Throughout our paper, DNS names are considered malicious if they are produced by DGA's. Non-DGA names, even those related to malignant activities (e.g. malware propagation), are labeled as benign names in the training set.

for differentiating between benign and malignant patterns, thus blocking communication between bots and their C&C servers. Notably, various classification algorithms have been investigated with promising results, including deep neural networks [11], [12], [13], [14], [15], [16] as well as Tree-based Classifiers (e.g. Random Forests - RF's), Support Vector Machines (SVM) and Naive Bayes [17], [18], [19], [20], [21].

The seeding strategy, the number of domain names produced by a bot and their structure are determined by the DGA family. Although there are various families with diverse characteristics, DGA's are grouped into the following four generation schemes [1] based on the technique utilized to produce domain names:

- *Arithmetic-based:* These algorithms generate sequences of random values. DGA names are constructed by concatenating the ASCII representations corresponding to these values or using them to locate characters within lists that constitute the DGA alphabet.
- *Wordlist-based:* DGA names are generated by randomly concatenating dictionary words. Thus, domain name randomness is reduced, rendering malicious name detection more complicated.
- *Hash-based:* Domain names are constructed by hashing alphanumeric strings and returning their hexadecimal representation.
- *Permutation-based:* They generate at random a domain name, which is subsequently permuted several times to produce multiple DGA names.

2) SHAPLEY ADDITIVE EXPLANATION (SHAP)

SHAP is a model-agnostic, post-hoc XAI method related to cooperative game theory. In cooperative games, players collaborate to achieve a pay-off, which is subsequently split based on participant contributions. Accordingly, features are considered as participants that tune a classifier and subsequently SHAP determines feature importance by estimating the effect of specific features on classification decisions when these features are present and absent.

SHAP delivers *global* and *local* explanations on ML model decisions, whereas various visualization tools facilitate interpretations, e.g. summary plots, dependence plots and force plots. Model-agnostic SHAP is typically based on the *KernelExplainer* [22] method; this approximates feature importance via a weighted linear regression model applied to input instances (sample points). SHAP time complexity mainly depends on the dataset size. Enabling execution within reasonable time frames may require clustering and/or subsampling a given dataset. This process extracts the eXplainability Background Instances (XBI's) used for tuning SHAP values and eXplainability Test Instances (XTI's) utilized for generalizing model interpretations.

B. RELATED WORK

Various approaches have been proposed for the detection of DGA names with promising results, e.g. [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [23], [24]. However, the aforementioned approaches emphasize on improving detection accuracy, but they do not deliver *global* and *local* model and feature interpretations.

Interpreting DGA name classifiers has recently attracted significant interest. In [25] neural network classifiers are interpreted based on their weights. A system for result visualization is also presented to facilitate model comprehension. However, interpretations rely on model-specific XAI methods applicable exclusively to deep learning models, whilst the total features are limited for visualization purposes. In [26] multi-class DGA name classifiers are developed based on features directly extracted from domain names and feature importance is assessed using various statistical methods. Nevertheless, [26] is limited to *global* explainability of DGA classifiers, thus neglecting model interpretations on specific DNS names. Moreover, the effect of different DGA schemes on model decisions is not addressed.

In [27], [28], and [29] SHAP and/or equivalent XAI techniques (e.g. Local Interpretable Model-Agnostic Explanation - LIME [30] and Counterfactual Explanations [31]) are employed to provide *global* and *local* interpretations on binary DGA name classifiers. Although the aforementioned approaches deliver promising results, they are limited mainly to tree-based ML classifiers. These approaches focus on interpreting how names are classified as benign or malicious, therefore neglecting how the characteristics of different DGA families affect classification decisions. Furthermore, feature calculation in [27] and [29] requires resource-intensive operations on databases involving historical data, e.g. IP reputation lists, *WHOIS* lookups and Time To Live (TTL) values from DNS responses. These are usually time-consuming and may raise privacy concerns.

C. KEY CONTRIBUTIONS

Our approach relies on SHAP for model-agnostic (regardless of the selected models) and post-hoc (after the learning procedure is completed) validation of DGA name classifier operation. Our models are based on features extracted entirely from given names, hence resource-intensive operations on privacy-sensitive historical DNS data are not required. We compare interpretations derived from tree-based models (i.e. XGBoost) and neural networks (i.e. MLP's) using both *global* and *local* explanations. Notably, we extend related approaches by analyzing how binary classifier feature rankings perform when facing diverse DGA schemes, e.g. following testing methods used in use cases related to radio communications and health systems [32], [33]. Finally, malicious DNS data used for training and interpreting our models are selected from DGArchive; we included 105 DGA families, a significantly higher number compared to [27], [28], and [29].

III. OVERVIEW

This section outlines the design principles of our analysis (subsection III-A) and provides a baseline description of our

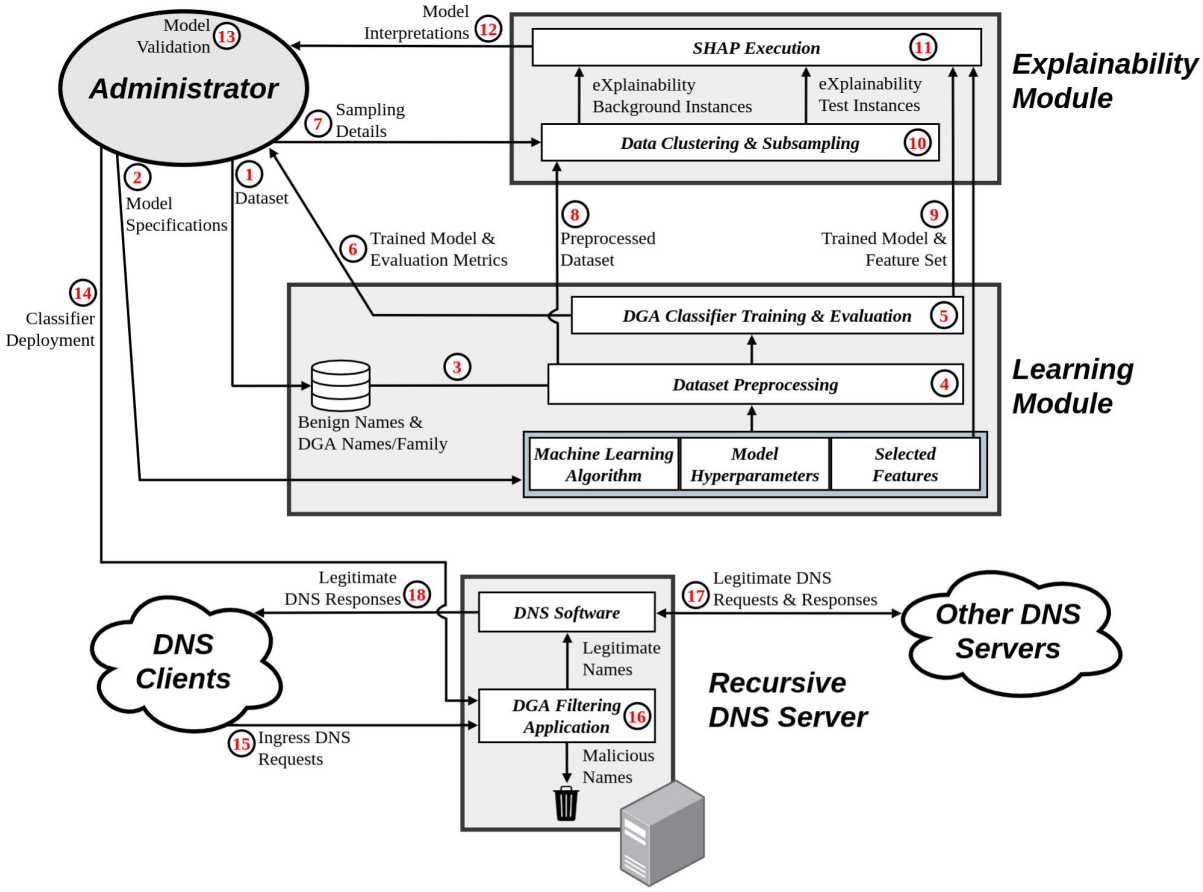


FIGURE 1. Baseline design.

proposed schema for developing and interpreting DGA name classifiers (subsection III-B).

A. DESIGN PRINCIPLES

The main design principles of our approach are:

- *Model-agnostic ML interpretations*: We leverage on the SHAP *KernelExplainer* [22] to interpret our DGA name classifiers independently of the underlying ML model. Therefore, we analyze the operation of tree-based and deep neural network classifiers in a unified manner.
- *Local and global interpretations*: Our approach relies on SHAP to rank feature contributions in classification decisions made on specific input instances for *local* explainability and lists of domain names for *global* explainability.
- *Analysis relying on various SHAP visualization tools*: Multiple SHAP visualization methods (i.e. summary, dependence and force plots) are employed to estimate feature importance, determine how feature values affect model decisions and investigate feature interactions.
- *Classification based on domain-specific features*: ML models are trained on features directly extracted from domain names without requiring costly database operations on historical data that may raise privacy concerns. Such features conceive the statistical and linguistic properties of DNS names, hence they are suitable for real-time DGA name classifications.
- *Explanations for diverse DGA schemes*: We assess the effect of different DGA family properties on feature contributions. This way we infer how the binary DGA name classifiers distinguish between legitimate and malicious DNS names for specific DGA schemes (i.e. arithmetic, wordlist, hash and permutation based).

B. BASELINE DESIGN

Fig. 1 depicts an overview of our approach for DGA traffic detection based on accurate and reliable classifiers. The purpose of the *Administrator* is to train supervised binary classifiers that effectively differentiate between benign and DGA names, validate their dependable operation via XAI techniques (specifically SHAP) and deploy filtering rules to drop botnet traffic.

The architecture of Fig. 1 consists of three components:

- *Learning Module*: Data are preprocessed and the necessary learning parameters are defined to train and evaluate DGA name classifiers.

- *Explainability Module*: SHAP is used to analyze and validate the operation of name classifiers developed by the *Learning Module*.
- *Recursive DNS Server*: Ingress DNS requests are inspected using the trained DGA name classifiers; those involving malicious names are dropped, while legitimate DNS traffic is forwarded for name resolution.

The *Administrator* initially selects the learning dataset that will be utilized for tuning DGA name classifiers (step 1). The selected data consist of benign and malicious (i.e. DGA generated) DNS names labeled for binary classification purposes. Malicious dataset labels include the DGA algorithm used for name construction; such information is typically available from reverse engineering efforts on DGA malware installed within infected hosts [34].

Details of the *Learning Module* operation are subsequently determined (step 2). The *Administrator* defines the model specifications required for tuning name classifiers, i.e. the ML algorithm, the model hyperparameters and the selected features. The learning dataset is then retrieved (step 3) and preprocessed (step 4) based on the selected features and ML model details. The *DGA Classifier* is subsequently trained and evaluated (step 5), while assessment results and tuned model parameters are returned to the *Administrator* (step 6).

