A cybersecurity data science: Machine learning in IoT network security

5 authors, including:

Lyubomir Gotsev
University of Library Studies and Information Technologies
12 PUBLICATIONS **5** CITATIONS
SEE PROFILE

Milena B. Dimitrova
University of Library Studies and Information Technologies
1 PUBLICATION **2** CITATIONS
SEE PROFILE

Boyan Jekov
University of Library Studies and Information Technologies
29 PUBLICATIONS **23** CITATIONS
SEE PROFILE

Eugenia Kovatcheva
University of Library Studies and Information Technologies
94 PUBLICATIONS **183** CITATIONS
SEE PROFILE

A Cybersecurity Data Science Demonstrator: Machine Learning in IoT Network Security

Lyubomir GOTSEV
“Computer Sciences” Department, University of Library Studies and Information Technologies
Sofia, 1784, Bulgaria

Milena DIMITROVA
“Computer Sciences” Department, University of Library Studies and Information Technologies
Sofia, 1784, Bulgaria

Boyan JEKOV
“Computer Sciences” Department, University of Library Studies and Information Technologies
Sofia, 1784, Bulgaria

Eugenia KOVATCHEVA
“Computer Sciences” Department, University of Library Studies and Information Technologies
Sofia, 1784, Bulgaria

Elena SHOIKOVA
“Computer Sciences” Department, University of Library Studies and Information Technologies
Sofia, 1784, Bulgaria

ABSTRACT

The punctilious understanding of where and how Data Science creates a value-added in IoT network security lies in the applied experimental session evaluating the performance of particular machine learning models for attack detection. Results stand as a base demonstrating the benefits of the emerging technologies integration for predicting threats issues. Furthermore, implementing machine learning to intelligent security systems deepens the need for a multi-disciplinary approach and data e-infrastructure to manage the whole lifecycle (Software Engineering end-to-end, including ML and Data DevOps).
Comparative performance analysis of the algorithms that have proven helpful in mitigating security in IoT domains such as Support Vector Machines, Random Forrest, Naïve Bayes, Logistic Regression, Decision Tree is presenting. The case study is accomplished by conducting experiments with the public available IoT-23 dataset containing labeled information of malicious and benign IoT network traffic. The benign scenarios were obtained from original hardware and not simulated. That allowed to be analyzed real network behavior. As a result, models produce accurate outputs usable to predict and detect vulnerabilities in IoT-based systems. Besides, the lab could be multiplicate for creating business and industrial demonstrators to present the advantages of developing intrusion detection tools featuring machine learning algorithms.

Keywords: Cybersecurity Data Science, Machine Learning, IoT Network Security, Business Demonstrator.

1. INTRODUCTION

IoT & Cybersecurity

EU Agency for Cybersecurity (ENISA) defines IoT as “a cyber-physical ecosystem of interconnected sensors and actuators, which enable intelligent decision making.”[1]. The novel paradigm supports digital transformation, tending industry to become more competitive and efficient by improving processes and developing innovative products or services. In other words, the concept transforms the business world and the way we live. That explains why IoT devices and networks have rapidly increased over the last decade. But, the more technology interest and impact, the more security risks and unwanted attention attract. The number of threats and attack scenarios affecting IoT networks is constantly growing. Safety implications are pertinent for such a broad ecosystem of interconnected services, devices, smart objects, and components. The security concerning landscape enlarges, including still immature ecosystem bearing a fragmentation of standards in a non-homogeneous IoT market with a promising ability to scale globally. Vulnerabilities of low-protected devices could be used for an attack vector against other critical infrastructures. Therefore, safety depends on protecting all actors involved, such as the devices themselves, cloud/edge services, communications, applications, and maintenance. Ensuring minimization of the addressed challenges is a high priority for Cybersecurity in IoT systems to increase user trust in their services.

Data-driven Paradigm (DDP) & Cybersecurity

The current pace of engineering development is exerting profound changes in economics, industries, science, and innovation consolidated around another novel paradigm: Big Data. Indeed, IoT is one of the sets of technology drivers for Big Data and both ecosystems interact and enhance each other. The need to unlock insights from D-Ocean and transform information into unexpected or undiscovered knowledge, giving new perspectives, predictions, or value, is the main objective in Data Science. It encompasses the knowledge discovery domain, enlacing Artificial Intelligence (incl. Machine & Deep Learning) and Advanced Analytics to extract functional patterns from Big Data. Exact that principle interlinks with security data for quantifying cyber risks and optimizing cybersecurity operations. It explains the current case study idea to develop, examine and describe data-driven cybersecurity models using machine learning techniques for anomaly detection in IoT networks.

