# AI-Based Anomaly Detection and Classification of Traffic Using Netflow

Gustavo Gonzalez Granadillo[1][a] and Nesrine Kaaniche[2][b]
[1]*Schneider Electric, DCR Security Department, Barcelona, Spain*
[2]*SAMOVAR, Telecom SudParis, Institut Polytechnique de Paris, France*

Keywords: Anomaly Detection, Network Traffic Behavior, Classification Algorithms, NetFlow.

Abstract: Anomalies manifest differently in network statistics, making it difficult to develop generalized models for normal network behaviors and anomalies. This paper analyzes various Machine Learning (ML) and Deep Learning (DL) algorithms employing supervised techniques for both binary and multi-class classification of network traffic. Experiments have been conducted using a validated NetFlow-based dataset containing over 31 million incoming and outgoing network connections of an IT infrastructure. Preliminary results indicate that no single model effectively detects all cyber-attacks. However, selected models for binary and multi-class classification show promising results, achieving performance levels of up to 99.9% in the best of the cases.

## 1 INTRODUCTION

The rapid evolution of cyber threats has necessitated equally advanced defense mechanisms, with artificial intelligence (AI) emerging as a cornerstone of modern cybersecurity strategies. Recent scientific research underscores AI's dual role: while it empowers attackers to launch sophisticated campaigns, it also provides defenders with unprecedented tools to detect, analyze, and mitigate threats (Rahman et al., 2025). Key conclusions from peer-reviewed studies and industry analyses reveal that AI-driven detection systems excel in identifying anomalies, automating responses, and adapting to dynamic attack vectors. However, challenges such as explainability, adversarial AI manipulation, ethical approaches, and the need for continuous innovation persist (Khanna, 2025), (Mohamed, 2023).

While AI has demonstrated its ability to reduce computational complexity, model training time, and false alarms, there is a notable limitation in the intrusion detection domain. Most studies focus on a limited number of algorithms, with the support vector machine being the predominant technique utilized (Wiafe et al., 2020).

In this paper, we propose to analyze an IT network traffic dataset containing Netflow data (e.g., protocols, source and destination IP, source and destination port, packets, etc.) aiming to classify legitimate and malicious traffic. The selected dataset has been created in 2017 by Hochschule Coburg, a cybersecurity research institute in Germany, and consists of 4 weeks of labeled internal and external communications, containing more than 31 million observations of normal traffic and cybersecurity incidents (e.g., Port Scans, Ping Scans, Brute Force, and Denial of Service). The objective of this work is to model internal communications to detect anomalies (malicious communications) with a high degree of accuracy that complements the information presented by commercial Intrusion Detection Systems (IDSs) and helps in decision making.

The contributions of this paper are summarized as follows:

- A binary classification analysis using supervised learning techniques to compare the performance of eight algorithms, enabling the selection of the most effective method for classifying legitimate versus malicious traffic.
- A multi-class classification analysis employing supervised learning techniques to predict, not only when the traffic is normal or malicious, but also to identify the specific type of anomaly associated to the malicious traffic (e.g., Brute Force, Denial of Service (DoS), Port Scan, or Ping Scan).

A set of experiments have been conducted using multiple features of flows in order to identify the most predictive variables and the best approach to detect

[a] https://orcid.org/0000-0003-2036-981X
[b] https://orcid.org/0000-0002-1045-6445



Granadillo, G. G. and Kaaniche, N.
AI-Based Anomaly Detection and Classification of Traffic Using Netflow.
DOI: 10.5220/0013552700003979
In *Proceedings of the 22nd International Conference on Security and Cryptography (SECRYPT 2025)*, pages 644-649
ISBN: 978-989-758-760-3; ISSN: 2184-7711

anomalies in a given traffic using NetFlow data. Results show a promising approach using Random Forest for the binary supervised classification approach and an ensemble of five ML algorithms for the multi-class supervised classification approach.

The remainder of this paper is structured as follows. Section 2 presents the methodology used to analyze the dataset and create the prediction models. Section 3 describes the experimentation and results obtained for the binary and multi-class classification approach using supervised learning techniques. Section 4 discusses related work. Finally, conclusions and perspective for future work are presented in Section 5.

# 2 AI-BASED ANOMALY DETECTION METHODOLOGY

This section describes the methodology followed to build, test, and validate our proposed AI-based anomaly detection models.

