# Machine Learning Approach for Cyberattack Detection and Prevention on IoT Networks

Janet M. Maluki
Department of Computing and Informatics - United States International University-Africa
P.O Box 14634 00800, Kenya

Jimmy K.N. Macharia
Professor of Information Systems, United States International University-Africa,
P.O Box 14634 00800, Kenya

Dalton Ndirangu Kaimuru, PhD
United States International University-Africa,
P.O. Box 14634 00800, Kenya

## ABSTRACT
A key part of securing IoT networks is detecting intrusions and stopping potential attacks before they cause harm. To achieve this, various security measures have been implemented, including firewalls, intrusion detection systems, antivirus software, and organizational security policies. This study adopts a systematic approach to detecting and preventing cyberattacks in IoT networks. It examines prior research, evaluates existing intrusion detection techniques, and applies these insights to develop a more effective and adaptable detection framework. This study examines intrusion detection techniques that incorporate machine learning and statistical methods. Building on a thorough analysis of existing intrusion detection systems, it introduces a novel model that enhances multiple cyberattack detection and prevention in IoT networks. The experimental results highlight the model's strong performance, achieving an impressive 98% accuracy. It also maintains a weighted average recall of 97%, precision of 96%, and an F1-score of 96% across various attack categories, demonstrating its reliability in detecting multiple cyberattacks.

## General Terms
IoT, Security, Cyberattack, Detection, Machine Learning, Algorithms

## Keywords
Intrusion, attacks, statistics, models, anomalous, classification, clustering, detection, framework, Internet of Things.

## 1. INTRODUCTION
The Internet of Things (IoT) has transformed device communication, driving advancements in industrial automation, smart cities, and healthcare [1]. However, the growing interconnectedness of IoT devices has also introduced new vulnerabilities, making these networks prime targets for cyber threats [2]. Ensuring the security of IoT systems is crucial, particularly in detecting and preventing cyberattacks that could compromise data availability, privacy, and integrity [3]. Preventing and detecting cyberattacks in IoT networks requires a combination of technologies and strategies, including firewalls, antivirus software, intrusion detection systems (IDS), and organizational security policies [4]. These tools play a crucial role in identifying malicious activity, blocking potential threats, and strengthening the resilience of IoT networks. However, the ever-evolving nature of IoT environments often challenges the effectiveness of these traditional security measures [5]. To address these challenges, advanced techniques leveraging statistical analysis and machine learning (ML) have been developed [4]. These methods offer scalable, adaptive, and efficient solutions for detecting and mitigating cyber threats in IoT networks [5]. This study systematically investigates ML and statistical approaches for intrusion detection in IoT environments, analyzing existing research and methodologies to identify the most effective strategies for enhancing network security. By integrating insights from prior research, this work aims to build a generic framework that addresses the unique security challenges posed by IoT environments. The outcomes of this research contribute to advancing IoT network security by offering innovative approaches to detecting and preventing cyberattacks. The integration of statistical and machine learning techniques highlights their potential to enhance the resilience of IoT networks while reducing vulnerabilities to emerging threats.

## 2. CYBERATTACK DETECTION IN IoT NETWORKS
With the growing number of connected devices, IoT networks are becoming increasingly vulnerable to cyber threats, making intrusion detection and prevention a critical area of research. As noted in [8], historical data can help distinguish legitimate users from malicious ones by analyzing behavior patterns. By identifying typical user activities, it becomes possible to detect significant deviations that may indicate potential security threats. However, as noted in [9], identifying a malicious user can be challenging, especially when the difference between normal and abnormal behavior is subtle, making certain violations difficult to detect. Research in [10] and [11] highlights two main approaches to intrusion detection: anomaly-based and misuse-based detection. Both methods have been instrumental in shaping the development of modern intrusion detection systems.

### 2.1 Anomaly-based intrusion detection
Anomalies, often referred to as outliers, exceptions, or irregularities, are data patterns that significantly deviate from a system’s expected behavior [6]. Anomaly detection aims to identify such deviations and flag them as potential security threats [7]. Detection methods range from simple threshold-based techniques to sophisticated statistical models. These profiles can be predefined, adaptive, or self-learning, enabling more accurate identification of unusual activities and improving the effectiveness of intrusion detection systems.
In the context of the Internet of Things (IoT), an anomaly refers to a measurable deviation from a system’s expected behavior, either on a local or global scale [12]. This definition underscores key aspects of IoT data, emphasizing that most collected data represent routine system activity [14]. Additionally, it recognizes that what is considered "normal" can evolve due to changing conditions over time. Figure 1 illustrates how anomalies are classified within standard system operations.

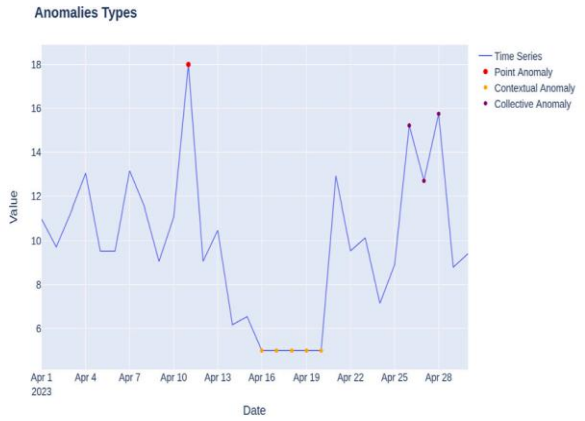


**Figure 1 Classification of anomalies in system operations**

## 2.2 Classification of anomalies

*Point Anomaly:* According to Fahrmann [7], a point anomaly refers to an individual data instance that significantly deviates from the rest, typically falling within a low-density value range. This means the data point stands out as an outlier when compared to the overall dataset. ***Contextual Anomaly***: As described in [8], a contextual anomaly occurs when a data instance appears unusual within a specific context but may not be considered abnormal in a different setting. In other words, its anomaly status depends on the surrounding data, and with additional context, it might be perceived as normal. Finally, ***Collective anomaly*** refers to a group of data instances that, when analyzed together, exhibit abnormal behavior. While each instance may appear normal on its own, their combined occurrence creates an unusual pattern that deviates from expected system behavior [6].