2. IoT CYBERSECURITY

Policies and Regulatory Initiatives

The European initiative is summarized in Baseline Security Recommendations for IoT [1]. The goal of this report produced by ENISA is to identify and analyze existing IoT security practices, guidelines, relevant industry standards, and research initiatives in IoT security for Critical Information Infrastructures (e.g., Industry 4.0, Machine-to-Machine communications, IoT updatability). Particular emphasis is given to existing EU policies and regulatory drives such as the Directive on Security of Network and Information Systems (NIS Directive), the EU General Data Protection Regulation (GDPR), the Internet of Things – An Action Plan for Europe, as well as the work of the Alliance for Internet of Things (AIOTI) and the Staff Working Document on ICT Standardization [1].

In response to a series of IoT-related cyber-attacks in 2016, the "Internet of Things Cybersecurity Improvement Act of 2017" is introduced by US senators. The main goal is to establish minimum cybersecurity requirements for connected devices.

Another significant research in the field is the NIST Framework for Improving Critical Infrastructure Cybersecurity [2]. The focus is primarily on information technology (IT), industrial control systems (ICS), cyber-physical systems (CPS), or connected devices, more generally, including the Internet of Things (IoT).

The few trials given underline the joint efforts to establish good practices and measures in the IoT security domain. Nevertheless, fragmentation of regulations and policies still stands as a complicated concern among various challenges.

Security Considerations

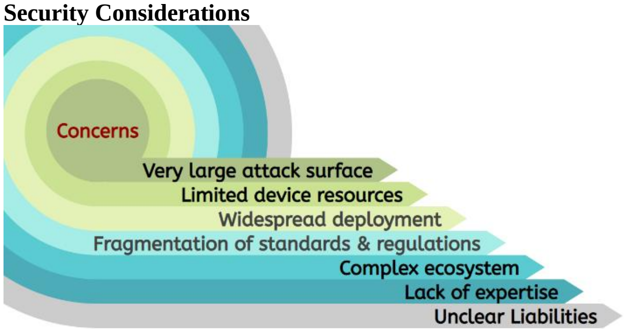


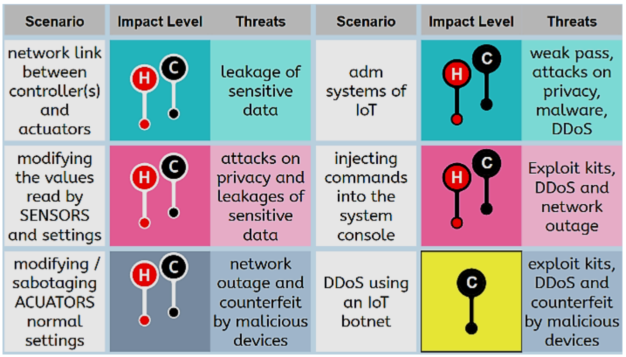
Figure 1. IoT Security Considerations

Although the main obstacles are already discussed, others are visualized in the diagram; further issues retard safety consolidation in IoT ecosystems. Time and budget pressure the development process to shorten the life cycle and pay more attention to functionality than to develop so-called security by design. Integration and updating the infrastructure, inherent complexity, heterogeneity and scalability count security challenges.

Critical IoT Attack Scenarios

Unauthorized access, malware, zero-day attack, data breach, denial of service (DoS), social engineering, or phishing have increased exponentially in recent years. ENISA [1] defines the potential impacts of different threats based on IoT experts' interviews, ranging from low through medium to high and critical. Various security concerns regarding IoT are inherited from the networking domain and are not entirely novel. Thus, the discussion focuses on the main security threats and the most critical attack scenarios (non-exhaustive listing) that affect IoT devices and networks and those closely related to the experimental session. The following figure illustrates some of the most critical methods, including potential impact levels and threats.