## 2.1 Data Access and Collection

Most of the existing datasets to classify network traffic are old and insufficient in terms of understanding the behavioral patterns of current cyber-attacks. A great number of research work is still performed using DARPA and KDD databases, which are at least 20 years old, and which lack instances related to new sophisticated attacks (Sarker et al., 2020), (Mubarak et al., 2021). The dataset used in this research has been created by the Corbug University of applied sciences and arts in Germany in 2017. The Coburg Intrusion Detection Data Sets (CIDDS) is a public and open dataset that can be downloaded from the official university's website[1]. There are two datasets available: CIDDS-001, with over 31 million flows; and CIDDS-002, with over 16 million flows, both containing NetFlow data with benign and malicious traffic.

For this analysis, we have selected the CIDDS-001 dataset which emulates a small business environment composed of an internal and external network that include typical IT clients and servers e.g., web, file, mail, backup, etc. The complete dataset has been captured during 4 weeks and has more than 31 million flows (observations) from which 89.66% are benign (flows labeled as *normal*) and 10.34% are malicious (flows labeled as *attacker* or *victim*). In addition, malicious flows are further labeled as *BruteForce*, *Dos*, *PingScan* or *PortScan* as shown in Table 1.

[1]https://www.hs-coburg.de/forschung/forschungsprojekte-oeffentlich/informationstechnologie/cidds-coburg-intrusion-detection-data-sets.html

Table 1: CIDDS-001 Dataset.

| Weeks | Week 1 | Week 2 | Week 3 | Week 4 |
|---|---|---|---|---|
| Normal | 7010897 | 8515329 | 6349783 | 6175897 |
| DoS | 1252127 | 1706900 | 0 | 0 |
| PortScan | 183511 | 82404 | 0 | 0 |
| PingScan | 3359 | 2731 | 0 | 0 |
| BruteForce | 1626 | 336 | 0 | 0 |

## 2.2 Data Preparation and Analysis

In order to clean the dataset and prepare it for its usage during the training and testing phases, we have performed the following actions:

- The *Flows* column has been removed as all observations present the same value.
- For the binary classification, *attackType*, *attackID*, and *attackDescription* have been removed; for the multi-class classification, the *class* column has been removed.
- *Src IP Addr*, and *Dst IP Addr* have been removed since these features may change in future observations.

New features have been created and some categorical features have been transformed into numeric values before they were used as input to the learning algorithms. The following transformations have been performed in the selected dataset:

- The *class* variable has been transformed into a binary feature with 0 representing the normal category and 1 representing an attack.
- The *Bytes* variable presents some observations with a non-numeric format (e.g., 1.0 M, which corresponds to 1 million bytes), this variable has been transformed into its corresponding numeric value (e.g., 1.0 M = 1000000).
- *Flags* and *Proto* variables have been transformed into their numeric values (e.g., .....F= 1, .A..S.=18, TCP=6, UDP=17).
- *Date first seen* has been split into four columns (Day, Hour, Minute, Second), since the data has been captured the same month and year, these two options were not considered in the transformation.
- Two new features have been created: (i) *Packets_speed*, the number of packets divided by the duration; and (ii) *Bytes_speed*, the number of bytes divided by the duration

After the transformation, all numeric variables have been standardized by using the mean and stan-

dard deviation of the whole dataset. The resulting values range between -3 and 3.

### 2.3 Model Tuning and Training

For binary classification we selected a sample of 10,486 observations (from which 88.2% are normal flows and 11.8% are attack flows). For the multi-class classification we selected a sample of week 2 containing 13.613 observations (from which 79.9% belong to the normal category and 21.1% belong to the attack category). This latter is further divided into DoS, Brute Force, Port Scan and Ping Scan.

During the training process, all algorithms are tuned to obtain the parameters and hyper-parameters that best fit the model. Logistic Regression has no parameters to tune but it has been used to identify the best set of predictable variables through the classic standard method Stepwise AIC and BIC. Several tests have been performed with various set of variables to which we applied cross-validation of 4 groups and 5 repetitions. As a result, the best model suggests the use of the following variables: *Tos*, *Flags*, *Duration*, *Bytes_speed*, *SrcPt*, *Packets*, *Hour*, and *Packets_speed*.

