Journal of Intelligent Systems and Internet of Things
Vol. 14, No. 02, PP. 25-35, 2025
DOI: https://doi.org/10.54216/JISIoT.140203

Intrusion Detection System in Wireless Sensor Networks Using Machine Learning

Zainab S. Idan[1,*], Ahmed Al-Fatlawi[1], Hussein Akeel Hussein Alaasam[2] ,Sajjad H. Hasan[1], Ahmed Ali Talib Al Khazaali[3]

[1]Department of Computer Techniques Engineering, College of Technical Engineering, University of Alkafeel, Najaf, Iraq
[2]College of Basic Education, University of Kufa, Najaf, Iraq
[3]Electrical and Computer Engineering, Altinbas University, Istanbul, Turkey
Emails: zainabsabah@alkafeel.edu.iq; ahmed.fatlawi@alkafeel.edu.iq; hussaina.alaasam@uokufa.edu.iq; sajad.hadi@alkafeel.edu.iq; ahmed.al-khazaali@ogr.altinbas.edu.tr

Abstract

Current industrial control systems are increasingly integrating with corporate Internet technology networks in order to fully utilize the abundant resources available on the Internet. The growing connection between industrial control systems and the internet has made them a desirable choice. Industrial control systems are in need of significant protection due to being a common target for a range of cyber-attacks. The use of the Internet of Things is currently increasing across industries due to its efficiency, and the Internet of Things is facing a security challenge. This document gives an overview of the intrusion detection system and the methods of the intrusion detection system. The purpose of this document is to examine intrusion detection methods and present the best method based on studies. Experimental results show that this system uses a combination of machine learning methods for high performance.

Received: March 07, 2024 Revised: June 12, 2024 Accepted: October 04, 2024

Keywords: Intrusion Detection System; Machine Learning; Wireless Sensor Networks; Internet of Things; Data Mining

1. Introduction

With the rapid development and expansion of intranet use, information security and network security has become one of the most important issues for the effective use of intranet technology. Basically, the information system provides reliability, availability, accuracy, integrity and usability of information. To create a secure system, access control methods, encryption, authentication, etc. are used. But these methods do not have the ability to deal with the problems of attacks, and it is not impossible to bypass and penetrate the current security defences such as firewalls. On the other hand, methods of attacking the network are changing rapidly and new methods are being invented. These methods have three salient features:

- There are no restrictions in terms of place and time.
- They are very secretive.
- They have a lot of complexity.

Computers and internet networks are essential in communication and information sharing in the modern world. In the meantime, criminals have gained access to significant data from specific facilities or individuals and have hacked into computer systems with the purpose of infiltrating, pressuring, or disrupting system operations. Terms like Intruder, Hacker, and Cracker are common in the computer world as they attempt to breach other systems and compromise their security. Hence, it is clear that ensuring information security and effectiveness in computer networks connected to external sources is essential.

Since it is not feasible to create computer systems (both hardware and software) that are completely free of security weaknesses, intrusion detection plays a crucial role in computer systems research. Intrusion Detection Systems (IDS) help system security administrators to identify intrusions and attacks. The main goal of an intrusion detection system is not to stop attacks, but simply to identify and potentially discover attacks and security vulnerabilities in the system or computer networks and inform the system administrator. Intrusion detection systems are frequently employed in conjunction with firewalls to enhance security.

Detection and prevention of intrusion is one of the main mechanisms in meeting the security of networks and computer systems. We do not consider intrusion detection systems as separate systems, but they can be used as subsystems of network equipment, operating systems and even services. To evaluate intrusion detection systems, special data is required, which in this project uses the NSL-KDD set.

The NSL-KDD dataset consists of 47 attributes and 5 classes, of which we have a normal class and 4 types of attack classes: R2L, U2R, DoS and Prob. Denial of Service (DoS) attack involves exploiting system resources by overusing them, leading to the denial of normal resource requests.

R2L(Remote-to-Local) attack: During an R2L attack, a perpetrator gains unauthorized access to the victim's computer remotely and proceeds to exploit the user's legitimate account by transmitting packets through the network. User-to-Root (U2R) attack: This attack is carried out on the victim machine with the goal of gaining root access. Probing attack: Computers are scanned in order to collect information or discover already identified weaknesses. To ensure network security, you must first understand the concept of network attack. A network attack can compromise network stability or compromise the security of information stored on computers connected to that network. The purpose of intrusion detection is to establish a system that automatically monitors network activity and detects attacks. When an attack is detected, the system administrator is notified and can react correctly.