Upon completion of the learning phase, the *Administrator* configures the *Explainability Module* by determining the reduced dataset instances required for SHAP execution (step 7). This step refers to the clustering and subsampling processes required for keeping the SHAP running time within feasible time periods. In steps 8 and 9 the *Learning Module* feeds the trained *DGA Classifier*, the selected features and the preprocessed dataset to the *Explainability Module*. This dataset is then clustered and subsampled (step 10) to derive the instances required for SHAP; the eXplainability Background Instances (XBI's) used in SHAP calculations for assessing feature importance and the eXplainability Test Instances (XTI's) consisting of the input sampling points used to eventually derive model interpretations. Note that, in our case XTI's were subsampled from the class of malignant DGA names since our purpose was to assess feature importance per DGA generation scheme.

After SHAP analysis is completed (step 11), the *Explainability Module* provides the *Administrator* with *global* and *local* model-agnostic interpretations of the trained classifiers (step 12). The *Administrator* gathers the *Learning* and *Explainability* module results to validate model operation (step 13). If the classifier accuracy and explanations are satisfactory, the *Administrator* deploys appropriate DGA filtering procedures within the *Recursive DNS Server* (step 14).

In step 15, ingress DNS requests from *DNS Clients* are inspected by the *Recursive DNS Server* (step 16). Malicious DNS requests are dropped, whereas legitimate ones are resolved by the *DNS Software*, e.g. BIND [35], installed within the *Recursive DNS Server* (steps 17 and 18).

IV. IMPLEMENTATION DETAILS

This section elaborates on feature selection (subsection IV-A), on the development and operations of the *Learning Module* (subsection IV-B) and on details pertaining to the *Explainability Module* (subsection IV-C).

A. SELECTED FEATURES

We leverage on feature values that are directly extracted from given domain names and denote linguistic properties (e.g. values denoting the number of vowels) and statistical measures (e.g. entropy values). Such features facilitate real-time DNS traffic inspection and limit sensitive data exchanges by not requiring storage of privacy-sensitive information. As already stated, we do not employ historical data features (e.g. time-based patterns of DNS responses and IP reputation measures), which typically require excessive processing resources and storing them may raise privacy concerns [17].

Prior to feature extraction valid DNS suffixes (one or multiple zone namespaces, e.g. "*.com*" and "*.gov.uk*") are removed from domain names as in [17]. These are not generated by DGA's, hence they are not meaningful to the learning process. Identification of valid DNS suffixes is based on the *Mozilla* public suffix list [36]. Note that removing these suffixes mapped multiple distinct names to common prefixes within the learning dataset, e.g. "google.com" and "google.fr" were both reduced to "google". As a result, classifiers are tuned towards accurately recognizing frequently requested DNS names; their appearance frequency within the dataset reflects specific trends of DNS queries resolved by *Recursive DNS Servers*.

The features used for DGA name classification are outlined in Table 1; feature selection was based on approaches available from the literature, e.g. [14], [17], [37]. In the following, features 44, 47, 48 and 50 are further analyzed:

- *Vowel_Freq (feature 44):* Determines the number of vowels included within the domain name, i.e. letters *a*, *e*, *i*, *o*, *u* and *y*; considering *y* as a vowel typically increases classification accuracy as reported in [20].
- *Reputation (feature 47):* Evaluates domain name *Reputation* defined as an indication of its legitimacy [38]; the higher the *Reputation* the more legitimate the name may appear. A method for measuring the reputation score of a domain name is the appearance frequency of N-grams (i.e. sequences of N consecutive characters) present in benign names and absent in malignant ones [39]. Estimating *Reputation* requires a preprocessing stage whereby a *whitelist* is constructed based on the N-grams derived from a set of legitimate DNS names (e.g. the *Tranco* list [40]). *Reputation* of a given domain name is evaluated by determining how many of its N-grams are included in the aforementioned *whitelist*. N values are selected between 3 and 7 characters as in [39]; unigrams (i.e. N = 1) and bigrams (i.e. N = 2) are excluded because most of them exist in both legitimate and malicious

TABLE 1. Selected Features for DGA name classification.

Sequence Number	Feature Name(s)	Description
1	Length	Length of the domain name
2	Max_DeciDig_Seq	Length of maximum decimal digit sequence
3	Max_Let_Seq	Length of maximum letter sequence
4 - 29	Freq_A, Freq_B, ..., Freq_Z	Frequency of letters A-Z within the domain name
30 - 39	Freq_0, Freq_1, ..., Freq_9	Frequency of digits 0-9 within the domain name
40	Spec_Char_Freq	Number of special characters (hyphens, dots) within the domain name
41	Ratio_Spec_Char	Fractional division of Spec_Char_Freq and Length
42	DeciDig_Freq	Number of decimal digits (0-9) within the domain name
43	Ratio_DeciDig	Fractional division of DeciDig_Freq and Length
44	Vowel_Freq	Number of vowels within the domain name
45	Vowel_Ratio	Fractional division of Vowel_Freq and Length
46	Max_Gap	Length of the longest domain name label
47	Reputation	Number of whitelisted N-grams (N = 3, ..., 7)
48	Words_Freq	Number of concatenated meaningful words within the domain name
49	Words_Mean	Average length of concatenated meaningful words obtained from feature 48
50	Entropy	Shannon Entropy of the domain name

names, thus affecting the learning process and hindering feature importance.

- *Words_Freq (feature 48):* Determines the number of meaningful words within given names. Words are extracted using the *Wordninja* Natural Language Processing (NLP) tool [41] similarly to [42]. *Wordninja* probabilistically splits strings into concatenated words based on the unigram frequency of words appearing within the English *Wikipedia*. As in [43], words shorter than 3 characters (e.g. pronouns and articles) are ignored as their effect to the learning process is not significant.
- *Entropy (feature 50):* Estimates domain name randomness using Shannon Entropy [17]. We used the standard definition of entropy:

$$H(X) = \sum_{x \in X} p(x) log_2 p(x) \tag{1}$$

where X is the set of characters included within a DNS name and $p(x)$ the frequency of character $x \in X$.

B. LEARNING MODULE

This module trains and evaluates supervised binary classifiers that differentiate between legitimate and DGA names. The labeled dataset comprised of benign and malicious names is retrieved and the *Learning Module* proceeds with dataset preprocessing by performing feature extraction. Pairwise feature correlations are calculated using the Pearson's Correlation Coefficient (PCC) statistical measure [44] to detect redundant features not contributing significantly to the learning process. Upon detecting pairs with PCC's exceeding a predefined threshold, a feature is randomly selected and evicted from the dataset, eventually accelerating the learning process without significant performance degradation.

The resulting dataset is randomly split into the training set (used for tuning the binary classifier) and the testing set (used for evaluating model generalization). Training and testing instances are scaled between 0 and 1 using Min-max normalization based on minimum and maximum values of training instances as in [45]. The *Learning Module* completes dataset preprocessing by balancing the number of benign and malicious class instances. Training set instances are oversampled using the Synthetic Minority Over-sampling Technique (SMOTE) [46], similarly to [45]. SMOTE synthetically generates instances following training set statistical properties to reduce imbalance between given classes.

Finally, the *Learning Module* trains and evaluates DGA name classifiers. We trained tree-based classifiers (i.e. Random Forest - RF, Gradient Boosting - GB, eXtreme Gradient Boosting - XGBoost, Adaptive Boosting - AdaBoost, Extremely Randomized Trees - ExtraTrees) and a deep neural network (i.e. Multi-Layer Perceptron - MLP). Tree classifiers were developed using *scikit-learn* [47] and XGBoost Python Package [48], whereas MLP's with *Keras* [49]. Model hyperparameters were fine-tuned using Grid Search, which exhaustively explores a subset of the ML algorithm hyperparameter space and selects the best performing classifier [50].

C. EXPLAINABILITY MODULE

This module analyzes the operation of DGA name classifiers using SHAP, eventually delivering *global* and *local* model-agnostic post-hoc interpretations to the *Administrator*.

The preprocessed dataset, the trained model and the selected features are initially retrieved from the *Learning Module*. The preprocessed dataset is then clustered and subsampled to limit SHAP analysis within reasonable time constraints [4]. The eXplainability Background Instances (XBI's) are obtained as the centroids of *K-means* clustering on the training set, whereas eXplainability Test Instances (XTI's) are derived by randomly subsampling the testing set. XBI's are used to tune SHAP values and XTI's to interpret decisions made by the DGA name classifiers.

Subsequently, SHAP *KernelExplainer* [22] is used to derive *global* and *local* interpretations by ranking features according to their contribution in classification decisions and determining interactions between them. SHAP offers various visualization tools to facilitate comprehension of interpretations [4], [6]. We relied on the following SHAP plots:

- *Summary plots*: Features are ranked in descending order according to their impact on model decisions. XTI's are mapped as instance dots based on their positive or negative contributions to model classifications, i.e. their SHAP values depicted in the horizontal dimension. Low and high values of features are additionally mapped on summary plots to depict their effect on classifier operation. SHAP relies on a color palette to distinguish feature values; extreme values are visualized using a pair of basic colors (e.g. blue and red), whereas basic color shades denote their intermediary values.
- *Dependence plots*: They demonstrate contributions of specific features on model decisions. XTI's are mapped as dots on a two-dimensional plot; the horizontal axis includes all possible values of an investigated feature, whereas the vertical axis depicts the corresponding SHAP values, i.e. their impact on model decisions. Dependence plots also visualize the correlation between the investigated feature and an additional one that mostly interacts with it. This interacting feature is determined by evaluating the joint effect of all possible feature pairs, therefore estimating their influence on classification accuracy using the Shapley interaction values [3], [51]. Low and high values of the interacting feature are depicted using the aforementioned color palette, thus facilitating conclusions of how feature interactions jointly affect classification decisions.
- *Force plots*: They demonstrate feature contributions on specific XTI's (typically single *local* instances). A pair of basic colors is used to discern model features according to whether they contribute positively (e.g. red) or negatively (e.g. blue) to classification decisions. Names and values of features mostly contributing to model decisions are included in the plot, whereas less important feature names and values are omitted. A decimal number (denoted with bold characters) corresponds to the final result returned by the binary classifier.

V. EVALUATION

This section includes the results of our experimental analysis. Subsection V-A describes the selected dataset and subsection V-B outlines the experimental testbed. Subsection V-C involves the *Learning Module* performance evaluation that assesses the accuracy of binary DGA name classifiers. Finally, subsection V-D includes the SHAP-based interpretations extracted by the *Explainability Module*.

A. DATASETS

ML models were evaluated using malicious and benign domain names, typically used for building DGA name classifiers. Our data were retrieved in Spring 2023.

Malicious DNS names were obtained from *DGArchive* [9], a moderated repository continuously updated with DGA names resulting from reverse engineering efforts on DGA malware code. We retrieved roughly 200 million domain names corresponding to 105 distinct DGA families pertaining to all generation schemes (i.e. arithmetic, wordlist, hash and permutation based). The total repository size and constraints of our experimental infrastructure rendered training of DGA name classifiers time-consuming and memory intensive. Therefore, we sampled *DGArchive* and randomly extracted 10,000 DNS names from each DGA family as in [52]; families involving less than 10,000 names were included without subsampling. Eventually, our dataset consisted of 600,775 DGA names, which were used to train, evaluate and interpret DGA name classifiers.

