Machine Learning techniques for Behavioral Feature Selection in Network Intrusion Detection Systems

Vicente Martinez[1], **Rodrigo Salas**[2], **Oliver Tessini**[3], **Romina Torres**[3]

[1] Instituto de Estadística, Facultad de Ciencias, Universidad de Valparaíso, Chile
[2] Escuela de Ingeniería C. Biomédica, Universidad de Valparaíso, Chile
[3] Facultad de Ingeniería, Universidad Andres Bello, Chile

Corresponding author: rodrigo.salas@uv.cl

Keywords: Machine Learning, Feature Selection, Intrusion Detection, UNSW-NB 15

Abstract

Information systems are prone to receiving multiple types of attacks over the network. Therefore, Network Intrusion Detection Systems (NIDSs) analyze the behavior of the network traffic to detect anomalies and eventual cyberattacks. The NIDS must be able to detect these cyberattacks in an efficient and effective manner based on a set of features where it is expected that the performance depends on both the selected features and the machine learning technique used. The main goal of this work is to identify the most relevant characteristics required to detect, with a high sensitivity and precision, between normal traffic and a network intrusion, together with the most relevant features associated to the identification of a specific type of attack. In this work, a comparative study of different decision tree-based machine learning techniques combined with several feature selection techniques in order to accomplish the goal. Random Forest and the XGBoost achieved a performance that reaches up to 98.5% in the F-measure when the complete set of features were used. Results show the performance was just slightly reduced to 98% when the 10 most relevant features were used. Moreover, results also show that the model using only the 10 most relevant features was able to separately identify the type of attack with a performance of at least 90% in the F-measure. We conclude that it is possible to obtain and rank a subset of the most relevant features that characterize the intrusion pattern in the network traffic in order to support the decision of how many features to include during runtime under a real network environment.

1 Introduction

Cyberattacks on Information Systems may compromise some of the three essential aspects of information security: confidentiality, integrity and availability of data [1]. This could negatively affect an organization, putting business continuity at risk, because many of those threats such as a denial of service (DoS) directly affect availability. A Network Intrusion Detection System (NIDS) uses a user behavior model to detect anomalies and eventual cyberattacks. The most frequent types of attacks are: denial of service, brute force, penetration attacks based on browsers that exploit for example vulnerabilities such as "cross-site scripting", attacks made by remote execution by command line, SSL attacks based on intercepting encrypted data to gain benefits, a back attack left by applications that expose remote access, among others.

Cyberattacks area carried out in seven phases [2]: 1) reconnaissance, 2) weapons, 3) delivery 4) exploitation, 5) installation, 6) command and control, and, 7) action on the target. The first phase is used to obtain information details from the target, for example: type of technology used or the email list. Additionally, it seeks to generate an attack strategy for the following phases. In the second phase, with the information stolen from the previous phase, payloads or malicious codes are created to take advantage of any vulnerability or weakness in the system. The payload is sent to the victim in the third phase and executed to exploit the vulnerabilities in the fourth phase. In phase five the malware is installed on the victim's system. In phase six, it is possible to take control of the victim's computer and steal confidential information. Finally, in the seventh phase, actions are taken on the target computer, either stealing information or leaving "botnets" to extract passwords or images, or even more to activate cameras or microphones.

A network intrusion detection system (NDIS) is used to detect malicious or anomalous activities in the network [3], where methodologies based on detecting anomalies in the normal behavior of the user or network traffic in general are highlighted. In particular, the methodologies to detect anomalies are carried out in two ways; one of them is through rules that look for patterns or signatures established within the analyzed traffic; the other is through an analysis of the user behavior, that is, any deviation from the normal profile of behavior is considered a potential attack [4]. The behavior profile of a user can be obtained based on different statistical indicators, such as: input, CPU use, use of inputs and outputs, document writing styles, among others [4]. Two phases are usually considered: the learning phase during which the model is built and the detection phase during which the current behavior is compared

