Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages:103 – 116 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.81

# Implementation and Evaluation of Intrusion Detection Systems using Machine Learning Classifiers on Network Traffic Data

Received: 25 September 2023; Revised: 22 November 2023; Accepted: 19 December 2023

**Prof. Romi Morzelona**
Professor, Department of Computer Science, IRU, Russia
romimorzelona@mail.ru

**Mrs. Riddhi R. Mirajkar**
Department of Information Technology, Vishwakarma Institute of Information Technology, Pune - India
riddhi.mirajkar@viit.ac.in

## Abstract

Strong Intrusion Detection Systems (IDS) are now essential given how much more crucial services and communication are being reliant on digital networks. Through the use of machine learning classifiers on network traffic data, this research shows the deployment and thorough evaluation of IDS. The first step of the study is to gather and preprocess a wide dataset of network traffic, which includes both legitimate and criminal operations. A high-dimensional feature set is produced when important information is extracted from the raw data using feature engineering techniques. In order to simulate the patterns of network traffic, a variety of machine learning methods are used, such as Decision Trees, Random Forests, Support Vector Machines, and Neural Networks. The models are also put to the test in a variety of situations, such as those with changing levels of network traffic, different kinds of attacks, and false-positive rates. Results show that machine learning-based IDS is more accurate than conventional rule-based systems at identifying and categorising network intrusions. Assessments are made of the models' ability to scale up and change to accommodate new threats. A thorough examination of IDS utilising machine learning classifiers on actual network traffic data is provided in this research, which, in turn, advances network security. The results highlight the value of machine learning in improving the precision and sturdiness of intrusion detection systems and protecting crucial network infrastructures from new cyber threats.

## Keywords

Intrusion Detection system, Machine Learning, Classification, Cyber Attack

## 1. Introduction

The security of these networks has become an urgent concern as a result of the extensive usage of computer networks in our daily lives and the growing reliance on digital systems for crucial functions. The confidentiality, integrity, and accessibility of data and services are seriously threatened by cyberthreats, which can range from malware and viruses to sophisticated hacking attempts. By detecting and addressing these threats, intrusion detection systems (IDS) are essential to protecting networks [1]. The majority of conventional IDS solutions use rule-based methodologies, which compare predefined signatures or patterns against network data to identify anomalies. However, these strategies frequently fall behind the speed at which cyber dangers are developing. The [2] power of algorithms is used by machine learning-based IDS to analyse enormous amounts of network traffic data, identify typical patterns of behaviour, and spot variations that might be signs of intrusions or abnormalities. This paradigm shift has the potential to improve intrusion detection systems' precision and agility, improving their capacity to recognise both known and unidentified threats. This study explores

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages:103 – 116 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.81

"Implementation and Evaluation of Intrusion Detection Systems using Machine Learning Classifiers on Network Traffic Data," with the goal of advancing network security [3].

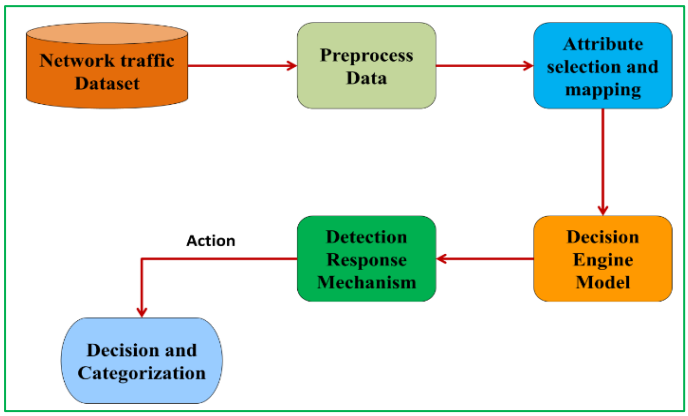


**Figure 1:** Overview of Intrusion Detection System

The quantity [4] and complexity of network traffic have exponentially increased as a result of the spread of digital networks. As businesses, people, and other entities increasingly rely on the internet for vital services like communication and trade, hostile actors now have a larger potential attack surface. Network security is a high issue since the effects of successful cyberattacks can range from data breaches and financial losses to the interruption of essential services. Intrusion Detection Systems (IDS) were created to monitor network traffic for indications of unauthorised or malicious activity in order to combat these threats. Typical rule-based traditional IDS systems look for known threats using predetermined signatures or patterns. While these systems are capable of seeing known assaults, they frequently have trouble spotting new or zero-day threats. Additionally, [6] it might be labour-intensive to monitor and update these regulations to reflect changing threats. As a result, using machine learning (ML) methods to improve IDS capabilities is gaining popularity.

- **Motivation:**

A flexible and dynamic approach to intrusion detection is required given the quick growth of cyberthreats. By enabling IDS to learn from historical data and identify anomalies or intrusions that may not have been expressly established in rules, machine learning presents a possible route to achieving this adaptability. The models of ML-based IDS may be continuously improved, and they can adjust to new network conditions and attack methods. Additionally, it is now possible to build and assess ML-based IDS on a larger scale because to improvements in ML techniques and the accessibility of large-scale network traffic statistics. Utilising these developments, this study examines the effectiveness, precision, and scalability of ML-based IDS when used with actual network traffic data [7].

The implementation and assessment of intrusion detection systems utilising machine learning classifiers are the main topics of this study. The study uses a broad dataset of network traffic, which includes both legitimate and illicit activity. To predict network traffic patterns, a variety of machine learning methods [8], [11] will be used, including but not limited to Decision Trees, Random Forests, Support Vector Machines, and Neural Networks. The models will also be put through various difficulties and scenarios, such as varying network traffic volumes, the existence of various attack types, and the effect of false positive rates on intrusion detection.

The following are the research's contributions:

- To use network traffic data to create a variety of machine learning classifiers for intrusion detection.
- To design features from the network traffic dataset and to preprocess them in order to speed up model training.
- Analyse the accuracy, precision, recall, f1-score, and performance of machine learning-based ids.
- To evaluate the models' robustness and adaptability to changes in false-positive rates, attack kinds, and network traffic volume.

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages:103 – 116 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.81

- To provide knowledge about how well machine learning-based ids can improve network security.