Legitimate DNS names were selected from *Tranco* [40], a public online service ranking domain names based on their popularity. *Tranco* merges data from various name ranking services, namely *Alexa*, *Cisco Umbrella*, *Majestic* and *Farsight*. Name rankings are calculated over long time periods (e.g. 30 days), thus mitigating the impact of abrupt daily fluctuations and/or list manipulation attempts. However, *Tranco* still contains a small percentage of DGA names that are frequently requested by large numbers of infected Internet devices (bots). Therefore, we filtered the *Tranco* dataset [53] by removing names included within *DGArchive*; these amounted to 0.57% of *Tranco* entries. We subsequently utilized the top-ranked 1 million entries from the remaining *Tranco* names similarly to [28]. Following [39] we used the first 100,000 to construct the *whitelist* pertaining to the *Reputation* feature (subsection IV-A); the remaining 900,000 were used to train and assess the DGA name classifiers.

The aforementioned name sets were labeled as benign and malignant without indicating specific families of malicious DGA names. Binary classifiers were selected instead of multi-class ones. Although multi-class classifiers may provide insight in specific DGA families, they are typically less accurate than binary ones in segregating benign and malignant names [11].

B. TESTBED OVERVIEW

Experiments were performed within our laboratory infrastructure. We utilized a Virtual Machine (VM) comprising of 8 virtual cores and 24GB physical memory. The hypervisor was a Dell PE R730 with Intel Xeon E5-2620 v3 2.4 GHz. Training of neural networks was accelerated using the NVIDIA GeForce GTX 1050 Ti 4GB [54] graphics card.

C. LEARNING MODULE

The *Learning Module* was evaluated by assessing (i) the pairwise correlation among selected features and (ii) the performance of supervised binary DGA name classifiers. Assessments were performed using the dataset of benign and malicious names described in subsection V-A.

Pearson's Correlation Coefficient (PCC) was utilized to detect highly correlated features. PCC's were calculated for all feature pairs and those exceeding 0.9 (by absolute value) were considered strongly correlated [55]. In such feature pairs, a feature was selected at random and evicted from

the dataset. In particular, *Ratio_DeciDig* was determined as strongly correlated to other features, hence it was removed from subsequent experiments.

We selected Random Forests (RF's), Gradient Boosting (GB), eXtreme Gradient Boosting (XGBoost), Adaptive Boosting (AdaBoost) and Extremely Randomized Trees (ExtraTrees) as indicative algorithms of tree-based classifiers; Multi-Layer Perceptrons (MLP) were selected as representative models of deep neural networks. Classifiers were trained and evaluated using the dataset described in subsection V-A. This dataset was randomly split into two parts using the *train_test_split* method of *scikit-learn* [10]; 80% was utilized as the training set and the remaining 20% as the testing set.

Grid Search was used to tune model hyperparameters. The number and maximum depth of RF, GB and XGBoost trees were varied as described in Table 2. The number of AdaBoost and ExtraTrees estimators were varied as described in the table. Similarly, multiple MLP configurations were considered by varying the hidden layers number, the neurons per layer, the batch size and the rate of dropout regularization layers placed between the hidden layers to reduce overfitting. Considered MLP hyperparameters are described in Table 3.

Based on the accuracy of ML models, classifier performance was assessed as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

where True Positives (TP's) are the correctly classified DGA names, True Negatives (TN's) are the correctly categorized benign names, False Positives (FP's) are the incorrectly classified benign names and False Negatives (FN's) are the misclassified malicious names.

Grid Search determined that among RF, GB, XGBoost, AdaBoost, ExtraTrees and MLP classifiers the best accuracy scores on the testing set were 94.67%, 94.66%, 94.81%, 92.32%, 94.67% and 94.51% respectively[2] as shown in Table 4. Their configuration details are summarized in tables 2, 3.

D. EXPLAINABILITY MODULE

The *Explainability Module* was evaluated based on SHAP interpretations derived on the trained models (subsection V-C) for the dataset described in subsection V-A. We investigated (i) the features used to discern benign and malicious names derived from multiple DGA families and, (ii) the most influential features utilized to differentiate specific DGA schemes.

Interpretations were derived for 105 DGA families of the *DGArchive* repository and are available from our *GitHub* repository [10]. However, for illustration purposes representative results are presented in this paper for 4 indicative DGA families pertaining to 4 diverse DGA schemes (see Section II-A1). Specifically, as in [59] results are presented for the following: (i) *DirCrypt* (arithmetic-based), (ii) *Matsnu* (wordlist-based), (iii) *Bamital* (hash-based) and (iv) *Volatile-Cedar* (permutation-based).

Similarly to [4], XBI's were selected as the cluster centroids resulting from *K-means* execution on the training set with K equal to 50. XTI's used for interpreting how name classifiers differentiate between benign and malicious names derived from all DGA families were obtained by randomly subsampling 250 DGA names from the testing set. Interpretations pertaining to specific DGA families were based on XTI's randomly subsampled from testing set entries of these specific families; families with less than 250 names were included without subsampling.

A greater number of XBI's and XTI's yielded in our extensive experiments insignificant interpretation improvements, while SHAP running time increased dramatically [10]. Using the aforementioned parameters, the *Learning Module* and the *Explainability Module* required approximately 2 days to complete their operation.

TABLE 2. Hyperparameter tuning of tree classifiers using grid search.

Hyperparameters	Considered Values	Best Classifier Value
Random Forest - RF		
Number of Trees	10, 20, ..., 200	200
Maximum Tree Depth	10, 20, ..., 200	50
Gradient Boosting - GB		
Number of Trees	10, 20, ..., 100	100
Maximum Tree Depth	10, 20, ..., 50	20
eXtreme Gradient Boosting - XGBoost		
Number of Trees	10, 20, ..., 200	100
Maximum Tree Depth	10, 20, ..., 200	20
Adaptive Boosting - AdaBoost		
Number of Trees	10, 20, ..., 1000	520
Extremely Randomized Trees - ExtraTrees		
Number of Trees	10, 20, ..., 500	260
Other Parameters: Default scikit-learn values		

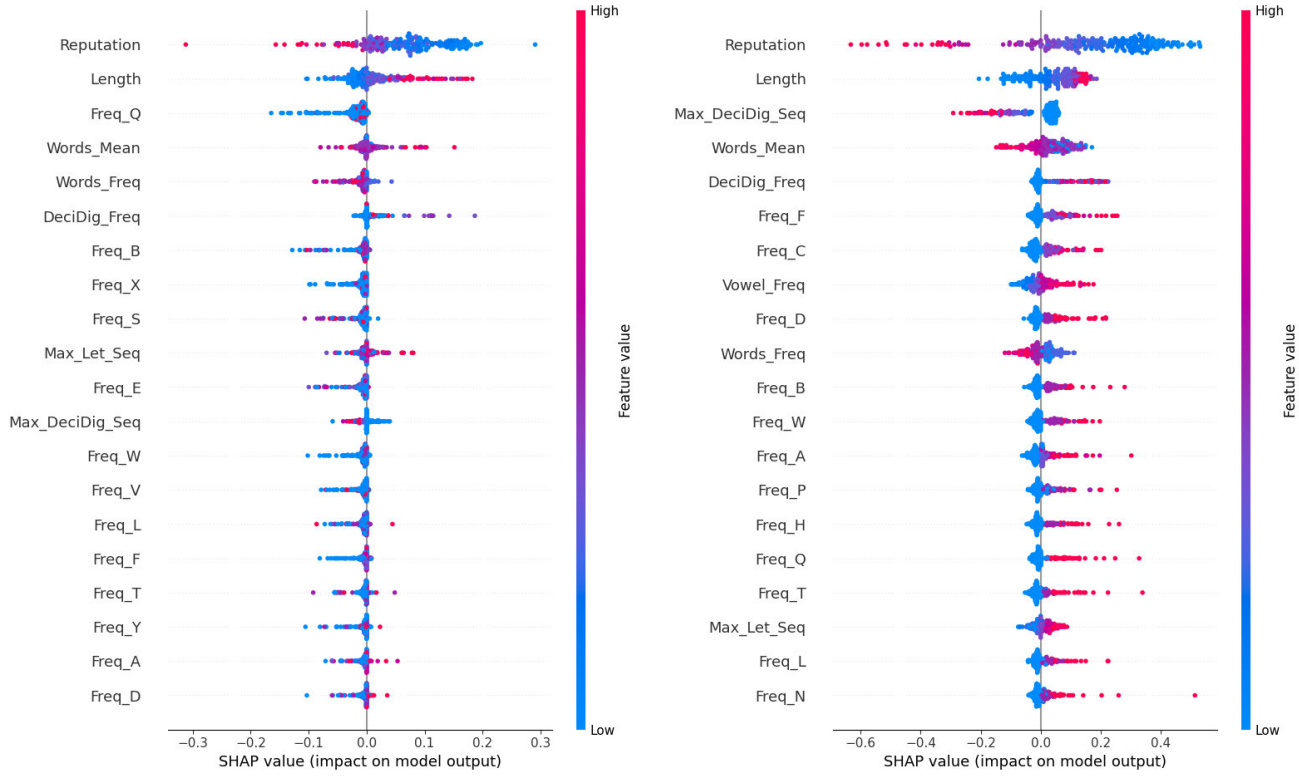
TABLE 3. Hyperparameter tuning of MLP classifier using grid search.

Hyperparameters	Considered Values	Best Classifier Value
Number of Hidden Dense Layers	1, 2, 3	3
Neurons per Hidden Dense Layer	100, 200, 300	Dense Layer 1: 300 Dense Layer 2: 200 Dense Layer 3: 200
Dropout Probability	0.2, 0.5	0.2
Batch Size	256, 512	512
Epochs	100 epochs with EarlyStopping [56]	
Loss Function	BinaryCrossentropy [57]	
Optimizer	Adam [58]	
Activation Functions	Input/Hidden Layers: ReLU Output Layer: Sigmoid	

TABLE 4. Accuracy of best classifiers.

Algorithm	Best Classifier Accuracy
Random Forest - RF	94.67%
Gradient Boosting - GB	94.66%
eXtreme Gradient Boosting - XGBoost	94.81%
Adaptive Boosting - AdaBoost	92.32%
Extremely Randomized Trees - ExtraTrees	94.67%
Multi-Layer Perceptron - MLP	94.51%

[2]Filtering repetitive name prefixes (see subsection IV-A) within the training and testing sets yielded comparable accuracy results, specifically 94.39% for XGBoost (best tree-based classifier) and 94.31% for the MLP neural network. Thus, we did not consider filtering them in our experiments pertaining to the *Explainability Module*.



