electronics

MDPI

Article

Evaluation of Explainable, Interpretable and Non-Interpretable Algorithms for Cyber Threat Detection

José Ramón Trillo [1,*,†], Felipe González-López [2,†], Juan Antonio Morente-Molinera [1,†], Roberto Magán-Carrión [2,†] and Pablo García-Sánchez [3,†]

[1] Department of Computer Science and Artificial Intelligence, Andalusian Research Institute in Data Science and Computational Intelligence, DaSCI, University of Granada, 18071 Granada, Spain; jamoren@decsai.ugr.es
[2] Department of Signal Theory, Telematics and Communications, University of Granada, 18071 Granada, Spain; feglez@ugr.es (F.G.-L.); rmagan@ugr.es (R.M.-C.)
[3] Department of Computer Engineering, Automation and Robotics—ETSIIT-CITIC, University of Granada, 18071 Granada, Spain; pablogarcia@ugr.es
[*] Correspondence: jrtrillo@ugr.es
[†] These authors contributed equally to this work.

Abstract

As anonymity-enabling technologies such as VPNs and proxies become increasingly exploited for malicious purposes, detecting traffic associated with such services emerges as a critical first step in anticipating potential cyber threats. This study analyses a network traffic dataset focused on anonymised IP addresses—not direct attacks—to evaluate and compare explainable, interpretable, and opaque machine learning models. Through advanced preprocessing and feature engineering, we examine the trade-off between model performance and transparency in the early detection of suspicious connections. We evaluate explainable ML-based models such as k-nearest neighbours, fuzzy algorithms, decision trees, and random forests, alongside interpretable models like naïve Bayes, support vector machines, and non-interpretable algorithms such as neural networks. Results show that neural networks achieve the highest performance, with a macro F1-score of 0.8786, but explainable models like HFER offer strong performance (macro F1-score = 0.6106) with greater interpretability. The choice of algorithm depends on project-specific needs: neural networks excel in accuracy, while explainable algorithms are preferred for resource efficiency and transparency, as stated in this work. This work underscores the importance of aligning cybersecurity strategies with operational requirements, providing insights into balancing performance with interpretability.

Keywords: cybersecurity; explainability; interpretability; artificial intelligence; classification

check for updates

Academic Editors: Li Pan, Ning Liu, and Conghui Zheng

Received: 30 June 2025
Revised: 24 July 2025
Accepted: 28 July 2025
Published: 31 July 2025

Citation: Trillo, J.R.; González-López, F.; Morente-Molinera, J.A.; Magán-Carrión, R.; García-Sánchez, P. Evaluation of Explainable, Interpretable and Non-Interpretable Algorithms for Cyber Threat Detection. *Electronics* **2025**, *14*, 3073. https://doi.org/10.3390/electronics14153073

1. Introduction

Rapid technological and digital advancements have propelled society into an era of unprecedented hyperconnectivity, marking the beginning of a fundamental transformation in how we generate, share, and protect information. Since its conception in the 1960s, the internet has evolved from a simple communication network among researchers to a global infrastructure supporting most human activities [1]. This evolution has enabled instantaneous access to massive volumes of information, connecting devices, automating processes, and facilitating a highly interdependent digital environment [2]. However, the infrastructure enabling the benefits of the digital age also brings new and complex challenges regarding information security and privacy. Cybersecurity has evolved in tandem with technology, continuously adapting to increasingly sophisticated and frequent security threats [3]. Initially, information security focused on protecting systems against physical intrusions and rudimentary malware [4]. Yet, as the internet expanded and the number of connected devices grew exponentially, cybercriminal tactics likewise became more intricate, ranging from distributed denial of service (DDoS) attacks to complex social engineering techniques, mainly aimed at exfiltrating sensitive data [5].

Today, in an environment where cybersecurity is essential to safeguard the integrity and confidentiality of information, one of the greatest challenges is the proliferation of cyber threats facilitated by the anonymity afforded by technologies such as virtual private networks (VPNs) and proxies [6]. While these services protect user privacy on unsecured networks, they also equip cybercriminals with tools to carry out malicious activities such as identity theft, malware distribution, and DDoS attacks [7]. Due to the opacity provided by anonymous IP addresses, the difficulty in tracing and mitigating these threats underscores the critical need for precise detection of IP addresses associated with anonymity services to bolster digital security.

In this context, machine learning (ML) algorithms emerge as a viable solution to identify complex patterns indicative of VPN and proxy usage [8]. However, the "black box" nature of many such algorithms poses significant challenges in terms of transparency and trustworthiness of their decisions [9]. This paper explores two crucial dimensions for addressing these issues within cybersecurity: interpretability, which enables an understanding of why a model makes specific decisions [10], and explainability, which elucidates which characteristics of the model influenced these decisions and how [9].

For example, a decision tree is considered interpretable because the decision path can be traced through explicit rules at each node, allowing direct human understanding. Similarly, fuzzy models such as HFER are explainable because they use rule-based logic that describes how inputs are transformed into outputs, even when the boundaries are soft or hierarchical. In contrast, neural networks are non-interpretable, as their inner structure involves multiple layers and parameters that cannot be easily understood or traced by human observers without external tools. These distinctions guide our classification of the models throughout the study.

Traditional rule-based systems often struggle to detect VPN and proxy usage because these services are specifically designed to bypass standard filtering and masking techniques. Machine learning (ML) provides a powerful alternative by enabling the automatic identification of complex and non-obvious patterns in large-scale, high-dimensional traffic data. In the context of VPN/proxy detection, ML can uncover behavioural signals—such as unusual attack timing, repetitive origin patterns, or atypical AS number distributions—that would be difficult to define explicitly through rules. Therefore, ML is employed in this study not only as a classifier but as a tool for uncovering hidden structures in anonymised threat data that can inform early detection and proactive security measures. To guide our research, we formulate the following key research questions (RQs):

- **RQ1:** To what extent do different ML algorithms balance predictive accuracy and interpretability when detecting anonymity service IPs?
- **RQ2:** Which machine learning models offer the highest degree of explainability without compromising detection efficiency with the proposed dataset?
- **RQ3:** Can a multi-criteria framework (e.g., a two-axis diagram) effectively categorise ML models based on their interpretability-performance trade-off?

In addressing these questions, we explore a variety of machine learning techniques—from decision trees to neural networks and support vector machines—evaluating them through the lens of both interpretability and explainability. Our methodology places particular emphasis on structuring a comparative analysis, including a visual representation to illustrate the interpretability-performance trade-off among the evaluated models clearly.

This paper is structured as follows: Section 2 describes the state of the art; Section 3 an analysis of the dataset to be addressed is to be carried out; Section 4 details the methodology for developing predictive models; Section 5 presents the algorithms employed and experimental results; Section 6 discusses the findings and their relevance in the cybersecurity context; and finally, Section 7 provides conclusions and suggests future research avenues in cybersecurity and machine learning.

