Review and Analysis of Cyberattack Detection Methods in Corporate Networks

Ioseb Kartvelishvili

Candidate of Technical Sciences, Professor, Georgian Technical University, Tbilisi, Georgia.

***Email**: s.kartvelishvili@gtu.ge*

Giorgi Kuchava

PhD (Engineering of Informatics), Associate professor, Georgian Technical University, Tbilisi, Georgia.

***Email:** kuchavagiorgi08@gtu.ge*

Demetre Chakvetadze

Master of Information Systems Management (DevOps), Business and Technology University, Tbilisi, Georgia.

***Email:** demetre.chakvetadze.1@btu.edu.ge*

Abstract

This paper presents and analyzes existing methods for detecting network attacks. The purpose of reviewing the principal existing attack-detection technologies is to examine the effectiveness of the attack-detection technologies currently available, along with their main advantages and shortcomings. On the basis of the methods discussed, it can be concluded that, in terms of software implementation and the quality of attack detection, the signature-based detection method remains one of the most effective approaches. The vast majority of contemporary systems rely solely on the signature-based method to identify the impact of an attack, or combine it with mechanisms for detecting anomalies in monitored network behavior.

Keywords: Intrusion Detection Systems, Signature Analysis Method, Statistical Analysis Method, Artificial Neural Network, Graphical Model, Biometric Method, Cluster Analysis Method.

1.Introduction

Today, the development of intrusion detection systems constitutes an active area of research, both theoretically and practically. For example, there exist neural-network-based intrusion detection systems that operate on the basis of analyzing web-server files. There are also systems based on neural networks that are used to detect anomalies in the behavior of users and network traffic. Certain works rely on the intellectual capabilities of neural networks, drawing on the foundations of fuzzy logic mechanisms and analogies with biological systems.

From a functional standpoint, the aforementioned systems exhibit certain shortcomings, which make it possible to link a typical attacker's behavioral model to events occurring within the network and the local computing environment. Information about network events is treated as external input data, and the essence of the models and systems developed for processing this data is thereby reduced accordingly. Issues concerning the development of a system for receiving input information fall outside the scope of work on attack-detection methods. This leads to a situation in which the quality of attack detection may suffer when only certain technical parameters of the network are taken into account.[1,2]

2.Methodology

Among the currently existing approaches to computer attack detection, two principal approaches can be distinguished: misuse detection and anomaly detection.

1. **Misuse detection** relies on a predefined model of attack (malicious behavior) and compares the flow of events within a system against known attack patterns. If the behavior of an object matches the description of a known attack, such behavior is then classified as an attack.

2. **Anomaly detection** relies on a model of the normal functioning of a system and identifies an anomalous intrusion within the flow of events as one that represents a significant deviation from baseline normal behavior.

Intrusion detection systems are characterized by the presence of Type I and Type II errors: a **Type I error (false positive)** occurs when normal, legitimate actions of the system are mistakenly identified as an attack. A **Type II error (false negative)** occurs when an actual attack, one that does not match the criteria of anomalous behavior or existing attack patterns, evades the system and remains undetected. In line with these approaches, various methods of attack detection are employed (Fig. 1, Fig. 2). For the purpose of illustrating the quantitative comparison of Type I and Type II errors, the following formulation may be adopted, according to which intrusion detection systems are evaluated on the basis of a confusion matrix, within which four basic quantities are distinguished: true positive (TP) a correctly detected attack, true negative (TN),

Fig. 1. Schematic diagram of a corporate network architecture with an integrated Firewall/IDS/IPS system.[2]