Category	Features
Flow Characteristic	*srcip*; *sport*; *dstip*; *dsport*; *proto*
Basic characteristics	*state*; *dur*; *sbytes*; *dbytes*; *sttl*; *dttl*; *sloss*; *dloss*; *service*; *sload*; *spkts*; *dpkts*
Content Features	*swin*; *dwin*; *stcpb*; *dtcpb*; *smeansz*; *dmeansz*; *trans_depth*; *res_bdy_len*;
Time Characteristics	*sjit*; *djit*; *stime*; *ltime*; *sinkpkt*; *dinpkt*; *tcprtt*; *synack*; *ackdat*;
General pourpose features	*is_sm_ips_ports*; *ct_state_ttl*; *ct_flw_http_mthd*; *is_ftp_login*; *ct_ftp_cmd*;
Connection features	*ct_srv_src*; *ct_srv_dst*; *ct_dst_ltm*; *ct_src_ltm*; *ct_src_dport_ltm*;*ct_dst_sport_ltm*; *ct_dst_src_ltm*
Output	*attack_cat*; *label*

Table 1. Types of Features extracted from Network Traffic and stored in the UNSW-NB-15 dataset.

with the normal model. Given privacy concerns, most models are based on system behavior rather than user behavior using features extracted from the traffic such as: protocols used, destination and duration of the connection, frequency, among others.

In general, a NIDS can be evaluated with metrics such as detection rate (which is defined as the ratio of malicious vectors detected and the actual number of malicious vectors), the incorrect detection rate (which percentage of malicious vectors miss-classified attacks over the total), performance (which measures the ability of the system to be able to act on network traffic without packet loss), completeness (defined as the ability to detect all attacks that compromise the network), among others. Irrelevant features are the main reason of false alarm and low detection rates [5]. Different techniques have been proposed to select the relevant ones during the last decade: [6] gives a complete survey about feature selection for reducing the data dimension without compromise the completeness or the performance in real environments. A ranking of eleven features for two datasets [5] was obtained using an hybrid method based on the central points of attribute values and association rule mining techniques, showing that were enough to reach similar normal/abnormal detection rate (when all the features were used) but only using a part of the processing time.

Different from previous works, in this article, we determine which are the ten most relevant features to be considered in order to maintain or improve the performance of the classifier for each type of attack. The rest of this work is divided as follows. Section 2 describes Material and Methods used to select the features. Section 3 discusses the results of our proposal. Section 4 presents the conclusion and we give some future works.

2 Materials and methods

In this section we provide a description of the data set used and how it was created. In addition, the machine learning methods are explained. These methods were implemented in python using the Scikit-Learn [7] and the XGBoost [8] toolboxes.

2.1 Description of the Data Set

Moustafa et al. [3] have published the UNSW-NB-15 dataset[1] where raw network packets where created by using the IXIA PerfectStorm tool in the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS) for generating a hybrid of real modern normal activities and synthetic contemporary attack behaviours. During two days, several attacks scenarios were generated in a controlled and monitored environment. The number of records in the training set is 175.341 records and the testing set is 82.332.

The UNSW-NB-15 dataset has nine types of attacks (Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms) where 42 numerical features and 5 categorical ones were gathered. Moreover, two additional output features were added: normal/attack, and the type of attack if applies. Table 1 shows the selected 47 features grouped in 5 categories besides of the output category with the two output variables (See [9] for further details).

2.2 Machine Learning Techniques

Machine learning techniques consist of a set of highly parameterized and non-linear data-driven models with the capacity of having a good generalization performance, i.e., the machine learning method has a good performance for new samples that were not used during the training phase. In this article, we considered the decision tree-based machine learning techniques that are listed below:

1. *Decision Tree* [10]: Is a machine learning technique whose architecture consists of a tree data structure, where each node is a test for an input variable, and the outcomes or results of this test are branches or edges that connects to another node. The algorithm generates a recursive partition of the input space that ends in a leaf with the final classification decision.

2. *Gradient Boosting* [11]: Is an ensemble of weak prediction models such as decision trees. The model is constructed in sequential stages using a boosting scheme to reduce the errors of the hard samples.

3. *Random Forests* [12]: Is an ensemble of randomized decision trees constructed with the Bootstrap Aggregating technique (Bagging), where the training sets are generated with random samples with replacements.

4. *XGBoost*[8]: Is a special variant of the Gradient Boosting that optimize the mechanism of finding the best feature split and to generate a new branch,. This technique

[1] UNSW-NB-15 dataset: https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/

uses the distributions of the feature to reduce the search space of possible feature splits.