## 2.3 Classification of anomalies detection

### *2.3.1 Protocol Anomaly Detection*

Protocol anomalies refer to deviations from established protocol standards and expected Internet behaviors in terms of format and operation [16]. These anomalies can occur across network, transport, and application layers, making them a key focus of protocol anomaly detection techniques [17]. This approach works by identifying unusual patterns during processes such as IP de-fragmentation and TCP reassembly, where inconsistencies or ambiguous conditions may arise [18]. Ensuring that the Intrusion Detection System (IDS) aligns with protocol standards helps minimize exceptions that could otherwise lead to misinterpretations or security vulnerabilities [19]. For an Intrusion Detection System (IDS) to effectively monitor application protocol behavior, it must have the ability to perform deep application protocol parsing, also known as decoding [17]. This process allows the IDS to analyze protocol structures in detail, ensuring that any deviations or suspicious activities are accurately identified. When analyzing application protocol behavior, various anomalies may indicate protocol inconsistencies or potential cyber threats. These anomalies include: (i) Invalid column values or unusual parameter combinations, (ii) The execution of unauthorized commands, (iii) Extremely short or excessively long field lengths, which may signal an attempt to exploit buffer overflow vulnerabilities, (iv) An unusually high occurrence of specific fields or directives, suggesting possible malicious activity, and (v) The use of a protocol or application on an unexpected port or for an unintended purpose [18].

### *2.3.2 Statistical Anomaly Detection*

Statistical anomaly detection involves continuously monitoring and analyzing patterns of legitimate user behavior over time. By establishing a baseline of normal activity, statistical methods can then be applied to compare newly observed behavior against this standard. If significant deviations are detected, the system flags them as potential anomalies with a high degree of confidence [19]. This approach helps identify unusual activities that may indicate security threats. Statistical anomaly detection methods include:

*a) Threshold detection*: This method entails placing user-independent thresholds for the frequency at which certain occurrences occur.

*b) Profile-based*: Identify shifts in the behavior of individual accounts and a profile of each user's activities is designed.

2.3.3 Application Payload Anomaly Detection

To support application anomaly detection efficiently, it remains essential to perform a detailed analysis of application protocols to define precise behavioral constraints [19]. Additionally, a thorough understanding of the application's semantics is necessary to enhance the accuracy and reliability of anomaly detection [20]. To identify application-level anomalies effectively, it is essential to understand the permissible encoding types for a given field and determine what other ancient guidelines have been set.

### *2.3.3 Application Payload Anomaly Detection*

Effective application anomaly detection requires an in-depth analysis of application protocols to establish precise behavioral constraints [19]. Additionally, gaining a comprehensive understanding of application semantics enhances the accuracy and reliability of detection mechanisms [20]. Identifying anomalies at the application level involves spotting permissible encoding types for specific fields and determining potential embedded applications. For instance, an anomaly may arise when shellcode unexpectedly appears in fields where it is not typically found [21]. A well-defined anomaly profile enables the detection of shellcode execution attacks without prior knowledge of the specific exploit code or confirmation of its presence in the system.

## 2.4 Signature-Based Intrusion Detection

Signature-Based Intrusion Detection Systems (SIDS), also known as the misuse detection approach, promotes a crucial component of an organization's security framework. This method detects known cyber threats by matching network traffic or host activity against predefined attack signatures or patterns [22]. SIDS relies on established detection rule sets, but for optimal effectiveness, only rules relevant to the specific operational environment should be activated [23]. The following section explores various techniques used to detect misuse.

*i) Expression matching*

The most basic type of misuse detection is expression matching, which looks for instances of patterns or signatures in an event stream (log entries, network traffic, etc.) [24]. Signatures are easy to create particularly when paired with protocol-aware field decomposition.

*ii) State transition analysis*

Attacks are modelled using state transition analysis as a network of states and transitions (matching events) [25]. Transitions may result from applying each observed event to instances of finite state machines, each represents an attack scenario.

iii) *Keystroke Monitoring:*

Keystrokes made by the user are used in this technique to identify when an attack has occurred. The approach uses

Pattern-matching for keystroke sequences suggesting an attack [26]. This method's drawbacks include the various ways to describe the same attack at the keystroke level and the general unavailability of user-typed keystrokes.

*iv) Expert Systems:*

According to [25] an expert system is a computer program that can represent and reason about a field with a wealth of information to provide guidance and solve difficulties. Expert system detectors encode attack knowledge as if-then implication rules. The if part of a rule specifies the prerequisites for an attack. When every condition on the left side of a rule is met, the actions on the right side of the rule are carried out, which could lead to the existence of an intrusion or the firing of more rules [27]. The primary benefit of creating if-then implication rules is that control thinking is kept outside problem-solution development.

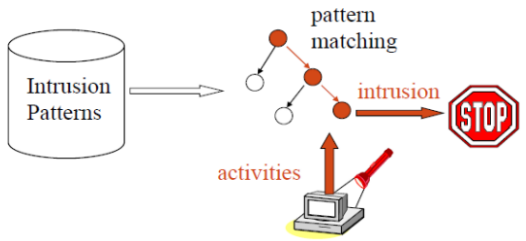


**Fig 2 Misuse detection system with pattern matching [9]**

# 3. DATA MINING APPROACHES FOR INTRUSION DETECTION

As the volume of digital documents continues to grow across multiple languages worldwide, data mining has gained significant traction in the field of knowledge discovery [10]. According to [11], data mining is an automated process used to extract meaningful and valuable insights from vast data repositories, making it an essential tool for handling large-scale information. The rapid advancements in data mining have led to the development of numerous algorithms derived from statistics, pattern recognition, machine learning, and database management [12]. These innovations have expanded the capabilities of data analysis, enabling more efficient and accurate knowledge extraction. In the context of this study, the following data mining techniques are particularly relevant:

## 3.1 Feature selection

Feature Selection (FS) is a crucial step in enhancing intrusion detection for IoT networks. It focuses on identifying the most relevant features while eliminating redundant or unnecessary ones to improve classification accuracy [32], [33]. This process becomes even more important when dealing with high-dimensional data, where using every available feature can be inefficient and may reduce model effectiveness due to limited data samples [34], [35]. The quality of FS directly impacts the performance of machine learning-based detection systems, ensuring that only the most meaningful attributes are used for accurately identifying cyber threats [36], [37]. In cybersecurity research, different feature selection (FS) techniques, including filter, wrapper, and embedded methods, are employed to optimize feature subsets and improve detection accuracy [38], [39]. Figures 3, 4, and 5 illustrate how key features were extracted from 14 attack files in the Edge-IIoT dataset using Chi-Square, Mutual Information, and Random Forest selection techniques. These methods help identify the most significant features, enhancing the efficiency of cyberattack detection. These statistical approaches help enhance cyberattack detection models by improving classification accuracy while decreasing computational complexity [40], [41], [42]. By examining each selected attribute, the system extracts meaningful insights, ranks features based on their importance and applies them strategically to boost detection accuracy and overall model performance.

## 3.2 Machine learning

Machine learning focuses on developing algorithms that improve automatically through experience. These algorithms are widely used in information filtering systems to identify user preferences and in data mining applications to uncover patterns within large datasets [34], [35]. The two primary machine learning techniques are clustering and classification, which are particularly effective in detecting hidden patterns without requiring prior knowledge of their structure [36]. Unlike traditional methods of cyberattack detection, machine learning approaches can dynamically adapt to complex data distributions, making them well-suited for cyber threat detection and anomaly identification in IoT networks [37], [38].