2. State of the Art

The advent of ubiquitous digital connectivity and the proliferation of sophisticated cyberattacks have necessitated a paradigmatic shift in cybersecurity paradigms. Traditional rule-based detection mechanisms, while effective in static and known threat scenarios, have proven inadequate in the face of dynamic, obfuscated, and increasingly anonymised threats, such as those perpetrated via VPNs or proxy services. Consequently, the integration of machine learning (ML) into cybersecurity frameworks has emerged as a promising strategy for detecting subtle and complex threat patterns [8].

Nonetheless, the deployment of ML in this domain introduces a critical trade-off between predictive performance and interpretability. While high-performing models—particularly deep neural networks—are adept at capturing intricate data patterns, their opaque decision-making processes raise serious concerns in contexts where transparency, auditability, and accountability are paramount [9]. These concerns have catalysed the emergence of explainable artificial intelligence (XAI), which seeks to render ML models comprehensible to human stakeholders without significantly compromising performance.

Models such as decision trees and random forests have historically been favoured for their inherent interpretability and ability to yield structured, logical decision paths. Random forests, in particular, have demonstrated high robustness in high-dimensional and noisy environments, making them well-suited for intrusion detection systems [11]. Likewise, probabilistic classifiers such as naïve Bayes offer computational efficiency and have proven effective in streaming and high-frequency data scenarios typical of cybersecurity environments.

Conversely, deep learning architectures (e.g., multilayer neural networks) dominate in terms of predictive accuracy, especially in complex classification tasks involving high-volume and temporally granular datasets [12]. These models, however, are often criticised as “black boxes”, lacking the semantic transparency required for operational deployment in security-sensitive or regulated industries [13].

In addition to these classical and post-hoc approaches, fuzzy logic-based models—such as the hierarchical fuzzy exception rules (HFER) framework—have gained prominence. These models combine rule-based transparency with the capacity to handle imprecise or ambiguous input data, thereby offering a middle ground between deterministic logic and statistical learning [14]. Such hybrid systems are particularly beneficial in threat scenarios characterised by uncertainty and evolving attack signatures.

Recent contributions reinforce the centrality of explainability in security-related ML applications. For instance, in [15] present a lightweight and explainable ensemble classifier for real-time anomaly detection in IoT environments, addressing the dual concerns of resource efficiency and model transparency. Similarly, [16] presents a comprehensive approach to Android malware detection using explainable machine learning techniques. The authors emphasise the importance of feature selection to enhance model interpretability, identifying key features that significantly contribute to malware classification. By reducing data dimensionality, the proposed method achieves high accuracy while maintaining transparency, facilitating trust and compliance in security applications.

To guide model selection, some studies advocate the use of multi-criteria taxonomies, such as two-dimensional maps plotting interpretability against performance. These frameworks enable security practitioners to align algorithm choice with specific operational requirements—balancing detection accuracy with the need for explainability and computational tractability [9,10].

Finally, the contemporary literature reflects a transition from purely performance-driven models toward hybrid, explainable, and context-aware approaches. This evolution is particularly salient in cybersecurity, where the legitimacy and effectiveness of automated systems hinge not only on their predictive prowess but also on their interpretability, auditability, and compliance with evolving regulatory standards.

3. Problem Formulation: The CrowdSec Challenge

The challenge presented by CrowdSec (Available online: https://www.crowdsec.net/ (accessed on 30 July 2025)) represents a critical advancement in cybersecurity, addressing a prevalent and increasingly sophisticated issue: the accurate detection of IP addresses associated with VPNs or proxy services often used to conceal malicious online activity. VPNs and proxy services are regularly utilized by threat actors to obfuscate their identities and locations, undermining the ability of organisations to detect, attribute, and mitigate cyber threats effectively. This layer of anonymity not only conceals the origins of malicious actions but also exacerbates the complexity of preventing unauthorised intrusions and various cybercrimes.

This study proposes a predictive methodology for identifying connections originating from VPN or proxy services, using attack reports provided by CrowdSec as a representative dataset. It is important to clarify that our methodology does not attempt to uncover the real IP addresses of users behind VPNs, as these services are explicitly designed to prevent such tracing. Instead, our focus lies in identifying whether an observed IP—often the public-facing endpoint of a VPN or proxy—is likely associated with anonymisation services. This is accomplished by analysing indirect signals, such as Autonomous System Number known to host VPN infrastructure, unusual temporal patterns in attack reports, and mismatches in geographic metadata. By learning from these features, machine learning models can effectively flag traffic that exhibits characteristics typical of anonymised sources without violating user privacy or relying on packet content inspection. Rather than integrating directly into the CrowdSec platform, the goal is to explore and validate a robust approach to threat detection based on advanced preprocessing, feature engineering, and machine learning techniques. By analysing patterns in attack signals and time series behaviours, the methodology aims to improve the identification of anonymised malicious traffic. Although the framework is not tested in a real-time setting, it lays the groundwork for future deployment scenarios by demonstrating the potential of combining structured data processing with model-driven insights. Ultimately, the contribution lies in the development of a flexible and generalizable pipeline for enhancing cyber threat detection using publicly available security datasets.

The implications of successfully implementing a precise detection model extend beyond the immediate utility of identifying VPN and proxy connections; it will also lay the foundation for future adaptive threat detection that evolves alongside changing adversarial tactics, thereby offering dynamic resilience. The inherent innovation in this project lies in the precision and efficiency of the detection model, combining high computational efficiency with advanced pattern recognition, making it a critical tool for the cybersecurity landscape of tomorrow.

The dataset employed in this study is structured with essential metadata, including the type of detected attack, IP addresses of the perpetrators, timestamps, and data from the reporting entities. This limited, but crucial, information poses both challenges and opportunities for feature engineering. This study utilises creative, novel approaches to feature engineering, creating nuanced time series features for each detected event that enable precise identification of anonymising connections (e.g., VPN or proxy) within the reported data.

To address the problem, we sourced an open dataset from Kaggle [17], specifically curated to align with CrowdSec's objectives of detecting VPN/proxy IP addresses associated with cyber threats.

A rigorous exploration of CrowdSec's dataset was undertaken to enhance the interpretability and predictive power of each variable, given the importance of nuanced feature representation in threat detection (See Table 1):

Table 1. Summary of dataset features.

Feature	**Type**	**Description**
`attack_time`	datetime64	Timestamp of the attack
`watcher_country`	Categorical	Country of the monitor
`watcher_as_num`	float64	AS number of the monitor
`attacker_country`	Categorical	Country of the attacker
`attacker_as_num`	float64	AS number of the attacker
`attack_type`	Categorical	Type of attack
`watcher_uuid_enum`	int64	Monitor ID
`attacker_ip_enum`	int64	Attacker IP ID
`label`	Binary (0/1)	VPN/proxy indicator

Each variable is described in detail below

- *attack_time*: Represented in date time64 format, capturing high-resolution timestamps for each attack. This temporal granularity supports fine-grained time series analyses, which are critical for uncovering attack frequency patterns—whether daily, weekly, or seasonal.
- *watcher_country*: This categorical feature denotes the monitoring entity's country of origin, enabling a spatial understanding of attack reporting distribution. Geographic categorisation assists in optimizing regionalised threat response strategies.
- *watcher_as_num*: A float64 variable representing the Autonomous System (AS) number of the reporting monitor. Recognising AS numbers provides insights into the network architecture supporting attack detection.
- *attacker_country*: The attacker's country is instrumental in identifying geo-located patterns and regional threat vectors, particularly in cases involving organised or state-sponsored actors.
- *attacker_as_num*: This float64 variable details the attacker's AS number. Knowing these identifiers allows for the tracing of origin infrastructure, revealing behavioural clusters of certain ASs.
- *attack_type*: A categorical classification of the attack type, this feature enables model differentiation based on threat modality, enhancing detection specificity for each attack vector.