Table 1. Sixth of the most critical IoT attack scenarios



Scenario	Impact Level	Threats	Scenario	Impact Level	Threats
network link between controller(s) and actuators	H C	leakage of sensitive data	adm systems of IoT	H C	weak pass, attacks on privacy, malware, DDoS
modifying the values read by SENSORS and settings	H C	attacks on privacy and leakages of sensitive data	injecting commands into the system console	H C	Exploit kits, DDoS and network outage
modifying / sabotaging ACUATORS normal settings	H C	network outage and counterfeit by malicious devices	DDoS using an IoT botnet	C	exploit kits, DDoS and counterfeit by malicious devices

High and critical impact levels are marked with H & C resp. In case the level varies from high to critical, both letters are noted. Ransomware, power source manipulation, exploitation of vulnerabilities in reading data, stepping stone attacks, exploiting protocol vulnerabilities are other severe examples with medium to high potential for compromising and damaging the ecosystem. Among IoT attack scenario criticality, ENISA identifies the top three: IoT administration system compromise, value manipulation in IoT devices and botnet/commands injection. Based on Mirai and Okiru malware, the last scenario is of particular attention as a target for the machine learning approach in IoT network security.

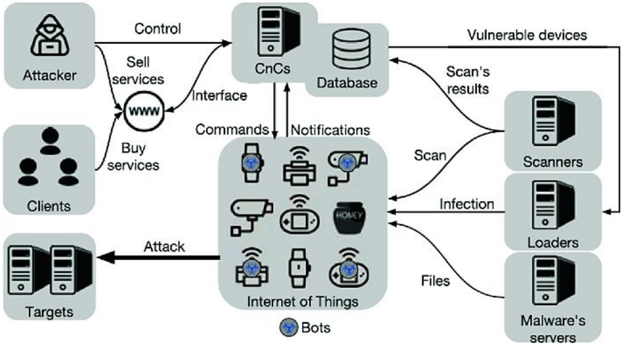


Figure 2. IoT botnet [3]

Mirai and Okiru turn networked devices into remotely controlled bots that can be used as part of a botnet in large-scale network attacks. The figure above illustrates the principal botnet mechanism. That scenario has a high impact level, critical cascade effect risk, high recovery effort, and complex detection. Therefore, for supporting and facilitating the detection process in IoT security, an advanced data-driven approach is very appropriate.

3. CYBERSECURITY DATA SCIENCE (CSDS)

Cross-Domain

An advanced data-driven approach to security issues refers to the umbrella term Cybersecurity Data Science. A few words about the knowledge discovery paradigm are relevant in the context of the Big Data phenomenon mentioned earlier.

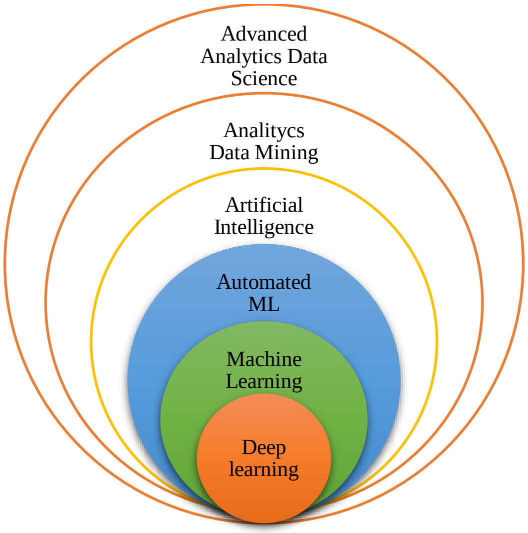


Figure 3. Knowledge Discovery Paradigm

Synergetic applying different tactics determine the significant relationship between the various research fields to find an optimal solution to a complex problem. For the current study, such a problem is defined in the Cybersecurity domain.