The decision tree-based machine learning techniques belongs to a family of models known as explainable machine learning models, where these models seek to understand the mechanism of how the inputs are mapped directly to the output space [13]. Moreover, these models are able to evaluate the contribution of the input elements, thus guaranteeing that the outputs are generated by a causality with the significant features [14]. Explainable machine learning models have been successfully applied in different areas such as fake-news detection [15], traffic accidents prediction [16] and rainfall-runoff modeling [17], just to name a few.

2.3 Feature Selection Techniques

The quality of the predictions of the machine learning algorithms depends mainly on the number of samples and the amount of relevant information contained in each variable to describe the phenomenon of interest [18]. The main stages of a feature selection process are: generation of subsets, evaluation of subsets, measurement of the quality level of the solution based on each subset, stopping criteria based on an evaluation function. Selecting only the relevant inputs will reduce the level of complexity of the generated data models [19].

In this work, the performances of the following feature selection methods are evaluated:

1. KBest-Chi2 (`chi2`): This technique applies the chi-squared test to select the K features with the highest score obtained by computing the chi-squared statistics. The chi-squared test measures the dependence between the stochastic variables and it is used to determine whether there is a statistically significant difference between the frequencies of two or more categories of a contingency table. The chi2 statistic is given by:

$$X^2 = \sum_{i}^{n} \sum_{j}^{m} \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \sim \chi^2_{(n-1)(m-1)} \qquad (1)$$

   where O_{ij} are the observed values and E_{ij} are the expected values. χ^2 is the chi-squared distribution with $(n-1)(m-1)$ degrees of freedom.

2. Recursive Feature Elimination (`RFE`): The RFE is a wrapper-type feature selection algorithm, where the machine learning method is initially trained with the whole set of features. Afterwards, recursively, the least important features are pruned from the set of features until the desired amount of features is reached.

3. Feature Importance (`FI`): is a technique that assigns a score to input characteristics based on how useful they are at predicting a target variable. Most decision trees-based machine learning are able to compute the feature importance score based on the Gini index [12, 20]:

$$I_G(\theta) = \sum_{T} \sum_{\tau} \Delta i_\theta(\tau, T) \qquad (2)$$

   where $\Delta i_\theta(\tau, T)$ is the decrease in the Gini impurity given by the optimal split of the node τ of the binary tree T. The Gini feature importance $I_G(\theta)$ indicates how often a particular feature θ was selected for a split and what is its discriminative value.

4. Hierarchical Clustering with Feature Importance (`HCFI`): this method combines the Ward's Hierarchical Clustering [21] and the Feature Importance techniques in sequential order. The hierarchical clustering is applied to the Spearman rank-order correlations using the Ward's minimum variance linkage on a condensed distance matrix. From each cluster, a single feature is selected as a representative characteristic. Afterwards the `FI` is applied to rank and select the most relevant features for predicting the target variable.

3 Results

3.1 Visualization of the dataset

In order to be able to visualize the highly dimensional dataset, a projection method called *t-distributed stochastic neighbor embedding* (t-SNE) was applied [22]. The t-SNE is an unsupervised nonlinear technique mainly used to improve the visualization regarding how a multiclass dataset is organized.

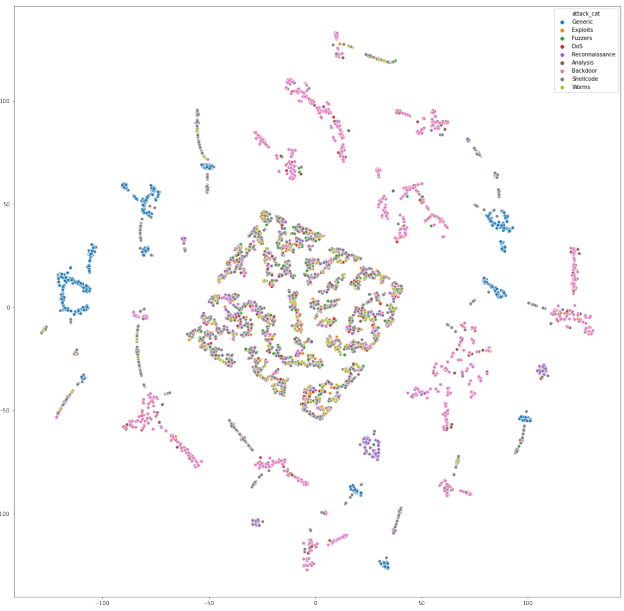

Figure 1. Two-dimensional projection of a subsample of the original dataset using the t-SNE method, where the colors symbolize a different type of attack.