### 2.4 Performance Evaluation

A variety of metrics are used to evaluate and analyze the performance of classification algorithms. Accuracy, precision, recall, and F1 scores are among the most popular performance indicators currently used in the literature (Osi et al., 2020). In addition, we have evaluated the anomaly detection rate of each classification model. Anomaly detection is used to define the ratio of the total number of correctly classified negative instances (e.g., anomalous connections due to attacks) over the total number of negative prediction.

## 3 EXPERIMENTATION AND RESULTS

This section describes the algorithms used for the binary and multi-class classification problems studied, as well as the main results obtained during the testing phase.

### 3.1 Binary Classification Using Supervised Learning Techniques

We have selected eight algorithms covering various supervised learning techniques, e.g., regression, decision trees, artificial neural networks, support vector machine, clustering, and Bayesian networks.

- **Logistic Regression (LR):** The best model uses the following eight variables: *Tos, Flags, Duration, Bytes_speed, SrcPt, Packets, Hour, Packets_speed* and provides failure rate of 0.0360 and an Area Under the Curve (AUC) of 0.9411.

- **Artificial Neural Networks (NNET):** The best model is composed of 15 artificial neurons, a *decay* of 0.01 and max iterations *maxit* of 100. It uses model averaging, where the same neural network model is fit using different random number seeds. The winner model provides a failure rate of 0.098 and an AUC of 0.9930.

- **Random Forest (RF):** The best model uses mtry=7, ntree=150 and sampsize=7000, resulting in a failure rate of 0.0098 and an AUC of 0.9930.

- **Gradient Boosting (GBM):** The best model uses n.trees=3000, interaction.depth=2, shrinkage=0.1, and n.minobsinnode=20, resulting in a failure rate of 0.0016 and an AUC of 0.9992.

- **eXtreme Gradient Boosting (XGBM):** The best model uses min_child_weight=5, eta=0.1, and nrounds=3000, resulting in a failure rate of 0.0020 and an AUC of 0.9995. Using this algorithm, variables such as *Tos* and *Hour* have a very low prediction power.

- **Support Vector Machine (SVM):** The best model uses C=100, resulting in a failure rate of 0.0353 and an AUC of 0.9437.

- **K-Nearest Neighbor (KNN):** The best model uses k=1, resulting in a failure rate of 0.0040 and an AUC of 0.9911.

- **Naïve Bayes (NB):**The best model has usekernel=True, fl=0, and adjust=1, resulting in a failure rate of 0.0917 and an AUC of 0.9874.

All models have gone through a second cross-validation process with 4 groups and 10 repetitions. Table 2 presents the confusion matrix and the results obtained for the accuracy, precision, recall, and F1-score metrics, as well as the average prediction time per million observations.

The best model in terms of prediction of both benign and malicious traffic is the Random Forest. We

Table 2: Testing Results for Binary Classification using Supervised Learning Techniques.

| Metrics | LR | NNET | RF | GBM | XGBM | SVM | KNN | NB |
|---|---|---|---|---|---|---|---|---|
| TP | 27,486,170 | 28,014,392 | 28,030,675 | 28,028,887 | 28,026,05 | 27,272,577 | 27,920,149 | 27,922,964 |
| FP | 451,444 | 468,968 | 29,312 | 68,923 | 46,655 | 366,613 | 2,353,599 | 483,397 |
| FN | 565,736 | 37,514 | 21,231 | 23,019 | 25,855 | 779,329 | 131,757 | 128,942 |
| TN | 2,784,583 | 2,767,059 | 3,206,715 | 3,167,104 | 3,189,372 | 2,869,414 | 882,428 | 2,752,630 |
| Accuracy | 0.9675 | 0.9838 | 0.9984 | 0.9971 | 0.9977 | 0.9634 | 0.9206 | 0.9804 |
| Recall | 0.9838 | 0.9835 | 0.9990 | 0.9975 | 0.9983 | 0.9867 | 0.9223 | 0.9830 |
| Precision | 0.9798 | 0.9987 | 0.9992 | 0.9992 | 0.9991 | 0.9722 | 0.9953 | 0.9954 |
| F1-Score | 0.9818 | 0.9910 | 0.9991 | 0.9984 | 0.9987 | 0.9794 | 0.9574 | 0.9892 |
| Anom_Det. | 0.8311 | 0.9866 | 0.9934 | 0.9928 | 0.9920 | 0.7864 | 0.8701 | 0.9553 |
| Pred_Time | 4.74 | 11.68 | 4.79 | 37.76 | 3.92 | 2.84 | 70.86 | 569.27 |

have built some ensemble models, but as the improvement is almost insignificant, we decided to discard it from the analysis and avoid more complex models. In general, all models provide a good performance, with scores ranging from 92.06% to 99.92%. The best models are those using decision trees algorithms (i.e., RF, GBM, and XGBM)