(a) Algorithm: eXtreme Gradient Boosting (XGBoost)

(b) Algorithm: Multi-Layer Perceptron (MLP)

FIGURE 2. **SHAP summary plots on XTI's including malicious names from all DGA families.**

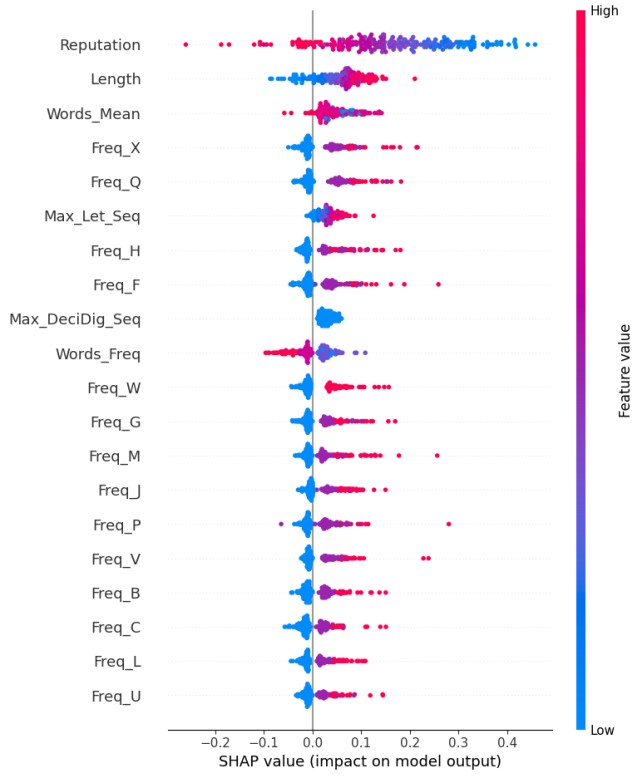
The following subsections present SHAP interpretations for XGBoost (which was the most accurate tree-based model) and the MLP deep neural network model. Interpretations are based on multiple SHAP plots: (i) summary plots pertaining to 250 XTI's from all DGA families (subsection V-D1), (ii) summary plots involving XTI's from selected DGA families (subsection V-D2), (iii) dependence plots pertaining to 250 XTI's from all DGA families (subsection V-D3), (iv) dependence plots including XTI's from specific DGA families (subsection V-D4) and (v) force plots for selected domain names (subsection V-D5). Legitimate and malicious name classes are denoted with numbers *0* and *1* respectively. Thus, negative SHAP values contribute to benign name classifications, whereas positive values to DGA name classifications.

1) XGBOOST AND MLP CLASSIFIER SUMMARY PLOTS FOR ALL DGA FAMILIES

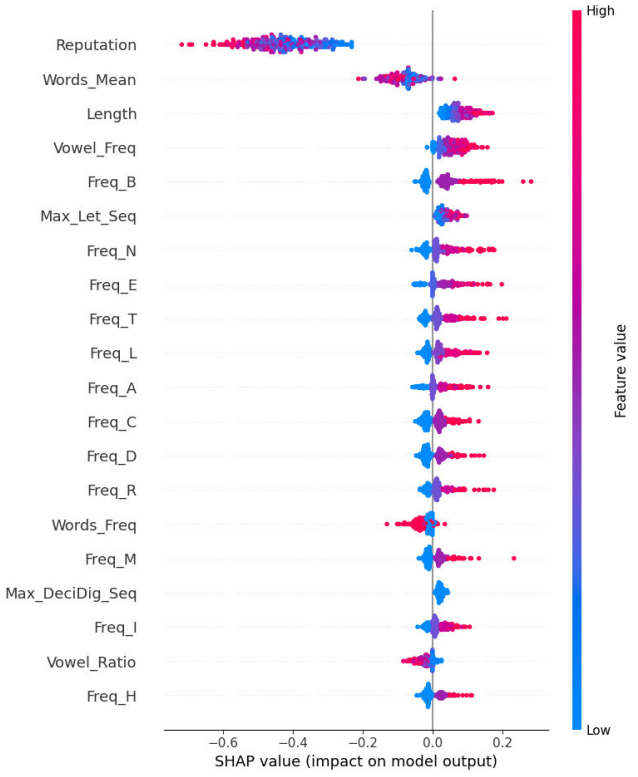
In this subsection SHAP summary plots are used to explain the operation of binary DGA name classifiers. Fig. 2 demonstrates XGBoost (Fig. 2a) and MLP (Fig. 2b) classification criteria for segregating malicious names from benign ones. Analysis was based on 250 XTI's, illustrated as colored dots in the horizontal dimension, from all DGA families. In these summary plots blue color is used to denote low feature values, whereas red color is utilized for high feature values (see subsection IV-C).

Fig. 2a depicts the 20 most influential features used by the XGBoost binary classifier. The most effective features are *Reputation*, *Length*, *Freq_Q*, *Words_Mean*, *Words_Freq* and *DeciDig_Freq* ranked in order of descending importance. High *Length* and *DeciDig_Freq* values favor malicious name classifications. Such behavior is related to lengthy names and high decimal digit frequencies, typically employed by most DGA's to avoid coincidence with legitimate registered domain names. As expected, high *Reputation* and *Words_Freq* values mostly point to benign name categorizations since the presence of many whitelisted N-grams and meaningful words are linked to legitimate names. *Max_DeciDig_Seq* contribution is significantly smaller compared to the impact of the aforementioned features; it is ranked 12th in terms of contribution to classification decisions. Finally, high feature values of *Words_Mean* may inconclusively affect both benign and malicious name classifications.

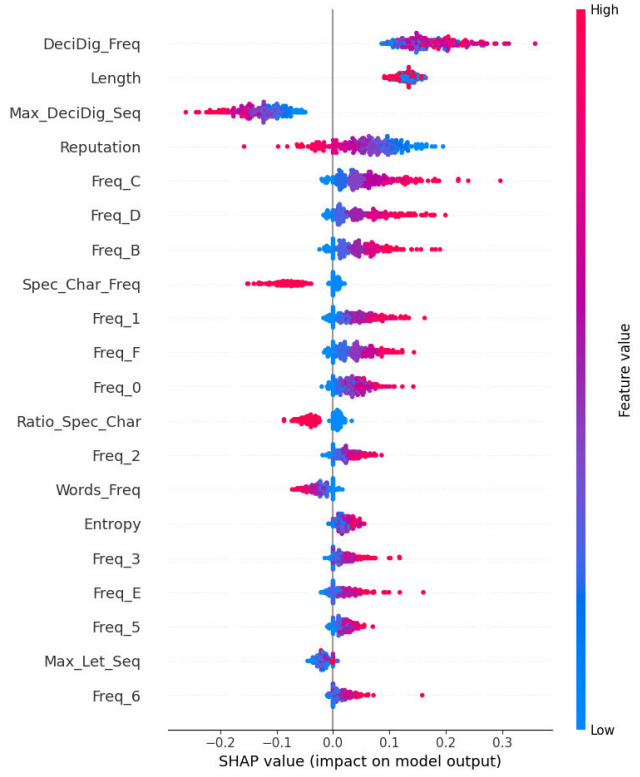
Fig. 2b shows that the most influential features used by the MLP classifier are *Reputation*, *Length*, *Max_DeciDig_Seq*, *Words_Mean* and *DeciDig_Freq*. Similarly to XGBoost, MLP relies predominantly on *Reputation* and *Length* features. *Max_DeciDig_Seq* was the 3rd most important feature for MLP with higher values pointing to benign name classifications. Recall that for XGBoost, *Max_DeciDig_Seq* was ranked 12th, a much lower significance level (Fig. 2a). Likewise, *Vowel_Freq* feature significantly affects MLP



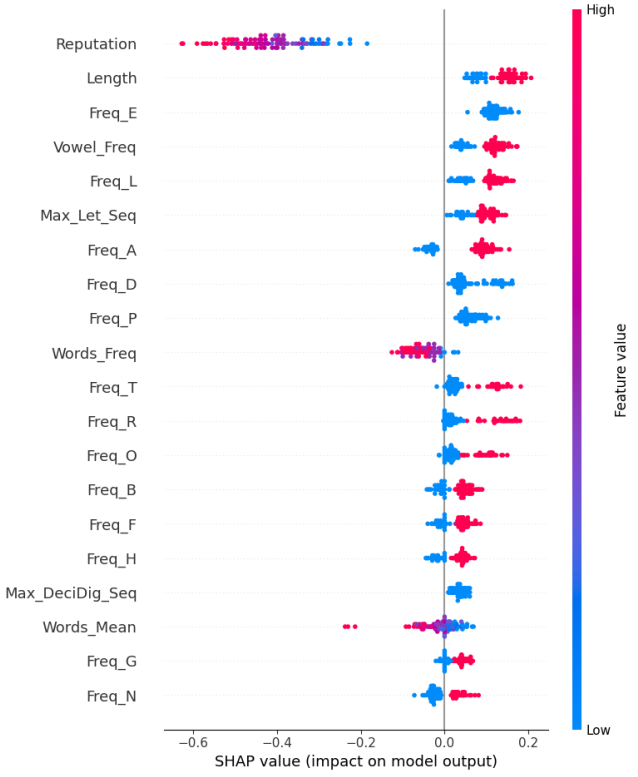
(a) DGA Family: *DirCrypt* (Arithmetic-based DGA)



(b) DGA Family: *Matsnu* (Wordlist-based DGA)



(c) DGA Family: *Bamital* (Hash-based DGA)



(d) DGA Family: *VolatileCedar* (Permutation-based DGA)

FIGURE 3. **SHAP summary plots derived for XTI's from specific DGA families (Algorithm: Multi-Layer Perceptron - MLP).**

decisions ranking as the 8th most influential feature, while XGBoost dependence on *Vowel_Freq* is not even among the 20 most significant features of Fig. 2a. This may be partially explained by the difference of XGBoost and MLP in modeling learning tasks. The former mainly relies on splitting training set instances based on dominant feature deviations; following boosting methods strong tree estimators are eventually constructed by iteratively improving weaker classifiers. The latter (MLP) tunes its weights during back propagation towards directions that linearly combine feature values, forming induced local fields that are further subjected to non-linear activation functions (e.g. ReLU, Sigmoid). Thus, XGBoost mainly relies on boosting methods based on significant feature deviations [60], while MLP on weighted feature differences.

2) MLP CLASSIFIER SUMMARY PLOTS FOR SELECTED DGA FAMILIES

This subsection addresses explanations pertaining to binary MLP classifiers tested for XTI's derived from specific DGA families. In Fig. 3 we present summary plots for 4 DGA families selected from 4 different generation schemes: (a) *DirCrypt* (arithmetic-based), (b) *Matsnu* (wordlist-based), (c) *Bamital* (hash-based) and (d) *VolatileCedar* (permutation-based). In Table 5 we list four indicative malicious names pertaining to each of the aforementioned DGA families; note that typical suffixes, e.g. "*.com*" and "*.info*", are not included in the table. These schemes and their respective families have the following properties [1]:

- Arithmetic-based DGA's (e.g. *DirCrypt*): Domain names are generated by concatenating randomly selected characters. *DirCrypt* is based on the 26 English alphabet letters to produce names between 8 and 20 characters. Names typically contain long consonant sequences and are characterized by increased randomness compared to benign names.
- Wordlist-based DGA's (e.g. *Matsnu*): Random dictionary words are concatenated to generate malicious domain names resembling legitimate ones. *Matsnu* forms long names between 12 and 24 characters by joining multiple dictionary words of relatively short length [61].
- Hash-based DGA's (e.g. *Bamital*): They rely on the hexadecimal representation resulting from hashing domain names. *Bamital* is based on *MD5* hash function to generate names consisting of 32 hexadecimal digits.
- Permutation-based DGA's (*VolatileCedar*): Multiple DGA names are produced by permuting a generated domain name that resembles legitimate names. Linguistic (e.g. number of vowels) and statistical properties (e.g. letter frequencies) of the initial malignant name are inherited by derived names.

