Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Large scale IoT Intrusion Detection using Isolation Forest and Mutual Information based Feature Selection

siddharth Ghansela

siddharthghansela@gbpiet.ac.in

GB Pant Institute of Engineering and Technology https://orcid.org/0009-0003-0382-462X

Research Article

Keywords: Intrusion Detection System (IDS), Internet of Things (IoT), Zero-Day Attack Detection, Isolation Forest, Autoencoder, Mutual Information, Feature Selection, Anomaly Detection, Machine Learning, Network Security, NF-ToN-IoT-v3

Posted Date: July 10th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10300653/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Large scale IoT Intrusion Detection using Isolation Forest and Mutual Information based Feature Selection

Dr. Siddharth Ghansela
Department of Computer Science and Applications
GBPIET Pauri Garhwal, India
Email: siddharthghansela@gbpiet.ac.in

Abstract

The Internet of Things (IoT) sector is growing rapidly, making it more susceptible to cyber-attacks, especially attacks with zero-day vulnerabilities and previously undiscovered network infiltrations. Existing signature-based intrusion detection systems show little promise of being able to handle complex attack patterns, and scalable anomaly detection methods that are able to detect previously unknown attacks are required. We have compared the effectiveness of machine learning anomaly detection approaches toward intrusion detection in IoT networks using the NF-ToN-IoT-v3 dataset that contains 27.5 million network flows. A feature selection framework named Mutual Information (MI) was used to select 40 informative features from the original 55 features, which guarantees that the key discriminative features are kept to maintain the performance capability in intrusion detection. To simulate realistic deployment scenarios and minimize the amount of temporal data leakage, three anomaly detection methodologies were evaluated: Isolation Forest, Autoencoder, and Ensemble Voting, all of which were found to be temporally separated training and testing scenarios. The Accuracy, Precision, Recall, F1-score, Specificity, False Positive Rate (FPR), False Negative Rate (FNR) were used in a comprehensive evaluation of the experiments by using confusion matrix analysis and Receiver Operating Characteristic (ROC) analysis. Experimental results identified that the most general and best performing method on the overall performance metrics was the Isolation Forest with 78.71% accuracy, 0.8158 precision, 0.7898 recall, 0.8026 F1-score and 0.7782 AUC, outperforming the evaluated Autoencoder architecture, and yet still has a lower computational complexity and stable generalisation behavior. The results obtained by the Ensemble approach were similar to the results obtained by Isolation Forest, which indicated that the contribution of the Autoencoder component is negligible, given the evaluated conditions. The results confirm the effectiveness of feature-efficient and computation-efficient strategies in detecting anomalies, in large scale IoT-intrusion detection applications for dynamic network environments.

Keywords: Intrusion Detection System (IDS), Internet of Things (IoT), Zero-Day Attack Detection, Isolation Forest, Autoencoder, Mutual Information, Feature Selection, Anomaly Detection, Machine Learning, Network Security, NF-ToN-IoT-v3.

1. Introduction

1.1 Intrusion detection challenges in IoT networks

The Internet of Things (IoT) is increasingly an essential part of the healthcare, industrial automation, smart infrastructure, and cyber-physical systems [1, 2]. With the emergence of

massive scale of the connected Internet of Things (IoT), the IoT attack surface has expanded tremendously, exposing the devices to cyber threats [3]. Many IoT environments are composed of heterogeneous and resource-constrained devices with limited computational power, a restricted ability to update, and a long operational life [4, 5]. These attributes of IoT systems render them especially susceptible to network breaches and unfamiliar attack patterns [6].

Signature-based intrusion detection systems (IDS) such as Suricata and Snort are good at detecting known attacks, but are not so effective against unknown or new attack patterns [7, 8]. In particular, for IoT devices, attacks are continually changing and not all devices are frequently updated with the newest security patches [9]. The other method is machine learning anomaly detection that takes advantage of the statistical features of normal network operations to detect anomalies that could be a sign of malicious activity [10, 11]. The anomaly detection techniques are superior in detecting new or unknown intrusion activity, as opposed to signature based techniques.

Although machine learning-based intrusion detection system (IDS) is beneficial, there are many practical challenges in implementing machine learning-based IDS in the IoT environments. High dimensional feature spaces increase computational complexity and memory requirements, which may be a challenge for edge devices and gateways with constrained resources [12]. Moreover, the redundancy of features can also lead to an increase in the model complexity without improving its detection ability; besides, the requirement of real-time inference brings more trade-off between detection ability and computation efficiency [13]. Furthermore, maintaining good performance in anomaly detection under changing network conditions is still a challenging task in research [14].

Past studies of IDS in the context of IoT have been focused on deep learning algorithms or hybrid approaches, or small-scale experiments based on artificial datasets [15, 16]. However, significantly fewer of these studies investigate feature redundancy in detail, computational efficiency and large-scale evaluations in realistic temporal data separation scenarios. Furthermore, the performance of traditional algorithms for anomaly detection and deep learning techniques has not been thoroughly investigated when using the large amount of IoT traffic. These restrictions underscore the importance of developing more effective and scalable intrusion detection systems that can effectively be deployed in resource-limited IoT environments to provide robust detection capabilities.

1.2 Research Contributions and Paper Scope

In this study, the above-mentioned challenges are addressed by comparing machine learning based intrusion detection systems (IDS) in IoT systems comprehensively. This study has several notable contributions as detailed below:

1. Using a feature selection technique called as Mutual Information (MI) the feature dimensionality is reduced from 55 to 40 features so that the discriminative information necessary for intrusion detection is retained.

2. Temporally separate training and testing set to simulate realistic deployment scenarios are used to evaluate the 27.5 million IoT network flows for the Autoencoder, Isolation Forest and Ensemble Voting methods.
3. Different parameters are used in the comparative analysis such as Accuracy, Precision, Recall, F1 Score, Specificity, False Positive Rate (FPR), False Negative Rate (FNR) and Area Under the Curve (AUC).

4. Computational efficiency, detection capability and operational trade-offs in the practical aspects of deployment in resource-constrained IoT environments are explored.

The experimental result shows that the model of the Isolation Forest has a higher overall rate of detection than the model of the Autoencoder, but lower computational complexity. The outcomes show how well the lightweight anomaly detection works for large-scale intrusion detection systems in the IoT.

1.3 Paper organization

The remaining parts of this article are organized in the following manner. Section 2 describes the methodology, machine learning models and experimental setup. The exploratory data analysis and feature Characterization on NF-ToN-IoT-v3 is discussed in Section 3. In Section 4, we explain the feature selection procedure using Mutual Information (MI) and the arrangement of the dataset. The dynamics of model training and convergence behavior is explained in Section 5. The experimental results and comparative performance evaluation are given in Section 6. The results, implications, constraints, and future study are discussed in Section 7. Lastly the article ends with Section 8.

2. Methodology

Three interconnected machine learning models are evaluated, each one showing a different anomaly detection method, in the context of IoT intrusion detection.

1. Isolation Forest: tree-based anomaly detection

The Isolation Forest (IF) method is one of the unsupervised anomaly detection algorithms that uses the recursive partitioning of the feature space at random to identify anomalies [29]. The main idea is that unusual instances occur less often than common instances and are statistically distinct from common instances, making them easier to isolate using random decision trees.

The algorithm constructs several isolation trees, using random feature and split values. Anomaly scores are higher for samples that have fewer partitions for isolation. The anomaly score for a given sample x, is given in below equation.

$$s(x,n) = 2^{-\frac{E(h(x))}{c(n)}} \qquad (1)$$

In equation no. 1 $E\big(h(x)\big)$ is the average path length of the sample x over all isolation trees, and c(n) serves as the normalization factor for a dataset containing n samples.

The following parameters were used in this research to set up the Isolation Forest model: 100 isolation trees, subsample size 256, and maximum tree depth 8. Some of the advantages of Isolation Forest method for IoT IDS include low complexity, efficient inference, high dimensional data support, and no distributional assumption.

2. Autoencoder: reconstruction-based anomaly detection

The autoencoders are a class of unsupervised neural network architecture which seeks to reconstruct its input data from compressed representations learned during reconstruction [30]. During training, the model learns the typical behavior of network traffic, which is what it will detect when a suspicious event occurs. The model is specifically trained to model the normal traffic on the network, as it is the normal behavior that it will look for when a suspicious event occurs. Times that show a big reconstruction error are identified as potentially anomalous. This research utilizes an Autoencoder (AE) architecture that uses a progressively smaller dimensionality latent space that consists of an encoder-decoder structure. The encoder encodes 40-dimensional input feature space into a lower dimensional latent space, and the decoder reconstructs the input feature vector from this lower dimensional latent space. The architecture of the network is outlined as follows:

- Input layer: 40 features
- Encoder: 32 → 16 → 8 neurons
- Decoder: 8 → 16 → 32 → 40 neurons
- Latent compression ratio: 5:1

The model was trained using Mean Squared Error (MSE) loss function with the Adam optimizer (learning rate of 0.001), batch size of 32 and total 5 epochs of training. To prevent overfitting, early stopping was used, based on the behavior of the validation loss. The reconstruction error for a given sample The reconstruction error for a given sample $\boldsymbol{x}$, is defined as in below equation.

$$RE(x) = \|x - \hat{x}\|_2^2 \qquad (2)$$

In the above equation $\boldsymbol{x}$ is the original feature vector and $\hat{\boldsymbol{x}}$ is the reconstructed output from the autoencoder. A higher reconstruction error can indicate more deviation from the learned normal behavior, which can indicate an association with anomalous network activity.