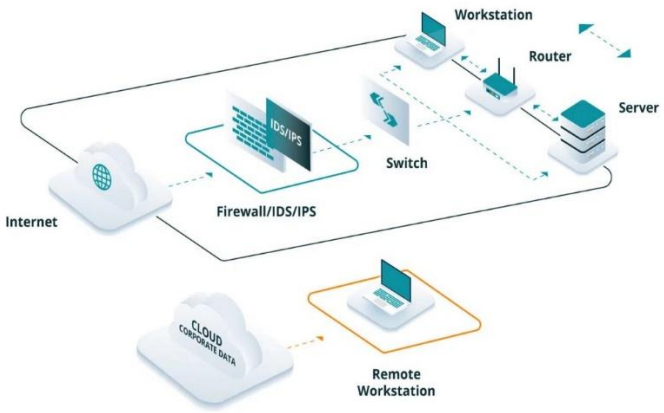
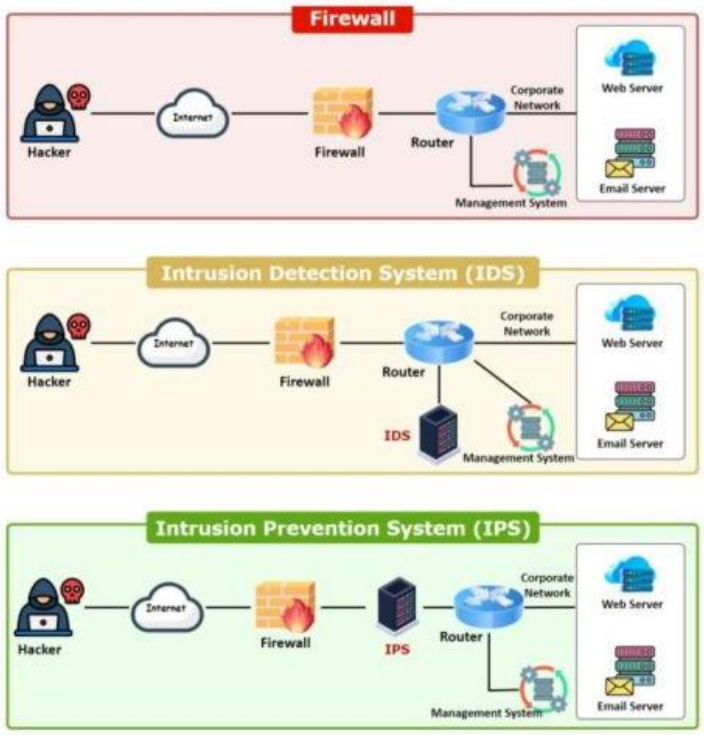


Fig. 2. Comparative placement of Firewall, Intrusion Detection System (IDS), and Intrusion Prevention System (IPS) within a corporate network architecture



a correctly disregarded normal action; false positive (FP); and false negative (FN). On the basis of these quantities, the principal computed metrics are as follows:

Table 1. Key Evaluation Metrics for Intrusion Detection Systems Based on the Confusion Matrix.

Metric	Formula	What It Indicates
Recall (TPR)	TP / (TP + FN)	What proportion of actual attacks the system correctly detects
FPR	FP / (FP + TN)	What proportion of legitimate actions are incorrectly flagged
Precision	TP / (TP + FP)	Of all instances labeled as "attack," how many are genuine
F1-score	2·(P·R) / (P + R)	The harmonic mean of Precision and Recall

Accuracy	(TP+TN) / (TP+TN+FP+FN)	Overall correctness rate (limited in usefulness for imbalanced data)
ROC-AUC	∫ TPR d(FPR)	The system's discriminative ability across all possible threshold values

In practice, Accuracy alone is insufficient, since normal actions in network traffic also substantially outweigh the presence of attacks (highly imbalanced data); therefore, when comparing methods, the joint consideration of Precision/Recall together with the F1-score plays the leading role.

Real comparison of methods is possible only on the basis of common, standardized datasets and testing thereon. The most frequently used datasets are:

Table 2. Standardized Datasets Commonly Used for Evaluating and Comparing Intrusion Detection Methods.