Each field has its role, as a part of the whole, in the novel value chain. Simulation of different aspects of human intelligence exhibited by machines exploits in the Artificial Intelligence domain, where Machine Learning plays a role of a model-fitting approach to achieve it. Deep Learning refers to a technique focused on multilayer neural network models for implementing ML in complex data over massive datasets. Analytics are grounded on the systematic processing and manipulation of data to uncover patterns, relationships between data, historical trends, and attempts at predicting future behaviors and events. An analysis is more related to inspecting, cleansing, transforming and modeling data. Data Mining refers to the intersection of machine learning, statistics, and database systems for extracting meaningful information from large amounts of data. These fields described the so-called Advanced Analytics as an umbrella term. Finally, data Science encompasses (all above) a compendium of principles, methods, and processes for extracting non-obvious and useful patterns from large datasets. That explains the significance of applying such a method for the detection of security threats.

Concept

Incorporating data science methods and behavioral analytics for various security incidents and intelligent decision-making in real-world applications stands behind Cybersecurity Data Science's concept [4].

Practice & Challenges

Some general aspects of CSDS as an emerging professional practice [4] can be summarized:

CSDS is data-focused, applies quantitative, algorithmic and probabilistic methods, attempts to quantify risk, focuses on producing and efficacious alerts, promotes inferential practices to categorize behavioral patterns, and ultimately optimizes cybersecurity operations.

The development of more rigorous scientific methods is one of the central challenges. In addition, the advancement of the CSDS domain demands the development of best practices resulting from experimentation, testing, and core research. Other challenges are data management: gathering, integrating, cleansing, transforming, and extrapolating critical measures from the fragmented, voluminous, and fast-streaming sources that underlie modern cyberinfrastructure.

Machine Learning

Machine learning-based security modeling is the core step where insights and knowledge are extracted from data through cybersecurity data science.

The main tasks involved are classification, prediction, monitoring and detection. There are various methods, but most can be categorized in supervised, unsupervised learning and reinforcement.

Supervised and unsupervised machine learning techniques focus mainly on data analysis problems, while reinforcement is preferred for comparison and supporting decision-making. The technique selection also depends on the nature of the available data. In cases where the input data is labeled, controlled training is preferred. Unsupervised algorithms are used when the outputs are not well defined, and the system itself has to establish the similarities between data. This method is designed to identify patterns and links in the data that people would miss.

Machine and deep learning in cybersecurity use-cases such as spam filtering, phishing email detection, malware and virus detection, spoofing and scanning detection, advanced authentication, network monitoring, and endpoint protection have pragmatic successes.

4. EXPERIMENTAL CASE STUDY

The primary goal is to establish and evaluate experimental sessions on applying ML models for threat detection in IoT network traffic. The results would act as a starting point for developing a demonstrator representing Machine Learning application capabilities in intelligent security systems. In addition, demonstrators are appropriate tools illustrating what, why and how it works, the advantages & disadvantages of presented methods to comprehensive business and industry stakeholders. The experiments occur in the infrastructure of the University Data Science CoE.

Design & Methods

Objectives:

- Evaluating ML models for network-based anomaly detection in IoT systems
- Evaluation of models' performance on samples with balanced and unbalanced data distribution
- High-performance classifier identification
- Resource utilization evaluation
- Adoption of experimental workflow for demonstrator development

Context:

- ML algorithms for improving detection, analysis and exposure of anomaly detection in IoT networks
- IoT Cybersecurity

ML problem:

- Multi-class classification

Data:

- IoT-23 public dataset, light labeled version

Infrastructure:

- ULSIT Experimental Lab, 6 CPU, 16 up to 32 GB RAM, OS Windows Server 2019

Platform, language & libraries:

- Anaconda
- Python 3.8
- Scikit, NumPy, Pandas, Matplotlib, Seaborn, Psutil.

Tasks:

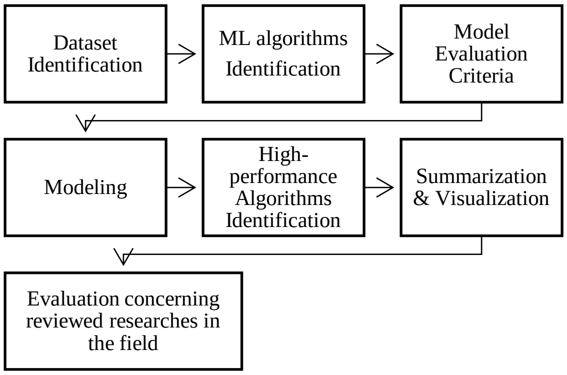

Dataset Identification
ML algorithms Identification
Model Evaluation Criteria
Modeling
High-performance Algorithms Identification
Summarization & Visualization
Evaluation concerning reviewed researches in the field

Figure 4. Tasks

Evaluation metrics:

- Accuracy, Precision, Recall, F1-score
- Resource utilization (Runtime, CPU, Memory)

Algorithm Families:

- Regression Algorithms, Decision Tree Algorithms, Ensemble Algorithms, Bayesian Algorithms, Instance-based Algorithms.