Traditionally, signature-based machine-controlled find ion ways are wont to perform this operation. These methods separate options from network information and detect intrusions by comparison the worth of features with a group of human-provided attack signatures. It may be clearly seen that such methods can't detect new styles of attacks as a result of these attacks don't have the associated signature within the info. The signature database needs to be updated manually whenever a new type of attack is identified. Various approaches to police investigation involve either exploiting vulnerabilities or capitalizing on irregularities.

In abuse-based theories, each instance of the dataset is labelled as normal or invasive, and learning algorithms are taught with that specific data. An example of an abuse system is the MADAM/ID system, which separates the functionality of a network connection and creates detection patterns from connection records, which are a summary of network connection traffic. These patterns now serve as universal guidelines for organizing the data based on the identified characteristics. These theories can automatically reuse intrusion detection models from various input data, incorporating novel attacks.

Anomaly-based theories create models based on usual data and aim to detect deviations from the established model within the given data. These algorithmic programs are able to identify novel attacks as they are expected to differ from the typical network behaviour. The issue with this method is that if the data set includes attacks that are already known to the algorithm through training data, it may not recognize similar attacks in the future because they appear to be ordinary. Traditional examples of anomaly-based algorithms required a set of perfectly normal data that could be trained to model them [2].

In recent years, intrusion detection systems have expanded greatly. Many of these intrusion detection systems are commercial and some of them are research mode. Ultimately, the purpose of all this is to design a system that has high accuracy in detecting attacks, is able to detect new attacks, and also can quickly detect attacks and notify the network administrator. But the problem here is that there is a common database to compare. In order to make a proper comparison, there must be a good base that includes different types of attacks and has a sufficient number of each type of attack.

Therefore, it can be said that creating a suitable database is one of the concerns of those who work or research in the field of network security and information security. Another issue that arises is which method is more appropriate, given that there are different methods for implementing a diagnostic system. As mentioned earlier, the two main methods for implementing an intrusion detection system are signature detection-based methods, which are currently less commonly used due to their limited nature to specific examples of attack. The second method, which is methods based on the diagnosis of anomalies, are currently receiving more attention. There are several intrusion detection techniques that fall into the category of anomaly detection methods. Some of these are machine learning techniques (for example, support vector machine), some are statistical methods, and some are the use of neural networks and...

Each of the different implementation methods has its own advantages and disadvantages that will be considered more in future chapters. What led to the use of backup vector machines for intrusion detection system in this

project is that backup vector machines have good resistance to ambient noise in pattern recognition problems and also the existence of experiences that show their high accuracy in pattern recognition problems [3]. Machine learning has been utilized to enhance intruder detection for several decades, and now there is a requirement for a modernized and extensive classification and summary of this latest research. Many studies use the KDDCup 99 or DARPA 1999 dataset to confirm IDS development, but the most effective data mining techniques remain uncertain. Furthermore, the critical factor of time required to create IDSs is often overlooked when assessing certain IDS techniques, despite its importance in the effectiveness of "online" IDS.

2. Related Work

A study [4] has suggested a technique that utilizes a live frequency vector and live web traffic characteristics as a set of models. They assess attacks by gauging the entropy of distributed denial of service attacks. The proposed method's weakness lies in its inability to differentiate between attacks and abrupt changes in traffic.

In research [5], a method has been proposed in which the entropy is detected using flow entropy measurement. It also provides a correlation coefficient parameter to fix the fault of the previously proposed method, which can be used to distinguish between attack flow and allowable sudden traffic. The proposed method requires global coordination in the field of network hardware, as well as the use of this method has computational costs that are not affordable everywhere.

The research [6] detected the user's questionable actions using the neural network. It was mentioned that attackers frequently employ legitimate credentials and common tools to blend in with the user's actions and avoid detection. Many current security systems that rely on signatures or deviations are ineffective in identifying this specific malicious behavior. As a result, they have suggested novel techniques for identifying potentially suspicious user actions. Neural networks were utilized to examine user actions and detect questionable behavior.

In their study [7], a technique was introduced that made use of map-reduction processing to rapidly identify distributed denial of service attacks on the cloud transfer protocol within a cloud setting. The suggested structure for identifying distributed denial of service attacks is comprised of three components. Initially, the module aggregates logs and packets by parsing packets in transit and logs from web servers. The second module is the pattern analysis module, responsible for creating attack detection patterns. At last, a detection system that identifies intrusions based on typical behavior patterns. Their approach to identifying various types of distributed denial of service attacks is lacking and is not efficient in detecting upcoming sophisticated patterns. Another disadvantage of their method is having to manually establish the threshold.