Dataset	Year	Characteristics
KDD Cup 99 / NSL-KDD	1999 / 2009	2009 A classic baseline dataset; the NSL-KDD version was introduced to address the problem of duplicate records
UNSW-NB15	2015	Contains nine modern attack categories, comprising a mixture of synthetic and real traffic
CICIDS2017	2017	Realistic, time-stamped traffic featuring DDoS,

		Brute-Force, Botnet, and other contemporary attacks
CSE-CIC-IDS2018	2018	An extended version of CICIDS2017, incorporating simulation of a large-scale infrastructure

Let us examine the existing methods for detecting network attacks:

Signature Analysis Method:

The signature analysis method is based on the fact that the majority of attacks on a system are known and unfold according to similar scenarios.

Attack signatures define the characteristics, conditions, and interrelationships of events that give rise to intrusion attempts or an actual intrusion. The maintenance of a database of intrusion signatures by a security system represents the simplest, yet at the same time the most widespread, means of implementing the signature method. During operation, the sequence of actions performed by a user or a program is compared against known signatures. A sequence of events corresponding to a signature may serve as an indication of an attempted security breach.

The signature method operates at the lowest level of abstraction and analyzes data that is either transmitted directly over the network or processed locally. The most common implementations are various fast-search algorithms.

Typical representatives that employ signature analysis methods include antivirus scanners, which work with a database of virus signatures, and the majority of network intrusion detection systems, which work with a database of network attack signatures. Signatures of network attacks may take the form of a specific sequence of bytes within network packets. The signature method is notable for offering the best speed of operation, although such methods are not adaptive. This group of methods is, by other criteria, globally robust and well-validated.

Owing to their specific nature, signature methods are used primarily for detecting known violations; nevertheless, there exist works that demonstrate the feasibility of applying them within an anomaly-detection approach as well.[4]

Widely used open-source and commercial IDS/IPS systems based on the signature analysis method include: Snort, Suricata, Zeek (Bro), OSSEC / Wazuh, and YARA.

Statistical Analysis Method:

The statistical analysis method is based on constructing a profile of the "normal" behavior of objects over a certain period of time. Normal behavior profiles are used for monitoring users, system activity, or network traffic. Subsequent observations are compared against the expected values of the normal behavior profile, which is constructed during the training period of the intrusion detection system. However, when operating systems that rely on training and profiles, the following problems arise:

- The construction of a profile is a poorly formalized and labor-intensive task, requiring a substantial amount of predefined preliminary work;
- Difficulties arise in determining the threshold values of statistical characteristics so as to reduce the probability of Type I or Type II errors;
- It is impossible to construct a static profile for systems whose characteristics change over time, or, indeed, some users' actions, owing to the nature of their work, are simply not amenable to profiling.

Artificial Neural Network:

An artificial neural network is a mathematical model constructed on the principle of organization underlying the biological neural networks inherent in living organisms. Methods based on the use of artificial neural networks fall within the class of analytical methods, which are built upon hypothetical principles governing the learning and functioning of thinking beings' brains, thereby making it possible to predict the values of certain variables in new observations on the basis of data from other observations.

With the aid of artificial neural networks, the contemporary world has addressed problems of pattern recognition, forecasting, optimization, associative memory, control, and

many others. Methods based on artificial neural networks offer the following advantages in connection with intrusion detection systems:

Advantages:

- The ability to work with incomplete or distorted data;
- The capacity to predict the subsequent development of malicious impact;
- Adaptability to previously unknown types of attacks;

However, like all analytical methods based on artificial neural networks, they also possess certain drawbacks, such as:

- Local stability;
- The complexity of the training process and sample selection;
- The impossibility of understanding the underlying decision-making logic (Fig. 2);