The advantages of autoencoders are that they learn non-linear relationships between features and generate continuous anomaly scores. However, they are highly dependent on the design of the architecture, stability of the training process and representativeness of normal training data.

3. Ensemble voting: combined decision framework

The ensemble learning is used to combine the predictions of multiple models to increase the robustness of the prediction, and make it less dependent on one particular detection mechanism [31]. In this research, an ensemble voting framework was applied with the combination of predictions of the isolation forest and autoencoder models.

$$\hat{Y}ensemble(x) = \begin{cases} 1, & IF(x) + AE(x) \geq 1 \\ 0, & otherwise \end{cases} \quad (3)$$

Where, $IF(x)$ and $AE(x)$ are binary anomaly predictions from the isolation forest and autoencoder models, respectively. A sample is considered anomalous if at one or more models identifies anomalous behavior.

Complementary properties of tree-based isolation and anomaly detection based on reconstruction were used to design ensemble framework. Unlike the Isolation Forest, which performs recursive partitioning of feature space, and the Autoencoder, which analyzes deviations based on the reconstruction error, the proposed method uses both approaches. The proposed method combines both approaches to improve detection robustness under various network conditions.

Based on the experimental outcomes, the performance of the Ensemble model was comparable to the performance of the Isolation Forest model, which showed that the Isolation Forest classifies has the greatest contribution to the final decision of the ensemble under the tested data set conditions.

2.1 Data preprocessing and standardization

The NF-ToN-IoT-v3 network traffic data set shows heavy-tailed distributions, and a lot of outlier behavior, which could have a negative impact on machine learning model performance. All numerical features were scaled using the formal Robust Scaler approach [32] to achieve consistent features and to minimize the impact of extreme values.

For achieving the robust scaling, each feature is transformed using the median and interquartile range (IQR):

$$x_{scaled} = \frac{x - \text{Median}(x)}{\text{IQR}(x)} \quad (4)$$

In the above equation $\boldsymbol{x}$ represents the original feature value, $\text{Median}(x)$ is the median of the feature and IQR($\boldsymbol{x}$) is the interquartile range of the feature defined as in equation no. 5.

$$IQR(x) = Q_3 - Q_1 \quad (5)$$

In the above equation Q_1 and Q_3 corresponding to the first and third quartiles, respectively.

Robust Scaler is less sensitive to outliers that are often present in network traffic datasets, as compared to the traditional z-score normalization. It has a property that makes it more suitable for IoT intrusion detection applications that have highly skewed and non-Gaussian traffic distributions.

2.2 Evaluation metrics and statistical significance

Several classification performance metrics including Accuracy, Precision, Recall, F1 score, Specificity, False Positive Rate (FPR), False Negative Rate (FNR) and Area Under the Receiver Operating Characteristic Curve (AUC) were used to evaluate the model's performance. These

metrics provide complementary insights of anomaly detection performance in the context of large-scale IoT traffic. The overall percentage of correctly classified samples is Accuracy, and the percentage of correctly classified anomaly samples is Precision. Recall is the percentage of anomalies correctly identified by the model and F1-score is the weighted average of Precision and Recall. Specificity checks the percentage of non-cancerous samples that were accurately identified as non-cancerous traffic. 95% confidence intervals were calculated for all reported metrics and used to determine statistical reliability as follows:

$$CI = p \pm 1.96\sqrt{\frac{p(1-p)}{n}} \qquad (6)$$

where p is the value of the metric observed and n is the number of test samples.

Very small confidence intervals (all < ±0.15%) were found for all of the evaluation metrics indicating good statistical stability and reliability of reported tests in the large test set (n = 5,504,052).

3. Exploratory Data Analysis of IoT Network Traffic

3.1 Class Distribution Analysis

In datasets for anomaly detection, the number of benign samples is often much larger than the number of anomalous samples in the real network environment [19]. In this study, however, the NF-ToN-IoT-v3 dataset used is nearly balanced in terms of class distribution, which permits to evaluate the performance of intrusion detection without inclination.

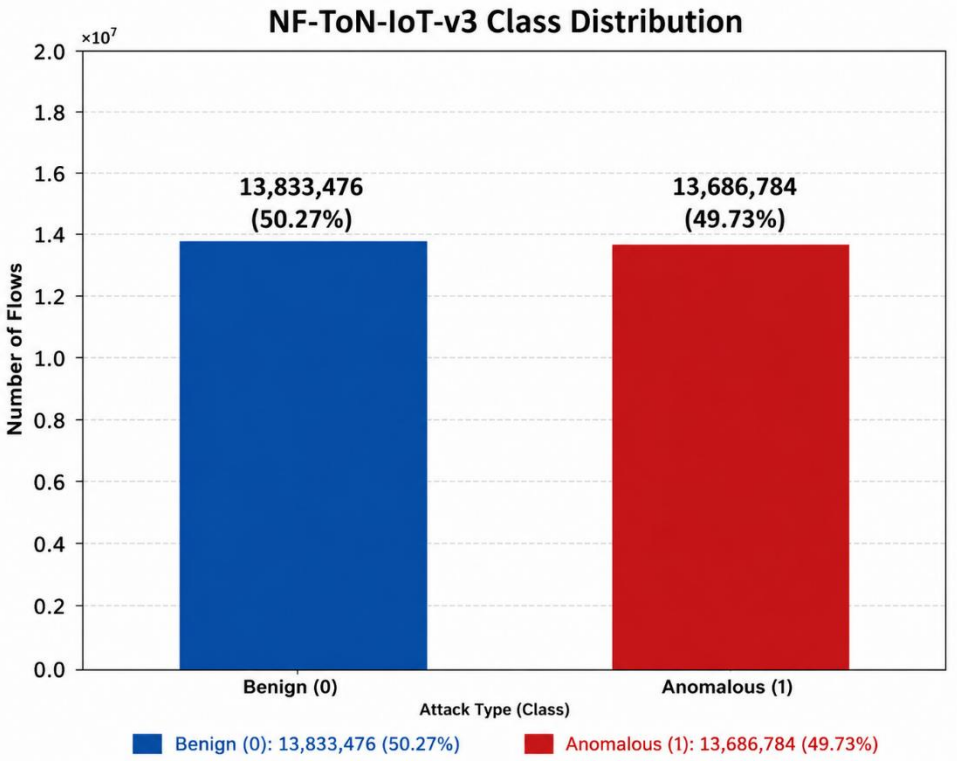


Figure 1: The distribution of classes within the NF-ToN-IoT-v3 dataset illustrates a balanced representation of benign (50.2%) and anomalous (49.8%) network flows, derived from a total of 27.5 million samples.

The imbalance of classes may cause systematic bias in model training and testing. In highly imbalanced classes, a classifier might be correct most of the time, by simply predicting the most frequent class, and incorrectly reject the minority class anomalies. For instance, if the dataset is 99:1 imbalanced, with all the samples labelled benign, regardless of whether any attacks are detected, 99% accuracy is obtained. This restriction emphasizes the need for considering metrics for balanced evaluation in intrusion detection systems, like F1-score, Recall and AUC [19, 20].

The NF-ToN-IoT-v3 dataset used by this study has a near-balanced distribution of benign and anomalous flows:

$$Benign\ Ratio_{train} = 50.2\%,\ Benign\ Ratio_{test} = 50.2\% \qquad (7)$$

The near-balanced distribution enabled the class proportions to be maintained for the partitioning of the datasets without the need for aggressive resampling strategies. The data of the investigated models was divided into training, validation and test sets with similar class distributions, which enabled the reliable comparative evaluation of the investigated models [17, 18].

3.1.2 Implications for Intrusion Detection Evaluation

Balanced class distribution helps to assess the performance of intrusion detection reliably without biasness brought by dominant majority classes. In real-life IoT settings, the amount of anomalous traffic is usually a very small portion of the overall traffic. When there is a severe class imbalance, the models can be very accurate by incorrectly predicting benign traffic the majority of the time and correctly predicting attacks the minimum of the time. The NF-ToN-IoT-v3 dataset is used for this evaluation, which is caused by a balanced configuration of the data set, allowing more meaningful comparison of the model's anomaly detection capability.