Classifiers

- Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), Gaussian Naïve Bayes (NB), Support Vector Machines (SVM).

Data

To achieve the research aim: study, apply and compare different machine learning models relevant for the timely detection of potential threats by monitoring IoT networks, a large dataset IoT-23 (published in January 2020, [5]) has been identified. The resource consists of labeled IoT network traffic for malicious and benign data created by the Avast AIC (Artificial Intelligence and Cybersecurity) laboratory. The set is designed for researchers to develop machine learning algorithms. IoT-23 has 20 malware captures/files (called scenarios) from various IoT devices and 3 captures (files) for benign IoT traffic.

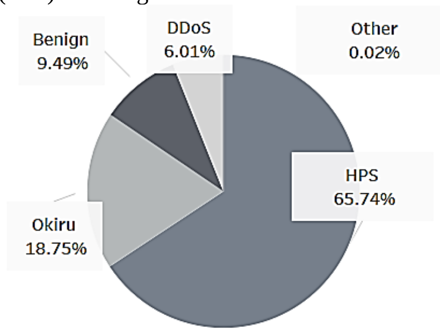


Figure 5. Distribution of main labeled flows

For each malicious scenario, a specific malware sample in a Raspberry Pi used several protocols and performed different actions. The benign scenarios were obtained of three IoT devices: a smart LED lamp, a home intelligent personal assistant, and a smart door lock. It is essential to mention that these IoT devices are real hardware and not simulated. That allowed to be analyzed real network behavior.

Each flow is representing by 21 features and 2 labels. The attributes are mixed in nature, with some being nominal, some numeric, and some taking on timestamp values.

The following tables represent detailed information about data attributes and labels.

Table 2. Attributes description

Attribute	Description	Type
ts	Timestamp of the capture.	int
uid	ID of the capture.	str
id_orig.h	Originating IP where the attack happened.	str
id_orig.p	Port used by the responder.	int
id_resp.h	IP address of the device on which capture happened.	str
id_resp.p	Port used from the response from the device on which the capture happened.	int
proto	Network protocol.	str
service	Application protocol.	str
duration	Duration of the transmission between device and attacker.	float
orig_bytes	Amount of data sent to the device.	int
resp_bytes	Amount of data sent by the device.	int
conn_state	State of the connection.	str
local_orig	? the connection originated locally.	bool
local_resp	Whether the response originated locally.	bool
missed_bytes	Amount of missed bytes in a message.	int
history	History of the state of the connection.	str
orig_pkts	Amount of packets sent to the device.	int
orig_ip_bytes	Amount of bytes sent to the device.	int
resp_pkts	Amount of packets sent from the device.	int
resp_ip_bytes	Amount of bytes sent from the device.	int
tunnel_parents	ID of connection if tunneled.	str
label	Type of capture, benign or malicious.	str
detailed-label	Type of the malicious capture	str

Table 3. Label explanation

Label	Description
Attack	An attack that cannot be identified.
Benign	Traffic that is not suspicious.
C&C	"Command and Control" type attacks take control of the device to perform various episodes in the future.
FD	The C&C server sends a file to the infected device
HB	This label indicates that packets sent on this connection track the infected host by the C&C server.
HB-A	The C&C server checks the status of the infected device while the verification method remains unidentified.
HB-FD	The C&C server checks the status of the infected device by sending small files.
Mirai	Connections have characteristics of a Mirai botnet. This label is added when the flows have similar patterns as the most commonly known Mirai attacks.
HPS	Information is gathered from a device for a future attack.
Torii	The Torii botnet performs the attack.
DDoS	The infected device is performing a DDoS attack.
Okiru	The Okiru botnet performs the attack.

Abbr: FD-File Download, HB-Heart Beat, A-Attack, HPS-HorizontalPortScan, DDS - denial of a service attack.