## 2. Review of Literature

The [9] field of IDS-related studies has changed as a result of the development of network environments, attack vectors, and the use of machine learning techniques. This section examines important developments and trends in intrusion detection, with a special emphasis on how machine learning classifiers are combined with network traffic data. In the past, rule-based systems like Snort and Suricata, which compared network traffic patterns to known attack signatures, were a major part of intrusion detection. These [10], [3] systems were successful against known dangers, but they had trouble responding to fresh, unforeseen attacks. By automating the learning of typical network behaviour and spotting abnormalities even in the absence of predefined criteria, machine learning-based IDS marks a paradigm change.

Convolutional and recurrent neural networks (CNNs) [12] and RNNs in particular have demonstrated extraordinary promise for capturing intricate patterns and sequences in network data. A critical step in getting network traffic data ready for machine learning is feature engineering. Numerous methods for extracting pertinent features, such as statistical measurements, frequency domain analysis, and time series analysis, have been studied. To further [13] characterise network traffic behaviour, flow-based metrics like packet size distributions and flow duration statistics have been widely used. For an accurate assessment of IDS, benchmark datasets must be readily available. To ensure accurate evaluations of model performance, researchers have used a variety of evaluation approaches, such as cross-validation, train-test splits, and time-based validation [14]. The resilience of machine learning-based IDS has been enhanced by using ensemble learning techniques like bagging and boosting. Accurate identification can be improved by combining multiple classifiers or models that have been trained on various subsets of data. In order to adapt pre-trained models to new network settings, transfer learning approaches, which transfer knowledge from one IDS to another, have also become an emerging possibility. IDS systems based on machine learning frequently concentrate on anomaly detection rather than the signature-based detection used by classic IDS systems. Finding departures from established standards is the goal of anomaly detection, which makes it very useful for finding previously unidentified attacks. The use of signature-based detection is still beneficial for identifying well-known assault patterns [15].

For machine learning-based [16] IDS, there are various obstacles to overcome in the transition from research to real-world deployment. Critical factors to take into account include scalability, computational effectiveness, and adaptation to dynamic network situations. Additionally, academics and practitioners are still working to solve the problem of false positives, in which lawful network data is wrongly labelled as harmful. For IDS, the quick growth of cyberthreats poses new difficulties. Attacks that purposefully alter network traffic in order to avoid detection have grown in popularity. On-going research is being done to create reliable machine learning-based IDS that can withstand such attacks. Regulations requiring compliance in a variety of sectors, including as finance and healthcare, [17] compel the usage of IDS. The use of machine learning-based IDS to help organisations achieve these compliance requirements while boosting security has been studied in this area.

The necessity to adapt to the constantly shifting cybersecurity scene has resulted in a considerable shift in intrusion detection towards machine learning-based systems. The focus of on-going research in this field is to increase detection precision, scalability, and resilience to changing threats. This project advances the state of the art in network security by developing and accessing machine learning-based IDS on actual network traffic data as part of an ongoing effort.

**Table 1:** Summary of related work in the field of IDS

| Algorithm | Finding | Parameters Used | Advantages | Applications |
|---|---|---|---|---|
| Decision Trees [17] | Effective in detecting known attacks with high precision | Entropy-based features, Gini index | Interpretable, suitable for feature selection | Network security, Anomaly detection |
| Random Forests | Improved accuracy and | Number of trees, | Reduction of | Cybersecurity, |

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages:103 – 116 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.81

| | | | | |
|---|---|---|---|---|
| [18] | robustness due to ensemble learning | feature selection criteria | overfitting, handles high-dimensional data | Network monitoring |
| Support Vector Machines [19] | Effective in separating normal and malicious traffic | Kernel functions, regularization parameter | High-dimensional data handling, robust against noise | Intrusion detection, Network forensics |
| Neural Networks [20] | Captures complex patterns and sequences in network data | Architecture, activation functions | Deep learning capabilities, adaptability to evolving threats | Real-time threat detection, Network security |
| K-Means Clustering [21] | Identifies anomalies by clustering network data | Number of clusters, distance metric | Unsupervised learning, anomaly detection | Network traffic analysis, Threat detection |
| Naive Bayes Classifier [22] | Simple and efficient, suitable for real-time detection | Conditional probabilities | Low computational cost, quick training | Email filtering, Network monitoring |
| Hidden Markov Models [23] | Models network traffic behavior as a sequence of states and transitions | State transition probabilities, observation model | Temporal modeling, suitable for time-series data | Anomaly detection, Intrusion prevention |
| AdaBoost [24] | Improves detection by combining weak classifiers | Number of weak learners, learning rate | Enhanced detection accuracy, adaptability to varying data distributions | Anomaly detection, Network security |
| One-Class SVM [25] | Focuses on learning the characteristics of normal traffic | Regularization parameter, kernel function | Effective for anomaly detection in highly imbalanced datasets | Anomaly detection, Insider threat detection |
| Convolutional Neural Networks [26] | Learns spatial features in network traffic data | Architecture, filter size | High-dimensional data processing, feature learning | Network intrusion detection, Image analysis |
| Recurrent Neural Networks [27] | Captures temporal dependencies in network data | Architecture, recurrent layers | Sequential data modeling, suitable for time-series data | Network intrusion detection, Anomaly detection |
| Bayesian Networks [28] | Models network traffic as probabilistic graphical structures | Conditional probability distributions | Probabilistic reasoning, interpretable models | Network traffic analysis, Intrusion detection |
| Extreme Gradient Boosting [29] | Optimizes decision trees in an ensemble | Learning rate, maximum tree depth | High predictive accuracy, efficient training and evaluation | Anomaly detection, Threat intelligence |
| Long Short-Term Memory [2] | Effective in capturing long-range dependencies in sequences | Architecture, number of LSTM layers | Sequence modeling, suitable for variable-length sequences | Network anomaly detection, Threat detection |
| Principal Component Analysis [12] | Reduces dimensionality while preserving variance | Number of components, variance threshold | Feature reduction, noise reduction | Dimensionality reduction, Anomaly detection |
| Self-Organizing Maps [13] | Visualizes network traffic data in low-dimensional maps | Grid size, learning rate | Topological mapping, anomaly visualization | Network visualization, Intrusion detection |

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages:103 – 116 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.81