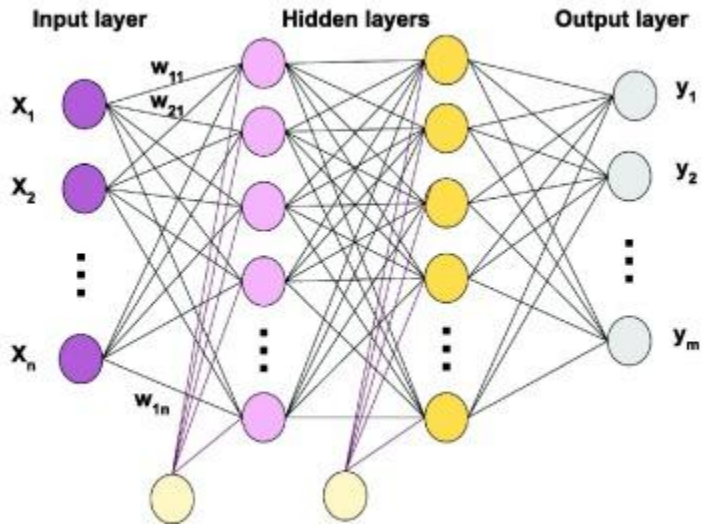


Fig. 3. Neural Network

The term "artificial neural network" collectively encompasses several architecturally distinct approaches, each of which serves a different purpose in the detection of network attacks. [6]

Table 3. Neural Network Architectures Employed in Intrusion Detection Systems.

Architecture	Principle	Application in IDS

CNN	Recognition of spatial patterns by means of convolutional filters	Analysis of features derived from representing packet payloads or traffic as "images"
RNN / LSTM	Retention of sequential, time-dependent data	Detection of the temporal dynamics of network flows and sequential anomalies
Autoencoder	Unsupervised learning based on reconstruction error	Anomaly detection without labeled data; a high reconstruction error indicates suspicion
GAN	Adversarial learning between a generator and a discriminator	Enrichment of rare attack classes with synthetic data; robustness testing
Transformer	Attention mechanism for long-range dependencies	Contextual analysis of long traffic sequences; the most recent research direction

Graphical Model:

A number of works exist in which information on possible attacks against computer networks is represented in the form of a graphical model. On the basis of the graphical structure, it is possible to determine all possible modeled attack scenarios. For example, in order to construct such a model, several sets may be defined: the set of vulnerabilities of the automated system — V, the set of methods for carrying out attacks — A, and the set of consequences of attacks — C. Each element (A, V, C) of the Cartesian product $A \times V \times C$ is then treated as a form of informational attack, carried out by an attacker A through the exploitation of vulnerabilities V, by means of a given method, resulting in consequence C.

Such models impose no restrictions on the possible sources and objects of an attack, nor on the methods of its execution, and are therefore universal in character; however, for an arbitrary network, it is difficult to formally define all of these sets. For this reason, graphical models are better suited to the modeling of various information subsystems and to the assessment of any given attack and its associated risk. The application of such methods within real-time intrusion detection systems is difficult, owing to the high computational complexity involved and the need for expert determination of the numerous sets of system parameters.

Biometric Method:

The biometric method encompasses a system for recognizing individuals on the basis of one or more physical or behavioral characteristics. Behavioral biometrics comprises methods that do not require specialized technical equipment, such as fingerprint scanning, retinal scanning, or voice recording. Behavioral biometric methods are employed in intrusion detection systems that rely on "keystroke dynamics" and the nature of mouse usage characteristic of each individual system user. These methods are based on the hypothesis that different users exhibit a distinct "handwriting" in the way they interact with data-entry interfaces. On the basis of a constructed profile of normal behavior for a given system user, deviations are identified that result from attempts by other individuals to work with the keyboard or other devices.

Cluster Analysis Method:

The essence of the cluster analysis method lies in representing the data of a system as a set of feature vectors and subsequently partitioning this set of feature vectors into clusters. Among the clusters, those data associated with normal behavior are distinguished, while the remainder are considered abnormal. Each particular method employs its own metric in order to partition feature vectors into clusters, or to assign a given feature vector to a specific cluster. The cluster analysis method makes it possible to construct adaptive systems. The cluster analysis method is stable within the particular system in which the data used to form the clusters was collected. For other systems, a repeated training procedure will be required, as will, potentially, a different set of feature vectors.