- *watcher_uuid_enum* and *attacker_ip_enum*: Unique identifiers for each monitoring entity and attacker IP, respectively, in int64 format. These variables enable precise tracking and correlation of monitoring and attack events across time and space.
- label: This binary label acts as the supervised learning target, distinguishing IP addresses associated with VPN/proxy use (1) from those not using these services (0).

4. Methodology of Explainable, Interpretable Algorithms and Non-Interpretable Algorithms

To address the classification of anonymised network traffic through machine learning, Section 4 is structured into two main subsections (see Figure 1). First, in Section 4.1, we describe the data preprocessing pipeline, which includes strategies for managing class imbalance, feature engineering, and normalization techniques designed to ensure the reliability and consistency of the input data. Second, in Section 4.2, we present the suite of machine learning models employed—ranging from explainable and interpretable algorithms to non-interpretable architectures—along with the evaluation metrics used to assess their effectiveness in detecting VPN and proxy-related IP addresses.

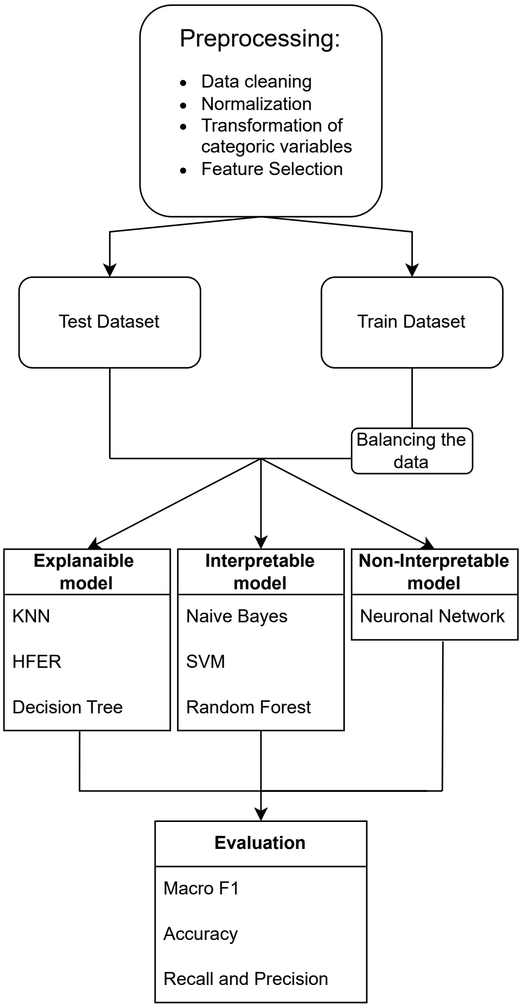

Figure 1. Diagram of the methodology followed in this work.

4.1. Data Preprocessing

The dataset used in this study was obtained from a publicly available repository on Kaggle [17]. Prior to training, a complete data preprocessing pipeline was applied to ensure quality input for the models and to support reliable and interpretable evaluation results.

For the construction of training and evaluation subsets, a hold-out method was applied. Since the evaluation labels were initially unavailable, the labelled portion of the dataset was partitioned following a stratified split approach inspired by [18], allocating 70% for training and 30% for model evaluation. Although the choice of this split ratio is widely adopted in the literature, it is acknowledged that it lacks empirical justification as being universally optimal. Nonetheless, this strategy preserves the original class distribution and provides a practical balance between data availability for training and evaluation.

As a result, 61,629,685 tuples were divided into 43,140,780 for training and 18,488,905 for evaluation. This partitioning enables both the development and validation of models under consistent statistical conditions, maintaining the integrity of the original dataset structure. The stratified nature of the split also ensures proportional representation of both classes, which is essential when evaluating model performance in the context of significant class imbalance.

Before proceeding with the analysis, extensive pre-processing of the data was carried out, including handling missing values, coding categorical variables, feature selection, normalization and the application of techniques to correct class imbalances. This last aspect is particularly relevant due to the marked imbalance in the dataset: class 0 (non-VPN/Proxy) accounts for 98.32% of the instances (60,594,448 records), while class 1 (VPN/Proxy) accounts for only 1.68% (1,035,237 records). To mitigate this problem, the study implemented several strategies in the training set, such as undersampling and SMOTE (Synthetic Minority Over-sampling Technique), aimed at improving the sensitivity of the model towards the minority class without negatively affecting its overall performance. The decision on which technique to apply was guided by preliminary validation experiments to assess the trade-off between model stability and minority-class recall. As a result, the diagram shown in Figure 1 is obtained. Although SMOTE proved useful in synthetically balancing the dataset, we acknowledge that oversampling alone may not fully capture the underlying variability and behavioural signatures associated with VPN/proxy traffic. The absence of alternative datasets or a controlled testbed setup limits the representativeness of minority class patterns, which could affect generalisation in real-world applications. These considerations are discussed further in the conclusions as avenues for future work.

4.2. Machine Learning Models

This section presents the machine learning algorithms employed for classifying IP addresses associated with VPN or proxy services. Each model is briefly described, highlighting its operational principles and applicability to the detection task.

To evaluate model performance, we used four commonly accepted metrics [12]:

- **Accuracy:** The proportion of correctly predicted instances among all predictions. In imbalanced datasets, this metric may be misleading, as high accuracy can be achieved by consistently predicting the majority class.
- **Precision:** The proportion of true positive predictions among all instances predicted as positive. It reflects how reliable the model is when it predicts a VPN/proxy connection.
- **Recall:** The proportion of true positives identified out of all actual positive instances. It indicates the model's ability to detect all VPN/proxy-related traffic.
- **Macro F1-score:** The harmonic mean of precision and recall, calculated for each class and then averaged. It is particularly suitable for imbalanced datasets, as it gives equal weight to both classes.

The following algorithms were evaluated:

- Explainable: Explainable algorithms are those whose internal decision-making logic can be transparently represented and understood by humans, often through rule-based or instance-based mechanisms. These models provide direct insight into how predictions are made, making them suitable for applications requiring high levels of transparency and traceability.