Figure 1 shows the projection according to the t-SNE technique by preserving the topological structure of the data. It can be seen several data fragments in the peripheral region of the

chart that belong to the same type of attack, this implies that the machine learning technique will be able to detect and separate well this group of data. However, in the central region, the data is very mixed, thus, the classification task will be more difficult.

3.2 Feature Selection for Intrusion Detection

In this section, we perform a comparative study of the decision tree-based machine learning techniques combined with different feature selection techniques for intrusion detection. To evaluate the performance we use the F1-measure consisting of the harmonic average between precision and recall.

$$Accuracy = \frac{TP+TN}{n}; \quad Precision = \frac{TP}{TP+FP}$$
$$Recall = \frac{TP}{TP+FN}; \quad F1 = 2\frac{Precision \cdot Recall}{Precision+Recall}$$

where n is the number of samples, TP is the true positive, FN is the false negative and FP is the false positive. Recall and sensitivity are used interchangeably.

Classifier Method		Feature Selection Method (Best 10 Features)			
	All	FI	HCFI	RFE	Chi2
Decision Tree	0.976	**0.975**	0.956	**0.975**	0.945
Gradient Boost.	0.963	**0.957**	0.927	**0.957**	0.911
Random Forest	0.984	**0.981***	0.962	0.980	0.954
XGBoost	0.985	0.947	0.942	**0.957**	0.948

Table 2. Evaluation of the F1 score for four different classifiers combined with different feature selection methods. Each classifier was trained with all the features and compared with the classifiers implemented with a selection of the best 10 characteristics.

Table 2 shows the resulting performance using the F1 score for the machine learning methods combined with the feature selection techniques. The results show that the *Random Forest* and the *XGBoost* achieved a performance that reaches up to 98.5% in the F1-measure when the complete set of features were used, however by selecting the 10 most relevant features the performance is slightly reduced to 98.1% for the Random Forest.

Table 3 shows the ranking of features obtained for the Random Forest and XGBoost by considering the complete set of features (All, 42 features), and by selecting 10 features with the FI, HCFI and RFE; together with the features selected using the KBest-Chi2 and those found by Moustafa in [5]. The selected features varies depending on both the machine learning and the feature selection methods, only the *ct_dst_src_ltm, ct_srv_src, dttl, ct_state_ttl, response_body_len, sload*, and *state* appears selected by at least 4 of the 8 feature selection techniques. There is a reduced number of variables that were not selected by any of the methods: {*dloss; dmean; dpkts; dur; is_ftp_login; trans_depth*}. In [5] found these two features relevants: *dwin* and *synack*, and they were not detected by techniques tested in this work. The KBest-Chi2 found the *dload,*

Feature Name	Classifier and Feature Selection Method: Random Forest FI.	Random Forest HCFI	Random Forest RFE	XGBoost FI	XGBoost HCFI	XGBoost RFE	Chi2	[5]
proto						4		
state	1			6		1		X
sbytes			4				7	
dbytes	3						6	
rate			7				5	
sttl			1			5		X
dttl		10		7	1	6		X
sloss						8		
service					5	7		
sload	4		5	3			3	
dload							4	
spkts	7			2				
swin						3		X
dwin								X
stcpb							2	
dtcpb							1	
smeansz			8					
response_body_len	5	1			7		10	
sjit		6			10		9	
djit				9				X
sinpkt					5	1	8	
dinpkt		4						
tcprtt	8		6					
synack								X
ackdat	9	5						
is_sm_ips_ports				10				
ct_state_ttl			3	4		10		X
ct_flw_http_mthd		7			8			
ct_ftp_cmd	10							
ct_srv_dst		2	9		9			X
ct_srv_src	2	3		1	2			
ct_dst_ltm				5	4			
ct_src_ltm			10	8				X
ct_src_dport_ltm	6							
ct_dst_sport_ltm		9			3	2		X
ct_dst_src_ltm		8	2		6	9		
Non-selected:	dloss; dmeansz; dpkts; dur; is_ftp_login; trans_depth							

Table 3. Ranking of the 10 most relevant characteristics obtained using four different feature selection methods and two machine learning classifiers: Random Forest and XGBoost. Features with lower values are more relevant than those with higher values. Empty space means that the feature was not selected.

dtcpb and *stcpb* relevant, meanwhile for the machine learning-based feature selection methods they are not relevants. The Random Forest with the Feature Importance shows the best performance results (98.1%).