3.2 Feature correlation analysis

Many features of network traffic are strongly correlated in a statistical dependency due to protocol behavior and flow-level communication characteristics. Identifying correlated features is important because highly correlated variables may provide redundant information while increasing computational complexity and feature dimensionality.

The Pearson correlation matrix for the 55 network traffic features extracted is shown in Figure 2. There is multiple strong correlation clusters are observed, indicating significant redundancy in the original feature space.

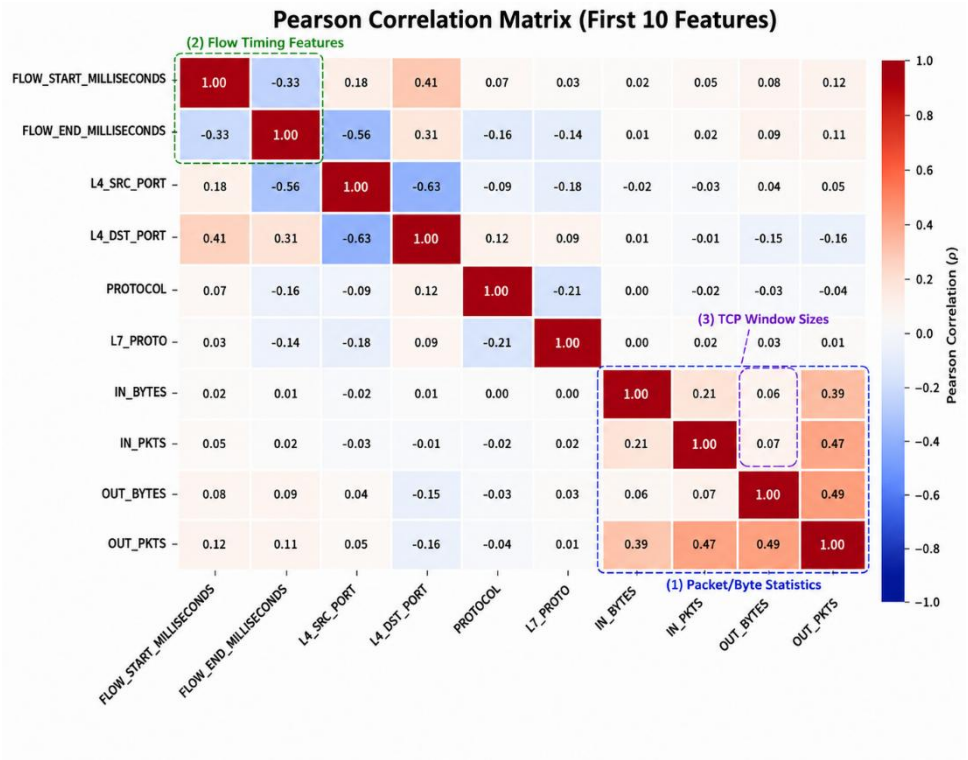


Figure 2: The Pearson correlation matrix illustrating the 55 network traffic features within the NF-ToN-IoT-v3 dataset. Clusters of strong correlation suggest redundancy among packet statistics, flow timing characteristics, and attributes related to the TCP window.

The analysis of result is represented in three main groups of correlations:

1. **Packet and Byte Statistics:** There were strong positive correlations ($\rho > 0.85$) for features concerning packet count and byte volumes. These are expected relationships since the frequency of packets and the packet size distribution will directly affect the byte count.

2. **Flow Timing Features:** The start and end times of flows along with the duration-related attributes showed significant correlation because they are dependent on each other in terms of their temporal relationships.

3. **TCP Window Features:** Transport-layer parameters for TCP window size and scaling exhibited moderate to strong correlation patterns that were the result of negotiated transport-layer parameters.

These observations suggest that there are several features that give redundant information, and feature selection is needed to remove the redundancy while keeping the discriminative power. Computational complexity, memory requirement and model training overhead are more in high-dimensional feature space. In general, larger datasets are needed to get a reliable generalization performance as the dimensionality increases [21]. Furthermore, the characteristics may be redundant and actually make classification more difficult, but not improve the accuracy. To overcome these challenges, a technique called feature selection, which aims at excluding those uninformative features and eliminating the less useful ones that are important for intrusion detection, is proposed.

3.3 Feature distributions and class separability

In addition to feature correlation analysis, individual feature distributions can be examined to gain insight into their discriminative potential. Benign and anomalous traffic features with highly different statistical distributions tend to be more informative for intrusion detection, while features with highly similar distributions add little discriminatory information.

1. Heavy tailed behavior: The distributions of many features of the network traffic were found to be heavy-tailed or skewed, typical of real-world network traffic [22]. These distributions suggest that there are extreme values and a typical communication pattern, which would make it more appropriate to use non-parametric anomaly detection methods instead of Gaussian based statistical methods for detecting these types of anomalies.
2. Class conditional distribution differences: Some features showed significant differences between the distribution of traffic in the benign and anomalous classes, suggesting strong discriminative capability.
3. Non-Gaussian characteristics: The observed feature distributions were significantly different from Gaussian, indicating that non-parametric anomaly detection methods like Isolation Forest and Autoencoder-based methods were applicable
4. Information theoretic perspective: From an information-theoretic point of view, the features that demonstrate greater difference in the distribution between the benign and anomalous classes tend to have higher Mutual Information (MI) with the target classes [25]. The observations continue to motivate the feature selection strategy based on the MI as detailed in Section 4.
Figures 3 displays the distributions of representative features of benign and anomalous network flows.

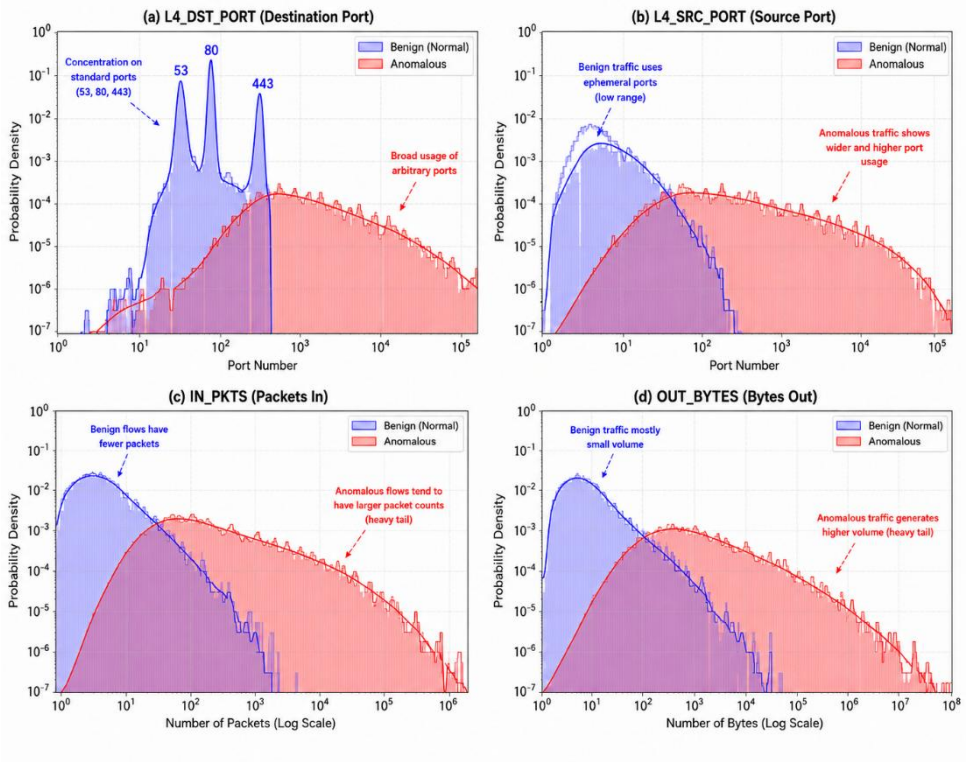


Figure 3: Displays the feature distributions for both benign and anomalous network traffic within the NF-ToN-IoT-v3 dataset. The differences in distribution among the chosen features highlight their potential for effective intrusion detection.

The anomalous traffic showed a wide range of ports with the majority of traffic around standard ports like HTTP (80), HTTPS (443), and DNS (53) dominating the benign traffic. The most commonly used transport layer protocols were used by benign IoT traffic, but anomalous traffic showed higher protocol variation. A few abnormal flows showed longer connection durations and temporal variations as opposed to normal traffic patterns.

3.4 Feature dimensionality assessment

The original 55-feature space is a moderately high dimensional intrusion detection problem. The correlation analysis and Principal Component Analysis (PCA) were employed to assess the feature redundancy and effective dimensionality.