Regarding the prediction time, we have noticed that some algorithms are much faster than others. For these evaluations, we have used a computer with an Intel(R) Core(TM) i7-10870H processor, 2.20GHz and 16GB of RAM with a Windows 11 64-bit Operating System. Additionally, it has an NVIDIA GeForce RTX 3060 graphics card with 6GB of display memory (and 8GB of shared memory).

The model created with the Random Forest algorithm, not only offers the best prediction results, but also performs the predictions very quickly (less than five seconds per million observations). The fastest of all algorithms is SVM (2.84 seconds per million observations) and the slowest is Naïve Bayes (9.49 minutes per million observations). The Random Forest model has been selected as the winner for the binary classification analysis using supervised learning techniques.

## 3.2 Multi-Class Classification Using Supervised Learning Techniques

During this phase, we tuned and trained six algorithms with cross-validation of 4 groups and 5 repetitions using all 14 variables present in the dataset. From the list of previous supervised learning algorithms we discarded logistic regression, as this is not a binary classification problem; and Naïve Bayes, as the prediction time was too high.

- **Artificial Neural Networks (NNET):** The best model is composed of 20 artificial neurons, a *decay* of 0.01, and *maxit* of 100. It uses model averaging and provides an accuracy of 0.9264.
- **Random Forest (RF):** The best model uses *mtry=6*, *ntree=300*, and *sampsize=default*, resulting in an accuracy of 0.9949. The most predictable variables are *DstPt*, *SrcPt*, *Flags*, *Packets*, and *Bytes*.
- **Gradient Boosting (GBM):** The best model uses *n.trees=500*, *interaction.depth=2*, *shrinkage=0.1*, and *n.minobsinnode=20*, resulting in an accuracy of 0.9958. The most predictable variables are *DstPt*, *Duration*, *Flags*, *Bytes*, *Packets_speed*, *Packets*, *SrcPt*, and *Proto*.
- **eXtreme Gradient Boosting (XGBM):** The best model uses *min_child_weight=5*, *eta=0.1*, and *nrounds=1000*, resulting in an accuracy of 0.9959. The most predictable variables are *DstPt*, *Flags*, *Bytes*, *Packets*, *Duration*, *SrcPt*, *Packets_speed*, *Proto*, and *Tos*.
- **Support Vector Machine (SVM):** The best model uses *C=500*, resulting in an accuracy of 0.9458.
- **K-Nearest Neighbor (KNN):** The best model uses *k=1*, resulting in an accuracy of 0.9696.

For the testing phase, we have evaluated the dataset from weeks 1 and 2 (as the other two weeks do not present anomalous observations). The metrics used in this analysis include Precision, Recall and F1-Score (Bex, 2021), as well as the time spent by each model in making predictions. Figures 1 and 2 show the preliminary results obtained in this phase.

As previously shown, the models that offer the highest performance in detecting most of the attacks are those using decision trees algorithms (i.e., RF, GBM, and XGBM). However, the detection of Ping Scans is very low (57% in the best of the cases for the F1-score), with algorithms (e.g., NNET) that fail to detect this category entirely. This is very likely due to the fact that there are not enough observations of this

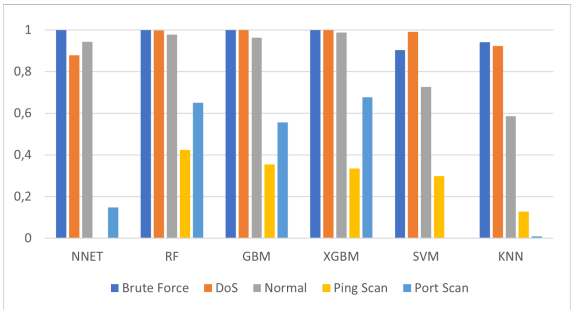


Figure 1: Precision.