Scenarios

The experimental study has a specific setup: two samples corresponding to two scenarios differing in data distribution and the number of threats for multi-class classification.

The first sample includes 4 labeled classes - regular traffic (benign), DDoS, Okiru, and HorizontalPortScan, and for each category are included 5 million records. Thus, in this variant, all flows are equally represented in the data.

The second sample includes the complete set of labeled classes (16). The number of flows for each type is limited to 5 million, and not all categories have that much data. For classes with less data, all available ones are included. In this scenario, the data distribution is unbalanced.

Applying scenarios aims to compare the models' performance on both data distribution types.

Table 4. Samples

Abbr	# Files	# Labeled classes	# Records per class / Total flows
S4	4	4	5 000 000 / 20 000 000
S16	16	16 (all)	5 000 000 * / > 20 000 000
* If the file contains fewer items than specified, all ones are count			

Workflow

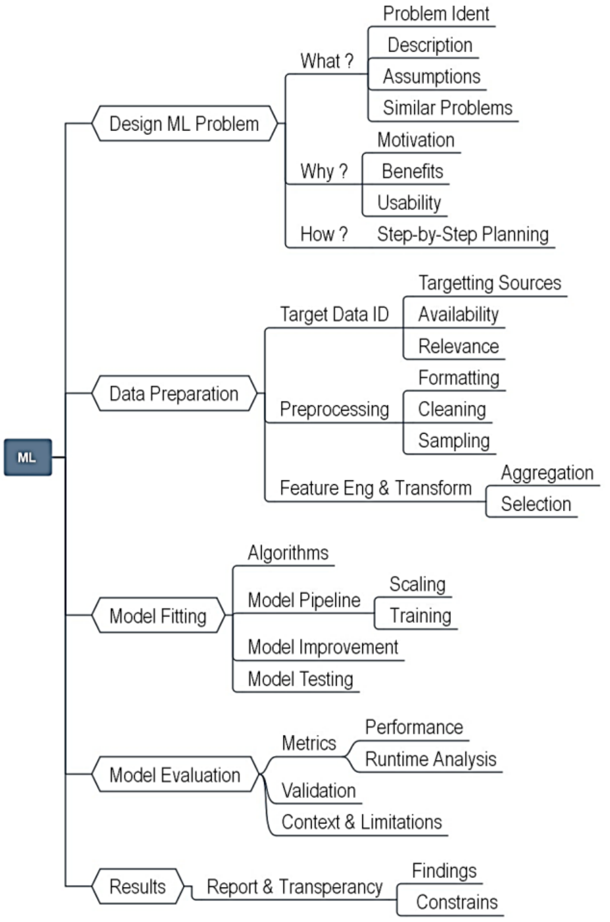


Figure 6. Experimental session workflow

Implementation

During the preparation and modeling stages, several comments are appropriate regarding the IoT-23 dataset.

Attribute "detailed-label" is the defined target variable showing the type of the malicious capture, while "label" refers to just the type of traffic (benign or malicious) and is removed. The missing value in the target variable marked as "-" were replaced with "benign". The unique identification of capture (uid) is unusable for network anomaly detection. A statistical correlation was applied to eliminate the data unrelated to the target variable, and two weakly related attributes, "ts" and "id.orig.h", were excluded. The other four attributes (local_orig, local_resp, missed_bytes, tunnel_parents) are considered minor due to many missing inputs. The correlation reveals insignificant differences in values between both samples (S4 & S16). Encoding of categorical features has been performed. Scaling is a step of model pipeline completion. At the end of described activities, features are reduced to fourteen.

Logistic Regression, Decision Tree, Random Forest, Gaussian Naïve Bayes and Support Vector Machines are used to build the ML models. They are implemented through a pipeline (scikit built-in method) in two steps: data scaling (standard scaler, min/max for improving SVM) and training the model.

5. RESULTS

Processing on a multi-class dataset requires adequate evaluation metrics: Accuracy, Precision, Recall, and **F1-score**. The weighted average version is appropriate for unbalanced sets. However, it makes more sense in the current session, analyzing two samples with balanced and unbalanced data distribution. Additionally, resource utilization metrics were calculated.

Each model is evaluated and discussed.