  - **k-Nearest Neighbours (KNN):** A non-parametric algorithm that assigns class labels based on the majority label of the k nearest data points. Simplicity and interpretability make KNN a useful baseline model. It was evaluated with $k = 3$ and $k = 5$ [11].
  - **Chi-square + KNN (Chi):** This model combines the feature selection approach proposed by Chi et al. [19] with a fuzzy logic-based classification system. The method focuses on identifying the most relevant features through rule-based heuristics, enabling the construction of interpretable and effective fuzzy classification models.
  - **HFER (Hierarchical Fuzzy Exception Rules):** A fuzzy rule-based approach that applies hierarchical logic to handle exceptions and borderline cases in classification tasks. It enhances interpretability while maintaining competitive performance [14].
  - **Decision Tree:** A tree-based model that recursively splits the dataset using features that result in the highest information gain. Decision trees are highly interpretable and useful for identifying decision logic explicitly [20].

- Interpretable: Interpretable algorithms offer a balance between predictive performance and understandability. While their internal workings may not be as directly explainable as rule-based systems, they maintain a structure that allows human analysts to interpret decisions through parameters, feature importance, or simplified representations.

  - **Random Forest:** An ensemble method composed of multiple decision trees. Each tree is trained on a random subset of data and features, and their predictions are aggregated by majority voting. This improves robustness and generalization [21].
  - **Naïve Bayes:** A probabilistic model based on Bayes' theorem and the assumption of conditional independence between features. Despite its simplicity, it often performs well in text and high-dimensional datasets [22].
  - **Support Vector Machines (SVMs):** A margin-based classifier that identifies the optimal hyperplane to separate classes. We evaluated SVM with linear, polynomial, and radial basis function (RBF) kernels to explore different types of boundaries [23].

- Non-interpretable: Non-interpretable algorithms, often referred to as black-box models, achieve high predictive accuracy by capturing complex, non-linear relationships in data. However, their internal mechanisms are opaque and difficult to interpret, which may limit their applicability in contexts requiring transparency, auditability, or regulatory compliance.

  - **Neural Networks:** Deep learning models with multiple hidden layers capable of capturing complex, non-linear patterns in data. The implemented architecture included seven blocks, each consisting of

    - Dense layer with 128 neurons and ReLU activation;
    - Batch normalization for improved training stability;
    - Dropout layer with 50% rate to reduce overfitting.

    The final output layer uses softmax activation for binary classification [13].

For the following pseudocode Algorithm 1, we outline the process of training a traditional machine learning classifier. The steps involve selecting features, handling missing data, and choosing the appropriate sampling and classification techniques. Based on the type of classifier selected (such as KNN, decision trees, or SVM), the model is trained on the provided dataset to generate a trained classifier. Below is the pseudocode:

Algorithm 1 Simplified Classification Pipeline

```
Input: Training dataset $D$
Output: Trained classifier model
Handle missing data and encode categorical variables in $D$
If needed, normalize $D$
If needed, apply undersampling or SMOTE based on validation performance and model characteristics
If needed, apply feature selection
Select and configure classification algorithm (e.g., KNN, SVM, RF, NB)
Train classifier on $D$
return Trained model
```

The pseudocode Algorithm 2 illustrates the process of training a neural network model. It includes essential steps such as feature selection, data preprocessing, and the configuration of the neural network architecture. The architecture incorporates dense layers, batch normalization, and dropout techniques to improve model performance and prevent overfitting. Finally, the network is trained using the provided dataset to generate a fully trained neural network model.

Algorithm 2 Neural Network Training Pipeline

```
Input: Training dataset $D$
Output: Trained neural network classifier
Handle missing data and encode categorical variables in $D$
if normalization is needed then
    Normalize $D$
end if
if sampling is needed then
    Apply undersampling or SMOTE based on validation performance and model characteristics
end if
if feature selection is needed then
    Apply feature selection
end if
Initialize neural network:
    Add 7 blocks of Dense(128, relu),BatchNormalization,Dropout(0.5)
    Add output layer: Dense(1, softmax)
Train the neural network with $D$
return Trained model
```

The results of each algorithm are then presented, highlighting their strengths and weaknesses in terms of accuracy and computational efficiency. Furthermore, the implications of these results for the effective detection of VPN/Proxy services in the context of cybersecurity are discussed.

5. Analysis of Results

In this section, we conduct a comprehensive analysis of the results obtained from the explainable, interpretable, and non-interpretable algorithms evaluated for the classification of IP addresses associated with VPN or Proxy services. Furthermore, we will provide recommendations based on the specific context of their implementation. In the realm of cybersecurity, it is imperative not only to identify threats accurately but also to do so in a manner that is understandable and efficient. The selection of the appropriate algorithm can significantly impact the ability to detect and mitigate attacks, particularly in contexts where adversaries employ advanced techniques to conceal their identities, such as through VPN and proxy services.

To effectively address this challenge, it is essential to evaluate the *macro F1-scores* of each algorithm, as this metric provides a balanced perspective on model performance in terms of *precision* and *recall*, without unduly favouring more frequent classes. The evaluation of *macro F1-scores* allows for an objective comparison of the inherent strengths and weaknesses of each algorithm. This metric is particularly valuable in scenarios characterised by class imbalance, such as the present study, where the majority of IP addresses are not associated with VPN or proxy services.

By comparing different algorithms, we can identify those models that offer higher precision, as well as those that excel in terms of interpretability and computational efficiency. This information enables us to formulate recommendations tailored to the project's specific needs. For instance, in contexts where precision is paramount, such as in detecting sophisticated attacks, algorithms with superior *macro F1-scores* will be prioritised. Conversely, in environments where interpretability and ease of implementation are of greater importance, explainable algorithms will be deemed more suitable.

Moreover, contextualised recommendations are essential to ensure that cybersecurity solutions are effective and practical. An algorithm that delivers high precision but proves difficult to interpret may not be the best choice in all cases, particularly when thorough auditing of the decision-making process is required to comply with security and privacy regulations.

The results highlight a clear distinction in algorithm effectiveness, as it can be shown in Figure 2 and in the scatter plot depicted in Figure 3. In this figure, neural networks achieve the highest performance, approaching the maximum value of 1.0, suggesting their ability to capture complex patterns and relationships in the data. Nevertheless, it is important to consider that complex data structures can often be transformed into simpler representations through dimensionality reduction or embedding techniques (e.g., PCA, autoencoders). Such transformations may allow less complex or more interpretable models to achieve higher performance by making key patterns more accessible. While this study did not explore these techniques, their integration could potentially bridge the performance gap between transparent and opaque models and thus remains a promising avenue for future work. Other models, such as decision trees, random forest, naïve Bayes, and linear SVM, perform moderately well, indicating they may capture simpler patterns effectively but might struggle with more intricate structures. On the other hand, polynomial and radial SVM show the lowest performance, close to 0.4, potentially due to suboptimal parameter settings, insufficient flexibility, or data characteristics that are poorly suited to these models. This comparison emphasises the importance of algorithm selection and configuration when addressing classification tasks, as performance can vary significantly depending on the method and its suitability to the dataset.