Table 4. Comparative Summary of the Principal Advantages of Intrusion Detection Methods.

Method	Principal Advantage
Signature Analysis	High speed and accuracy with respect to known attacks
Statistical Analysis	Well suited for long-term monitoring of traffic behavior
Artificial Neural Network	Adapts to previously unknown types of attacks
Graphical Model	Universal, applicable to any attack scenario
Biometric Method	Requires no additional technical equipment
Cluster Analysis	Adaptive; does not require labeled data

The introduction of machine learning methods into intrusion detection has given rise to a new, specific risk: the detection model itself may become the object of an attack:

Evasion attack — the attacker deliberately alters the characteristics of malicious traffic to a minimal degree, such that it appears normal to the model, while in reality the effect of the attack is preserved.

Poisoning attack the deliberate introduction of malicious records into the training data, which corrupts the model's training process and reduces the reliability of its future detections.

Model extraction / inversion an attempt, through repeated querying of the model, to reconstruct its internal logic or the properties of its training data.

To mitigate these risks, the following are employed: adversarial training (training the model with the inclusion of perturbed examples), robust feature selection, and the joint

(ensemble) decision-making of several independent models, which renders the deception of a single model less sufficient for misleading the system as a whole.[7]

In recent years, the field of network attack detection has developed significantly as a result of the active adoption of Deep Learning and artificial intelligence methods. Contemporary research indicates that, rather than a single method, hybrid approaches are increasingly employed, ones that combine the advantages of several algorithms simultaneously.

Deep-learning-based methods make it possible to recognize complex, non-linear patterns in traffic and to identify so-called zero-day attacks;

Hybrid models (Random Forest, XGBoost in combination with other algorithms) increase detection accuracy and reduce false positives compared with individual methods;

Explainable AI (XAI) provides transparency in decision-making logic, thereby addressing the principal shortcoming of traditional neural networks;

Integration with SIEM/SOAR systems and threat intelligence data ensures real-time correlation of events and automated response;

Zero Trust architecture and User and Entity Behavior Analytics (UEBA) broaden the scope of application of biometric and behavioral methods;

Federated Learning makes it possible for several organizations to jointly train a model without compromising data confidentiality.

3.Results and Discussion

To move from a purely theoretical comparison to a practical verification, five representative detectors were selected — one for each of the method groups discussed above — which were actually implemented in Python and evaluated on the same, publicly available test data, in order to compare the theoretical claims presented above against real numerical results.

Dataset: NSL-KDD (KDDTrain+ / KDDTest+)

Training / Test: 125,973 / 22,544 records

Tools: Python 3, scikit-learn 1.8

The NSL-KDD test set is deliberately constructed to include attack subtypes not present in the training set, which renders it a fairly useful, if modest, means of testing a model's ability to generalize to unknown attacks — a matter directly connected to the comparative discussion of signature-based and anomaly-based detection methods presented above.

- Decision Tree a fast, rule-based classifier, employed here as an analogue of signature-/rule-based detection.

- Gaussian Naive Bayes a lightweight probabilistic model, representing the statistical analysis approach.

- MLP Neural Network*(2 hidden layers, 32/16 neurons) represents the artificial neural network method.

- Random Forest (100 trees) represents a hybrid/ensemble approach.

- Isolation Forest**, trained without attack labels represents unsupervised anomaly detection.