3.3 Feature Selection for a type of attack

In this section we will explore by a simulation study which are the most relevant features for the identification of each type of attack. To this aim, the dataset was separated into several subsets, where in each one all the samples from a specific attack were considered and an equal amount of data from other types of attacks were randomly selected. Each data set was separated into a training set and a test set, where a binary classifier was adjusted with the training set to detect the attack of interest. Due to the results obtained by the classifiers in the detection of an attack event versus a normal one, the *Random Forest* classifier was chosen, since it was the one that showed the best results according to the Table 2.

Table 4 shows the results obtained by comparing the perfor-

mances according to the F1-measure obtained by the Random Forest classifier according to the type of technique of selection of characteristics used. The table shows the performances obtained by considering all the features (All, 42 features), and by selecting 10 features with the FI, HCFI, RFE and Chi2 techniques. As it was expected, the best set of features and best feature selection technique vary depending on the type of attack. In most cases, the feature selection technique that presented the best results was FI (6 cases), followed by RFE (3 cases). It is interesting to note that in 6 cases out of 9 (Analysis, Backdoor, Exploits, Fuzzers, Reconnaissance and Shellcode) the performance of the classifier improved when was made with the subset of features selected. On the other hand, in two cases (DoS and Generic) the techniques presented similar results.

Attack	All 42	FI 10	HCFI 10	RFE 10	Chi2 10
Analysis	0.927	**0.929**	0.926	0.928	0.910
Backdoor	0.948	**0.952**	0.944	0.944	0.911
DoS	0.902	0.895	0.901	0.900	0.901
Exploits	0.900	**0.902**	0.900	0.901	0.899
Fuzzers	0.935	**0.938**	0.933	0.937	0.911
Generic	0.999	0.998	0.998	0.998	0.998
Reconnaissance	0.937	**0.942**	0.936	0.938	0.938
Shellcode	0.949	0.958	0.957	**0.972**	0.926
Worms	0.957	0.936	0.926	0.946	0.907

Table 4. Performance results of the classifiers for each type of attack. Evaluation of the *F1-score* for the *Random Forest* classifier combined with different feature selection methods. The classifier was trained with all the features and compared with the selection of the best 10 features.

Afterwards, for each type of attack, the classifiers together with the feature selection technique that obtained the best results were selected. With these selected machine learning technique, a ranking of the 5 most relevant features were made for each type of attack, where the importance were measured by using the Feature Importance metric of the trained Random Forest classifier. Table 5 shows these selected features. The first thing that is observed is that the set of most relevant characteristics is different according to the type of attack. For example, to study the Analysis attack it is suggested to observe the features: *smean, service, sbytes, proto* and *ct_srv_dst*. An interesting aspect that can be seen in the table is the appearance of the features of *dur, dmean* and *dtcpb* as relevant to detect attacks such as *DoS, Exploits* and *Worms* respectively, but which, however, were not identified as relevant features to detect an attack vs. normal traffic (see table 3). Another relevant aspect to observe is the importance of some features to detect types of attacks, where for example *sbytes* is important for all the attacks, *smean* is relevant in 8 of the 9 attacks, *service* in 6 of 9 attacks, and *proto* in 4 of 9 attacks.

4 Conclusion and Future Work

It is not a easy task to understand when the network is under attack. Even more, if several features must be observed. Therefore, identifying the relevant characteristics could help to gain understanding of the network behavior model when is under attack or under a particular kind of attack. In this work we present a subset of only 25% of the features that should be observed in order to 1) detect an intrusion and 2) classify the type of attack in order to create a defense response. We show that using a specific subset of features by each model we achieved almost the same performance results.

As future work, it is expected to compare these results with the results obtained with deep learning methods which can automatically select the characteristics. Furthermore, it is expected to build a prototype of a semi-autonomous IDS with self-adaptation capability to monitor the obsolescence of the models, and to determine at runtime when to retrain and replace the models without the need to restart the IDS.

Acknowledgements

The work of Romina Torres has been partially supported by the grant DI-02-19/R of the Universidad Andres Bello, Chile.

References

[1] N. Moustafa and J. Slay, "The significant features of the UNSW-NB15 and the KDD99 data sets for network intrusion detection systems," in *2015 4th International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS)*, (Los Alamitos, CA, USA), pp. 25–31, IEEE Computer Society, nov 2015.