Although Figure 4 shows that most algorithms achieve high accuracy scores—often exceeding 0.9—this metric must be interpreted with caution, particularly in imbalanced datasets. In such cases, high accuracy can be misleading, as it may mask poor performance on minority classes. While neural networks appear to deliver the highest accuracy, approaching 1.0 in some instances, this alone does not necessarily reflect superior classification performance. Therefore, complementary metrics such as precision, recall, or F1-score are essential for a more nuanced and reliable evaluation of model effectiveness, especially in tasks where class imbalance is a concern.

In addition to neural networks, other algorithms such as HFER3.5, random forest, naïve Bayes, and linear support vector machines (SVM) also show commendable performance, achieving relatively high accuracy scores. These algorithms display significant effectiveness in handling the data and task requirements. Their strong results suggest they are well-equipped to deal with diverse data patterns and underlying complexities.

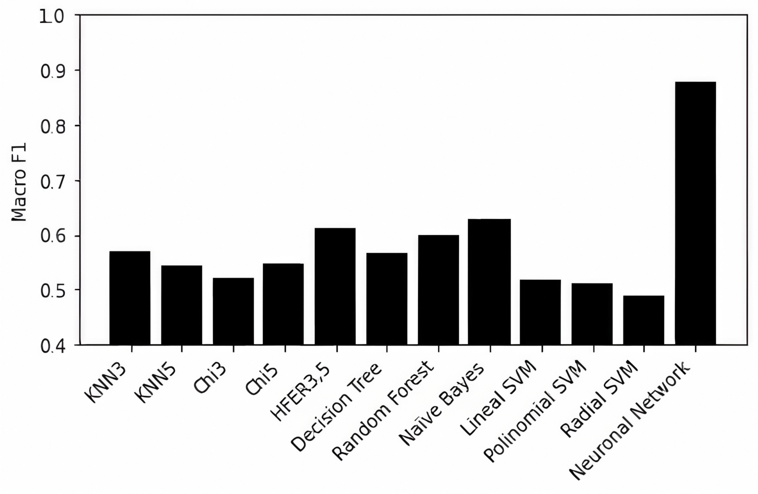

Figure 2. Macro F1-score of the algorithms with its best selection of hyperparameters.

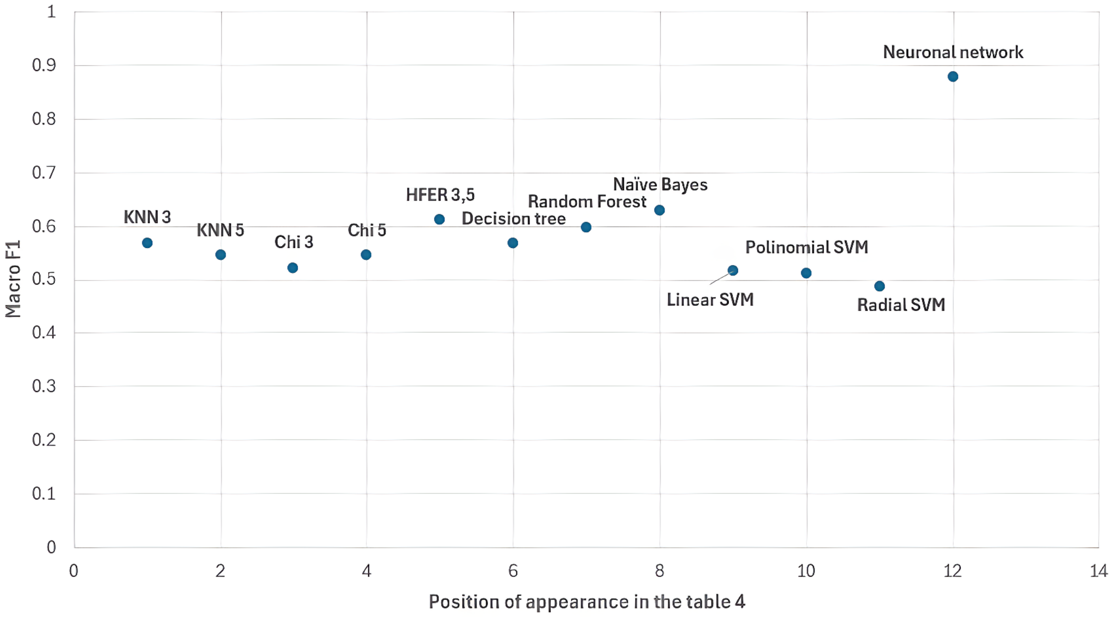

Figure 3. Scatterplot of the algorithms as a function of their Macro F1-score.

Given the scale of the dataset used in this study—over 61 million labelled instances—computational efficiency becomes a critical factor in the practical deployment of threat detection models. Therefore, we provide a theoretical analysis of the computational complexity and empirical feasibility of the evaluated algorithms.

Explainable algorithms such as KNN and decision trees exhibit relatively low training complexity. KNN has negligible training time but incurs a prediction cost of $O(n \cdot d)$ per query, which may hinder real-time scalability. Decision trees, on the other hand, typically require $O(n \cdot d \cdot log(n))$ for training and $O(log(n))$ for prediction, making them suitable for rapid inference tasks. HFER, a fuzzy rule-based system, introduces moderate computational overhead due to rule generation but remains tractable with hierarchical pruning mechanisms.

Interpretable algorithms like naïve Bayes and linear SVM offer efficient training and prediction. Naïve Bayes has linear time complexity $O(n \cdot d)$ for both phases, and SVMs with linear kernels scale better than their polynomial or RBF counterparts, which become impractical in large-scale settings due to their higher training complexity (between $O(n^2)$ and $O(n^3)$).

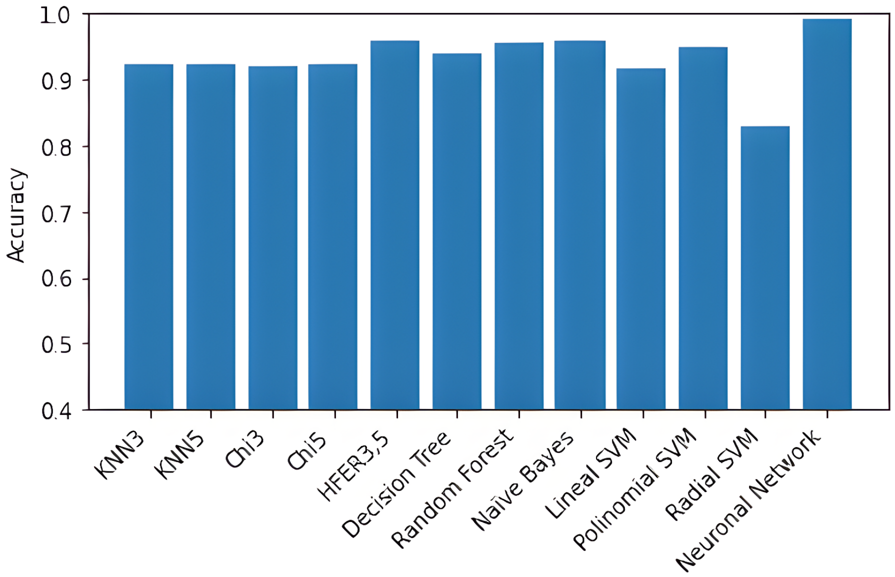

Figure 4. Accuracy of the algorithms with its best selection of hyperparameters.