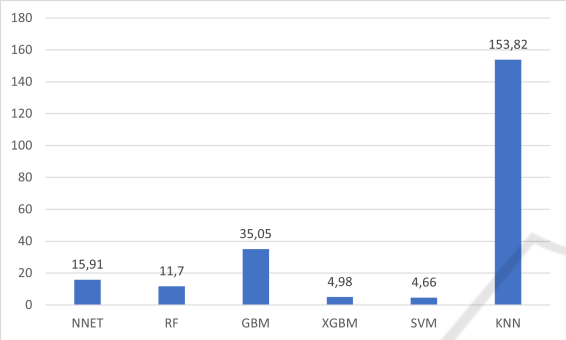


Figure 2: Average Prediction Time.

class in the training dataset. Similarly, the detection of brute force attacks reaches 77% in the best of the cases for the F1-score, with algorithms (e.g., SVM) that fail to detect this category entirely.

In terms of prediction time, the fastest algorithm is the SVM, which perform predictions with a rate of 4.66 seconds per million observations. The slowest algorithm is the KNN, with a rate of 2.56 minutes per million observations.

Aiming to improve the detection rate of the categories for Ping Scans and Brute Force attacks, we developed ensemble models using the majority voting method, where predictions are based on the class predicted by the majority of models. Since we have six models, we built ensembles of two, three, four, five, and six models. The winner model in this phase is an ensemble composed of five models (i.e., XGBM, GBM, RF, NNET, and SVM). This ensemble improves the prediction of Ping Scans to 62.17% and Brute Force attacks to 83.67% while keeping the prediction of all other classes higher than 97%.

## 4 RELATED WORK

AI-based techniques have been widely proposed in the literature as a viable approach for network anomaly detection. Deplace et al. (Delplace et al., 2020), present Machine Learning (ML) and Deep Learning (DL) models for the detection of botnets using Netflow data sets. The results show that the Random Forest (RF) classifier performs the best in 13 different scenarios with an accuracy greater than 95%. One of our previous works (Gonzalez-Granadillo et al., 2019) uses the One-CLass SVM algorithms to classify network traffic as anomalous and normal based on unlabeled NetFlow data. Such approaches provide promising results, although they are limited to the type of dataset used in the training and testing processes.

Several researchers (Garg et al., 2019), (Gu et al., 2019) developed ML models to improve the performance of Intrusion Detection Systems. The focus has been on developing optimized features and improving classifiers to reduce false alarms. However, although the results are outstanding (reaching more than 99% in terms of accuracy), most of these works have been tasted and validated against outdated datasets, such as DARPA (1998), and KDD (1999), for which the characteristics and volume of attacks have changed significantly since that time.

Current research (Rabbani et al., 2021) proposes to perform the feature extraction by using collection tools such as Bro IDS and Argus or to complement features extracted with NetFlow. Although the authors extracted 49 features with their approach, Random Forest reached optimal performance with 11 features, whereas Logistic Regression did it with 18 features. Results demonstrated more than 99% of detection accuracy and 93.6% of categorization accuracy, with the inability to categorize three out of eight attacks due to feature similarities and unbalanced data.

Our work differs from previous works in the way that it defines new features to be used for evaluating the performance of several ML algorithms for binary and multi-class detection using supervised learning techniques. In total we have evaluated eight algorithms and have compared against potential ensembles to detect the best model that classifies network traffic. To overcome the dataset issues, we used a big and updated dataset developed for the training and evaluation of several cyber-attacks.

## 5 CONCLUSION AND FUTURE WORK

In this paper we have analyzed multiple machine learning algorithms to detect network traffic anomalies (i.e., Brute-Force attacks, DoS attacks, Port Scans, and Ping Scans) based on the analysis of a variety of NetFlow features. Two main analysis have been carried out in this paper: (i) Binary classification

using supervised learning algorithms; and (ii) Multi-class classification using supervised learning algorithms.

The best performance is obtained with the Random Forest, in the binary classification using supervised learning techniques; and an ensemble of five algorithms (i.e., XGBM, GBM, Random Forest, Neural Networks and SVM) in the multi-class classification using supervised learning techniques.