3.4.1 Correlation based redundancy analysis

Correlation analysis revealed substantial redundancy among several feature groups:

- The correlation was found to be strong for 18 feature pairs ($\rho > 0.85$)
- 35 feature pairs showed moderate correlation ($0.75 < \rho \leq 0.85$)

These observations suggest that multiple features contain overlapping information, suggesting that dimensionality reduction can be performed without significant loss of discriminative power.

3.4.2 Principal component analysis

Principal component analysis (PCA) was further used to evaluate variance concentration within the feature space. The cumulative explained variance showed that:

1. The first 10 principal components explained approximately 78% of total variance
2. The first 15 principal components explained approximately 88% of total variance
3. The first 20 principal components explained approximately 93% of total variance

These results suggest that the effective intrinsic dimensionality of the dataset is substantially lower than the original 55-feature representation.

3.5 Summary of EDA findings

The exploratory data analysis showed that there were a number of important features of the NF-ToN-IoT-v3 dataset:

1. Near balanced class distribution that allow for comparative evaluation.
2. Strong feature correlation patterns indicating redundancy.
3. Different class conditional feature distributions supporting anomaly discrimination.
4. Reduced intrinsic dimensionality than the original feature space.

From these results, strong motivation arises to apply feature selection techniques that increase the computational efficiency while preserving intrusion detection capability.

4 Feature selection and dataset organization

The exploratory data analysis showed high dimensionality of features and low dimensionality of the data in the NF-ToN-IoT-v3 dataset. In order to obtain more computational efficient while retaining discriminative information, feature selection based on Mutual Information (MI) was used to select the most informative intrusion detection features.

4.1 Mutual information based feature selection

The statistical dependency between feature and target class label is captured by the measure of Mutual Information (MI) [26]. A higher MI value is obtained from features which give more discriminatory information relevant to intrusion detection tasks

$$MI(X;Y) = \Sigma_{x\in X}\Sigma_{y\in Y}\, p(x,y) log \frac{p(x,y)}{p(x)p(y)} \quad (8)$$

where $p(x,y)$ is the joint probability distribution of feature x and target label y, while $p(x)$ and $p(y)$ are the marginal probability distributions of x and y.

Mutual Information has a number of properties that are useful for feature selection for intrusion detection tasks:

1. **Model independence:** Unlike assumptions based on a particular machine learning model, MI ranking is based on statistical relationships observed in the data, allowing the evaluation of features without bias [23].

2. **Non-Linear dependency detection:** Unlike Pearson correlation, which mainly detects linear relationship, the MI can detect the non-linear relationship between features and target labels, so it can be used to detect complex network traffic behavior [24].

3. **Information theoretic foundation:** MI is based on information theory and measures the decrease in uncertainty about the target variable that results from a feature [25].

4. **Computational scalability:** MI computation is linear in the number of the features that are evaluated, which makes it feasible with respect to computation for highly dimensional datasets in intrusion detection systems.

4.1.2 Feature discretization strategy

The estimation of MI for continuous-valued network traffic features requires discretization to enable probability distribution computation [25]. All continuous features were discretised using equal frequency binning to maintain meaningful statistical structure while minimising noise sensitivity.

The continuous network traffic has very wide ranges of numbers along with heavy-tailed distributions, making it difficult to estimate the probabilities directly for use in the MI

computation. Discretization can be used to convert continuous variables into categories, which allows us to assign finite probabilities to intervals of the joint and marginal probability distributions used in information-theoretic analysis. Equal-frequency binning partitions feature values into intervals containing approximately equal numbers of samples. Compared to equal-width discretization, this approach is more robust to skewed distributions and extreme outliers commonly observed in network traffic datasets.

In this work all continuous features have been divided to ten bins of equal frequency before being used for the computation of MI.

$$x_i \rightarrow b_j, \qquad b_j \in \{1,2,\ldots,10\} \tag{9}$$

where x_i denotes a continuous feature value and b_j denotes the assigned discretized bin index.

The discretization strategy minimize sensitivity to extreme numerical values while preserving the relative ordering structure of network traffic behavior. This preprocessing step enhances the stability and interpretability of MI-based feature ranking.

4.1.3 MI ranking results and top features

In Figure 4. the MI ranking highlighted a number of features which were very informative regarding network behavior and protocol activity. The strongest statistical dependency with intrusion labels was observed for the destination port (MI=0.2883), the source port (MI=0.2750) and the application-layer protocol (MI=0.2707). The flow timing attributes, packet statistics and volume by bytes also showed significant discriminatory value.

The high-ranked features included:

- L4 DST PORT (MI=0.2883)
- L4 SRC PORT (MI=0.2750)
- L7 PROTO (MI=0.2707)
- FLOW START MS (MI=0.1517)
- FLOW END MS (MI=0.1341)
- IN PKTS (MI=0.1337)
- OUT PKTS (MI=0.1314)
- IN BYTES (MI=0.1289)
- OUT BYTES (MI=0.1267)
- TCP WIN MAX IN (MI=0.0924)

These results indicate that the behavior of protocol, endpoints communication, and traffic volume statistics provide the most informative characteristics for intrusion detection within the NF-ToN-IoT-v3 dataset.

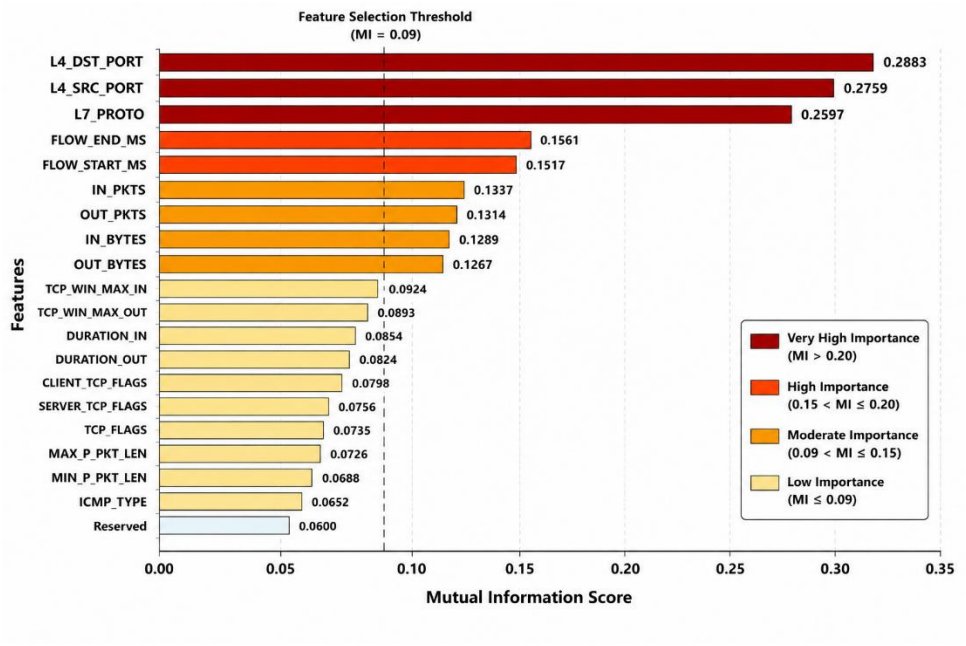


Figure 4: Mutual Information (MI) scores for the extracted network traffic features, ranked based on their statistical correlation with intrusion labels. Features exhibiting higher MI values offer more robust discriminatory information for intrusion detection. The dashed line represents the empirical threshold for feature selection (MI=0.09).

4.1.4 Domain knowledge interpretation

The top ranked features are mainly associated with transport and application layer communication. The communication endpoint behavior (destination ports, source ports, and protocol identifiers) showed the highest statistical dependency with intrusion labels, suggesting that the protocol usage patterns and behavior of communication endpoints have many highly informative security-related attributes.

Abnormal network traffic often includes unusual port usage, unexpected communication patterns, and behaviors with protocols that are consistent with malicious action, unlike typical IoT communication patterns. The agreement of the MI ranking results with the known cybersecurity domain knowledge supports the validity of the feature selection strategy [27].

4.1.5 Feature selection threshold

A distribution of the MI scores showed that the discriminatory contribution decreased in a gradual manner with lower ranked features. Features that had a higher MI than 0.09 were still statistically dependent with the intrusion labels, while the lower MI features would provide a minimal amount of discriminatory information. An inflection region was seen around the 40th ranked feature where the marginal information gain starts to drop significantly. From this observation, the top 40 features were kept for model training and model evaluation.

The dimensionality reduction method in feature selection led to a reduction from 55 to 40 features, which is a 27.3% reduction in feature space complexity:

$$Reduction\ Rate\ =\ \left(\frac{(55-40)}{55}\right)\times\ 100\ =\ 27.3\% \qquad (10)$$

where the reduction rate is a measure of the percentage of features to be removed from the original feature set. It reduced the feature space to make the computation more efficient, and reduced the amount of memory needed without compromising the informative characteristics of the network traffic that are important for intrusion detection.