Neural networks, while achieving the highest predictive performance, exhibit the highest computational cost. Their training complexity typically lies in the order of O($n \cdot d \cdot h \cdot e$), where h is the number of neurons per layer and e the number of epochs. Training on over 40 million samples required high memory availability and prolonged processing times, making them less suitable for real-time or low-resource scenarios without GPU acceleration.

In conclusion, explainable and interpretable models such as decision trees, HFER and naïve Bayes provide a favourable trade-off between detection quality and resource consumption, which supports the resource efficiency claim made in the summary.

On the other hand, polynomial SVM, although performing slightly less well than the aforementioned methods, still maintains competitive accuracy levels. Its performance is respectable, reflecting its ability to capture some level of complexity, but it falls short of matching the top performers. Meanwhile, Radial SVM demonstrates the weakest performance across the board, with an accuracy score approaching 0.8. While still functional, this result indicates that Radial SVM struggles with the task at hand, potentially due to its limitations in handling complex data structures or the nature of the problem being addressed. To ensure reproducibility and facilitate further experimentation, all code and artefacts related to model development and hyperparameter tuning have been made publicly available (Available online: https://github.com/jrtrillo/TFM-ciberseguridad (accessed on 24 July 2025)).

To ensure a thorough and rigorous analysis, we systematically explored a selection of hyperparameters for each algorithm, thoroughly evaluating the entire parameter space. The optimal hyperparameter selection for each algorithm, which was determined through extensive experimentation and fine-tuning, is detailed in Table 2. Although Table 2 reports only the best-performing preprocessing configuration for each model, it is important to note that all combinations of the considered techniques (i.e., SMOTE, undersampling, feature selection, and normalization) were tested independently and jointly during experimentation. The configurations shown are those that achieved the highest macro F1-score in each case. Including all possible results would have significantly increased the length and complexity of the manuscript, so only the optimal setting per model is presented for clarity. Nevertheless, a complete ablation-style breakdown may be considered in future work to quantify the impact of each individual preprocessing component.

Table 2. Preprocessing the best selection of hyperparameters. The check mark means that it has been selected, and the cross means that it has not been selected.

	Selection of Preprocessing Techniques			
	Feature Selection	**SMOTE**	**Undersampling**	**Data Normalization**
KNN 3	✓	✓	X	X
KNN 5	✓	✓	X	X
Chi 3	✓	X	✓	X
Chi 5	✓	X	✓	X
HFER 3.5	✓	✓	X	✓
Decision Tree	✓	✓	X	X
Naïve Bayes	X	✓	X	X
Linear SVM	X	✓	X	✓
Polynomial SVM	X	✓	X	X
Radial SVM	✓	X	✓	X
Neuronal Network	✓	✓	X	✓

Table 2 summarises the combination of preprocessing techniques applied to each model. The columns represent the following operations:

- **Feature Selection :** Identifies and retains only the most relevant features to reduce dimensionality and improve model generalisation.
- **SMOTE (Synthetic Minority Over-sampling Technique):** A strategy to synthetically generate new instances of the minority class in order to balance the dataset.
- **Undersampling:** Reduces the number of majority class samples to mitigate class imbalance by equalising class proportions.
- **Data Normalization:** Transforms feature values to a common scale (typically [0,1]) to ensure consistency across different models, especially those sensitive to feature magnitudes.

Following the identification of the optimal configurations, we proceeded to evaluate the performance of the algorithms across multiple metrics. In addition to the primary evaluation metric, the *macro F1-score*, we incorporated several other important performance measures, including *accuracy*, *recall*, and *precision* for each class. By considering these complementary metrics, we were able to gain a more comprehensive understanding of how well the algorithms performed in various aspects of the classification task. Each metric provides unique insights into different facets of model performance: *accuracy* reflects overall correctness, *precision* assesses the relevance of positive predictions, and *recall* evaluates the model's ability to identify all relevant instances. This multifaceted evaluation approach, detailed in Table 3, allowed for a more informed and holistic assessment of the algorithms' effectiveness, ensuring that our conclusions were based on a well-rounded analysis of their performance. To further enhance reproducibility, Table 4 summarises the best-performing hyperparameter configurations used for each algorithm after tuning.

This section addresses **RQ1**, as the analysis—particularly the results presented in Table 2 and Figure 2—offers a comprehensive comparative evaluation of diverse machine learning algorithms, emphasizing the inherent trade-off between predictive accuracy and interpretability in the context of detecting IP addresses linked to anonymity services.

Table 3. The best results for the different metrics after running each algorithm with different combinations of pre-processing and parameter settings. The highlighted value is the highest value for the metric.

	Accuracy	**Precision (0)**	**Recall (0)**	**F1-Score (0)**	**Precision (1)**	**Recall (1)**	**F1-Score (1)**	**Macro F1**
KNN 3	0.9223	0.9296	0.9908	0.9592	0.4966	0.1075	0.1768	0.5680
KNN 5	0.9240	0.9341	0.9880	0.9603	0.3356	0.0800	0.1292	0.5447
Chi 3	0.9210	0.9330	0.9859	0.9587	0.2171	0.0525	0.0845	0.5216
Chi 5	0.9248	0.9349	0.9880	0.9607	0.3327	0.0803	0.1294	0.5451
HFER 3.5	0.9576	0.9670	0.9896	0.9782	0.4052	0.1735	0.2430	0.6106
Decision tree	0.9382	0.9480	0.9887	0.9679	0.3653	0.1071	0.1657	0.5668
Random Forest	0.9561	0.9662	0.9889	0.9774	0.3624	0.1549	0.2171	0.5972
Naïve Bayes	0.9581	0.9662	0.9909	0.9784	0.4831	0.1963	0.2792	0.6288
Linear SVM	0.9174	0.9296	0.9856	0.9568	0.2068	0.0478	0.0776	0.5172
Polynomial SVM	0.9508	0.9658	0.9839	0.9748	0.0762	0.0366	0.0495	0.5121
Radial SVM	0.8292	0.8369	0.9875	0.9060	0.3798	0.0383	0.0695	0.4878
Neuronal network	**0.9910**	**0.9932**	**0.9976**	**0.9954**	**0.8611**	**0.6830**	**0.7618**	**0.8786**

Table 4. Optimal hyperparameter configuration for each model.