In the following we analyze specific feature contributions using summary plots derived by experimenting with malignant XTI's, randomly subsampled from the aforementioned DGA schemes:

TABLE 5. Indicative names per DGA family.

DGA Family	Scheme	Indicative Name (Prefix)
DirCrypt	Arithmetic	iwqvktutvmptevjbnzy
Matsnu	Wordlist	chickenpriceresearch
Bamital	Hash	b7a8b33957a2f95105353aa1873aebda
VolatileCedar	Permutation	shplayergetadobaefl

- For *DirCrypt*, Fig. 3a shows that *Reputation* and *Length* are the most important features (higher SHAP values) followed by *Words_Mean*, *Freq_X*, *Freq_Q* and *Max_Let_Seq*. As expected, high *Length* and *Max_Let_Seq* values favor the malicious class since the typically long names and absence of digits discern *DirCrypt* names from benign ones. On the contrary, high values of *Reputation* and *Words_Freq* favor legitimate name classifications since *DirCrypt* names contain less whitelisted N-grams and meaningful words. High feature values of *Words_Mean* may be inconclusive, whereas lower *Words_Mean* values point to malignant (DGA) name categorizations.
- Regarding *Matsnu*, Fig. 3b shows that the most important features in terms of SHAP values are *Reputation*, *Words_Mean*, *Length*, *Vowel_Freq*, *Freq_B* and *Max_Let_Seq*. *Reputation* exclusively contributes to benign name classifications (negative SHAP values) since many whitelisted N-grams may be present in both legitimate and *Matsnu* names, therefore favoring misclassifications (FN's) of DGA XTI's. High *Words_Mean* values point to benign name classifications (FN's); this is expected as *Matsnu* concatenates dictionary words that are typically short [61], thus higher *Words_Mean* values (mean length of meaningful words within the name) may mislead the classifier towards benign name classifications. *Reputation* and *Words_Mean* influence is mainly counterbalanced by *Length*, *Vowel_Freq* and *Freq_B* values. High *Length* values point to malicious name classifications since *Matsnu* names are typically longer than benign names. Although vowels are typically present in both benign and *Matsnu* names, high *Vowel_Freq* values enable DGA name categorizations (TP's); *Matsnu* names are usually longer than benign ones, hence they typically include more vowels. Letter *B* was found in various *Matsnu* XTI's, thus high *Freq_B* favors TP's.
- Regarding *Bamital*, Fig. 3c shows that *DeciDig_Freq*, *Length*, *Max_DeciDig_Seq* and *Reputation* mainly affect model decisions. High *DeciDig_Freq* values (i.e. total frequency of decimal digits 0-9) and high frequencies of specific hexadecimal digits (e.g. *Freq_C*, *Freq_D*, *Freq_B* and *Freq_1*) contribute significantly to TP's since *Bamital* names exclusively consist of such characters. As expected, impact of *Length* is very important since *Bamital* names follow *MD5* hash function statistical properties and their size is fixed (i.e. 32 characters), thus clearly distinguishing them from benign names. High *Max_DeciDig_Seq* (i.e. maximum

digit sequence) values point to misclassifications of DGA names as benign (FN's) since long decimal digit sequences are usually not present in *Bamital* names; hash function results are typically uniform, therefore short decimal digit sequences are followed by hexadecimal digits. High *Reputation* values erroneously favor the class of benign names (FN's) as the frequency of whitelisted N-grams within *Bamital* names is usually limited.

- For *VolatileCedar*, Fig. 3d shows that *Reputation* is the most important feature exclusively favoring legitimate classifications (FN's) with negative SHAP values; the initial name used by *VolatileCedar* resembles benign names, therefore many DGA N-grams may be included within the *Reputation whitelist*. The effect of *Reputation* is mainly counterbalanced by features *Length*, *Freq_E*, *Vowel_Freq*, *Freq_L* and *Max_Let_Seq*. As a permutation-based DGA, *VolatileCedar* is characterized by specific feature values, which act as signatures for discerning malicious names from benign ones.

3) XGBOOST AND MLP CLASSIFIER DEPENDENCE PLOTS FOR ALL DGA FAMILIES

In this subsection SHAP dependence plots are used to investigate pairwise feature relationships, thus complementing our analysis based on summary plots. Fig. 4 depicts XGBoost and MLP classifier dependence plots on malignant XTI's subsampled from all DGA families. Plots are provided for 4 features of interest, i.e. *Reputation*, *Entropy*, *Max_DeciDig_Seq* and *Words_Mean*. Interacting features are determined by SHAP using Shapley interaction values (subsection IV-C); red and blue colors denote high and low values of interacting features respectively, while these values are depicted normalized between *0* and *1* (subsection IV-B). Interactions pertaining to features of interest are summarized below:

- *Reputation Interactions*: Fig. 4a and Fig. 4b show that *Reputation* significantly influences classifications. Namely, *Reputation* interacts with *Length* for XGBoost and *DeciDig_Freq* for MLP. However, combined *Reputation* and interacting feature values do not clearly affect classification decisions because, as shown in Fig. 2, the impact of *Reputation* is significantly higher than that of *Length* and *DeciDig_Freq*.
- *Entropy Interactions*: As expected from the summary plots of subsection V-D1, Fig. 4c and Fig. 4d show that *Entropy* values are not significant for both XGBoost and MLP classifiers. Although higher *Entropy* values may favor malicious name categorizations for MLP's, their SHAP values are considerably low, therefore *Entropy* effect is counterbalanced by more influential features.
- *Max_DeciDig_Seq Interactions*: As already mentioned in subsection V-D1, values of *Max_DeciDig_Seq* feature are not significant for XGBoost (Fig. 4e). For MLP, Fig. 4f depicts that *Max_DeciDig_Seq* significantly impacts classifications and interacts with *Length*. Long sequences of decimal digits, i.e. high *Max_DeciDig_Seq* values, combined with shorter names, i.e. low *Length* values favor benign name classifications. This is expected as several DGA families alternate letters and decimal digits, thus long digit sequences are not formed.
- *Words_Mean Interactions*: Fig. 4g and Fig. 4h show that *Words_Mean* values affect both XGBoost and MLP classifiers. However, although high *Words_Mean* values favor legitimate name classifications (FN's) for MLP, for XGBoost high *Words_Mean* values mainly point to malicious name categorizations. Explicit correlations between *Words_Mean* and other interacting features are not evident in our experiments.

4) MLP CLASSIFIER DEPENDENCE PLOTS FOR SELECTED DGA FAMILIES

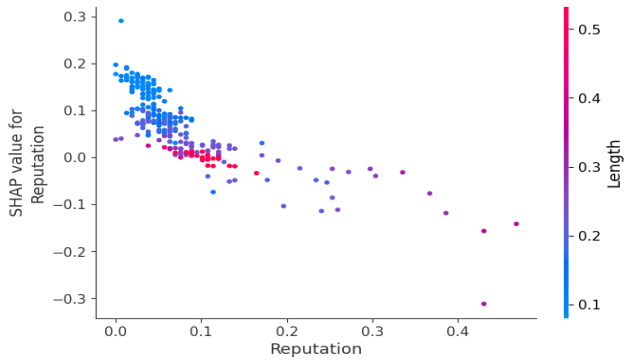
In this subsection we present indicative dependence plots for MLP's mapping eXplainability Test Instances (XTI's) for dominant features per DGA scheme (see subsection V-D2). Notably, in Fig. 5 we indicatively present dependence plots pertaining to *DirCrypt* and *Bamital*.

DirCrypt XTI's in Fig. 5a and Fig. 5b show that *Reputation* and *Length* features interact with *Max_Let_Seq* and *Reputation* respectively. High *Reputation* and *Max_Let_Seq* values favor benign class categorizations (FN's), while high *Length* values favor TP's. Such effect of *Reputation* and *Length* on model classifications is expected since *DirCrypt* names are typically long and randomized, thus they stand out from benign names. Note that *Length* effect on model decisions increases at a smaller rate as *Reputation* values increase.

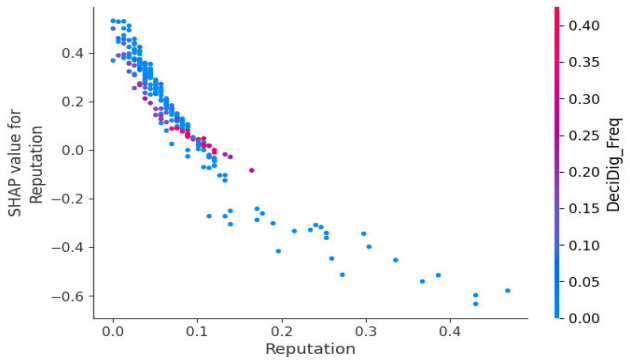
Bamital XTI's in Fig. 5c and Fig. 5d show that *DeciDig_Freq* interacts with *Spec_Char_Freq*, while *Max_DeciDig_Seq* with *Length*. Increasing *DeciDig_Freq* favors TP's, with its influence increasing (higher SHAP values) for higher values of the interacting feature (*Spec_Char_Freq*). This is expected because *Bamital* names consist of hexadecimal digits, thus decimal digits constitute their majority. Moreover, as in Fig. 5d, increased *Max_DeciDig_Seq* values favor FN's since *Bamital* follows the statistical properties of *MD5* hash function with hexadecimal digits uniformly distributed across domain names. Therefore, long decimal digit sequences typically favor benign name misclassifications.

5) MLP CLASSIFIER FORCE PLOTS FOR LOCAL EXPLAINABILITY

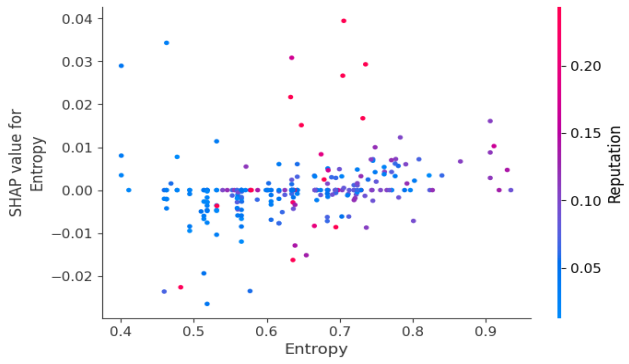
In this subsection force plots are used to analyze the operation of binary MLP classifiers pertaining to specific inputs (*local* explainability). Force plots are particularly helpful for understanding False Positives (FP's) and False Negatives (FN's) in classification of specific benign and DGA names. In these plots, features dominantly influencing name classifications are depicted along with their values. Red color denotes features favoring malicious name categorizations and blue colors those contributing to benign name classifications. A bold decimal value corresponds to the classifier output.