### *3.2.1 Clustering*

Clustering is a fundamental technique in data mining that combines data points into distinct groups based on their similarities and shared attributes [36], [37]. As an unsupervised learning method, it uncovers hidden structures in datasets without requiring predefined labels [38]. Various clustering approaches exist, each using diverse strategies to enhance data partitioning and pattern recognition [39], [40].

*i) Hierarchical clustering*

Hierarchical clustering arranges datasets through a stepwise, iterative process rather than grouping all data points simultaneously [37], [38]. This method progressively merges or splits clusters based on similar measures, resulting in a structured hierarchy. Hierarchical clustering is further labelled into distinct approaches, including:

*a) Division clustering*

In divisive clustering, the dataset starts as a single cluster and is recursively split until each data point forms its cluster, following a top-down hierarchical structure.

*b) Agglomerative Clustering*

Initially, it reflects each data point as an individual cluster and iteratively merges the closest clusters based on predefined criteria, following a bottom-up approach from leaf to root.

*ii) Partitional clustering*

Partitional clustering segments data points into k distinct groups based on specific significance criteria, ensuring optimal separation and similarity within each cluster [39]

*iii) K-Mean Clustering method:*

This algorithm groups data into clusters by reducing the distance between each point and the respective cluster centroid. It has three main variations: k-means for numerical data, k-medoids for categorical datasets, and k-prototypes for mixed data types [40].

a) K-mean: Applied to sets of numerical data.
b) K-media: Applied to categorical datasets
c) K-prototype: Applied to both numerical and categorical datasets.

*iv) Fuzzy C Mean Clustering:*

This clustering method not only evaluates the distance between data points and cluster canters but also incorporates membership values, allowing data points to belong to multiple clusters with varying degrees of association [40].

*v) QT Clustering*

Quality Threshold (QT) clustering groups data points based on a predefined cluster approach. It ensures high-quality clusters by identifying large groups whose diameters do not exceed a user-specified threshold, maintaining consistency and reliability in cluster formation [41]

### *3.2.2 Classification*

A data item is classified into a few pre-established categories. Typically, these algorithms produce "classifiers" in the form of rules or decision trees [13]. This technique can be used in intrusion detection to collect enough "normal" and "abnormal" audit data for a user or program. A classification algorithm can then be used to learn a classifier that identifies whether the audit data belongs to the abnormal or normal class. [14]. In classification-based IDS, all traffic is classified by IDS as either malicious or normal [15]. However, reducing false positives, (classification of benign traffic as malicious) and false negatives (the classification of malicious traffic as normal) is the difficult part of the classification-based IDS [16]. In intrusion detection systems, classification methods include fuzzy logic, neural networks, genetic algorithms, and inductive rule generation.

## 3.3 Statistical Techniques

This method compares events statistically according to a predefined list of parameters [44]. Statistical methods are referred to as "top-down" learning and are used once the relationships between the data are established, using mathematics to help with the search.

The three fundamental categories of statistical methods are decision trees, nonlinear, and linear [18]. Statistical models test the obtained system and network data for attack analysis. Operational, Average and Standard Deviation, Multivariate, Markovian, and Time Series models are the most used models. Different periods, such as the day of the week, the month, the year, or per-host or per-service basis, can be used to compute statistical trends.

Denning (1987) [45] discussed some of the issues and solutions associated with statistical measurements to identify abnormalities. The operational model, mean and standard deviation model, multivariate model, Markov process model, and time series model are the five statistical measurements she described. The IDS's rules use these measures to identify intrusions.

*Operational model:* An intrusion is indicated when the operational model surpasses a predetermined threshold. The security policy typically establishes the threshold. For instance, the security policy may stipulate that a password should be reported if three or more trials are unsuccessful. [46].

*Mean and standard deviation model*: This model indicates incursion if it deviates from the mean ± threshold stdev [47]. In this instance, the threshold is distinct from the last one in those four and is typically employed since, under a normal distribution, about 100% of the data should fall inside that range.

The *multi-variate model*: In which activity correlation is employed. For instance, the CPU time and I/O that software uses. It's possible that only observing CPU usage is insufficient to identify an intrusion [48].

*Hidden Markov model:* HMM is a modest kind of dynamic Bayesian network and is a statistical tool for modelling sequential observations. The Markov chain model: Where activities are viewed as events, and the likelihood that an event will occur is determined by its past [49]. For instance, if a programmer often uses a set of commands to modify, compile, link, and run an application, then the IDS can determine what commands are expected because the same set of commands is always expected. An intrusion is suspected if an unusual command occurs, and an IDS will raise an alarm. (visible) that depend probabilistically on a hidden sequence of occurrences (hidden states) [50]. The study by [51], describes a state-of-the-art technique that uses Hidden Markov Models (HMM) to detect advanced online threats. According to the findings, these attacks involve several steps and may occur over an extended period. Certain acts may be interchangeable within each step. To conceal the intrusion, an intruder may purposefully choose a series of acts within a step.

Other cases can entail inconsistent action sequences (due to background noise) or the offender's inexperience [52]. An intrusion detection system must be able to manage some of these uncertainties. HMMs are ideally suited to tackle the multi-step attack problem [53]. Authors [54], and [55] directly compare HMMs to two other traditional methods, decision trees and neural nets, and demonstrate that HMMs detect these intricate intrusions significantly better than neural networks and generally better than decision trees.

From [45], the author notes that the Hidden Markov Model" assumes that the state variables are hidden and correspond to phenomena that are, perhaps, fundamentally unobservable," and as such, should perform well in modelling user actions. He concluded HMM and the instance-based learner, trained using the same data, performed comparably.

## 3.4 Profiles

There are three categories for profiles: activity, template, and abnormality. The IDS represents the associated activity profile when an audit record is created [17]. Depending on the model and value, it may generate an anomaly profile and trigger an alarm. The activity profile is made using a profile template if it doesn't exist.

Creating profiles is the most challenging aspect of IDSs, though templates are retained [18]. A template comprises the previously listed fields in a data structure. The IDS will not identify activity profiles when a new user is created in the system and will instead generate the necessary ones using the appropriate template profiles upon the user's initial login [19]. Except for subjects, every field in the template is duplicated in the new activity profile. Each Single subject can use profiles. The frequently used data structures for profiles: Name, Subject, Object, Action pattern, Resource-usage-pattern, Exception-pattern, Time, Variable type, Threshold, and Value. The profile's three main components are name, Subject, and Object.