The main limitation of our proposed approach is in terms of explainability. While some models (e.g., Logistic Regression, Random Forest, Naive Bayes) provide indicators of the prediction power of each variable, some others (e.g., neural networks, SVM, ensembles) are limited in terms of explainability / interpretability, preventing us from identifying the key feature or group of features that contributes the best in classifying the connections. It is, therefore, not possible to assign weights to each feature based on their contribution to accurately classify the data.

The analysis performed in this paper is also limited to the use of NetFlow data. Although some approaches suggest the use of datasets that combine Argus with tools such as Wireshark or Bro IDS (Rabbani et al., 2021), ours requires the input data feeding the model to be transformed using NetFlow. There are several advantages of this approach over packets, such as PCAP (packet capture) dumps, e.g., keeping only certain information from network packet headers and not the whole payload. Therefore, the processing and analysis of the data yields more interesting performance results (since a single flow can represent thousands of packets) enabling almost real-time analysis.

Future work will consider using unsupervised learning approaches and a training sample with a greater number of attack instances (especially *Ping Scan]* and *Brute Force*) to see if it is possible to improve detection rates on these classes. Having a balanced dataset would also solve the issue. In addition, it is important to evaluate datasets with other network attacks (e.g., botnet, malware, man-in-the-middle, phishing, etc.) to verify up to which extent the developed models are able to detect attacks different from those existing in the training dataset.

# REFERENCES


Bex, T. (2021). Comprehensive guide to multiclass classification metrics. In Towards data Science. Available at https://towardsdatascience.com/comprehensive-guide-on-multiclass-classification-metrics-af94cfb83fbd.

Delplace, A., Hermoso, S., and Anandita, K. (2020). Cyber attack detection thanks to machine learning algorithms. ArXiv preprint, availabble at https://arxiv.org/abs/2001.06309.

Garg, S., Kaur, K., Kumar, N., Kaddoum, G., Zomaya, A. Y., and Ranjan, R. (2019). A hybrid deep learning based model for anomaly detection in cloud datacentre networks. *IEEE Trans. Netw. Serv. Manag.*

Gonzalez-Granadillo, G., Diaz, R., Medeiros, I., Gonzalez-Zarzosa, S., and Machnicki, D. (2019). Lads: A live anomaly detection system based on machine learning methods. In *Security and Cryptography, SECRYPT*.

Gu, T., Chen, H., Chang, L., and Li, L. (2019). Intrusion detection system based on improved abc algorithm with tabu search. *IEEE Trans. Electr. Electron. Eng.*, 14.

Khanna, S. (2025). AI in cybersecurity: A comprehensive review of threat detection and prevention mechanisms. *International Journal of Sustainable Devlopment in Field of IT*, 17.

Mohamed, N. (2023). Current trends in ai and ml for cybersecurity: A state-of-the-art survey. *Cogent Engineering*, 10(2):2272358.

Mubarak, S., Habaebi, M. H., Islam, M. R., Rahman, F. D. A., and Tahir, M. (2021). Anomaly detection in ics datasets with machine learning algorithms. *Computer Systems Science & Engineering*.

Osi, A. A., Abdu, M., Muhammad, U., Ibrahim, A., Isma'il, L. A., Suleiman, A. A., Abdulkadir, H. S., Sada, S. S., Dikko, H. G., and Ringim, M. Z. (2020). A classification approach for predicting COVID-19 patient's survival outcome with machine learning techniques. MedRxiv, the preprint server for heatlh sciences.

Rabbani, M., Wang, Y., Khoshkangini, R., Jelodar, H., Zhao, R., Ahmadi, S. B. B., and Ayobi, S. (2021). A review on machine learning approaches for network malicious behavior detection in emerging technologies. *Entropy*, 23:529.

Rahman, M., Uddin, I., Das, R., Saha, T., Haque, E. S. M., Shatu, N. R., and Shafiq, S. I. (2025). Application of artificial intelligence in detecting and mitigating cyber threats. *International Research Journal of Innovations in Engineering and Technology (IRJIET)*, 9:17–26.

Sarker, I. H., Kayes, A. S. M., Badsha, S., Alqahtani, H., Watters, P., and Ng, A. (2020). Cybersecurity data science: an overview from machine learning perspective. *Journal of Big Data*, 7(41).

Wiafe, I., Koranteng, F. N., Obeng, E. N., Assyne, N., and Wiafe, A. (2020). Artificial intelligence for cybersecurity: A systematic mapping of literature. *IEEE Open Access Journal*, 8.