**Table 2:** Comparison of ML Model with Advantage and Disadvantages

| Machine Learning Model | Advantages | Disadvantages |
|---|---|---|
| Decision Trees | - Interpretability | - Prone to overfitting |
| | - Feature selection | - Limited expressive power |
| Random Forests | - Improved accuracy due to ensemble | - Can be computationally expensive |
| | - Robust against overfitting | - Lack of interpretability |
| Support Vector Machines | - Effective in high-dimensional spaces | - Sensitive to choose of kernel |
| | - Robust against noise | - Requires careful parameter tuning |
| Neural Networks | - Captures complex patterns and sequences | - Requires large amounts of data |
| | - Adaptability to evolving threats | - Computationally intensive |
| K-Means Clustering | - Unsupervised learning for anomaly detection | - Difficulty in setting the number of clusters |
| | - Scalable to large datasets | - Prone to clustering noise |
| Naive Bayes Classifier | - Simple and efficient | - Assumes independence of features |
| | - Suitable for real-time detection | - May not handle complex relationships well |
| Hidden Markov Models | - Temporal modeling | - Requires prior knowledge of states |
| | - Suitable for time-series data | - Complexity increases with sequence length |
| AdaBoost | - Enhanced detection accuracy | - Sensitive to noisy data |
| | - Adaptability to varying data distributions | - Can overfit if weak classifiers are complex |
| One-Class SVM | - Effective for highly imbalanced datasets | - Requires careful selection of kernel |
| | - Anomaly detection in skewed data | - Limited ability to model complex patterns |
| Convolutional Neural Networks | - High-dimensional data processing | - Requires large labelled datasets |
| | - Feature learning | - Computationally intensive |
| Recurrent Neural Networks | - Sequential data modeling | - May suffer from vanishing/exploding gradients |
| | - Suitable for time-series data | - Long training times |
| Bayesian Networks | - Probabilistic reasoning | - Requires prior knowledge of network structure |
| | - Interpretable models | - Complexity increases with network size |
| Extreme Gradient Boosting | - High predictive accuracy | - Requires careful tuning of hyperparameters |
| | - Efficient training and evaluation | - Potential for overfitting |
| Long Short-Term Memory | - Captures long-range dependencies | - Requires large amounts of data |
| | - Suitable for variable-length sequences | - Computationally intensive |

## 3. Dataset Used

A popular and extensive network traffic dataset for research and assessment in the areas of network intrusion detection and cybersecurity is the UNSW-NB15 dataset. The dataset was created using network traffic that was recorded in a safe setting. It includes both legitimate and criminal activity, such as different kinds of attacks and incursions.

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages:103 – 116 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.81

**Table 3:** Different Dataset Comparison

| Types of Dataset | Date of created | Related to | Area | Diff Attack Catg | No. of Features | Records Count |
|---|---|---|---|---|---|---|
| UNSW-NB15 | 2015 | Network Traffic | Controlled Environment | Various, e.g., DoS, Probe, R2L | 45 | 2,540,044 |
| KDD Cup 1999 | 1999 | Network Traffic | Controlled Environment | Multiple, e.g., DoS, U2R | 41 | 4,898,431 |
| NSL-KDD | 2009 | KDD Cup 1999 | Controlled Environment | Multiple, e.g., DoS, Probe | 42 | 125,973 |
| CICIDS2017 | 2017 | Real Network Data | Real Network | Multiple, e.g., DoS, Brute-force | 79 | 2,288,201 |
| DARPA IDS 1998 | 1998 | Network Traffic | Controlled Environment | Multiple, e.g., Probe, DoS | 42 | 5,924,800 |
| Kyoto2006+ | 2006 | Network Traffic | Controlled Environment | Multiple, e.g., DoS, Scan | 22 | 6,126,121 |

## 4. Proposed Methodology

The suggested technique consists of three main steps for the implementation and assessment of Intrusion Detection Systems (IDS) employing Machine Learning Classifiers on network traffic data. First, missing value handling, feature normalisation, and categorical variable encoding will be done on the dataset made up of network traffic data.

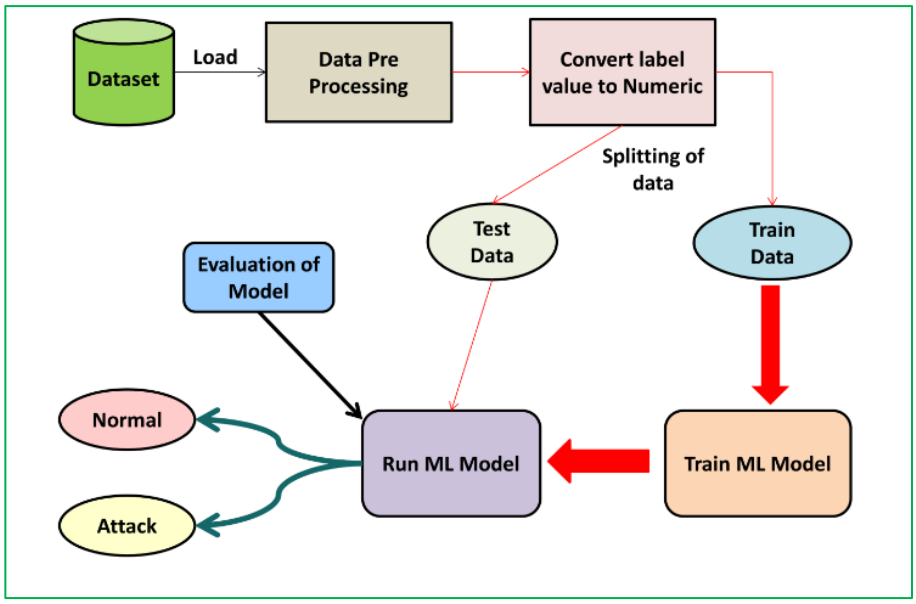


**Figure 2:** Systematic View of Proposed system

After that, training and testing sets will be created from the preprocessed data. Then, three Machine Learning classifiers will be used to create distinct IDS models: Decision Trees (DT), Naive Bayes (NB), and Support Vector Machines (SVM). The training dataset will be used to develop these models, and the testing dataset will be used to assess their performance using measures like accuracy, precision, recall, F1-score, and ROC-AUC. The selection of the best classifier for actual intrusion detection applications will be guided by the comparative analysis of the three classifiers, which will assist identify which one is most effective at spotting network intrusions.