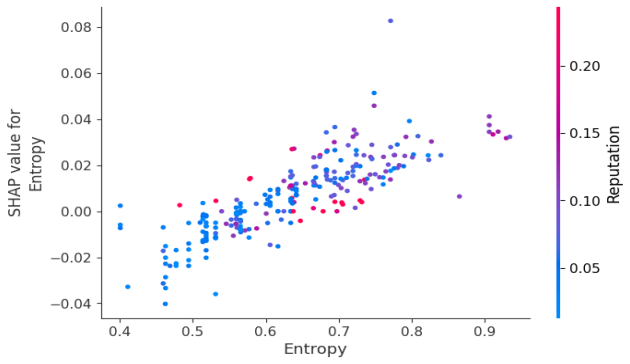
(a) Feature: *Reputation*, Model: XGBoost



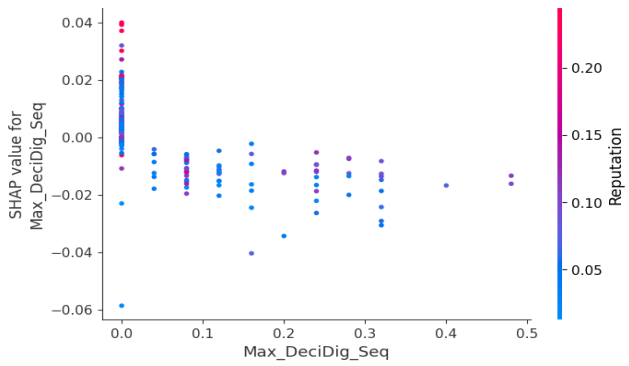
(b) Feature: *Reputation*, Model: MLP



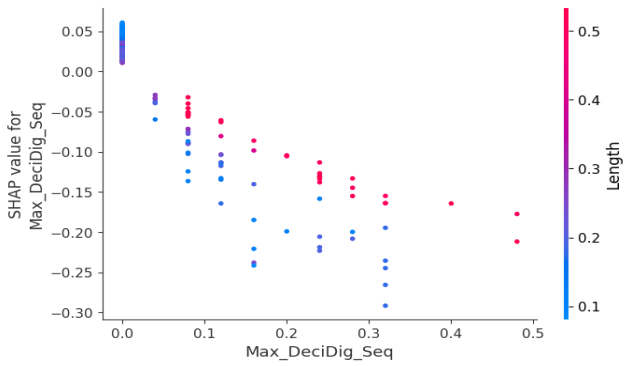
(c) Feature: *Entropy*, Model: XGBoost



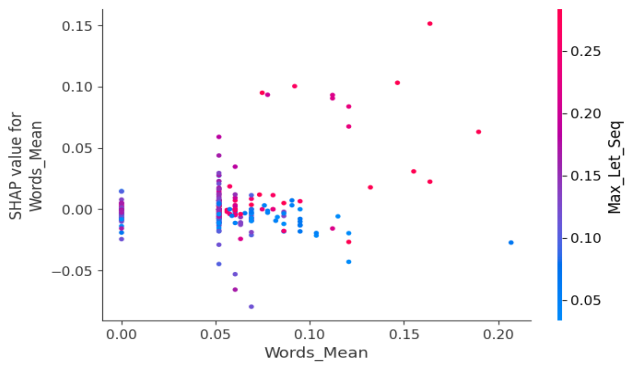
(d) Feature: *Entropy*, Model: MLP



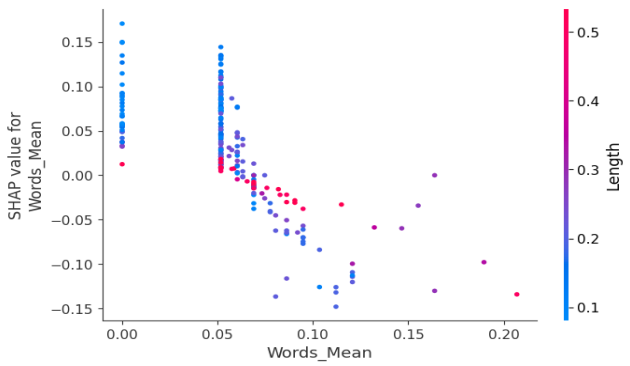
(e) Feature: *Max_DeciDig_Seq*, Model: XGBoost



(f) Feature: *Max_DeciDig_Seq*, Model: MLP

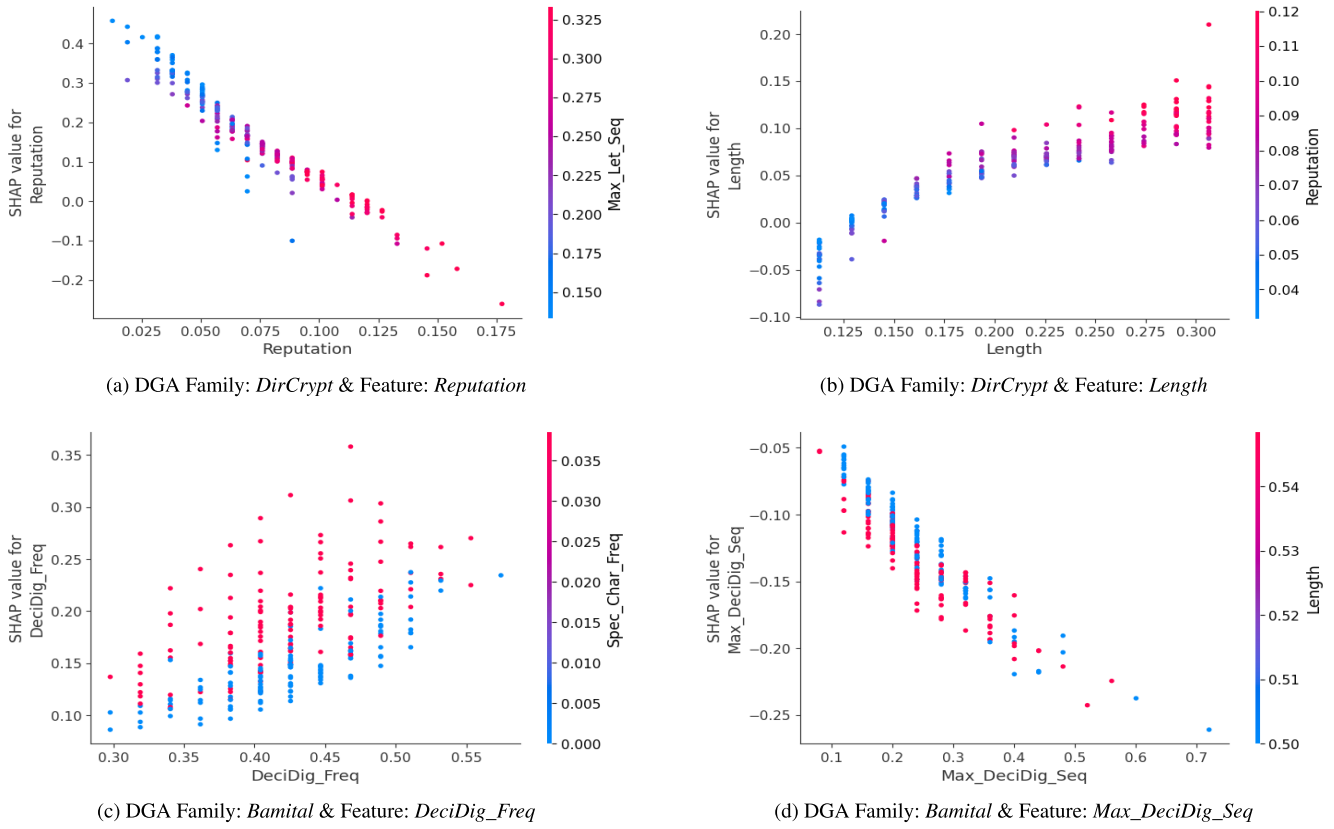


(g) Feature: *Words_Mean*, Model: XGBoost



(h) Feature: *Words_Mean*, Model: MLP

FIGURE 4. **SHAP dependence plots derived for XTI's including malicious DNS names from all DGA families.**



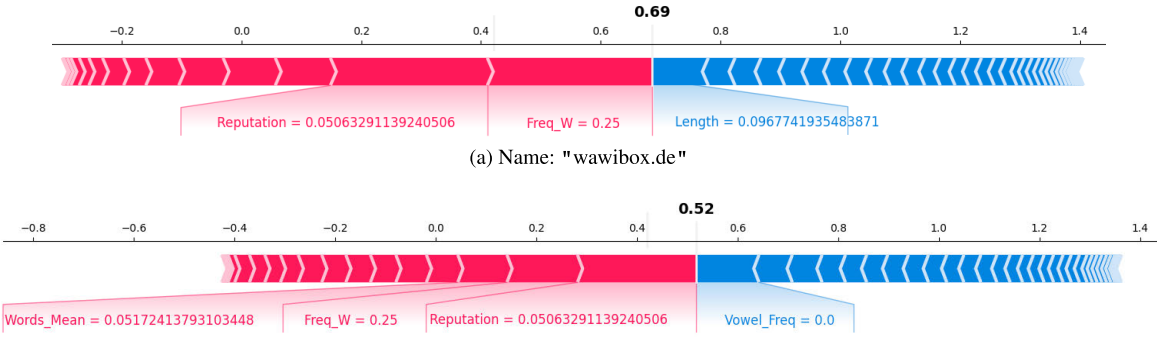
(a) DGA Family: *DirCrypt* & Feature: *Reputation*

(b) DGA Family: *DirCrypt* & Feature: *Length*

(c) DGA Family: *Bamital* & Feature: *DeciDig_Freq*

(d) DGA Family: *Bamital* & Feature: *Max_DeciDig_Seq*

FIGURE 5. **SHAP dependence plots derived for XTI's from specific DGA families (Algorithm: Multi-Layer Perceptron - MLP).**



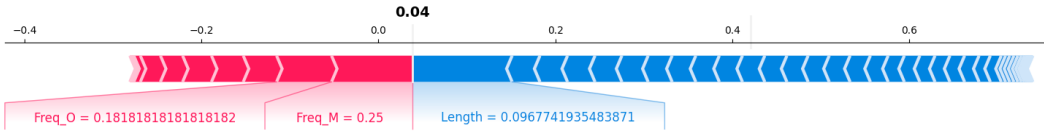
(a) Name: "wawibox.de"

(b) Name: "rvwgm2wrld2.xyz"

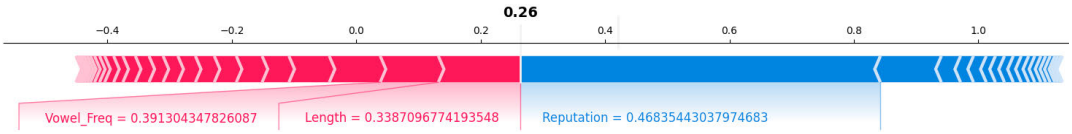
FIGURE 6. **Force plots pertaining to benign (non-DGA) names incorrectly classified as DGA.**

Fig. 6 depicts force plots pertaining to MLP FP's, i.e. benign (non-DGA) names incorrectly categorized as DGA. Fig. 6a shows that name ''wawibox.de'' is perceived as DGA, mainly because of the high frequency of letter *W* and the low *Reputation* value. For this particular name *Freq_W* and the absence of many whitelisted N-grams override the effect of *Length* that favors benign name classifications. Fig. 6b shows that name ''rvwgm2wrld2.xyz'', which is frequently used for malware propagation [62] but is not produced by DGA's, is misclassified as DGA. This is attributed to the low *Reputation* value, the high frequency of letter *W* and the low *Words_Mean* value, although the zero *Vowel_Freq* value might point to non-DGA name classification.