Algorithm	Best Hyperparameters
KNN 3	$k = 3$, Euclidean distance
KNN 5	$k = 5$, Euclidean distance
Chi 3	$k = 3$, Chi-squared
Chi 5	$k = 5$, Chi-squared
HFER 3.5	3 fuzzy labels for the base rules, 5 fuzzy labels for the exception rules
Decision Tree	max_depth = 10, criterion = ’entropy’
Random Forest	n_estimators = 100, max_depth = 12, criterion = ’gini’
Naïve Bayes	GaussianNB (default parameters)
Linear SVM	C = 1.0, kernel = ’linear’
Polynomial SVM	C = 1.0, degree = 3, kernel = ’poly’
Radial SVM	C = 1.0, gamma = 0.1, kernel = ’rbf’
Neural Network	7 layers: Dense(128, ReLU), dropout = 0.5, optimizer = Adam, batch_size = 512, epochs = 10

6. Discussion

In this study, we conducted a comparative analysis of explainable and interpretable algorithms based on their *macro F1-scores*, evaluating their advantages and drawbacks depending on the application context. Explainable algorithms like KNN, fuzzy logic models, decision trees, and random forests offer inherent transparency in decision-making processes, which is valuable in applications where interpretability is a priority. Among these, random forest stands out with a *macro F1-score* of 0.5972, indicating a balance of *accuracy* and interpretability that makes it suitable for real-world deployment where model transparency is required. KNN variants, despite being resource-efficient and straightforward, showed lower scores, such as KNN3’s 0.5680 and KNN5’s 0.5447, suggesting that while useful for resource-constrained environments, they may fall short in more demanding scenarios. Fuzzy logic models, especially HFER with a 0.6106 score, demonstrated robust interpretability and moderate performance, aligning well with applications that demand transparent yet reasonably accurate threat detection.

Beyond the comparative analysis of performance metrics, it is important to connect these results back to the core problem that motivates this study: identifying VPN/proxy-based anonymised IP traffic in real-world cybersecurity scenarios. Our findings suggest that models like neural networks, despite their limited interpretability, can effectively flag traffic patterns typically associated with anonymity services due to their high sensitivity to complex behavioural indicators such as anomalous AS numbers or atypical geolocation dynamics. On the other hand, explainable models like HFER and decision trees, though less accurate, are better suited for environments where traceability and operational transparency are essential, such as real-time security audits or compliance-driven monitoring. This highlights that model selection is not only a technical decision but a strategic one, depending on the specific needs of the cybersecurity infrastructure.

On the other hand, interpretable algorithms like naïve bayes, SVM, and neural networks demonstrated a significant edge in terms of performance, especially in complex data structures, where interpretability alone is less critical. Neural networks achieved the highest *macro F1-score* of 0.8786, outperforming all other models and emphasizing their capability to capture intricate, non-linear data relationships critical in cybersecurity. Naïve Bayes, with a *macro F1-score* of 0.6288, exhibited strong interpretability and adaptability for dynamic data, adding value in fast-paced cybersecurity environments. While SVMs performed comparably lower, with scores ranging from 0.4878 to 0.5172, they retain their strength in high-dimensional scenarios, which are common in security analytics, but with limited transparency due to the complexity of kernel-based decision boundaries.

When prioritizing *accuracy*, especially for scenarios demanding sophisticated pattern recognition, interpretable algorithms, particularly neural networks, clearly excel, though they require considerable computational resources and offer limited insight into the prediction rationale. This characteristic can restrict their applicability in cases where decision transparency is crucial for compliance or trust. In contrast, explainable models such as decision trees and random forests offer a compelling balance for implementations that demand clarity, efficiency, and ease of interpretation. Decision trees, scoring 0.5668, and random forest, with a slightly higher 0.5972, provide straightforward interpretability while maintaining sufficient *accuracy*, making them advantageous for scenarios where decisions need to be easily understandable and verifiable by human analysts. Fuzzy logic models, specifically HFER, maintain a level of flexibility and performance suitable for applications where ambiguity in data exists, providing a middle ground between complexity and transparency.

The findings of this study indicate that a flexible approach combining both types of algorithms could maximize the effectiveness of cybersecurity solutions. Hybrid systems could employ explainable algorithms for continuous monitoring, where transparency is key and interpretable algorithms for detailed analysis in response to specific incidents. Such an adaptive framework, aligning with the dynamic and diverse threat landscape of cybersecurity, allows for optimal model selection based on operational demands, enhancing both the robustness and responsiveness of threat detection systems. Ultimately, these findings directly answer RQ2 by identifying the models that offer the best explainability–performance balance. In particular, the HFER model (macro F1-score = 0.6106) emerges as the most effective explainable algorithm, combining rule-based transparency with acceptable detection capability. Among interpretable models, Random Forest (macro F1-score = 0.5972) also demonstrates high detection efficiency while maintaining a moderate level of interpretability. Therefore, both models represent suitable options when explainability is required without severely compromising classification performance. Furthermore, **RQ3** is addressed through the conceptual framework articulated in the Introduction and substantiated by the bidimensional representation in Figure 3, which serves as a cogent tool for classifying machine learning models according to the inherent trade-offs between their interpretability and predictive performance.

7. Conclusions

This study addresses the challenge of enhancing threat detection in cybersecurity through the use of explainable and interpretable machine learning algorithms, focusing on the accurate classification of IP addresses associated with VPN and Proxy services. These addresses are commonly exploited by malicious actors to conceal their identities and facilitate harmful activities. This work extends to applications in next-generation firewalls (NGFWs), which integrate traditional firewall capabilities with advanced technologies like deep packet inspection (DPI), intrusion prevention systems (IPS), and threat intelligence, allowing for efficient identification and blocking of both known and unknown threats. Additionally, NGFWs support SSL/TLS inspection to decrypt and analyse encrypted traffic, providing an added layer of security by revealing hidden threats in encrypted communications. They also incorporate zero-trust access models, where access is granted based on user verification and contextual factors rather than solely perimeter security.

Our analysis highlights the role of specific algorithms: KNN, though simplistic, proved effective in balanced data scenarios, while decision trees provided high interpretability but required pruning to avoid overfitting. Fuzzy logic algorithms excelled in managing ambiguity in complex datasets, and random forests offered robust *accuracy* with substantial feature importance interpretation. SVMs, while challenging to interpret, demonstrated efficacy in high-dimensional spaces, essential in cybersecurity contexts, while naïve Bayes showed agility in real-time data updates, and neural networks, though opaque, yielded insights through feature importance analyses.

While foundational, this study opens avenues for future work in combining explainable and interpretable algorithms for more robust threat detection. Priority areas include advancing feature engineering to better capture attacker behaviour dynamics, incorporating ensemble techniques like stacking and blending to bolster model *accuracy* and interpretability and developing real-time deployment solutions in intrusion detection systems. Enhancing neural network transparency with post-hoc explainability methods such as LIME (local interpretable model-agnostic explanations) or SHAP (Shapley additive explanations) could significantly improve understanding of their decision-making processes. Although not used in this study, these techniques are highly recommended for use in future work aiming to make deep learning models more interpretable in cybersecurity environments. Furthermore, embedding real-time adaptive learning models would enable quicker responses to evolving threats, enhancing overall resilience. Additionally, future work should consider the collection of enriched datasets through controlled testbeds or the integration of heterogeneous sources with annotated VPN/proxy traffic. This would address the limitations of relying solely on oversampling techniques such as SMOTE and enhance the robustness and generalisability of the proposed models.