[2] F. Garba, "The anatomy of a cyber attack: Dissecting the cyber kill chain," *Scientific and practical cyber security journal*, 03 2019.

[3] N. Moustafa and J. Slay, "The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," *Information security journal: A global perspective*, vol. 25, no. 1-3, pp. 18–31, 2016.

[4] J. Peng, K.-K. R. Choo, and H. Ashman, "User profiling in intrusion detection," *J. Netw. Comput. Appl.*, vol. 72, pp. 14–27, Sept. 2016.

[5] N. Moustafa and J. Slay, "A hybrid feature selection for network intrusion detection systems: Central points," *arXiv preprint arXiv:1707.05505*, 2017.

[6] R. Abdulhammed, H. Musafer, A. Alessa, M. Faezipour, and A. Abuzneid, "Features dimensionality reduction approaches for machine learning based network intrusion detection," *Electronics*, vol. 8, no. 3, p. 322, 2019.

[7] 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," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

	Ranking of Features				
Type of Attack	1st	2nd	3rd	4th	5th
Analysis	smean	service	sbytes	proto	ct_srv_dst
Backdoor	service	smean	sbytes	sload	proto
DoS	smean	sbytes	ct_srv_src	**dur**	rate
Exploits	sbytes	smean	proto	sttl	**dmean**
Fuzzers	ct_srv_dst	sbytes	service	smean	ct_srv_src
Generic	sbytes	smean	service	ct_dst_sport_ltm	proto
Reconnaissance	ct_src_dport_ltm	sbytes	smean	ct_dst_src_ltm	ct_srv_dst
Shellcode	ct_src_dport_ltm	service	sbytes	dbytes	ct_dst_src_ltm
Worms	sbytes	smean	service	ct_srv_src	**dtcpb**

Table 5. Ranking of the 5 most relevant characteristics obtained for each type of attack by using the Random Forest classifier and the best feature selection method.

[8] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, (New York, NY, USA), pp. 785–794, ACM, 2016.

[9] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 military communications and information systems conference (MilCIS)*, pp. 1–6, IEEE, 2015.

[10] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Wadsworth International Group, 1984.

[11] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of Statistics*, vol. 29, pp. 1189–1232, 2000.

[12] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[13] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM computing surveys (CSUR)*, vol. 51, no. 5, pp. 1–42, 2018.

[14] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, *et al.*, "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information Fusion*, vol. 58, pp. 82–115, 2020.

[15] E. Puraivan, E. Godoy, F. Riquelme, and R. Salas, "Fake news detection on twitter using a data mining framework based on explainable machine learning techniques," *11th International Conference on Pattern Recognition Systems*, pp. 1–6, 2021.

[16] C. Parra, C. Ponce, and R. Salas, "Evaluating the performance of explainable machine learning models in traffic accidents prediction in california," *39th International Conference of the Chilean Computer Science Society (SCCC)*, pp. 1–8, 2020.

[17] Y. Morales, M. Querales, H. Rosas, H. Allende-Cid, and R. Salas, "A self-identification neuro-fuzzy inference framework for modeling rainfall-runoff in a chilean watershed," *Journal of Hydrology*, vol. 594, p. 125910, 2021.

[18] S. Chabert, T. Mardones, R. Riveros, M. Godoy, A. Veloz, R. Salas, and P. Cox, "Applying machine learning and image feature extraction techniques to the problem of cerebral aneurysm rupture," *Research Ideas and Outcomes*, vol. 3, p. e11731, 2017.

[19] A. Veloz, R. Salas, H. Allende-Cid, H. Allende, and C. Moraga, "Identification of lags in nonlinear autoregressive time series using a flexible fuzzy model," *Neural Processing Letters*, vol. 43, pp. 641–666, June 2016.

[20] B. H. Menze, B. M. Kelm, R. Masuch, U. Himmelreich, P. Bachert, W. Petrich, and F. A. Hamprecht, "A comparison of random forest and its gini importance with standard chemometric methods for the feature selection and classification of spectral data.," *BMC Bioinform.*, vol. 10, 2009.

[21] J. Ward, "Hierarchical grouping to optimize an objective function.," *Journal of the American Statistical Association*, vol. 58, pp. 236–244, 1963.

[22] L. v. d. Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.