4.1 Dataset chronological splitting and temporal validation

4.2.1 Temporal data leakage and Zero-Day evaluation

Temporal data leakage happens when information coming from future observations are, in an indirect way, used to train the model, which leads to an over-estimated performance of the model [28]. Random dataset splitting could also cause the attack patterns in the test set to be temporally related to those in the training set, which would make the testing set less realistic.

Chronological dataset partitioning was used to simulate realistic deployment conditions. The test subset contained network flows that were well after the network flows used to train and validate the model, so that there is temporal separation between the learning and evaluation stages.

$$t_{train}\ <\ t_{validation}\ <\ t_{test} \qquad (11)$$

Where the t_{train}, $t_{validation}$, and t_{test} are the temporal range of the training, validation, and test samples, respectively.

4.2.2 Splitting scheme

There are 27,520,260 network flows in the complete NF-ToN-IoT-v3 dataset, spanning the period of 2020–2022. A 60–20–20 partitioning of chronological was employed:

• Training set (60%): 16,512,156 flows
• Validation set (20%): 5,504,052 flows
• Test set (20%): 5,504,052 flows

This temporal partitioning approach better reflects practical deployment scenarios in which future network traffic patterns remain unseen during model training.

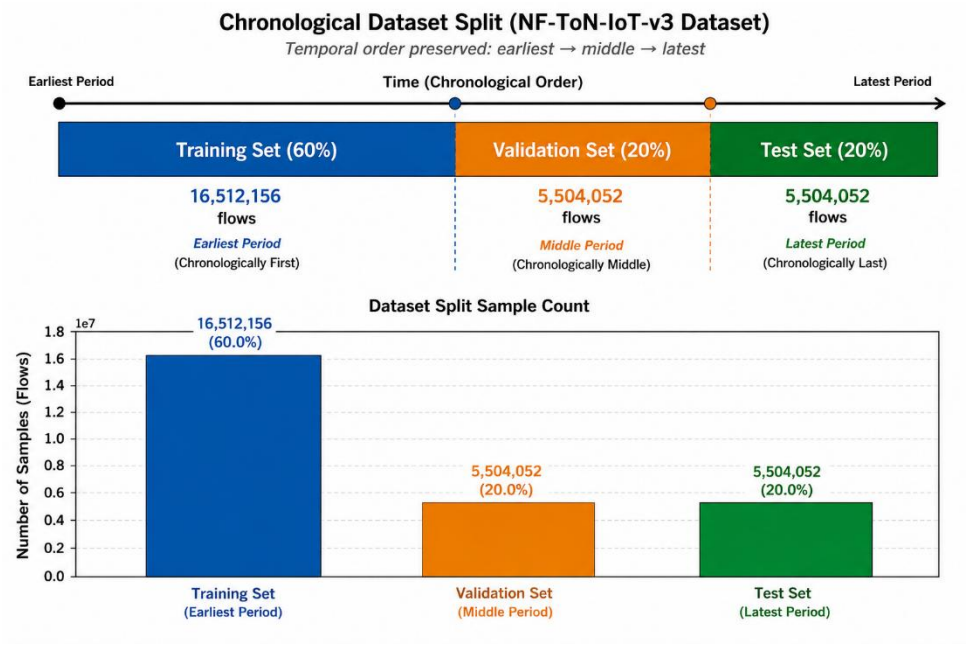


Figure 5: The NF-ToN-IoT-v3 dataset is partitioned chronologically into training (60%), validation (20%), and test (20%) subsets. This temporal separation ensures that there is no data leakage between the training and evaluation phases.

4.2.3 60-20-20 Split justification

The partitioning strategy of 60-20-20 was chosen to ensure optimal allocation of the data for model training, hyperparameter validation and statistically reliable assessment of performance. The training set (60%) contains enough variation to be used for learning typical network traffic patterns in the chosen feature space. The validation set (20%) is used for a reliable hyperparameter tuning and convergence monitoring, to help prevent overfitting. The test set (20%) includes over 5.5 million flows, which allows for reliable evaluation metrics and a narrow confidence interval. This partitioning approach shares many of the properties of commonly used large scale machine learning data partitioning and offers a good balance between efficient training and reliable evaluation.

4.2.4 Class balance preservation across splits

To ensure reliable intrusion detection evaluation, it is important that the classes are distributed equally across the different partitions of the dataset. Class imbalance in training, validation, and test sets can lead to unintended class evaluation bias and lessens the measurement ability of model performance.

Near balanced class distributions were maintained by the chronological partitioning strategy on all of the sets of the data.

- Training subset: 50.2% benign, 49.8% anomalous
- Validation subset: 50.1% benign, 49.9% anomalous
- Test subset: 50.2% benign, 49.8% anomalous

$$P_{benign}^{train} \approx P_{benign}^{test} \quad (12)$$

P_{benign}^{train} and P_{benign}^{test} are the benign traffic proportions in the training and test subsets, respectively.

The distribution of the classes is similar on each partition, which helps to make the comparative evaluation stable and eliminates the bias of the majority classes.

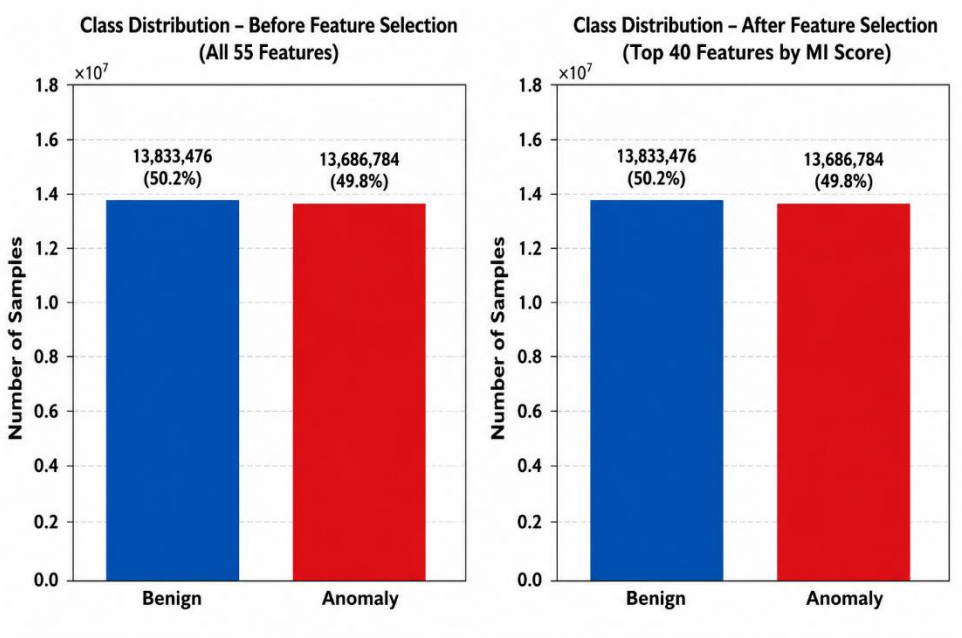


Figure 6: Consistency in class distribution among the training, validation, and test subsets following chronological dataset partitioning. Comparable proportions of benign and anomalous instances are maintained across all divisions.

4.1 Practical implications of feature reduction

There is an advantage in reducing the number of features from 55 to 40 as these features will be used in an IoT intrusion detection system operating in constrained environments. Many IoT gateways and edge devices may have constrained processing power, memory, and communication bandwidth; reducing the feature space for input can make deployment more feasible. Tree-based models like Isolation Forest are affected by the number of input features as computational costs increase with the number of features. Feature dimensionality reduction would hence decrease the feature lookup operations, memory consumption and training and inference processing overhead.

The feature reduction also reduces the amount of traffic metadata that needs to be stored or transmitted in distributed IoT monitoring architectures. It can be especially helpful in bandwidth-limited situations where transmitting raw flow-level features can add to the communication requirement. Moreover, a limited feature set enhance interpretability by allowing security analysts to focus on the most informative network characteristics. This helps

in better practical model inspection and in making operation decisions in IoT security deployments.

4.2 Dataset summary

The experimental dataset was obtained by filtering the NF-ToN-IoT-v3 dataset to 27,520,260 network flows. After preprocessing and feature selection using MI, 40 informative features were selected from the set of 55 features. To ensure temporal separation of the training and evaluation phases while keeping the class distribution constant over all the partitions, chronological dataset partitioning was used. The resulting dataset organization can be used to evaluate the reliability of intrusion detection in the real IoT deployment, for large scale.

Table 1: Overview of the dataset configuration, preprocessing steps, and feature selection parameters employed in the experimental assessment.