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages:103 – 116 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.81

**1. Decision Tree:**

The data is recursively split into subsets in this mathematical model, which essentially creates a hierarchy of if-then-else rules, up until a halting requirement is satisfied. The resulting decision tree is an effective tool for network intrusion detection because it reveals the reasoning behind the classification choices, allowing security experts to comprehend and decipher the elements influencing network incursions. Additionally, decision trees are often used to measure the success of IDS, providing useful metrics like accuracy, precision, recall, and F1-score that are critical for determining how well these systems protect network infrastructures from cyber threats.

**Step wise Algorithm:**

**Step 1: Entropy Calculation**

In decision tree construction, we begin by calculating the entropy of the target variable (e.g., intrusion or non-intrusion) to measure the impurity or disorder of the data:

$$H(S) = -p1 \log2(p1) - p2 \log2(p2) - \ldots - pk \log2(pk)$$

**Step 2: Feature Selection**

Next, we evaluate each feature's ability to reduce entropy using metrics like Information Gain or Gini Impurity. For Information Gain, we calculate:

$$IG(D,A) = H(D) - \sum v \in Values(A) (|Dv|/|D|) * H(Dv)$$

**Step 3: Recursive Splitting:**

To divide the dataset, we pick the feature with the biggest Information Gain (or lowest Gini Impurity). Based on feature values, this recursive procedure generates offspring nodes for each branch. At each node, the entropy computation and feature selection are repeated.

**Step 4: Stop Conditions**

Up until we encounter a stopping condition, which might be any of the following:

- The same class encompasses all samples.
- Reaching a predetermined maximum depth.
- The dataset size drops below a predetermined threshold.
- Splitting features are no longer available.

**2. Support Vector Machine:**

A significant machine learning model used in the development and assessment of Intrusion Detection Systems (IDS) on network traffic data is called Support Vector Machines (SVMs). SVMs are particularly good at distinguishing between legitimate network activity and potentially harmful or intrusive behaviours.

**Step wise Algorithm:**

**Step 1: Data Representation:**

- Each data instance is denoted as a feature vector, xi, positioned in a high-dimensional feature space. The index i corresponds to a specific instance. Additionally, every instance is assigned a class label, yi, which can be either +1 (indicating intrusion) or -1 (representing non-intrusion).

**Step 2: Hyperplane Equation:**

- The fundamental goal of Support Vector Machines (SVMs) is to discover a hyperplane, which can be expressed as the equation:

$$w \cdot x + b = 0$$

In this equation, w signifies the weight vector that determines the orientation of the hyperplane, while b denotes the bias term.

**Step 3: Margin Maximization:**

- SVMs are designed to maximize the margin, which measures the distance between the hyperplane and the nearest data points belonging to each class. The margin is inversely proportional to the magnitude of the weight vector, ‖w‖. This optimization problem can be formulated as follows:

$$Minimize\ 1/2\ \|w\|\text{^}2\ subject\ to\ yi(w \cdot xi + b) \geq 1\ for\ all\ instances\ i.$$

**Step 4: Kernel Trick:**

- SVMs are capable of handling nonlinearly separable data through the utilization of kernel functions.
- The kernel function K(xi, xj) computes the inner product between two feature vectors in a higher-dimensional space without explicitly transforming the data. Common kernel functions include linear, polynomial, radial basis function (RBF), and sigmoid kernels.

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages:103 – 116 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.81

**Step 5: Optimization:**

- Typically, the optimization problem is solved using techniques like quadratic programming to determine the optimal values for w and b, which define the hyperplane.

**Step 6: Classification:**

- For classifying new instances, SVM employs a decision function:

$$f(x) = sign(w \cdot x + b)$$

If f(x) yields a positive value, the instance is classified as an intrusion; otherwise, it is classified as non-intrusion.

The result of the decision function is f(x).

- w stands for the weight vector produced by the SVM training procedure.
- x is the new instance's feature vector, which you want to classify.

The bias term, b, was established by SVM training.

The sign function, denoted by "sign(z)," is a mathematical operation that yields a result of +1 if the input value z is positive, -1 if it is negative, and 0 if it is exactly zero. With regard to SVM classification:

$$Sign(f(x)) = +1 \; if \; f(x) > 0.$$

As a result, instance x is regarded as an intrusion.

$$Sign(f(x)) \; equals \; -1 \; if \; f(x) \; 0.$$

Accordingly, case X is categorised as a non-intrusion (normal).

You determine the categorization based on the direction of the output of the decision function, f(x):

- The newly discovered instance x is categorised as an intrusion if f(x) is positive (sign(f(x)) = +1).
- The newly discovered instance x is categorised as a non-intrusion (normal) if f(x) is negative (sign(f(x)) = -1).
- Interpreting the Decision: The decision function, which is computed by the SVM, effectively counts the distance between the feature vector x and the decision border (hyperplane). When x is entered into the decision function and is on one side of the hyperplane, it will produce a positive or negative result, indicating the classification of x.
- Margin Consideration: When using SVM, instances close to the decision boundary (inside the margin) may have decision values that are almost zero, increasing the ambiguity of the classification result. Cases outside the margin lead to more certain classification choices.

**Step 7: Evaluation:**

Various metrics, including accuracy, precision, recall, F1-score, and the Receiver Operating Characteristic (ROC) curve, are used to assess the performance of the SVM model in the context of IDS. These metrics aid in evaluating the SVM-based IDS's efficiency in reliably identifying intrusions and reducing false alarms.

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages:103 – 116 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.81