Table 5. Classifiers evaluation metrics in scenarios

		NB		SVM		LR		DT		RF	
		S04	**S16**	**S04**	**S16**	**S04**	**S16**	**S04**	**S16**	**S04**	**S16**
Accuracy		0.58	0.58	0.72	0.74	0.76	0.75	1.00	1.00	1.00	1.00
Precision	W	0.75	0.76	0.68	0.70	0.74	0.73	1.00	1.00	1.00	1.00
Recall	W	0.58	0.58	0.72	0.74	0.76	0.75	1.00	1.00	1.00	1.00
F1 Score	W	**0.51**	**0.48**	**0.68**	**0.70**	**0.73**	**0.72**	**1.00**	**1.00**	**1.00**	**1.00**
Runtime	sec	2.63	8.91	0.71	1.24	0.72	1.23	0.86	1.10	48.04	66.42
CPU	%	100	100	222.3	425.9	219.5	430.5	99.9	100	100	100
Memory	%	4.43	5.68	5.31	5.33	7.25	8.72	4.26	11.81	11.14	18.79

Classifiers

The Naïve Bayes classifier (NB) has the worst performance than the other models observed in both scenarios. However, the precision is significantly higher than the recall, which means fewer false-positive results.

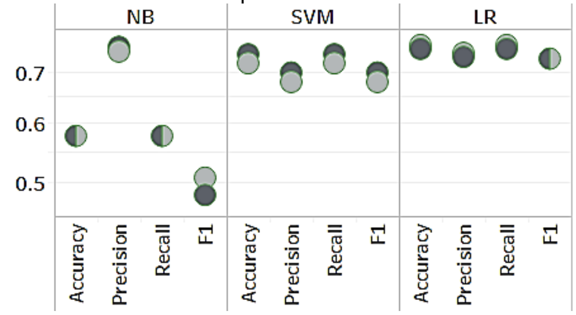


Figure 7. Metrics variation on S4 (light tone) & S16 (dark tone)

Regarding the execution time for prediction, the model trained on unbalanced data (S16) is relatively high but improves approximately three times in the other sample (S04).

CPU usage remains the same in both cases, limited to a single logical core. Memory consumption is relatively low.

Support vector machines (SVM) perform better than NB. To accelerate implementation, a min/max scaler is utilized, which reflects on prediction time drastically. Values are better when the classifier is applying to an unbalanced set than a balanced one. No significant difference was observed between the precision and recall. NB achieves higher results in precision than SVM in both samples, while about recall, it is precisely the opposite. The prediction time is short, making vector machines faster than NB, especially for the multi-class classification in S16. The SVM executes an average of three logical cores with an increase to five. In contrast, the NB uses only one logical core. The memory utilization varies from 5% to 11%.

Logistic regression (LR) achieves slightly better results than SVM. LR performs better on balanced data with fewer classes to predict. NB achieves similar precision results. The performance time is short, significantly faster compared to NB and with a similar time compared to SVM. LR utilizes three logical cores for computation, with the cores increasing to 5, identical to SVM. Memory usage ranging from 6% to 11%.

The decision tree classifier (DT) achieves high results, nearly 1, in both scenarios. However, in terms of precision and recall, the two values are merely the same. Time performance takes from 0.86 to 1.14 seconds, ranking it alongside some of the fastest algorithms in the study - SVM and LR**.** One logical core, similar to NB, is used to calculate the prediction, with the memory load from 4% to 12%.

The random forest classifier (RF) achieves results similar to that of the DT. Furthermore, precision and recall take the same values. Execution time varies from 33 to 66 seconds, making it the slowest one in the study. An increase in memory usage is observed, ranging from 6% to 19%, but only one logical core is utilized.

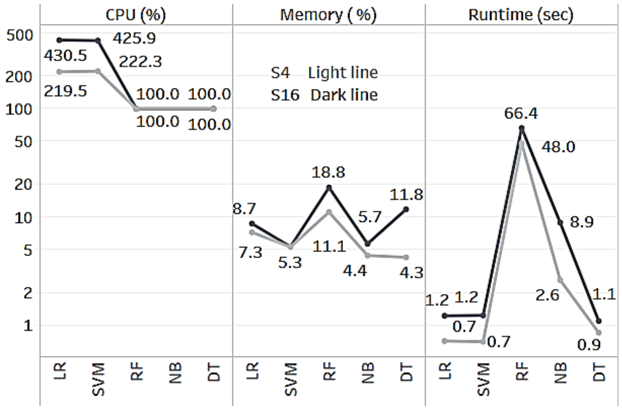


Figure 8. Resource utilization on S4 and S16

6. CONCLUSIONS

Although performance information was discussed, classifiers capabilities for precise threat identification needs comment.