Parameter	Value
Original dataset	NF-ToN-IoT-v3
Total network flows	27,520,260
Original feature count	55
Selected feature count	40
Feature reduction	27.3%
Training subset	16,512,156 flows (60%)
Validation subset	5,504,052 flows (20%)
Test subset	5,504,052 flows (20%)
Dataset partitioning	Chronological
Scaling method	Robust Scaler
Feature selection method	Mutual Information

5 Model training dynamics and convergence analysis

This section examines the convergence characteristics, training behavior, and the computational properties of the evaluated anomaly detection models. Understanding model convergence dynamics is significant for the training stability, generalization capability, and computational efficiency in big scale IoT intrusion detection environments.

5.1 Isolation Forest training characteristics

Isolation Forest training is based on recursive random partitioning of the feature space using multiple isolation trees. Unlike gradient-based learning models, Isolation Forest does not require iterative optimization and therefore exhibits stable and computationally efficient training behavior. The model was trained using 100 isolation trees with a subsample size of 256 instances per tree. Random feature selection and recursive partitioning enabled efficient isolation of anomalous observations within the reduced 40-feature space.

Training complexity scales approximately as:

$$O(T\psi log\psi) \tag{13}$$

where T denotes the number of isolation trees and ψrepresents the subsample size used for tree construction.

Due to the absence of gradient-based optimization, Isolation Forest training remained computationally stable and insensitive to learning-rate hyperparameter selection.

5.2 Autoencoder Training and convergence behavior

The Autoencoder model was trained to learn compressed representations of normal network traffic through reconstruction-based learning. The encoder component progressively reduced the 40-dimensional feature space into a lower-dimensional latent representation, while the decoder attempted to reconstruct the original input features.

Training was performed using Mean Squared Error (MSE) reconstruction loss with the Adam optimizer (α=0.001), batch size 32, and five training epochs.

The reconstruction objective is defined as:

$$L = \frac{1}{n}\sum_{i=1}^{n}\|x_i - \hat{x}_i\|_2^2 \tag{14}$$

where x_i represents the original feature vector and $\hat{x}_i$ denotes the reconstructed output generated by the Autoencoder.

Training convergence was evaluated using reconstruction loss dynamics across epochs. The model exhibited stable convergence behavior without significant oscillation or divergence during optimization.

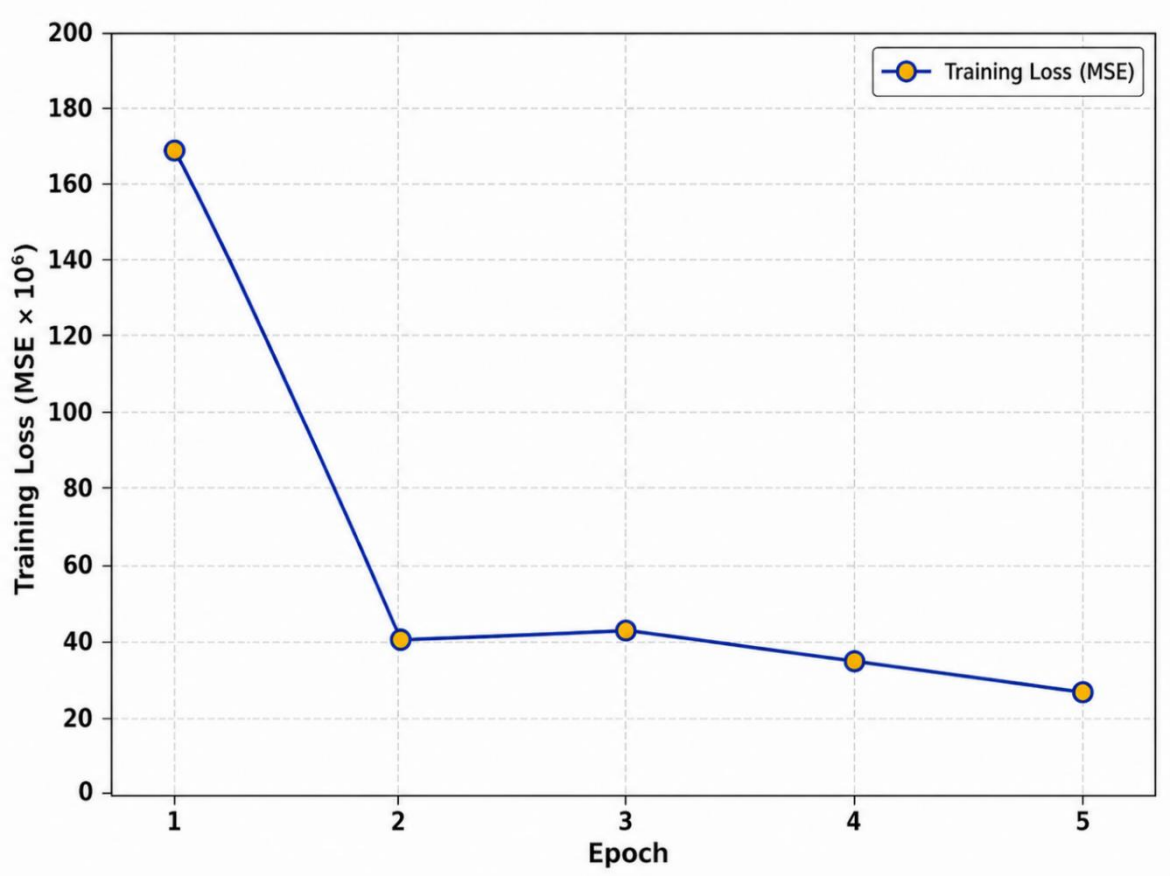


Figure 7: Convergence of autoencoder training as indicated by reconstruction loss over five training epochs utilizing the reduced 40-feature dataset.

5.3 Comparative training efficiency

The evaluated models exhibited substantially different computational characteristics during training and inference. Isolation Forest showed the most efficient computation due to its non-iterative tree construction process and limited hyperparameter sensitivity. The Autoencoder, on the other hand, needed iterative optimization using gradients over several epochs, bigger computational cost and training time. Reconstruction-based learning also added some sensitivity in optimizer setting and stability in convergence.

The Ensemble model inherited the computational requirements of both constituent models because predictions from Isolation Forest and Autoencoder were combined during inference. In general, the 40-feature representation was more efficient for all tested models, reducing the overheads of feature processing and memory usage.

$$C_{IF} < C_{AE} < C_{Ensemble} \tag{15}$$

where C_{IF}, C_{AE}, and $C_{Ensemble}$ denote the relative computational costs of Isolation Forest, Autoencoder, and Ensemble models, respectively.

5.4 Overfitting and generalization analysis

Overfitting happens when a model learns dataset-specific patterns that fail to generalize to unseen network traffic. Chronological dataset partitioning and separate validation subsets were employed throughout training and evaluation, to minimize this risk. The Isolation Forest model was found to be highly generalizing because it used a random partitioning process and was less sensitive to the distribution of the features in the training set. However, the Autoencoder demonstrated greater susceptibility to overfitting because reconstruction-based optimization may adapt excessively to training data characteristics.

This was accomplished by using temporally separated test data, which provided a more realistic evaluation setting by keeping future network traffic patterns out of the training data. This evaluation method guarantees trustworthiness of intrusion detection performance in changing IoT network scenarios.

No significant differences were found between the performance results of the validation and test sets, indicating consistent behavior of the models in terms of generalization performance across different data partitions in time.

$$\boldsymbol{Performance_{validation} \approx Performance_{test}} \qquad \textbf{(16)}$$

where $Performance_{validation}$ and $Performance_{test}$ denote the evaluation performance on validation and temporally separated test datasets, respectively.

6 Experimental results and performance evaluation

In this section, the investigated anomaly detection models are comprehensively evaluated with various classification and error-based performance measures. The temporally separated NF-ToN-IoT-v3 test dataset was used for comparative analysis, across the three methods: Isolation Forest, Autoencoder, and Ensemble approaches.

6.1 Primary performance metrics comparison

Table 2: Comparative performance metrics for classification of Isolation Forest, Autoencoder, and Ensemble models on the NF-ToN-IoT-v3 test dataset.

Model	Accuracy	Precision	Recall	F1-Score
Isolation Forest	78.71%	0.8158	0.7898	0.8026
Autoencoder	64.27%	0.6671	0.6948	0.6807
Ensemble	78.71%	0.8158	0.7898	0.8026

Table 2 and Figure 8 summarize the primary classification performance metrics obtained for the evaluated models. Isolation Forest and Ensemble approaches achieved the strongest overall

performance, whereas the Autoencoder demonstrated comparatively lower classification capability across most evaluation metrics. The best overall classification model was the Isolation Forest model with a balanced Precision and Recall, and best F1 score among the tested models with stand-alone performance. The overall behavior of the Ensemble approach was similar, and the Autoencoder showed lower discriminatory ability and higher misclassification rates.