Finally, the integration of NGFWs with DPI, IPS, threat intelligence, SSL/TLS inspection, and zero-trust models lays a strong defence against sophisticated threats. Emphasizing digital forensics and automated AI-driven responses further bolsters incident response capabilities. This study underscores the importance of ethical considerations in data privacy, proposing compliance with standards like GDPR to ensure responsible handling of cybersecurity data.

Author Contributions: Conceptualization, J.R.T. and P.G.-S.; Methodology, J.R.T., F.G.-L., R.M.-C. and P. G.-S., Software, J.R.T., J.A.M.-M., R.M.-C. and P.G.-S.; Formal analysis, J.R.T., F.G.-L. and J.A.M.-M.; Investigation, J.R.T. and F.G.-L.; Writing—original draft, J.R.T., J.A.M.-M., R.M.-C. and P.G.-S.; Writing—review & editing, J.R.T., F.G.-L., J.A.M.-M., R.M.-C. and P.G.-S.; Supervision, J.A.M.-M.; Funding acquisition, R.M.-C. and P.G.-S. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been supported by projects PID2022-139297OB-I00 and PID2023-147409NB-C21, funded by MICIU/AEI/10.13039/501100011033 and ERDF/EU. Authors also are supported by projects C-ING-165-UGR23, C-ING-027-UGR23 and C-ING-300-UGR23 co-funded by the Regional Ministry of University, Research and Innovation and the European Union under the Andalusia ERDF Program 2021-2027.

Data Availability Statement: The data is availability in https://www.kaggle.com/competitions/vpn-classification/data (accessed on 24 July 2025).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Warner, M. Cybersecurity: A pre-history. *Intell. Natl. Secur.* **2012**, *27*, 781–799. [CrossRef]
2. Thames, L.; Schaefer, D. *Cybersecurity for Industry 4.0*; Springer: Berlin/Heidelberg, Germany, 2017. [CrossRef]
3. Craigen, D.; Diakun-Thibault, N.; Purse, R. Defining cybersecurity. *Technol. Innov. Manag. Rev.* **2014**, *4*, 13–21. [CrossRef]
4. Singer, P.W.; Friedman, A. *Cybersecurity: What Everyone Needs to Know*; OUP: New York, NY, USA, 2014. [CrossRef]
5. Mirkovic, J.; Reiher, P. A taxonomy of DDoS attack and DDoS defense mechanisms. *ACM SIGCOMM Comput. Commun. Rev.* **2004**, *34*, 39–53. [CrossRef]
6. Pavlicek, A.; Sudzina, F. Use of virtual private networks (VPN) and proxy servers: Impact of personality and demographics. In Proceedings of the 2018 Thirteenth International Conference on Digital Information Management (ICDIM), Berlin, Germany, 24–26 September 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 108–111. [CrossRef]

7. Mittal, M.; Kumar, K.; Behal, S. Deep learning approaches for detecting DDoS attacks: A systematic review. *Soft Comput.* **2023**, *27*, 13039–13075. [CrossRef] [PubMed]
8. Sarker, I.H.; Kayes, A.; Badsha, S.; Alqahtani, H.; Watters, P.; Ng, A. Cybersecurity data science: An overview from machine learning perspective. *J. Big Data* **2020**, *7*, 41. [CrossRef]
9. Arrieta, A.B.; Díaz-Rodríguez, N.; Del Ser, J.; Bennetot, A.; Tabik, S.; Barbado, A.; García, S.; Gil-López, S.; Molina, D.; Benjamins, R.; et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion* **2020**, *58*, 82–115. [CrossRef]
10. Cano, A.; Zafra, A.; Ventura, S. An interpretable classification rule mining algorithm. *Inf. Sci.* **2013**, *240*, 1–20. [CrossRef]
11. Kilincer, I.F.; Ertam, F.; Sengur, A. Machine learning methods for cyber security intrusion detection: Datasets and comparative study. *Comput. Netw.* **2021**, *188*, 107840. [CrossRef]
12. Magán-Carrión, R.; Urda, D.; Díaz-Cano, I.; Dorronsoro, B. Towards a reliable comparison and evaluation of network intrusion detection systems based on machine learning approaches. *Appl. Sci.* **2020**, *10*, 1775. [CrossRef]
13. Petersen, P.; Voigtlaender, F. Optimal approximation of piecewise smooth functions using deep ReLU neural networks. *Neural Netw.* **2018**, *108*, 296–330. [CrossRef] [PubMed]
14. Trillo, J.R.; Fernandez, A.; Herrera, F. Hfer: Promoting explainability in fuzzy systems via hierarchical fuzzy exception rules. In Proceedings of the 2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Glasgow, UK, 19–24 July 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–8. [CrossRef]
15. Wu, J.; Xu, X.; Liao, X.; Li, Z.; Zhang, S.; Huang, Y. Intelligent diagnosis method of data center precision air conditioning fault based on knowledge graph. *Electronics* **2023**, *12*, 498. [CrossRef]
16. Christakis, I.; Tsakiridis, O.; Kandris, D.; Stavrakas, I. A Kalman Filter Scheme for the Optimization of Low-Cost Gas Sensor Measurements. *Electronics* **2024**, *13*, 25. [CrossRef]
17. alpacads. Binary Classification of VPN Proxy IP Address. Kaggle. 2023. Available online: https://kaggle.com/competitions/vpn-classification (accessed on 24 July 2025).
18. Blum, A.; Kalai, A.; Langford, J. Beating the hold-out: Bounds for k-fold and progressive cross-validation. In Proceedings of the Twelfth Annual Conference on Computational Learning Theory, Santa Cruz, CA, USA, 7–9 July 1999; pp. 203–208. [CrossRef]
19. Chi, Z.; Yan, H.; Pham, T. *Fuzzy Algorithms: With Applications to Image Processing and Pattern Recognition*; World Scientific: Singapore, 1996; Volume 10. [CrossRef]
20. Freund, Y.; Mason, L. The alternating decision tree learning algorithm. In Proceedings of the ICML, Bled, Slovenia, 27–30 June 1999; Volume 99, pp. 124–133.
21. Kulkarni, A.D.; Lowe, B. Random forest algorithm for land cover classification. *Int. J. Recent Innov. Trends Comput. Commun.* **2016**, *4*, 58–63.
22. Chen, S.; Webb, G.I.; Liu, L.; Ma, X. A novel selective naïve Bayes algorithm. *Knowl.-Based Syst.* **2020**, *192*, 105361. [CrossRef]
23. Dong, J.x.; Krzyżak, A.; Suen, C.Y. A fast svm training algorithm. In Proceedings of the International Workshop on Support Vector Machines, Niagara Falls, ON, Canada, 10 August 2002; Springer: Berlin/Heidelberg, Germany, 2002; pp. 53–67.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.