## 3.5 Proposed Structural Framework Design

The Internet has significantly transformed modern life, offering vast opportunities alongside increasing cybersecurity threats [58]. Cyber intruders are generally classified into two categories: outsiders and insiders. Outsiders operate externally, targeting systems through email-based spam attacks or attempting to bypass firewalls to compromise internal networks [59]. In contrast, insiders are legal users who exploit their access privileges, impersonate higher-level users, or misuse confidential data to facilitate unlawful access from external sources. Addressing these threats requires robust security mechanisms for potential cyberattack detection.

To enhance network security, this study introduces a structural design framework for intrusion detection and prevention in IoT networks. While detecting known attacks is essential, the identification of unknown threats is equally critical. Anomaly detection techniques play a vital role in uncovering these unknown attacks. Since each intrusion detection method offers distinct advantages in identifying cyber threats, the proposed framework leverages a statistical approach and machine learning as depicted in Fig. 6 to improve cyberattack detection and response effectiveness.

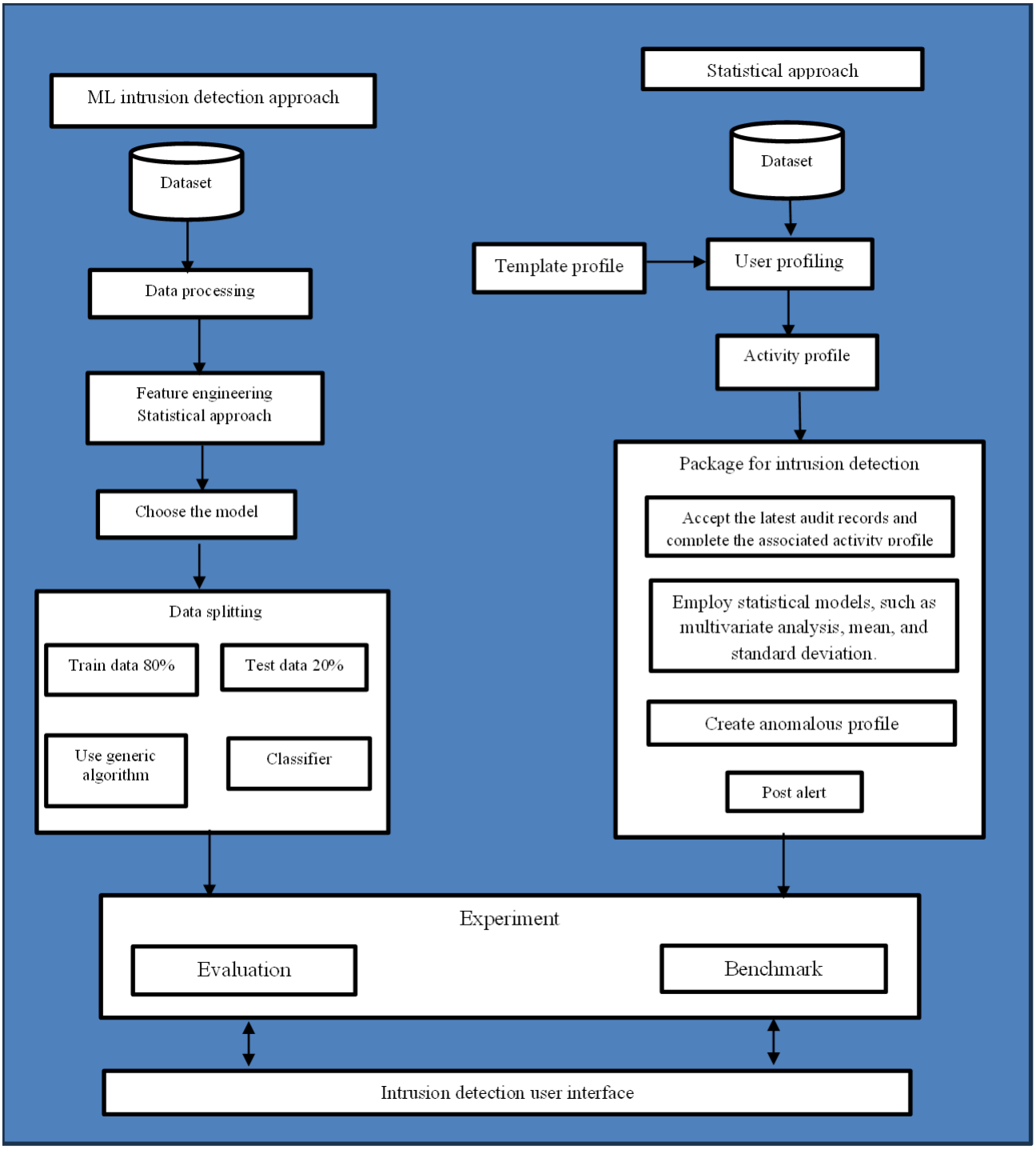


**Fig 3: Proposed structural framework design for cyberattack detection in IoT network**

The suggested approach considers IoT network traffic capturing and IoT system audit logs. Algorithms for supervised learning can be used to determine if an activity is harmful or normal. Network packet data sets are classified to detect attacks. The use of supervised learning to develop network traffic rules is suggested in the paper. These rules help differentiate standard connections from abnormal ones. The study follows a two-step approach: first, a statistical method is used to identify the most relevant features, and then supervised learning is applied to identify attack patterns effectively. The optimal features are used to form rules for detecting various cyberattacks using Random Forests. This permits the overview of higher levels of generality and thus higher detection rates.

An initial sample of randomly selected participants is used to begin the process. After that, the population changes over several generations as participants' attributes steadily improve, as seen by an increase in fitness value. The network is trained using the supervised approach to identify the unknown attacks as the last phase. Define a decent fitness function that offers incentives to the appropriate kind of participants. To enhance the grouping outcomes, the study considers all pertinent criteria. Our fitness function can be found using:

Fitness = Error rate + Entropy measure + Rule consistency

A rule's classification is a consequent portion if it applies to a specific case. If they don't match, there's no classification. An

individual is formed from a set of rules, and the rule voting corresponds to the instance that determines the final labelling rule set prediction. The classifier assigns equal weight to each matched rule. The error rate represents the percentage of misclassified instances within the training dataset. Classification was performed on network data using anomaly detection based on supervised learning. The proposed model was evaluated using threshold-based metrics, including Accuracy, True Positive Rate (TPR), False Positive Rate (FPR), Precision, F1-Score, and Recall.

## 4. FINDINGS

The Edge-IIoT dataset initially comprised 14 distinct files, each corresponding to a specific attack type. To achieve research objectives, these files were combined into a single dataset. Feature selection was performed, which played a vital role in model performance improvement in terms of accuracy, reducing overfitting, and speeding up training by retaining only the most relevant features.