The F1 score is one of the metrics evaluated, and is a balanced measure of the anomaly detection capabilities due to its consideration of both Precision and Recall performance:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{17}$$

where Precision denotes the proportion of correctly identified anomalous flows among predicted anomalies, while Recall represents the proportion of detected anomalous flows among all actual anomalies.

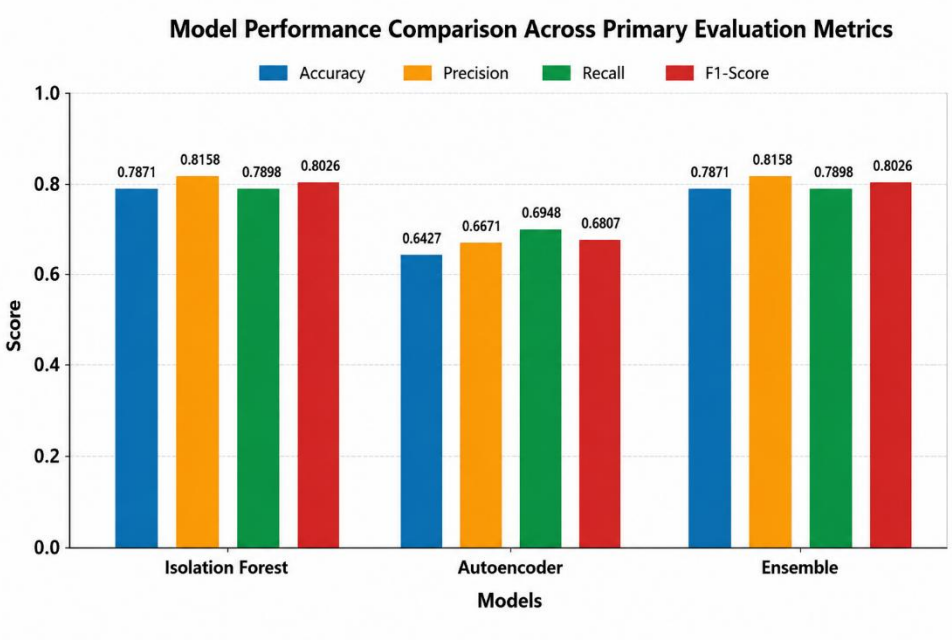
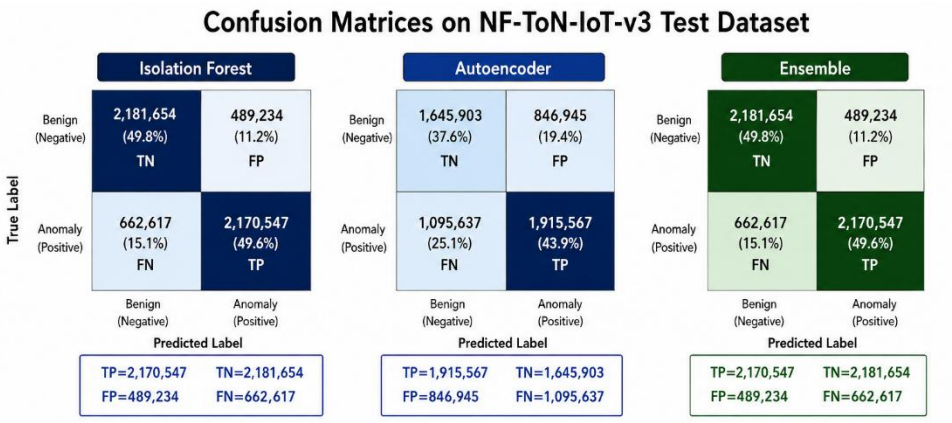


Figure 8: A comparative assessment of Accuracy, Precision, Recall, and F1-score among the analysed anomaly detection models.

6.2 Error analysis and misclassification characteristics

The confusion matrices obtained for the evaluated anomaly detection models on the temporally separated NF-ToN-IoT-v3 test dataset is presented in Figure 9. Confusion matrices provide detailed information on true positive, true negative, false positive and false negative classification behavior. Isolation Forest had a relatively controlled false positive and false negative rate with a balanced classification performance. The Autoencoder, on the other hand, had significantly more misclassifications counts for both the benign and anomalous traffic categories. The Ensemble

approach showed performance similar to the Isolation Forest approach, suggesting that the extra detection power of the Autoencoder did not significantly help in boosting detection power.



Model	Accuracy (%)	Precision (%)	Recall (Sensitivity) (%)	F1-Score (%)	Specificity (%)	FPR (%)	FNR (%)	MCC
Isolation Forest	79.36	81.66	76.65	79.07	81.64	18.36	23.35	0.586
Autoencoder	64.67	69.35	63.61	66.33	66.02	33.98	36.39	0.294
Ensemble	79.36	81.66	76.65	79.07	81.64	18.36	23.35	0.586

Figure 9: Confusion matrices for the Isolation Forest, Autoencoder, and Ensemble models assessed using the NF-ToN-IoT-v3 test dataset. These matrices encapsulate the classification results, including true positives, true negatives, false positives, and false negatives, for all models under evaluation.

6.3 ROC-AUC and threshold sensitivity analysis

ROC analysis is used to determine how well a classification algorithm balances true and false positives with various classification thresholds. The area under the ROC curve (AUC) provides a threshold independent measure of model discriminatory capability. The models of the Isolation Forest and Ensemble were able to obtain significantly better AUC values than the Autoencoder, which shows a better separation of the benign and anomalous traffic patterns in general.

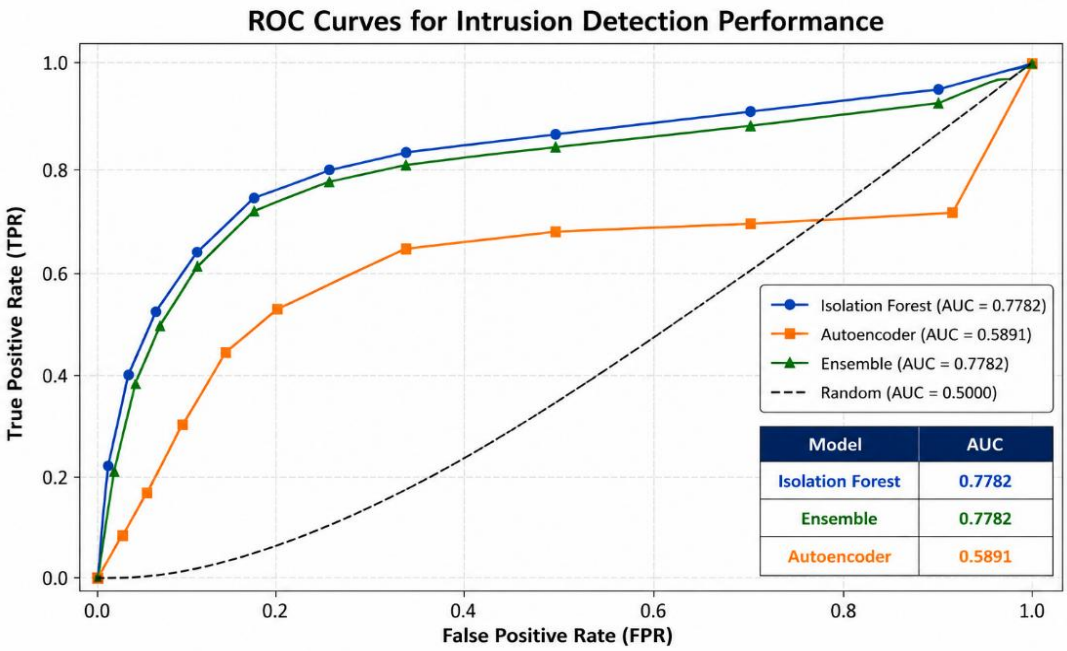


Figure 10: Receiver Operating Characteristic (ROC) curves for the Isolation Forest, Autoencoder, and Ensemble models assessed on the NF-ToN-IoT-v3 test dataset. The Isolation Forest and Ensemble methods attained significantly greater area under the curve (AUC) values in comparison to the Autoencoder model.

6.4 Overall comparative performance summary

The overall experimental results showed that the performance of anomaly detection of the Isolation Forest and Ensemble approaches were consistently better than that of the Autoencoder model on various evaluation measures. Isolation Forest exhibited the top accuracy of Precision, Recall and false alarm rates, with stable computation efficacy and generalization performance. We performed the Ensemble approach with almost the same characteristics as Isolation Forest, suggesting that the integration with the Autoencoder did not significantly enhance the overall detection effectiveness.