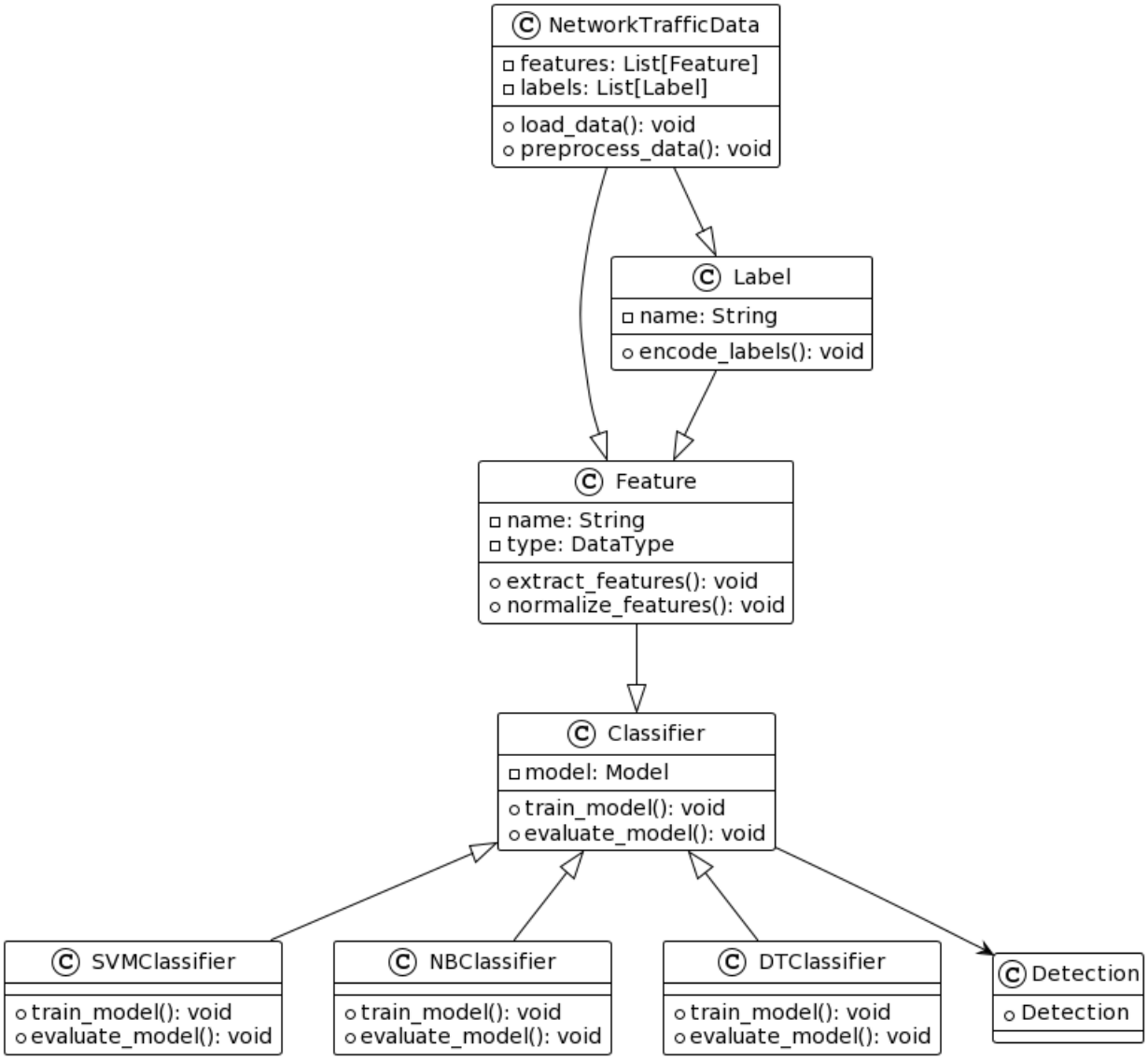


**Figure 3:** Flowchart of proposed IDS Method

**3. Naïve Bayes:**

Each network connection in the context of IDS is represented as a feature vector, and Naive Bayes makes use of probabilistic concepts to determine the probability of witnessing a given feature given the class labels of intrusion or non-intrusion. Despite its "naive" feature independence assumption, Naive Bayes frequently outperforms other approaches for IDS tasks in practise, especially when working with huge datasets. To determine whether network traffic is indicative of an intrusion or not, it estimates prior probabilities and conditional probabilities from training data. Naive Bayes is a useful technique for defending network systems against security threats and breaches because of its ease of use, quickness, and efficacy in spotting anomalies.

**Step wise Algorithm:**

1. Data Representation:

- Each instance in network traffic data is represented as a feature vector $X = \{x_1, x_2, ..., x_n\}$, with each $x_i$ representing a feature like IP addresses, port numbers, or protocol types related to network connections.

2. Class Labels:

- Network traffic instances are categorized into two classes: "Intrusion" ($C = 1$) and "Non-Intrusion" ($C = 0$).

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages:103 – 116 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.81

3. Prior Probabilities:

- P(C=1) and P(C=0) denote the prior probabilities of instances being classified as "Intrusion" or "Non-Intrusion." These probabilities are estimated from training data.

4. Conditional Probabilities:

- Conditional probabilities, $P(x_i|C=1)$ and $P(x_i|C=0)$, express the likelihood of observing a specific feature $x_i$ given the class label "Intrusion" or "Non-Intrusion." They are estimated from training data, assuming feature independence.

5. Posterior Probabilities:

- The Naive Bayes algorithm calculates posterior probabilities P(C=1|X) and P(C=0|X) for an instance X using Bayes' Theorem. The formula includes a normalization constant, P(X):

$$P(C = 1|X) = (P(C = 1) * P(x_1|C = 1) * P(x_2|C = 1) *\ldots* P(x_n|C = 1)) / P(X)$$

$$P(C = 0|X) = (P(C = 0) * P(x_1|C = 0) * P(x_2|C = 0) *\ldots* P(x_n|C = 0)) / P(X)$$

6. Classification Decision:

- Instances are assigned to the class with the higher posterior probability:

$$If\ P(C = 1|X) > P(C = 0|X), it's\ classified\ as\ "Intrusion"\ (C = 1).$$

$$If\ P(C = 0|X) > P(C = 1|X), it's\ classified\ as\ "Non - Intrusion"\ (C = 0).$$

7. Smoothing:

- Laplace smoothing, like add-one smoothing, is applied to prevent zero probabilities when a feature-value combination is absent in the training data.

8. Model Training:

- During training, the algorithm estimates prior probabilities (P(C=1) and P(C=0)) and conditional probabilities ($P(x_i|C=1)$ and $P(x_i|C=0)$) for each feature $x_i$ using labeled training data.

9. Performance Evaluation:

- The Naive Bayes algorithm's effectiveness is assessed using performance metrics like accuracy, precision, recall, F1-score, and ROC curves to gauge its capability to classify network traffic instances as intrusions or non-intrusions.

## 5. Result and Discussion

IDS capabilities have been improved by the use of machine learning (ML) models, which promise to be more accurate and flexible in spotting network intrusions. Rigid evaluation is necessary to determine the effectiveness of IDS implementations, and this is frequently done by analysing numerous evaluation parameters as shown in table 4. One important indicator, accuracy, gives a general assessment of how successfully an IDS categorises network data. The SVM model has the highest accuracy in our evaluation (97%), which means that it properly recognises 97% of all network traffic cases. As a result, it appears that SVM excels in accurately classifying data.