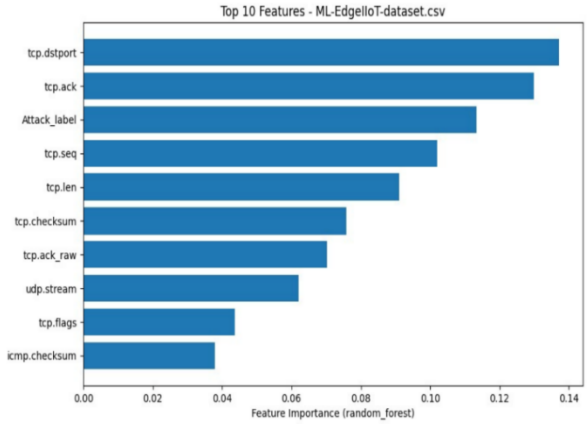


**Fig 4 Top 10 features selected using Random Forest (Author)**

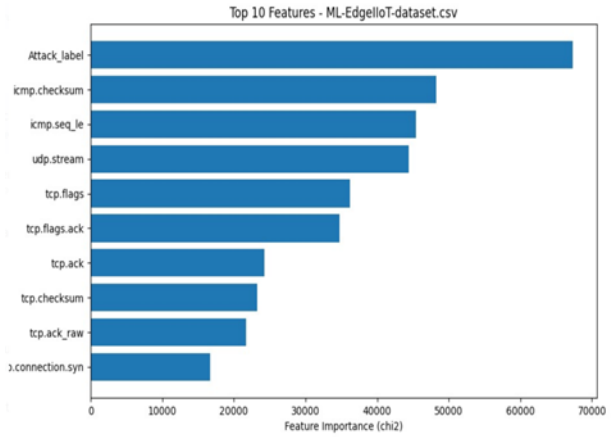


**Figure 5 Top 10 features selected using chi2 (Author)**

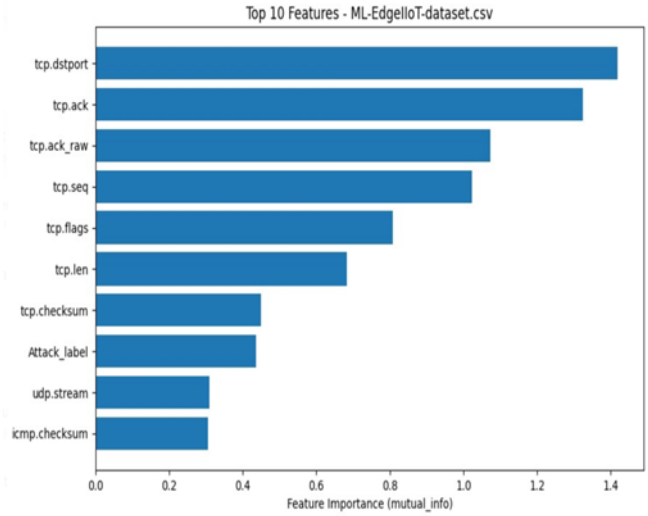


**Fig 6 Top 10 features selected using Random Forest (Author)**

The proposed model was trained using a Random Forest and 13 optimal features were selected using an integrated statistical feature selection approach. These key features included tcp. dstport, tcp. ack, attack_label, tcp. len, tcp. checksum, tcp.ack_raw, udp. Stream, tcp. Flags, ICMP. checksum, ICMP.seg _ len, tcp_ flags. ack, and tcp. Connection _syn. Figure 7 indicates the Classification report of the proposed model.

Classification Report:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Backdoor_attack | 1.00 | 0.97 | 0.98 | 5047 |
| DDoS_HTTP_Flood_attack | 0.94 | 0.89 | 0.92 | 45794 |
| DDoS_TCP_SYN_Flood_attack | 1.00 | 1.00 | 1.00 | 403833 |
| MITM_attack | 1.00 | 1.00 | 1.00 | 280 |
| OS_Fingerprinting_attack | 1.00 | 0.54 | 0.70 | 192 |
| Password_attack | 0.95 | 0.99 | 0.97 | 210670 |
| Port_Scanning_attack | 0.96 | 1.00 | 0.98 | 4501 |
| Ransomware_attack | 0.98 | 0.90 | 0.94 | 2198 |
| SQL_injection_attack | 0.96 | 0.72 | 0.82 | 10325 |
| Uploading_attack | 0.97 | 0.85 | 0.91 | 7372 |
| Vulnerability_scanner_attack | 1.00 | 0.97 | 0.98 | 29415 |
| XSS_attack | 0.97 | 0.75 | 0.85 | 3119 |
| accuracy | | | 0.98 | 722746 |
| macro avg | 0.98 | 0.88 | 0.92 | 722746 |
| weighted avg | 0.98 | 0.98 | 0.98 | 722746 |

**Figure 7 The classification report of each attack type**

The results demonstrate that the model effectively detects various attack types. The Backdoor attack achieves near-perfect performance with a precision of 1.00, recall of 0.97, and an F1-score of 0.98 across 4,952 samples, indicating minimal false negatives. ICMP, TCP SYN, and UDP Flood attacks are detected with perfect scores (precision, recall, and F1-score of 1.00) over 582,767, 403,853, and 639,476 samples, respectively, highlighting the model's robustness in identifying these threats. Additionally, despite a limited dataset of 241 samples, the model maintains a perfect classification score, though the small sample size necessitates cautious interpretation.

The overall performance metrics were evaluated using both macro average and weighted average. The macro average provides an unweighted meaning across all classes, ensuring equal treatment of each class. In contrast, the weighted average accounts for class imbalance by assigning greater weight to groups with more samples. Table 1 presents the comprehensive evaluation metrics for the proposed model.

**Table 1 The proposed model performance**

| Metric | Value |
|---|---|
| Accuracy | 0.9705 |
| Macro Average Precision | 0.92 |
| Macro Average Recall | 0.87 |
| Macro Average F1-Score | 0.89 |
| Weighted Average Precision | 0.9681 |
| Weighted Average Recall | 0.9705 |
| Weighted Average F1-Score | 0.9691 |

Table 1 demonstrates the model's strong overall performance, with an accuracy of 0.9705, indicating that approximately 97.05% of predictions are correct. This highlights the model's high effectiveness in accurately classifying instances.

**Macro Averages**: The model demonstrates strong performance across all classes, with an average precision of 0.92, meaning 92% of predicted positive instances were correctly classified. The recall score of 0.87 indicates that, on average, 87% of actual positive instances were accurately identified. Additionally, the F1-score of 0.89, which balances precision and recall, reflects the model's effectiveness when treating each class equally.

***Weighted Averages***: With precision (0.9681), recall (0.9705), and F1-score (0.9691), these metrics account for the support (number of instances) in each class, providing insight into the model's performance on larger classes. The high weighted scores, closely aligning with overall accuracy, indicate that the model excels in correctly classifying most instances.