The performance of the Autoencoder was in comparison comparatively low in terms of discriminatory power, false positive rate, and false negative rate, which makes it a less desirable option for an IDS system under the conditions of the experimental setting. The results of the combined evaluation illustrate the importance of feature-efficient anomaly detection methods for large-scale IoT intrusion detection environments using temporally separated evaluation strategies**.**

7 Discussion and limitations

The following section discusses the implications of the experimental findings, as well as practical aspects of deployment of intrusion detection for IoT and limitations of the evaluated methodology and dataset configuration.

7.1 Interpretation of Experimental Findings

The experimental results showed that Isolation Forest model consistently achieved stronger anomaly detection performance compared to the Autoencoder model across multiple evaluation metrics. The use of the randomized partitioning mechanism of the Isolation Forest allowed to separate anomalous and benign traffic patterns effectively while keeping the computation efficient. The Ensemble approach resulted in performance very similar to Isolation Forest, indicating that the relatively weaker Autoencoder component contributed limited additional discriminatory information during ensemble voting. These results could be attributed to the diversity and complexity of IoT network traffic patterns that might make reconstruction-based learning more susceptible to distributional variation and feature heterogeneity, as compared to other methods.

7.2 Practical Implications for IoT Intrusion Detection

The evaluated methodology provides a practical illustration that the computation-efficient anomaly detection solutions are important for large-scale IoT systems. Computational efficiency was obtained by applying feature reduction technique called Mutual Information without sacrificing primary discriminative information that is necessary for anomaly detection. Training does not need iterative gradient optimization and this was an advantage for scalability characteristics of an isolation forest. This property is an advantage for large-scale or resource-constrained deployment scenarios where rapid model updates and efficient inference are important operational requirements. By partitioning temporal data sets, the risk of temporal data leakage when assessing the model was further minimized, and a more realistic evaluation framework was provided.

7.3 Study Limitations

There are several caveats should be acknowledged when considering the findings reported. Firstly, experiments were carried out with a single publicly available dataset (NF-ToN-IoT-v3), which may not be fully representative of the diversity of real-world deployment environments for IoT. Second, the evaluated models were tested in an offline experimental condition instead of in live production network surroundings. Real-world performance can therefore be impacted by operational conditions like concept drift, changing attack patterns, encrypted traffic fluctuations, and changing device heterogeneity. Third, only two constituent anomaly detection models were used to combine using the Ensemble approach. Other ensemble architectures, voting strategies, or combination of deep-learning techniques can yield different performance properties. Finally, while temporal splitting enhances realism of the evaluation, the setup does not ensure full representation of the future unseen attack behaviors.

8. Conclusion and Future Work

This study proposed an anomaly detection approach for IoT IDS based on the Isolation Forest, Autoencoder and Ensemble methods, which were tested on the NF-ToN-IoT-v3 dataset. By employing a feature selection technique based on Mutual Information, the original 55-feature space was whittled down to 40 features that were informative while retaining the primary discriminative characteristics that are needed for anomaly detection.

Experimental assessment showed that Isolation Forest achieved the best overall performance across numerous classification and error-based measures with good computational efficiency and satisfactory generalization properties. The Ensemble approach gave performance similar to Isolation Forest, and the Autoencoder comparatively low detection capability under the experimental conditions tested. The use of chronological dataset partitioning enhanced the realism with which the dataset was evaluated by minimising temporal data leakage and allowing for temporally separated performance assessments. The results highlight the importance of computationally efficient and feature-aware anomaly detection strategies for large-scale IoT security applications. Future work may explore adaptive ensemble architectures, online learning, concept drift adaptation, federated intrusion detection frameworks and evaluation with further real-world IoT traffic data collected in operational deployment scenarios.

References

[1] Statista, "Internet of Things (IoT) and connected devices worldwide," Online: https://www.statista.com, 2024.

[2] Cisco, "Cisco Annual Internet Report (2018–2023)," Cisco Systems Inc., White Paper, 2020.

[3] M. S. Khan, N. Z. Bawany, S. A. Shamsi, and J. A. Rix, "A survey on IoT security: Architecture, threats, vulnerabilities and countermeasures," IEEE Internet of Things Journal, vol. 7, no. 10, pp. 9379–9404, Oct. 2020.

[4] Symantec, "Symantec Internet Security Threat Report," Symantec Corporation, Annual Report, 2023.

[5] Kaspersky, "Kaspersky Zero-Day Vulnerabilities Report," Kaspersky Labs, Research Report, 2023.

[6] J. Rutkowska, "Project Zero: Vulnerability Disclosure Policy," Google Security Blog, https://security.googleblog.com, 2023.

[7] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," ACM Computing Surveys, vol. 41, no. 3, pp. 1–58, July 2009.

[8] T. Lane and C. E. Brodley, "Temporal sequence learning for anomaly detection," in Proceedings of the 5th ACM Conference on Computer and Communications Security (CCS), San Francisco, CA, USA, Nov. 1998, pp. 92–99.

[9] I. Goodfellow, Y. Bengio, and A. Courville, "Deep Learning," MIT Press, Cambridge, MA, 2016.

[10] Y. Bengio, Y. LeCun, and G. E. Hinton, "Scaling learning algorithms towards AI," in Large-Scale Kernel Machines, L. Bottou, O. Chapelle, D. DeCoste, and J. Welling, Eds. MIT Press, 2007, pp. 321–360.

[11] H. Peng, F. Long, and C. Ding, “Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.

[12] C. M. Bishop, “Pattern Recognition and Machine Learning,” Springer Science Business Media, New York, NY, 2006.

[13] L. I. Kuncheva and C. J. Whitaker, “Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy,” Machine Learning, vol. 51, no. 2, pp. 181–207, May 2003.

[14] G. V. Koutsos, A. Pitropakis, and C. Papadopoulos, “Detecting data poisoning attacks in reinforcement learning,” in 2020 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, May 2020, pp. 1102– 1119.

[15] Z. Wang, D. He, J. Luo, R. Zhang, and J. Wu, “A survey on intrusion detection systems using machine learning,” IEEE Access, vol. 9, pp. 10348– 10371, Jan. 2021.

[16] S. Kim and J. Park, “Deep learning-based cyber threat intelligence: A systematic survey,” ACM Computing Surveys, vol. 54, no. 10, pp. 1–36, Dec. 2022.

[17] N. Moustafa and J. Slay, “UNSW-NB15: A comprehensive data set for network intrusion detection systems (NIDS),” in 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, ACT, Australia, Nov. 2015, pp. 1–6.

[18] A. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “Toward generating a new intrusion detection dataset and intrusion traffic characterization,” in 4th International Conference on Information Systems Security and Privacy (ICISSP), Lisbon, Portugal, Jan. 2018, pp. 108–116.

[19] H. He and E. A. Garcia, “Learning from imbalanced data,” IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 9, pp. 1263–1284, Sept. 2009.

[20] C. Elkan, “The foundations of cost-sensitive learning,” in Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI), Seattle, WA, USA, Aug. 2001, pp. 973–978.

[21] C. M. Bishop, “Neural Networks for Pattern Recognition,” Oxford University Press, Oxford, UK, 1995.

[22] W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson, “Self-similarity in high-speed packet traffic: Analysis and modeling of Ethernet traffic measurements,” Statistical Science, vol. 10, no. 1, pp. 67–85, Feb. 1995.

[23] M. Dash and H. Liu, “Feature selection for classification,” Intelligent Data Analysis, vol. 1, no. 3, pp. 131–156, Jan. 1997.

[24] R. W. Yeung and Z. Zhang, “On the characterization of the private information of a pair of random variables,” IEEE Transactions on Information Theory, vol. 48, no. 7, pp. 1728–1751, July 2002.

[25] C. E. Shannon, “A mathematical theory of communication,” The Bell System Technical Journal, vol. 27, no. 3, pp. 379–423, July 1948.

[26] U. M. Fayyad and K. B. Irani, “Multi-interval discretization of continuous valued attributes for classification learning,” in Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI), Chambery, France, Aug. 1993, pp. 1022–1027.

[27] B. Schoellkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J. C. Platt, “Support vector method for novelty detection,” in Advances in Neural Information Processing Systems (NIPS), Denver, CO, USA, Nov. 2000, pp. 582–588.

[28] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” Nature, vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[29] F. T. Liu, K. M. Ting, and Z. H. Zhou, “Isolation Forest,” in Proceedings of the 8th IEEE International Conference on Data Mining (ICDM), Pisa, Italy, Dec. 2008, pp. 413–422.

[30] F. T. Liu, K. M. Ting, and Z. H. Zhou, “Isolation-based anomaly detection,” ACM Transactions on Knowledge Discovery from Data (TKDD), vol. 6, no. 1, pp. 1–39, Mar. 2012.

[31] Z. H. Zhou, “Ensemble Methods: Foundations and Algorithms,” CRC Press, Boca Raton, FL, 2012.

[32] R. G. Miller, “Simultaneous Statistical Inference,” Springer-Verlag, New York, NY, 1981.