Fig. 7 depicts force plots pertaining to MLP FN's, i.e. DGA names incorrectly classified as benign. Fig. 7a shows that *Length* values have a major effect on misclassifying name ''nomodum.info'', generated by the *Simda* arithmetic-based DGA family, despite the high frequencies of letters *M* and *O* that favor malicious name classifications. Fig. 7b shows that name ''californiatransferable.ru'' originating from the *Gozi* wordlist-based DGA family is classified as benign because *Reputation* values point to benign name classifications. This counterbalances the effect of name



(a) Name: "nomodum.info", DGA Family: *Simda*



(b) Name: "californiatransferable.ru", DGA Family: *Gozi*

FIGURE 7. Force plots pertaining to malicious names incorrectly classified as benign.

length and the high presence of vowels that point towards DGA names.

VI. CONCLUSION AND FUTURE WORK

We investigated XAI methods for interpreting DGA name classifiers that detect malicious DNS messages used by bots to communicate with Command & Control (C&C) servers. We addressed defense mechanisms based on ML classifiers and analyzed their operation via the SHapley Additive exPlanation (SHAP) algorithm that provides *global* and *local* interpretations in a model-agnostic, post-hoc manner.

To that end, we first configured tree-based and deep neural network binary classifiers for differentiating between benign DNS names and malicious names produced by DGA's. We trained and evaluated classifiers based on supervised ML algorithms, specifically Random Forests (RF's), Gradient Boosting (GB), eXtreme Gradient Boosting (XGBoost), Adaptive Boosting (AdaBoost), Extremely Randomized Trees (ExtraTrees) and Multi-Layer Perceptrons (MLP's). These relied on features directly extracted from domain name datasets, thus eliminating time-consuming and privacy-sensitive operations on repositories of historical data. Classifiers were trained using up-to-date and inclusive datasets. Legitimate names originated from *Tranco*, an online service ranking top Internet sites; we selected the 1 million most popular names. Malicious instances were sampled from the *DGArchive* repository, which reports 105 DGA families from 4 different generation schemes; we randomly selected 600,775 DGA names.

Our SHAP-based evaluation analyzed the features used by our trained XGBoost (determined as the most accurate tree-based model) and MLP deep neural network classifiers to segregate benign and DGA name instances. We investigated how DGA families and their different underlying algorithmic generation schemes (i.e. arithmetic, wordlist, hash or permutation based) affect the features that specifically influence classification decisions. Relying on multiple SHAP visualization tools (summary, dependence and force plots) we provided *global* and *local* interpretations on sampled dataset instances. Specifically, we ranked feature importance, investigated the effect of feature values on model decisions and determined their interactions. Using up-to-date and extensive datasets, we conclude that our SHAP-based analysis enables interpretations of XGBoost and MLP name classifiers, attacked by well-known diverse DGA schemes. Such methods may facilitate ML adoption within networking environments where interpretations for black-box schemes are required.

TABLE 6. Abbreviations.

Acronym	Definition
AdaBoost	Adaptive Boosting
BiLSTM	Bidirectional Long Short-Term Memory
C&C	Command & Control
CNN	Convolutional Neural Network
DGA	Domain Generation Algorithm
DNS	Domain Name System
ExtraTrees	Extremely Randomized Trees
FN	False Negative
FP	False Positive
GB	Gradient Boosting
GDPR	General Data Protection Regulation
LIME	Local Interpretable Model-Agnostic Explanation
LSTM	Long Short-Term Memory
ML	Machine Learning
MLP	Multi-Layer Perceptron
NLP	Natural Language Processing
PCC	Pearson's Correlation Coefficient
RF	Random Forest
SHAP	SHapley Additive exPlanation
SMOTE	Synthetic Minority Oversampling Technique
TN	True Negative
TP	True Positive
TTL	Time To Live
VM	Virtual Machine
XAI	eXplainable Artificial Intelligence
XBI	eXplainability Background Instance
XGBoost	eXtreme Gradient Boosting
XTI	eXplainability Test Instance

We plan to extend our SHAP-based interpretations to address additional deep neural network models. These include Convolutional Neural Networks (CNN's), Long Short-Term Memory (LSTM) networks and/or Bidirectional LSTM (BiLSTM) networks that may be employed for DGA name classification [13]. Alternative XAI approaches, e.g. LIME [30] and Counterfactual Explanation [31], will also be considered. The proposed scheme may be further adapted to unsupervised deep learning models, e.g. Autoencoders. Finally, the proposed approach will be extended to multi-domain infrastructures using Federated Learning [63] for collaborative DGA name detection, similarly to [64], [65]. Therefore, privacy-aware model interpretations will be derived without sharing attack and benign data.

ACKNOWLEDGMENT

The authors sincerely thank the reviewers for providing valuable guidance during the peer review process. They also thank Daniel Plohmann and the Cyber Analysis and Defense Department, Fraunhofer FKIE, for granting access to DGArchive.

ABBREVIATIONS

Paper abbreviations are listed in Table 6.

REFERENCES

[1] D. Plohmann, K. Yakdan, M. Klatt, J. Bader, and E. Gerhards-Padilla, "A comprehensive measurement study of domain generating malware," in *Proc. USENIX Secur. Symp.*, Austin, TX, USA, Aug. 2016, pp. 263–278.

[2] *General Data Protection Regulation—GDPR*. Accessed: Jun. 1, 2023. [Online]. Available: https://gdpr-info.eu/

[3] C. Molnar. (2022). *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. Accessed: Jun. 1, 2023. [Online]. Available: https://christophm.github.io/interpretable-ml-book/

[4] L. Gianfagna and A. Di Cecco, *Explainable AI With Python*. Berlin, Germany: Springer, Apr. 2021, pp. 1–202, doi: 10.1007/978-3-030-68640-6.

[5] S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions," in *Proc. Int. Conf. Neural Inf. Process. Syst. (NIPS)*, Long Beach, CA, USA, Dec. 2017, pp. 4768–4777.

[6] *SHAP GitHub Repository*. Accessed: Jun. 1, 2023. [Online]. Available: https://github.com/slundberg/shap

[7] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, San Francisco, CA, USA, Aug. 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[8] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[9] *DGArchive Repository Access Portal*. Accessed: Jun. 1, 2023. [Online]. Available: https://dgarchive.caad.fkie.fraunhofer.de/welcome/

[10] *Paper GitHub Repository*. Accessed: Jun. 1, 2023. [Online]. Available: https://github.com/nkostopoulos/dga-explainability

[11] J. Woodbridge, H. S. Anderson, A. Ahuja, and D. Grant, "Predicting domain generation algorithms with long short-term memory networks," 2016, *arXiv:1611.00791*.

[12] B. Yu, J. Pan, D. Gray, J. Hu, C. Choudhary, A. C. A. Nascimento, and M. De Cock, "Weakly supervised deep learning for the detection of domain generation algorithms," *IEEE Access*, vol. 7, pp. 51542–51556, 2019, doi: 10.1109/ACCESS.2019.2911522.

[13] J. Namgung, S. Son, and Y.-S. Moon, "Efficient deep learning models for DGA domain detection," *Secur. Commun. Netw.*, vol. 2021, pp. 1–15, Jan. 2021, doi: 10.1155/2021/8887881.

[14] K. H. Park, H. M. Song, J. D. Yoo, S.-Y. Hong, B. Cho, K. Kim, and H. K. Kim, "Unsupervised malicious domain detection with less labeling effort," *Comput. Secur.*, vol. 116, May 2022, Art. no. 102662, doi: 10.1016/j.cose.2022.102662.

[15] B. Yu, J. Pan, J. Hu, A. Nascimento, and M. De Cock, "Character level based detection of DGA domain names," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Rio de Janeiro, Brazil, Jul. 2018, pp. 1–8, doi: 10.1109/IJCNN.2018.8489147.

[16] L. Yang, G. Liu, Y. Dai, J. Wang, and J. Zhai, "Detecting stealthy domain generation algorithms using heterogeneous deep neural network framework," *IEEE Access*, vol. 8, pp. 82876–82889, 2020, doi: 10.1109/ACCESS.2020.2988877.

[17] S. Schuppen, D. Teubert, P. Herrmann, and U. Meyer, "FANCI: Feature-based automated NXDomain classification and intelligence," in *Proc. USENIX Secur. Symp.*, Baltimore, MD, USA, Aug. 2018, pp. 1165–1181.

[18] M. Antonakakis, R. Perdisci, Y. Nadji, N. Vasiloglou, A. N. Saeed, and L. Wenke, "From throw-away traffic to bots: Detecting the rise of DGA-based malware," in *Proc. Usenix Conf. Secur. Symp.*, Bellevue, WA, USA, Aug. 2012, pp. 491–506.

[19] L. Bilge, S. Sen, D. Balzarotti, E. Kirda, and C. Kruegel, "Exposure: A passive DNS analysis service to detect and report malicious domains," *ACM Trans. Inf. Syst. Secur.*, vol. 16, no. 4, pp. 1–28, Apr. 2014, doi: 10.1145/2584679.

[20] A. O. Almashhadani, M. Kaiiali, D. Carlin, and S. Sezer, "MaldomDetector: A system for detecting algorithmically generated domain names with machine learning," *Comput. Secur.*, vol. 93, Jun. 2020, Art. no. 101787, doi: 10.1016/j.cose.2020.101787.

[21] C. Marques, S. Malta, and J. Magalhães, "DNS firewall based on machine learning," *Future Internet*, vol. 13, no. 12, p. 309, Nov. 2021, doi: 10.3390/fi13120309.

[22] *SHAP KernelExplainer*. Accessed: Jun. 1, 2023. [Online]. Available: https://shap-lrjball.readthedocs.io/en/latest/generated/shap.KernelExplainer.html

[23] X. Sun, M. Tong, J. Yang, X. Liu, and H. Liu, "HinDom: A robust malicious domain detection system based on heterogeneous information network with transductive classification," in *Proc. USENIX Int. Symp. Res. Attacks, Intrusions Defenses (RAID)*, Beijing, China, Sep. 2019, pp. 399–412.

[24] K. Bartos, M. Sofka, and V. Franc, "Optimized invariant representation of network traffic for detecting unseen malware variants," in *Proc. USENIX Secur. Symp.*, Austin, TX, USA, Aug. 2016, pp. 807–822.

[25] F. Becker, A. Drichel, C. Muller, and T. Ertl, "Interpretable visualizations of deep neural networks for domain generation algorithm detection," in *Proc. IEEE Symp. Visualizat. Cyber Secur. (VizSec)*, Oct. 2020, pp. 25–29, doi: 10.1109/VizSec51108.2020.00010.

[26] A. Drichel, N. Faerber, and U. Meyer, "First step towards EXPLAINable DGA multiclass classification," in *Proc. 16th Int. Conf. Availability, Rel. Secur.*, Aug. 2021, pp. 1–13.