# 5. DISCUSSION OF THE RESULTS

The evaluation of the cyberattack detection model, trained using the Random Forest algorithm with improved statistical feature selection, demonstrates robust overall performance while exhibiting varying effectiveness across different attack categories.

For backdoor attacks the model demonstrates exceptional performance, achieving perfect precision (1.00) and high recall (0.97), leading to an F1-score of 0.98 across 4,952 instances. This indicates that nearly all backdoor attacks are correctly identified, with minimal false negatives and no false positives.

For DDoS attacks, the model performs remarkably well in three out of four attack types. ICMP, TCP SYN, and UDP Flood attacks achieve perfect detection (precision, recall, and F1-score all at 1.00) across a significantly large number of instances. However, HTTP Flood attacks show slightly lower performance, with a precision of 0.93, recall of 0.87, and an F1-score of 0.90 over 46,159 samples, indicating a minor misclassification rate for this specific attack variant.

For MITM attacks, despite the limited sample size (241 instances), the model achieves flawless classification with perfect precision, recall, and F1 score.

Similarly, OS Fingerprinting attacks are detected with high precision (0.98) and strong recall (0.91), leading to an F1 score of 0.94. While performance is strong, there remains minor room for enhancement.

The model also demonstrates robust detection of Password attacks (precision 0.95, recall 0.99, F1-score 0.97) and Port Scanning attacks (precision 0.95, recall 1.00, F1-score 0.98), confirming its reliability in identifying these threats.

For Ransomware attacks, the model achieves perfect precision (1.00) but has a lower recall (0.88), leading to an F1 score of 0.94. This suggests that while false positives are nearly nonexistent, some actual ransomware instances are misclassified.

The most challenging attack types appear to be SQL Injection and XSS attacks. SQL Injection attacks demonstrate high precision (0.96) but suffer from low recall (0.71), resulting in an F1-score of 0.81. Similarly, XSS attacks have a precision of 0.96 but a recall of only 0.75, leading to an F1-score of 0.84. These lower recall values indicate that a significant number of these attacks go undetected, posing a potential security risk.

Uploading attacks demonstrate strong performance, achieving a precision of 0.97, recall of 0.85, and an F1-score of 0.91. Meanwhile, Vulnerability Scanner attacks are detected with near-perfect accuracy, boasting a precision of 1.00, recall of 0.97, and an F1-score of 0.98, indicating highly reliable detection.

Our proposed model integrates statistical feature selection techniques with the Random Forest algorithm to enhance cyberattack detection in IoT networks. Optimal features are selected using Random Forest, Information Gain, and Chi-Square methods, considering CPU time and program input/output. Moving forward, our research will focus on implementing ensemble learning and evaluating its effectiveness in detecting multiple cyberattacks while reducing false alarms, considering the resource limitations of IoT networks.

# 6. CONCLUSION

This study introduces a structured framework design to improve cyberattack detection in IoT networks through an integrated feature selection and machine learning approach. The proposed model leverages statistical techniques such as Random Forest, Information Gain, and Chi-Square to identify the most relevant features, thereby enhancing classification accuracy. Experimental results demonstrate strong overall performance with high precision, recall, and F1 scores across multiple attack categories. While the model exhibits near-perfect detection for several attack types, challenges persist in identifying individual threats, such as SQL Injection and XSS attacks, where recall values remain comparatively lower. These findings stress the crucial role of feature engineering in improving detection accuracy while also underscoring the need for further refinement to enhance the detection of harder-to-classify attacks.

# 7. RECOMMENDATION

The study findings indicate a potential for further enhancement to improve performance. Future research can explore this direction to refine and enhance the model.

- Enhance Low-Recall Detection: Optimize recall for SQL Injection and XSS by refining feature selection, adjusting thresholds, or adding relevant features.
- Expand Dataset: Increase data volume and diversity to improve generality across attack types.
- Explore Advanced Models: Investigate deep learning techniques like CNNs and RNNs for better pattern recognition.
- Real-World Deployment: Test the model in IoT environments to assess adaptability and effectiveness.
- Hybrid Detection: Combine anomaly- and signature-based methods for robust threat identification.

# 8. DECLARATION

This is my work, and it hasn't been submitted to another publication.

# 9. REFERENCES

[1] A. N. Ayesh, “Enhancing Urban Living in Smart Cities Using the Internet of Things (IoT),” Int. Acad. J. Sci. Eng., vol. 11, no. 1, pp. 237–246, 2024, Doi: 10.9756/iajse/v11i1/iajse1127.

[2] R. Lakhani, “Cybersecurity Threats in Internet of Things (IoT) Networks: Vulnerabilities and Defense

Mechanisms," vol. 12, no. 11, pp. 25965–25980, 2023, doi: 10.18535/ijecs/v12i11.4779.

[3] Y. Lu, "Security and Privacy of Internet of Things: A Review of Challenges and Solutions," J. Cyber Secure. Mobil., vol. 12, no. 6, pp. 813–844, 2023, doi: 10.13052/jcsm2245-1439.1261.

[4] K. Mahanta and H. B. Maringanti, "Security in the Internet of Things ( IoT ): Developing intrusion detection systems for IoT devices and networks and addressing the unique security challenges posed by this connection," Proc. Int. Conf. Artif. Intell. 5G Commun. Netw. Technol., no. May, pp. 570–576, 2023.

[5] A. Alaa Hammad, M. Adnan Falih, S. Ali Abd, and S. Rashid Ahmed, "International Journal of Computing and Digital Systems Detecting Cyber Threats in IoT Networks: A Machine Learning Approach," no. December 2024, doi: 10.12785/ijcds/1571020041.

[6] F. Alwahedi, A. Aldhaheri, M. A. Ferrag, A. Battah, and N. Tihanyi, "Machine learning techniques for IoT security: Current research and future vision with generative AI and large language models," Internet Things Cyber-Physical Syst., vol. 4, no. December 2023, pp. 167–185, 2024, doi: 10.1016/j.iotcps.2023.12.003.

[7] Z. Hasan, H. R. Mohammad, and M. Jishkariani, "Machine Learning and Data Mining Methods for Cyber Security: A Survey," Mesopotamian J. CyberSecurity, vol. 2022, no. January, pp. 47–56, 2022, doi: 10.58496/MJCS/2022/006.

[8] W. Hilal, S. A. Gadsden, and J. Yawney, "Financial Fraud: A Review of Anomaly Detection Techniques and Recent Advances," Expert Syst. Appl., vol. 193, p. 116429, 2022, doi: 10.1016/j.eswa.2021.116429.

[9] H. Taherdoost, "Security and Internet of Things: Benefits, Challenges, and Future Perspectives," Electron., vol. 12, no. 8, pp. 1–22, 2023, doi: 10.3390/electronics12081901.