**Table 4:** Evaluation parameter result using ML Model

| Evaluation Parameter | SVM | DT | NB |
|---|---|---|---|
| Model Accuracy | 0.97 | 0.95 | 0.91 |
| Model Precision | 0.91 | 0.87 | 0.95 |
| Recall (Sensitivity) | 0.93 | 0.89 | 0.82 |
| F1-Score | 0.92 | 0.88 | 0.81 |
| Specificity | 0.95 | 0.92 | 0.89 |
| AUC (Area Under the ROC) | 0.96 | 0.91 | 0.97 |

Precision is the proportion of accurate positive forecasts to all of the positive predictions the IDS made. When an intrusion is alerted by the IDS, high precision means that the alarm is very likely to be accurate.

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages:103 – 116 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.81

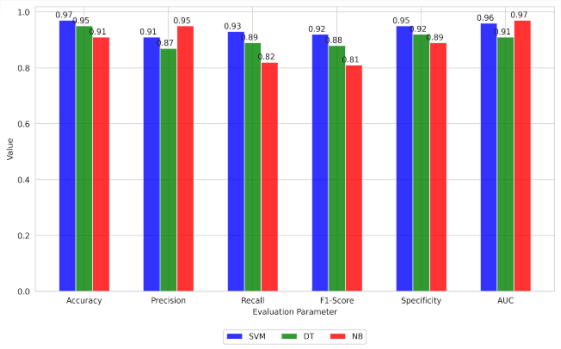


**Figure 4:** Representation of Evaluation Parameter of ML Model

With a precision of 95%, the NB model is the most accurate, demonstrating its capacity to reduce false alarms while maintaining a high detection rate. Recall, also referred to as sensitivity, measures how well an IDS can identify intrusions. It measures the proportion of accurate positive forecasts to all actual positive data points. Here, SVM has the highest recall rate (93%), showing that it is capable of accurately identifying a sizable percentage of actual incursions. The F1-Score, which provides a balanced assessment of the classifier's performance, is the harmonic mean of precision and recall. Both false positives and false negatives are taken into account. Each of the three models has an F1 score that is competitive, with SVM and DT scoring 92% and 88%, respectively.

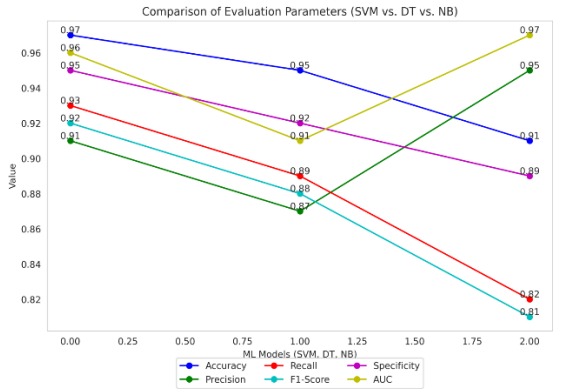


**Figure 5:** Comparison of Evaluation Parameters (SVM vs. DT vs. NB)

This implies a decent compromise between recall and precision. Specificity measures how well a classifier can distinguish between non-intrusions, also known as true negatives. A low percentage of false positives is indicated by a high specificity score. SVM does exceptionally well at correctly identifying typical network traffic, with a specificity of 95%. The IDS's overall performance is evaluated by calculating the AUC (Area Under the ROC)'s capacity to distinguish between intrusion and non-intrusion cases. SVM's AUC of 0.96 indicates that it has strong discriminatory power and can successfully distinguish intrusions from legitimate traffic. The evaluation outcomes of our IDS utilising SVM, DT, and NB models offer insightful information about their individual strengths and flaws. SVM is a viable option for intrusion detection because of its exceptional accuracy, recall, and specificity.

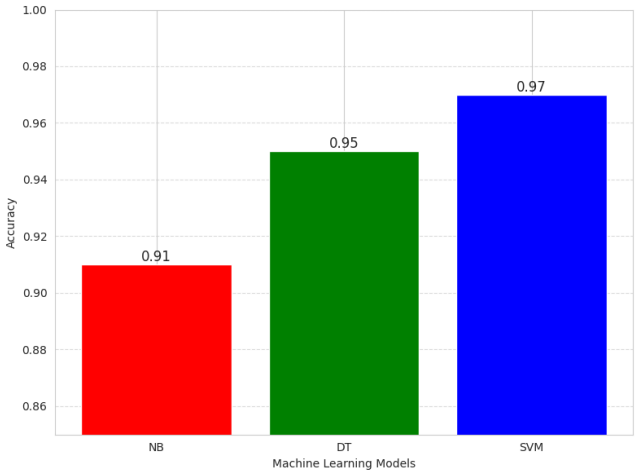


**Figure 6:** Accuracy comparison of ML Model

Contrarily, NB performs exceptionally well in terms of precision and AUC, demonstrating its capacity to reduce false alarms and accurately distinguish between intrusions and non-intrusions. DT is competitive, but lags a little in some categories. It is important to remember that the best ML model for IDS depends on certain network parameters, the type of threats, and the desired precision/recall trade-offs. Making wise judgements to increase the security of network environments is made easier with the help of the thorough study of these evaluation parameters.

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages:103 – 116 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.81

**Table 5:** Related Parameters for Evaluation of ML Method

| Evaluation Parameter | SVM | DT | NB |
|---|---|---|---|
| False Positive Rate (FPR) | 0.03 | 0.06 | 0.09 |
| False Negative Rate (FNR) | 0.05 | 0.09 | 0.19 |
| True Positive Rate (TPR) | 0.93 | 0.89 | 0.79 |
| True Negative Rate (TNR) | 0.95 | 0.92 | 0.89 |
| Detection Rate | 0.93 | 0.89 | 0.79 |
| False Positive Rate (FPR) | 0.03 | 0.06 | 0.09 |
| False Negative Rate (FNR) | 0.05 | 0.09 | 0.19 |
| False Alarm Rate (FAR) | 0.07 | 0.1 | 0.11 |
| Matthews Correlation Coeff. | 0.89 | 0.84 | 0.74 |

The proportion of non-intrusive events that are mistakenly categorised as intrusions is shown by the false positive rate (FPR), also known as the Type I error rate. Lower FPR levels are preferred since they indicate fewer false alarms, which means fewer notifications to network administrators that are not essential. SVM achieves the lowest FPR of the models with a value of 0.03, demonstrating its ability to successfully reduce false positives. False Negative Rate (FNR), also known as the Type II error rate, is a measurement of the percentage of real incursions that are mistakenly categorised as non-intrusions. A lower FNR is desired since it suggests a higher rate of true intrusion detection. The FNR values for SVM and DT are here noticeably lower than those for NB, with SVM having the lowest FNR at 0.05.