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier, IsolationForest

from sklearn.neural_network import MLPClassifier

from sklearn.naive_bayes import GaussianNB

from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, roc_auc_score

y_train / y_test: 0 = normal, 1 = attack (binarized NSL-KDD labels)

dt = DecisionTreeClassifier(max_depth=10, random_state=42).fit(X_train, y_train) nb = GaussianNB().fit(X_train_scaled, y_train)

mlp = MLPClassifier(hidden_layer_sizes=(32,16), max_iter=60).fit(X_train_scaled, y_train)

rf = RandomForestClassifier(n_estimators=100, max_depth=12, n_jobs=-1).fit(X_train, y_train)

iso = IsolationForest(contamination=0.35, n_jobs=-1).fit(X_train_scaled)

unsupervised # Evaluation (repeated for each model)

pred = model.predict(X_test)

score = model.predict_proba(X_test)[:, 1]

f1 = f1_score(y_test, pred) auc = roc_auc_score(y_test, score)

Table 5. Comparative Performance Results of the Implemented Detection Models on the NSL-KDD Test Set.

Model	Accuracy	Precision	Recall	F1	AUC	Training Time
Decision Tree	0.772	0.967	0.621	0.756	0.758	0.54 s
Naive Bayes	0.771	0.911	0.663	0.768	0.830	0.08 s
Neural Network (MLP)	0.794	0.971	0.658	0.785	0.941	22.2 s
Random Forest	0.770	0.966	0.617	0.753	**0.967**	6.3 s
Isolation Forest (unsupervised)	**0.798**	0.875	**0.754**	**0.810**	0.882	0.67 s