[10] T. Sobh, "An Artificial Immune System for Detecting Network Anomalies Using Hybrid Immune Theories," J. ACS Adv. Comput. Sci., vol. 0, no. 0, pp. 0–0, 2024, doi: 10.21608/asc.2024.258634.1021.

[11] P. Satam, "Anomaly Based Wi-Fi Intrusion Detection System," Proc. - 2017 IEEE 2nd Int. Work. Found. Appl. Self* Syst. FAS*W 2017, pp. 377–378, 2017, doi: 10.1109/FAS-W.2017.180.

[12] J. C. S. Sicato, S. K. Singh, S. Rathore, and J. H. Park, "A comprehensive analyses of intrusion detection system for IoT environment," J. Inf. Process. Syst., vol. 16, no. 4, pp. 975–990, 2020, doi: 10.3745/JIPS.03.0144.

[13] D. Fahrmann, L. Martin, L. Sanchez, and N. Damer, "Anomaly Detection in Smart Environments: A Comprehensive Survey," IEEE Access, vol. 12, pp. 64006–64049, 2024, doi: 10.1109/ACCESS.2024.3395051.

[14] S. Trilles, S. S. Hammad, and D. Iskandaryan, "Anomaly detection based on Artificial Intelligence of Things: A Systematic Literature Mapping," Internet of Things (Netherlands), vol. 25, no. April, p. 101063, 2024, doi: 10.1016/j.iot.2024.101063.

[15] M. Landauer, S. Onder, F. Skopik, and M. Wurzenberger, "Deep learning for anomaly detection in log data: A survey," Mach. Learn. with Appl., vol. 12, no. April, p. 100470, 2023, doi: 10.1016/j.mlwa.2023.100470.

[16] M. H. Thwaini, "Anomaly Detection in Network Traffic using Machine Learning for Early Threat Detection," Data Metadata, vol. 1, pp. 1–16, 2022, doi: 10.56294/dm202272.

[17] R. Foorthuis, On the nature and types of anomalies: a review of deviations in data, vol. 12, no. 4. Springer International Publishing, 2021. doi: 10.1007/s41060-021-00265-1.

[18] K. C. Nalavade, "Using Machine Learning and Statistical Models for Intrusion Detection," Int. J. Comput. Appl., vol. 175, no. 31, pp. 14–21, 2020, doi: 10.5120/ijca2020920854.

[19] P. Schummer, A. del Rio, J. Serrano, D. Jimenez, G. Sánchez, and Á. Llorente, "Machine Learning-Based Network Anomaly Detection: Design, Implementation, and Evaluation," AI, vol. 5, no. 4, pp. 2967–2983, 2024, doi: 10.3390/ai5040143.

[20] Peng Zhou, "Payload-based Anomaly Detection for Industrial Internet Using Encoder Assisted GAN," in 2020 IEEE 6th International Conference on Computer and Communications, 2020, pp. 669–673.

[21] A. Chatterjee and B. S. Ahmed, "IoT anomaly detection methods and applications: A survey," Internet of Things (Netherlands), vol. 19, no. June, p. 100568, 2022, doi: 10.1016/j.iot.2022.100568.

[22] B. Nawaal, U. Haider, I. U. Khan, and M. Fayaz, "Signature-Based Intrusion Detection System for IoT," Cyber Secur. Next-Generation Comput. Technol., no. November, pp. 141–158, 2024, doi: 10.1201/9781003404361-8.

[23] A. Abbas, M. A. Khan, S. Latif, M. Ajaz, A. A. Shah, and J. Ahmad, "A New Ensemble-Based Intrusion Detection System for Internet of Things," Arab. J. Sci. Eng., vol. 47, no. 2, pp. 1805–1819, 2022, doi: 10.1007/s13369-021-06086-5.

[24] G. Rekha, S. Malik, A. K. Tyagi, and M. M. Nair, "Intrusion detection in cyber security: Role of machine learning and data mining in cyber security," Adv. Sci. Technol. Eng. Syst., vol. 5, no. 3, pp. 72–81, 2020, doi: 10.25046/aj050310.

[25] A. Meleshko and V. Desnitsky, "The Modeling and Detection of Attacks in Role-Based Self-Organized Decentralized Wireless Sensor Networks," Telecom, vol. 5, no. 1, pp. 145–175, 2024, doi: 10.3390/telecom5010008.

[26] Z. Yang, Z. Sarwar, I. Hwang, R. Bhaskar, B. Y. Zhao, and H. Zheng, "Can Virtual Reality Protect Users from Keystroke Inference Attacks?," 2023, [Online]. Available: http://arxiv.org/abs/2310.16191

[27] M. S. Hammad, R. E. N. Altarazi, R. N. Al Banna, D. F. Al Borno, and S. S. Abu-naser, "A Proposed Expert System for Diagnosis of Migraine," vol. 7, no. 6, pp. 1–8, 2023.

[28] J. Sen and S. Mehtab, "Machine Learning Applications in Misuse and Anomaly Detection," Secur. Priv. From a Leg. Ethical, Tech. Perspect., pp. 1–22, 2020, doi: 10.5772/intechopen.92653.

[29] I. E. Salem, M. M. Mijwil, A. W. Abdulqader, M. M. Ismaeel, A. Alkhazraji, and A. M. Z. Alaabdin, "Introduction to The Data Mining Techniques in Cybersecurity," Mesopotamian J. CyberSecurity, vol. 2022, pp. 28–37, 2022, doi: 10.58496/MJCS/2022/004.

[30] R. R. Asaad and R. M. Abdulhakim, "The Concept of Data Mining and Knowledge Extraction Techniques," Qubahan Acad. J., vol. 1, no. 2, pp. 17–21, 2021, doi: 10.48161/qaj.v1n2a43.

[31] C. Singh, "Machine Learning in Pattern Recognition," Eur. J. Eng. Technol. Res., vol. 8, no. 2, pp. 63–68, 2023, doi: 10.24018/ejeng.2023.8.2.3025.

[32] M. Mohamed, A. Abdullah, A. M. Zaki, F. H. Rizk, M. M. Eid, and E. M. El El-Kenway, "Advances and Challenges in Feature Selection Methods: A Comprehensive Review," J. Artif. Intell. Metaheuristics, vol. 7, no. 1, pp. 67–77, 2024, doi: 10.54216/jaim.070105.

[33] M. Kumar, C. Sharma, S. Sharma, N. Nidhi, and N. Islam, "Analysis of Feature Selection and Data Mining Techniques to Predict Student Academic Performance," 2022 Int. Conf. Decis. Aid Sci. Appl. DASA 2022, no. March, pp. 1013–1017, 2022, doi: 10.1109/DASA54658.2022.9765236.