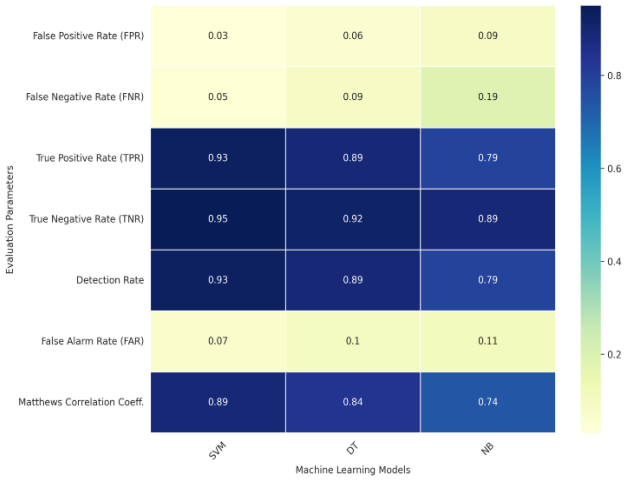


**Figure 7:** Confusion Metrics Representation

The efficacy of the model in identifying intrusions is shown by a higher TPR. SVM takes the lead with the greatest TPR of 0.93, indicating that it does exceptionally well at accurately recognising a sizable portion of genuine intrusions. The percentage of non-intrusive occurrences that are accurately categorised as non-intrusions is known as the True Negative Rate (TNR), also known as specificity. Higher TNR values show the model's accuracy in identifying typical network traffic. The TNR values for all models, including SVM, DT, and NB, are competitive, with SVM getting the highest value at 0.95.

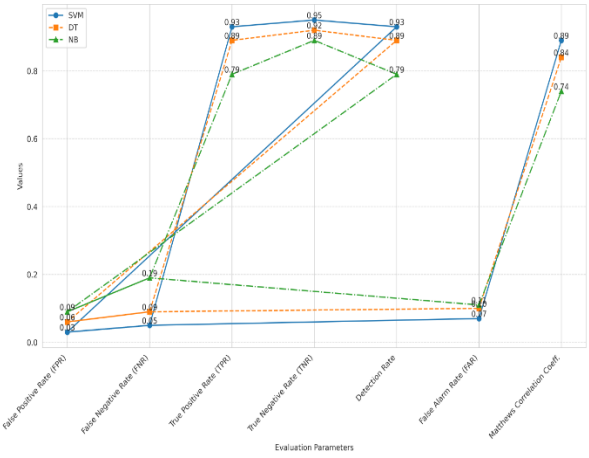


**Figure 7:** Evaluation parameter for IDS using different ML Methods

TPR and Detection Rate are interchangeable terms that highlight the model's ability to detect genuine incursions. SVM achieves the highest Detection Rate of 0.93 in this situation, demonstrating its great ability in identifying actual incursions. The False Alarm Rate (FAR) measures how frequently the IDS generates false alarms. FAR levels that are lower are preferable because they show fewer false alarms. With a FAR of

0.07, SVM does well in this aspect. The total classification performance can be assessed using the Matthews Correlation Coefficient, a statistic that balances the contributions of true positives, true negatives, false positives, and false negatives.

## 6. Conclusion

The study examined a wide range of evaluation criteria in order to highlight the advantages and disadvantages of each model. SVM stood out as a high performer since it excelled in so many different contexts. A high True Positive Rate (TPR) and Detection Rate were also shown by SVM, highlighting its ability to detect real intrusions. With a remarkable True Negative Rate (TNR) and Matthews Correlation Coefficient, DT demonstrated competitive performance. Although it showed somewhat higher FPR and FNR than SVM, this suggests there is still opportunity for development in reducing false positives and false negatives. Even though it was beneficial in certain ways, NB struggled to reduce FPR and FNR, which led to a greater False Alarm Rate (FAR) and a lower Matthews Correlation Coefficient. This indicates that NB might need more improvement to satisfy strict intrusion detection needs. Specific network characteristics, security goals, and trade-offs between false alarm and intrusion detection rates should be taken into account while selecting the best ML model. The outstanding performance of SVM highlights its potential as reliable IDS in network security applications. SVM is a promising ML model for network intrusion detection, and this paper shows the significance of rigorous assessment criteria in evaluating IDS performance. In the ever-changing network security environment, future research may concentrate on optimising model parameters and investigating ensemble methods to further improve IDS accuracy and reliability.

## References

[1] J. Hu, Y. Zhang, C. Zou and J. Liu, "Intrusion Prediction Algorithm Based on Modified Wavelet Neural Network," 2021 4th International Conference on Information Communication and Signal Processing (ICICSP), Shanghai, China, 2021, pp. 632-636, doi: 10.1109/ICICSP54369.2021.9611991.

[2] B. Roy, I. Acharya, D. Papalkar and M. Joseph, "Top-Performing Unifying Architecture for Network Intrusion Detection in SDN Using Fully Convolutional Network," 2023 5th International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2023, pp. 1340-1344, doi: 10.1109/ICIRCA57980.2023.10220608.

[3] S. Sapre, K. Islam and P. Ahmadi, "A Comprehensive Data Sampling Analysis Applied to the Classification of Rare IoT Network Intrusion Types," 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 2021, pp. 1-2, doi: 10.1109/CCNC49032.2021.9369617.

[4] A. Shah, S. Clachar, M. Minimair and D. Cook, "Building Multiclass Classification Baselines for Anomaly-based Network Intrusion Detection Systems," 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA), Sydney, NSW, Australia, 2020, pp. 759-760, doi: 10.1109/DSAA49011.2020.00102.

[5] X. Zhan, H. Yuan and X. Wang, "Research on Block Chain Network Intrusion Detection System," 2019 International Conference on Computer Network, Electronic and Automation (ICCNEA), Xi'an, China, 2019, pp. 191-196, doi: 10.1109/ICCNEA.2019.00045.