[27] N. Aslam, I. U. Khan, S. Mirza, A. AlOwayed, F. M. Anis, R. M. Aljuaid, and R. Baageel, "Interpretable machine learning models for malicious domains detection using explainable artificial intelligence (XAI)," *Sustainability*, vol. 14, no. 12, p. 7375, Jun. 2022, doi: 10.3390/su14127375.

[28] H. Suryotrisongko, Y. Musashi, A. Tsuneda, and K. Sugitani, "Robust botnet DGA detection: Blending XAI and OSINT for cyber threat intelligence sharing," *IEEE Access*, vol. 10, pp. 34613–34624, 2022, doi: 10.1109/ACCESS.2022.3162588.

[29] G. Piras, M. Pintor, L. Demetrio, and B. Biggio, "Explaining machine learning DGA detectors from DNS traffic data," 2022, *arXiv:2208.05285*.

[30] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should i trust you?' Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, San Francisco, CA, USA, Aug. 2016, pp. 1135–1144, doi: 10.1145/2939672.2939778.

[31] S. Wachter, B. Mittelstadt, and C. Russell, "Counterfactual explanations without opening the black box: Automated decisions and the GDPR," *Harv. JL Tech.*, vol. 31, no. 2, p. 841, 2018.

[32] O. Ayoub, N. Di Cicco, F. Ezzeddine, F. Bruschetta, R. Rubino, M. Nardecchia, M. Milano, F. Musumeci, C. Passera, and M. Tornatore, "Explainable artificial intelligence in communication networks: A use case for failure identification in microwave networks," *Comput. Netw.*, vol. 219, Dec. 2022, Art. no. 109466, doi: 10.1016/j.comnet.2022.109466.

[33] 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. 22nd IEEE Int. Symp. Cluster, Cloud Internet Comput. (CCGrid)*, Taormina, Italy, May 2022, pp. 933–939, doi: 10.1109/CCGrid54584.2022.00113.

[34] *Johannes Bader GitHub Repository*. Accessed: Jun. 1, 2023. [Online]. Available: https://github.com/baderj/domain_generation_algorithms

[35] *BIND 9 DNS Software*. Accessed: Jun. 1, 2023. [Online]. Available: https://www.isc.org/bind/

[36] *Mozilla Public Suffix List*. Accessed: Jun. 1, 2023. [Online]. Available: https://publicsuffix.org/

[37] D. Yan, H. Zhang, Y. Wang, T. Zang, X. Xu, and Y. Zeng, "Pontus: A linguistics-based DGA detection system," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Waikoloa, HI, USA, Dec. 2019, pp. 1–6, doi: 10.1109/GLOBECOM38437.2019.9014040.

[38] M. Antonakakis, R. Perdisci, D. Dagon, W. Lee, and N. Feamster, "Building a dynamic reputation system for DNS," in *Proc. USENIX Secur. Symp.*, Washington, DC, USA, Aug. 2010, pp. 273–290.

[39] H. Zhao, Z. Chang, G. Bao, and X. Zeng, "Malicious domain names detection algorithm based on N-Gram," *J. Comput. Netw. Commun.*, vol. 2019, pp. 1–9, Feb. 2019, doi: 10.1155/2019/4612474.

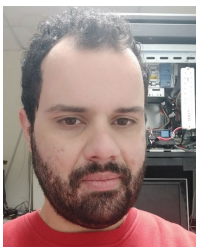
[40] V. Le Pochat, T. Van Goethem, S. Tajalizadehkhoob, M. Korczynski, and W. Joosen, "Tranco: A research-oriented top sites ranking hardened against manipulation," 2018, *arXiv:1806.01156*.

[41] *Wordninja GitHub Repository*. Accessed: Jun. 1, 2023. [Online]. Available: https://github.com/keredson/wordninja

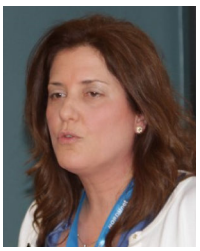
[42] J. J. Koh and B. Rhodes, "Inline detection of domain generation algorithms with context-sensitive word embeddings," in *Proc. IEEE Int. Conf. Big Data*, Seattle, WA, USA, Dec. 2018, pp. 2966–2971. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8622066
[43] C. Patsakis and F. Casino, "Exploiting statistical and structural features for the detection of domain generation algorithms," *J. Inf. Secur. Appl.*, vol. 58, May 2021, Art. no. 102725, doi: 10.1016/j.jisa.2020.102725.
[44] W. Kirch, "Pearson's correlation coefficient," in *Encyclopedia of Public Health*, vol. 1, Berlin, Germany: Springer, 2008.
[45] T. Zebin, S. Rezvy, and Y. Luo, "An explainable AI-based intrusion detection system for DNS over HTTPS (DoH) attacks," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 2339–2349, 2022, doi: 10.1109/TIFS.2022.3183390.
[46] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.
[47] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Apr. 2011.
[48] *XGBoost Python Package*. Accessed: Jun. 1, 2023. [Online]. Available: https://xgboost.readthedocs.io/en/stable/index.html
[49] *Keras GitHub Repository*. Accessed: Jun. 1, 2023. [Online]. Available: https://github.com/keras-team/keras
[50] P. Liashchynskyi and P. Liashchynskyi, "Grid search, random search, genetic algorithm: A big comparison for NAS," 2019, *arXiv:1912.06059*.
[51] S. M. Lundberg, G. G. Erion, and S.-I. Lee, "Consistent individualized feature attribution for tree ensembles," 2018, *arXiv:1802.03888*.
[52] A. Drichel, J. von Brandt, and U. Meyer, "Detecting unknown DGAs without context information," in *Proc. 17th Int. Conf. Availability, Rel. Secur.*, Aug. 2022, pp. 1–12, doi: 10.1145/3538969.3538990.
[53] *Tranco Website: A Research-Oriented Top Sites Ranking Hardened Against Manipulation*. Accessed: Jun. 2023. [Online]. Available: https://tranco-list.eu/
[54] *GeForce GTX 1050 Ti Specifications*. Accessed: Jun. 1, 2023. [Online]. Available: https://www.nvidia.com/en-gb/geforce/graphics-cards/geforce-gtx-1050-ti/specifications/
[55] *Correlation Analysis*. Accessed: Jun. 1, 2023. [Online]. Available: https://sphweb.bumc.bu.edu/otlt/mph-modules/bs/bs704_correlation-regression/bs704_correlation-regression2.html
[56] *EarlyStopping (Keras Documentation)*. Accessed: Jun. 1, 2023. [Online]. Available: https://keras.io/api/callbacks/early_stopping/
[57] *BinaryCrossentropy Loss Function (TensorFlow Documentation)*. Accessed: Jun. 1, 2023. [Online]. Available: https://www.tensorflow.org/api_docs/python/tf/keras/losses/BinaryCrossentropy
[58] *Adam Optimizer (Keras Documentation)*. Accessed: Jun. 1, 2023. [Online]. Available: https://keras.io/api/optimizers/adam/
[59] *DGArchive—A Deep Dive Into Domain Generating Malware*. Accessed: Jun. 1, 2023. [Online]. Available: https://www.botconf.eu/2015/dgarchive-a-deep-dive-into-domain-generating-malware/
[60] R. E. Schapire and Y. Freund, *Boosting: Foundations and Algorithms*. Cambridge, MA, USA: MIT Press, 2012, doi: 10.7551/mitpress/8291.001.0001.
[61] *Matsnu Reverse Engineered Code*. Accessed: Jun. 1, 2023. [Online]. Available: https://github.com/andrewaeva/DGA/blob/master/dga_algorithms/Matsnu.py
[62] *eu.rvwgm2wrld2.xyz—How to Get Rid It?* Accessed: Jun. 1, 2023. [Online]. Available: https://easysolvemalware.com/eu-rvwgm2wrld2-xyz-how-to-get-rid-of-it/
[63] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, Ft. Lauderdale, FL, USA, Feb. 2017, pp. 1273–1282.
[64] A. Drichel, B. Holmes, J. von Brandt, and U. Meyer, "The more, the better: A study on collaborative machine learning for DGA detection," in *Proc. 3rd Workshop Cyber-Security Arms Race*, Nov. 2021, pp. 1–12, doi: 10.1145/3474374.3486915.
[65] M. Dimolianis, D. K. Kalogeras, N. Kostopoulos, and V. Maglaris, "DDoS attack detection via privacy-aware federated learning and collaborative mitigation in multi-domain cyber infrastructures," in *Proc. IEEE 11th Int. Conf. Cloud Netw. (CloudNet)*, Paris, France, Nov. 2022, pp. 118–125, doi: 10.1109/CloudNet55617.2022.9978815.

NIKOS KOSTOPOULOS received the engineering degree from the National Technical University of Athens (NTUA), Athens, Greece, in 2017, where he is currently pursuing the Ph.D. degree. His research interests include the liaison of big data methods, programmable data planes, and machine learning algorithms for defending against attacks targeting (e.g. DDoS attacks) or abusing (e.g. DGA's) the normal operation of DNS.

DIMITRIS KALOGERAS received the engineering and Ph.D. degrees from the National Technical University of Athens (NTUA), Athens, Greece, in 1991 and 1996, respectively. He is currently a Senior Researcher with the Institute of Communication and Computer Systems (ICCS), NTUA. His research spans several aspects of advanced network architectures. He is also involved in the operational aspects of the Greek School Network, where he is applying DGA traffic detection mechanisms.



DIMITRIS PANTAZATOS received the bachelor's degree in economics and the master's degree in informatics from the University of Piraeus (UniPi), Piraeus, Greece, in 2013 and 2016, respectively. He is currently pursuing the Ph.D. degree with the National Technical University of Athens (NTUA). His research interests include the application of big data and machine learning algorithms pertaining to computer networking and online education.



MARIA GRAMMATIKOU received the engineering and Ph.D. degrees from the National Technical University of Athens (NTUA), in 1996 and 2001, respectively. She is currently a Senior Researcher and a Teaching Fellow with the School of Electrical and Computer Engineering, NTUA. Her research interests include future internet architectures with an emphasis on 5G/6G networking, network security, online education, and cloud computing.

VASILIS MAGLARIS received the engineering degree from the National Technical University of Athens (NTUA), Athens, Greece, in 1974, and the Ph.D. degree from Columbia University, New York, NY, USA, in 1979. From 1979 to 1989, he held industrial and academic positions in the USA in advanced electronic communications. From 1981 to 1990, he was with the Faculty of EE/CS, Polytechnic University—now NYU Tandon/Poly, Brooklyn, NY, USA. From 1990 to 2019, he was a Professor with the School of Electrical and Computer Engineering, NTUA, teaching and performing research with the Network Management and Optimal Design (NETMODE) Laboratory that he established. He was responsible for the development of the NTUA Campus LAN and of GRNET, the National Research and Education Network (NREN), Greece. From 2004 to 2012, he was the GÉANT Policy Committee Chairperson, the governance body of the advanced internet serving the 37 NRENs of the extended European Research Area. From July 2012 to June 2013, he was the General Secretary for Research and Technology appointed by the Greek coalition government. In 2020, the NTUA Senate conferred upon him the title of Professor Emeritus, enabling him to continue his teaching and research activities beyond his retirement.