[34] I. H. Sarker, "Machine Learning: Algorithms, Real-World Applications and Research Directions," SN Comput. Sci., vol. 2, no. 3, pp. 1–21, 2021, doi: 10.1007/s42979-021-00592-x.

[35] A. F. A. H. Alnuaimi and T. H. K. Albaldawi, "An overview of machine learning classification techniques," BIO Web Conf., vol. 97, pp. 1–24, 2024, doi: 10.1051/bioconf/20249700133.

[36] T. ALASALI and Y. ORTAKCI, "Clustering Techniques in Data Mining: A Survey of Methods, Challenges, and Applications," Comput. Sci., no. June 2024, doi: 10.53070/bbd.1421527.

[37] P. Shetty and S. Singh, "Hierarchical Clustering: A Survey," Int. J. Appl. Res., vol. 7, no. 4, pp. 178–181, 2021, doi: 10.22271/allresearch.2021.v7.i4c.8484.

[38] J. Landaburu, "済無No Title No Title No Title," J. GEEJ, vol. 7, no. 2, pp. 1–23, 2016, [Online]. Available: http://www.joi.isoss.net/PDFs/Vol-7-no-2-2021/03_J_ISOSS_7_2.pdf

[39] S. Pitafi, T. Anwar, and Z. Sharif, "A Taxonomy of Machine Learning Clustering Algorithms, Challenges, and Future Realms," Appl. Sci., vol. 13, no. 6, 2023, doi: 10.3390/app13063529.

[40] C. A. Buckner et al., "We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists TOP 1 %," Intech, vol. 11, no. Tourism, p. 13, 2016, [Online]. Available: https://www.intechopen.com/books/advanced-biometric-technologies/liveness-detection-in-biometrics

[41] A. Rachwał et al., "Determining the Quality of a Dataset in Clustering Terms," Appl. Sci., vol. 13, no. 5, pp. 1–20, 2023, doi: 10.3390/app13052942.

[42] D. Phiri, M. Simwanda, V. Nyirenda, Y. Murayama, and M. Ranagalage, "Decision tree algorithms for developing rulesets for object-based land cover classification," ISPRS Int. J. Geo-Information, vol. 9, no. 5, pp. 1–16, 2020, doi: 10.3390/ijgi9050329.

[43] P. Dini, A. Elhanashi, A. Begni, S. Saponara, Q. Zheng, and K. Gasmi, "Applied Sciences Overview on Intrusion Detection Systems Design Exploiting Machine Learning for Networking Cybersecurity," 2023.

[44] L. Boero, M. Cello, M. Marchese, E. Mariconti, T. Naqash, and S. Zappatore, "Statistical fingerprint-based intrusion detection system (SF-IDS)," Int. J. Commun. Syst., vol. 30, no. 10, 2017, doi: 10.1002/dac.3225.

[45] T. Lappas and K. Pelechrinis, "Data Mining Techniques for (Network) Intrusion Detection Systems," Dep. Comput. Sci. Eng. UC Riverside, Riverside CA 92521, 2007.

[46] Y. Hu, A. Yang, H. Li, Y. Sun, and L. Sun, "A survey of intrusion detection on industrial control systems," Int. J. Distrib. Sens. Networks, vol. 14, no. 8, 2018, doi: 10.1177/1550147718794615.

[47] M. N. Martinez and M. J. Bartholomew, "What does it 'mean'? A review of interpreting and calculating different types of means and standard deviations," Pharmaceutics, vol. 9, no. 2, 2017, doi: 10.3390/pharmaceutics9020014.

[48] M. R. Ahmed, S. Islam, S. Shatabda, A. K. M. Muzahidul Islam, M. Towhidul, and I. Robin, "Intrusion Detection System in Software-Defined Networks Using Machine Learning and Deep Learning Techniques-A Comprehensive Survey," Ieee, no. December, pp. 1–47, 2023, doi: 10.36227/techrxiv. 17153213.v1.

[49] A. Goswami, G. Choudhury, H. K. Sarmah, and A. Begum, "'Markov Chain' - The Most Invaluable Contribution of A. A Markov Towards Probability Theory and Modern Technology: A Historical Search," Int. J. Innov. Res. Sci. Technol., vol. 7, no. 3, 2020.

[50] S. N. Eshun and P. Palmieri, "De-anonymisation of real-world location traces: two attacks based on the hidden Markov model," J. Locat. Based Serv., vol. 18, no. 3, pp. 272–301, 2024, doi: 10.1080/17489725.2024.2385312.

[51] A. Ahmadian Ramaki, A. Rasoolzadegan, and A. Javan Jafari, "A systematic review on intrusion detection based on the Hidden Markov Model," Stat. Anal. Data Min., vol. 11, no. 3, pp. 111–134, 2018, doi: 10.1002/sam.11377.

[52] R. Gaharwal, P. Kumar, and U. Dwivedi, "Xournals Xournals Detection techniques for Intrusion Detection System Xournals," vol. 01, no. 01, pp. 16–20, 2019.

[53] S. Ingale, M. Paraye, and D. Ambawade, "A Survey on Methodologies for Multi-Step Attack Prediction," Proc. 4th Int. Conf. Inven. Syst. Control. ICISC 2020, no. Icisc, pp. 37–45, 2020, doi: 10.1109/ICISC47916.2020.9171106.

[54] M. Rabbani et al., "A review on machine learning approaches for network malicious behavior detection in emerging technologies," Entropy, vol. 23, no. 5, pp. 1–41, 2021, doi: 10.3390/e23050529.

[55] A. Mishra, Y. I. Alzoubi, M. J. Anwar, and A. Q. Gill, "Attributes impacting cybersecurity policy development: Evidence from seven nations," Comput. Secur., vol. 120, 2022, doi: 10.1016/j.cose.2022.102820.

[56] O. Watts, G. E. Henter, T. Merritt, Z. Wu, and S. King, "From HMMS to DNNS: Where do the improvements

come from?," ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc., vol. 2016-May, pp. 5505–5509, 2016, doi: 10.1109/ICASSP.2016.7472730.

[57] G. Alter, "Reflections on the Intermediate Data Structure (IDS)," Hist. Life Course Stud., vol. 10, no. 3, pp. 71–75, 2021, doi: 10.51964/hlcs9570.

[58] Z. Azam, M. M. Islam, and M. N. Huda, "Comparative Analysis of Intrusion Detection Systems and Machine Learning-Based Model Analysis Through Decision Tree," IEEE Access, vol. 11, no. June, pp. 80348–80391, 2023, doi: 10.1109/ACCESS.2023.3296444.

[59] A. Yadav, N. Thaker, D. Makwana, N. Waingankar, and P. Upadhyay, "Intruder Detection System: A Literature Review," SSRN Electron. J., 2021, doi:10.2139/ssrn.3866777.