Fig.3: ROC Curves, NSL-KDD Test Set

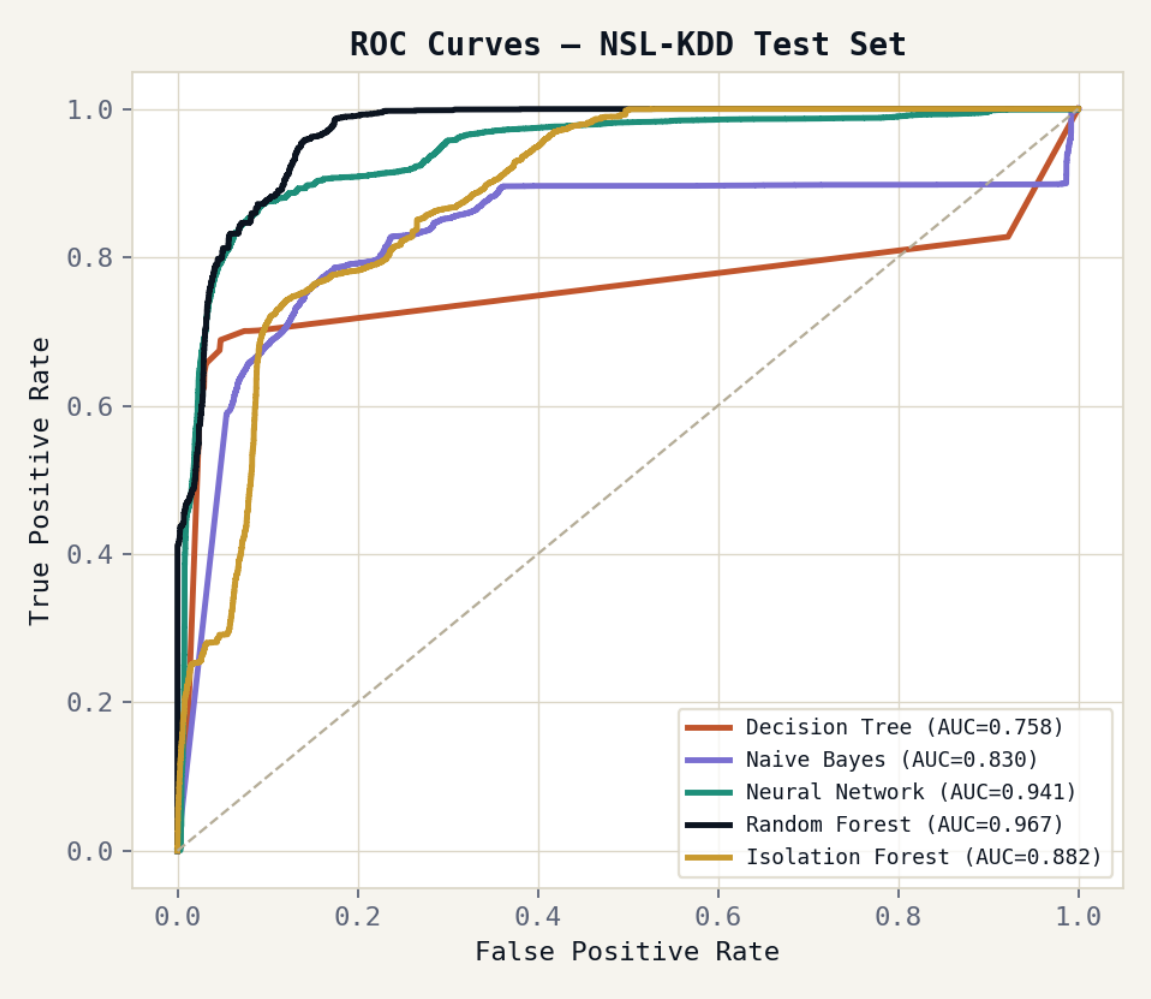
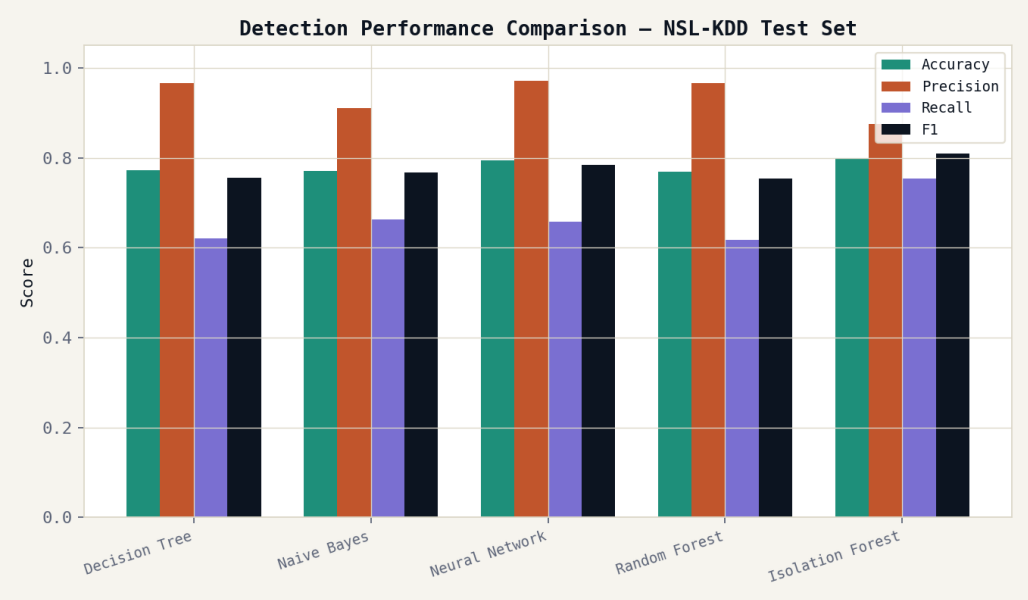


Fig.4: Accuracy / Precision / Recall / F1 by Module



The results obtained confirm the theoretical reasoning presented above. Decision Tree and Random Forest achieve the highest Precision (0.97) among all five models: when they flag something as an attack, this is confirmed to be correct in the great majority of cases. Their Recall, however, is by contrast the lowest in the group (0.62) that is, they fail to detect approximately 38% of actual attacks.

Isolation Forest, on the other hand, despite having had no access whatsoever to attack labels during training, achieves the highest Recall (0.75) and the best overall F1-score (0.81), which constitutes empirical confirmation of the principal advantage of the anomaly-based approach: the ability to detect attack types that have never previously been studied.