Decision tree and random forest classifiers achieve the best performance metrics in identifying different anomalies in network traffic on IoT-23. These results are observed in all cases considered and in all well-presented classes. When analyzing the implementation of DT, it was found that the F-measure has the highest values for the following threats: Attack, Benign, C&C, C&C-HB, C&C-HB-Att, C&C-HPS, DDoS, HorizPortScan, Okiru. The other seven threats are presented with too few flows (below 100), reflecting the metric's low value. Therefore, it is impossible to assess whether their identification is challenging for the classifier or the lack of data.

Although RF is accurate and precise, it is the slowest and consumes the most memory. Therefore, the classifier is relatively inappropriate in a limited resource environment for real-time detection.

NB performance is insufficient compared to the others but achieves a very high result in identifying a specific malware – Okiru. A considerable difference in favor of precision versus recall is observed. In a corresponding context, that means significantly fewer false-positive results.

SVM and logistic regression have similar behavior, with very comparable results, especially when applying to an unbalanced sample. Both algorithms are high-speed and identify DDoS almost flawlessly but use a sizeable computational resource (CPU%). They are relevant for determining certain classes when the shortest execution time is required, and the environment is not high limited in computing power. SVM is sensitive to feature scaling, as this affects much in the model construction time.

The outputs for the classes with a few flow captures are heterogeneous and satisfactory conclusions cannot be drawn.

The experimental session has a specific design: two samples corresponding to two scenarios differing in the distribution of data and the number of threats for multi-class classification. Therefore, comparing this project with other similar approaches, particularly for implementation on IoT-23, is not so relevant. Nevertheless, the results are in line with other corresponding researches [6][7][8].

Considerations have to be taken: training or classification time could be misleading, as it depends heavily on the hardware platform (i.e., CPU, GPU, RAM) and software libraries used (i.e., optimized or not). Insufficient data for some classes is another limitation. Finally, better non-linear classifiers performance needs further analysis.

The experimental case study achieved the defined objectives. Furthermore, the results and session design stand as starting points for developing a demonstrator illustrating the capabilities of applying machine learning methods for improving IoT network security.

ACKNOWLEDGEMENT

The research is supported by the Project "BG05M2OP001-1.001-0003 "Center of Excellence in Informatics and Information and Communication Technologies" funded by the OP "Science and education for smart growth "2014-2020", EU through the European structural and investment funds.

REFERENCES

[1] European Union Agency for Cybersecurity, **Baseline security recommendations for IoT in the context of critical information infrastructure**s, Publications Office of the EU, 2017, ISBN: 978-92-9204-236-3.

[2] NIST **Framework for Improving Critical Infrastructure Cybersecurity,** DOI: 10.6028/NIST.CSWP.04162018

[3] A. Marzano, D. Alexander, et al. "**The evolution of Bashlite and Mirai IoT botnets**.", 2018 IEEE ISCC, Brazil, 2018

[4] Sarker, I.H., Kayes, A.S.M., Badsha, S. et al**. "Cybersecurity data science: an overview from a machine learning perspective."** J Big Data Vol.7, 2020.

[5] S. Garcia, A. Parmisano, & M. J. Erquiaga, "**IoT-23: A labeled dataset with malicious and benign IoT network traffic**" (Version 1.0.0) [Data set], 2020, Zenodo.

[6] N. Stoian, "**Machine Learning for Anomaly Detection in IoT networks: Malware analysis on the IoT-23 Data set."**, 2020.

[7] Mahmudul Hasan, Md. Milon Islam, et al., **"Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches**," Internet of Things, Vol.7, 2019.

[8] R. Ahmad, I. Alsmadi "**Machine learning approaches to IoT security: A systematic literature review**," Internet of Things, Vol. 14, 2021.