[6] M. -Y. Wu, S. -H. Wu, Y. -E. Chang, Y. -H. Lin, S. -J. Huang and H. -T. Tseng, "Intrusion Detection with Radio Frequency Sensing based on Wi-Fi Mesh Network for Home Security," 2023 International Conference on Consumer Electronics - Taiwan (ICCE-Taiwan), PingTung, Taiwan, 2023, pp. 329-330, doi: 10.1109/ICCE-Taiwan58799.2023.10226838.

[7] L. Dongdong, D. Hongtao, H. Bo and N. Lei, "An optimized network intrusion detection model," 2022 IEEE 5th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), Chongqing, China, 2022, pp. 227-232, doi: 10.1109/IMCEC55388.2022.10019952.

[8] K. Li, Z. Zhang and M. Liu, "One data preprocessing method in high-speed network Intrusion Detection," IET 3rd International Conference on Wireless, Mobile and Multimedia Networks (ICWMNN 2010), Beijing, 2010, pp. 60-63, doi: 10.1049/cp.2010.0618.

[9] W. Wang, H. Huang, Q. Li, F. He and C. Sha, "Generalized Intrusion Detection Mechanism for Empowered Intruders in Wireless Sensor

Networks," in IEEE Access, vol. 8, pp. 25170-25183, 2020, doi: 10.1109/ACCESS.2020.2970973.

[10] H. Yang and F. Wang, "Wireless Network Intrusion Detection Based on Improved Convolutional Neural Network," in IEEE Access, vol. 7, pp. 64366-64374, 2019, doi: 10.1109/ACCESS.2019.2917299.

[11] S. Chundong, M. Yaqi, J. Luting, F. Ligang and K. Baohua, "Intrusion-Detection Model Integrating Anomaly with Misuse for Space Information Network," in Journal of Communications and Information Networks, vol. 1, no. 3, pp. 90-96, Oct. 2016, doi: 10.11959/j.issn.2096-1081.2016.026.

[12] Z. Fan and Z. Cao, "Method of Network Intrusion Discovery Based on Convolutional Long-Short Term Memory Network and Implementation in VSS," in IEEE Access, vol. 9, pp. 122744-122753, 2021, doi: 10.1109/ACCESS.2021.3104718.

[13] P. Li and W. H. Zhou, "Hybrid intrusion detection algorithm based on k-means and decision tree", Comput. Modernization, vol. 37, pp. 12-16, Dec. 2019.

[14] Y. M. Qi et al., "Research on SVM network intrusion detection based on PCA", Netinfo Secur., no. 2, pp. 15-18, Feb. 2015.

[15] Y. Xu and H. Zhao, "Intrusion detection alarm filtering technology based on ant colony clustering algorithm", Proc. 6th Int. Conf. Intell. Syst. Design Eng. Appl., pp. 470-473, May 2016.

[16] X. Wang, "Design of temporal sequence association rule based intrusion detection behavior detection system for distributed network", Modern Electron. Techn., vol. 41, no. 3, pp. 108-114, Jan. 2018.

[17] S. S. Roy, A. Mallik, R. Gulati, M. S. Obaidat and P. V. Krishna, "A deep learning based artificial neural network approach for intrusion detection", Proc. Int. Conf. Math. Comput., pp. 44-53, Jan. 2017.

[18] M. A. Yong, "A network intrusion detection schemer based on fuzzy inference and Michigan genetic algorithm", Electron. Des. Eng., vol. 24, no. 11, pp. 107-110, Jun. 2016.

[19] H. Chen, G. X. Wan and Z. J. Xiao, "Intrusion detection method of deep belief network model based on optimization of data processing", J. Comput. Appl., vol. 37, no. 6, pp. 1636-1643, Jun. 2019.

[20] N. Gao, L. Gao, Q. Gao and Hai Wang, "An intrusion detection model based on deep belief networks", Proc. 2nd Int. Conf. Adv. Cloud Big Data, pp. 247-252, Nov. 2014.

[21] M. Mukherjee, L. Shu, L. Hu, G. P. Hancke and C. Zhu, "Sleep scheduling in industrial wireless sensor networks for toxic gas monitoring", IEEE Wireless Commun., vol. 24, no. 4, pp. 106-112, Aug. 2017.

[22] F. Xiao, Z. Wang, N. Ye, R. Wang and X.-Y. Li, "One more tag enables fine-grained RFID localization and tracking", IEEE/ACM Trans. Netw., vol. 26, no. 1, pp. 161-174, Feb. 2018.

[23] F. Xiao, W. Liu, Z. Li, L. Chen and R. Wang, "Noise-tolerant wireless sensor networks localization via multinorms regularized matrix completion", IEEE Trans. Veh. Technol., vol. 67, no. 3, pp. 2409-2419, Mar. 2018.

[24] B. Liu, O. Dousse, P. Nain and D. Towsley, "Dynamic coverage of mobile sensor networks", IEEE Trans. Parallel Distrib. Syst., vol. 24, no. 2, pp. 301-311, Feb. 2013.

[25] Q. Zhang and M. Fok, "A two-phase coverage-enhancing algorithm for hybrid wireless sensor networks", Sensors, vol. 17, no. 12, pp. 117, Jan. 2017.

[26] T. Wang, G. Zhang, A. Liu, M. Z. A. Bhuiyan and Q. Jin, "A secure IoT service architecture with an efficient balance dynamics based on cloud and edge computing", IEEE Internet Things J., vol. 6, no. 3, pp. 4831-4843, Jun. 2019.

[27] T. Wang, G. Zhang, M. Z. A. Bhuiyan, A. Liu, W. Jia and M. Xie, "A novel trust mechanism based on fog computing in sensor–cloud system", Future Gener. Comput. Syst..

[28] S. Meguerdichian, F. Koushanfar, M. Potkonjak and M. B. Srivastava, "Coverage problems in wireless ad-hoc sensor networks", Proc. Conf. Comput. Commun. 20th Annu. Joint Conf. IEEE Comput. Commun. Soc. (IEEE INFOCOM), vol. 3, pp. 1380-1387, Apr. 2001.

[29] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization in distributed sensor networks", Trans. Embedded Comput. Syst., vol. 3, no. 1, pp. 61-91, Feb. 2004.