The MLP neural network is positioned between these two extremes, achieving the second-highest AUC (0.941), which reflects the strong discriminative ability discussed above; this, however, comes at the cost of the longest training time 22 seconds as against under 1 second for the other models, which represents a clear illustration of the accuracy–adaptability–speed trade-off described in the methods-comparison table. [8]

4. Conclusion

It should be noted that, in different situations, when processing data of differing character, certain methods make it possible to resolve the problem of detecting malicious impact better than others. The signature method is effective for analyzing the content of traffic but does not permit the detection of unknown impacts. The statistical method and the artificial neural network are well suited for analyzing traffic characteristics and detecting anomalies therein, particularly for network objects characterized by stable features over time, such as, for example, servers that provide certain network services. The biometric method makes it possible to resolve the issue of information input with respect to devices according to the characteristics of their operation, and so forth. Such an impact, however, may itself constitute a discernible sign of yet another, more complex impact.

In this connection, it would be advisable to develop an approach to detecting attacks on information systems that would allow for the joint use of arbitrary methods for attack detection, taking into account their respective advantages. In many cases, however, reliance is placed not on a single specific method, but rather on the combined, joint application of the approaches enumerated above.

A specific direction for further research is the development of a hybrid Ensemble architecture combining signature-based and anomaly-based detection, in which a fast signature filter e.g., at the level of Suricata intercepts known attacks at the very first stage, while the remaining, filtered traffic is passed on to an anomaly detector based on LSTM/Autoencoder. The validity of such a model should be tested on contemporary test datasets similar to CICIDS2017/CSE-CIC-IDS2018, through the joint minimization of F1-score and FPR, with robustness verified by means of adversarial testing.

References

1. ი. ქართველიშვილი, ა. ბიჩნიგაური, ზ. ბერიძე, "აკადემიური საზოგადოების როლი კიბერუსაფრთხოების ცნობიერების ამაღლებაში," *საქართველოს ტექნიკური უნივერსიტეტის შრომები*, თბილისი: გამომცემლობა „ტექნიკური უნივერსიტეტი", 2025, ISBN 978-9941-520-86-0.
2. ი. ქართველიშვილი, ა. ბიჩნიგაური, ლ. შონია, "ფიშინგისა და მავნე კოდის მქონე ვებგვერდების პრევენციის ეფექტური მექანიზმის მოდელის შემუშავება და რეალიზაცია ვებბრაუზერის გარემოში," *საქართველოს ტექნიკური უნივერსიტეტის შრომები*, თბილისი: გამომცემლობა „ტექნიკური უნივერსიტეტი", 2023, ISBN 978-9941-512-06-3.
3. ი. ქართველიშვილი, ლ. შონია, "ვირტუალური კერძო ქსელის (VPN) აგების კონცეფცია, ქსელის ფუნქციები და მათი კლასიფიკაცია," *მართვის ავტომატიზებული სისტემები*, საქართველოს ტექნიკური უნივერსიტეტი, №2(26), თბილისი, 2018.
4. K. Zetter, *Countdown to Zero Day: Stuxnet and the Launch of the World's First Digital Weapon*. Crown, 2014, 406 p.
5. J. Seitz, *Black Hat Python: Python Programming for Hackers and Pentesters*. No Starch Press, 2014, 171 p.
6. M. Sikorski, A. Honig, *Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software*. No Starch Press, 2012, 800 p.
7. R. Boyle, R. Panko, *Corporate Cybersecurity*. Pearson, 2025, 646 p.
8. დ. ჩაკვეტაძე, "Machine Learning მეთოდების გამოყენება ქაოსის ექსპერიმენტების შედეგების პროგნოზირებაში," ბიზნესისა და ტექნოლოგიების უნივერსიტეტი, 2026, 55 გვ.