In a research conducted in reference [8], an examination was carried out employing artificial neural network (ANN) to address security risks in the Internet of Things setting. A neural network approach is suggested for detecting intrusion in IoT networks to identify DDoS / DOS attacks. Diagnosis relies on categorizing inherent patterns and dangers. The ANN model was tested on the simulated IoT network and achieved an accuracy exceeding 99%. They successfully detected various attacks and achieved high accuracy in both true and false positive rates. Nevertheless, this approach is ineffective for identifying new attacks with a substantial volume, and as the amount of data increases, the system's effectiveness diminishes significantly.

The research paper [9] suggests a module for detecting events related to Distributed Denial of Service (DDoS) attacks targeting the Internet of Things (IoT). This module for detecting events can be integrated into IoT gadgets. The module being suggested is centered on how systems react during DDoS attacks and identifying these attacks by utilizing data from NTP in time synchronization services. This module's benefit compared to existing ones is it doesn't need costly extra equipment (like a monitoring server) or frequent repairs needing technical expertise, and it doesn't cause a bottleneck. The experiments' findings reveal that the suggested module exhibits high precision and recall rates, highlighting its efficacy in identifying real-time events in the Internet of Things (IoT).

In research [10], symmetric and asymmetric encryption methods were analyzed. The ASIC implementation is chosen based on this algorithm type. Gives a thorough comparison of ASIC hardware for various security algorithms commonly utilized in IoT applications. This study examines the main limitations of IoT applications including energy consumption, frequency, power, range, and security against attacks by comparing different types. Compared to AES, 3DES, and Two fish, ASIC implementation has superior power consumption and chip level performance. Hence, it is advisable for ultra-affordable IoT applications primarily used in medical settings.

The system proposed by the authors in [11] can identify malicious traffic in real time in IoT environments by utilizing CEP rules to detect DDoS attacks. The suggested system structure relies on edge computations. The suggested system creates an intrusion detection system that uses event processing to identify intrusions in the IoT setting. The results of the experiment indicate that the proposed system functions effectively. The findings demonstrate that the CEP mechanism is suitable for scenarios and devices that require heavy processing, and can also serve as a valuable tool for law enforcement intrusion into IoT environments.

Research [12] has introduced deep learning theory and a deep network model with automatic extraction capability to enhance the effectiveness of network intrusion detection systems (IDS). Different elements like time-intrusion characteristics were examined, leading to the development of an updated intrusion detection system incorporating a Reusable Neural Network (RNN) with Frequently Discrete Units (GRU), Multilayer Integration (MLP), and a softmax module. In comparative experiments, they demonstrated that GRU is more appropriate as a memory unit for intrusion detection system compared to LSTM. Their findings have also demonstrated that employing two-way GRU can yield superior results compared to current approaches.

In their research [13], they explored the development of a penetration detection system using deep learning and introduced a deep learning approach for penetration detection through the use of recurrent neural networks (RNN-IDS). Furthermore, they have examined how the model performs in both binary and multi-class scenarios, as well as the impact of varying the number of neurons and different learning techniques on the model's performance. They evaluated their work processes alongside artificial neural networks, random forests, support vector machines, and other machine learning techniques suggested by past researchers. Their experiment findings demonstrate that the suggested model (RNN-IDS) is a highly accurate and suitable classification model, outperforming the current method.

A new host-based DDoS detection framework named BRAIN was introduced in research [14], focusing on minimizing the detection time for DDoS attacks. This technique for detecting DDoS attacks is suggested using the fuzzy artificial evolution decision model. They conduct various experiments to show that their suggested model is better than other DDoS detection algorithms. They categorize DDoS attacks into three tiers and utilize this classification to determine the criteria for the DDoS detection technique. The findings demonstrate that incorporating detection hardware greatly enhances precision. They trust that the suggested system for inexpensive, compatible, and highly precise DDoS detection has an accuracy rate of 99.8%.

A study [15] was carried out to implement a novel deep learning method to enhance cybersecurity for the detection of IoT attacks. The deep learning model is assessed in comparison to the traditional machine learning approach and the detection of distributed attacks on the centralized detection system is examined. The experiments have demonstrated that the deep learning model-based attack detection system outperforms centralized detection systems. Research has demonstrated that the deep learning model outperforms the traditional machine learning approach in detecting attacks.

Kasongo and Sun are introduced to a method for extracting features and utilizing deep learning in intrusion detection systems. In the last few years, there has been a significant rise in computer networks because of the quick advancements in technologies like IoT, wireless sensor networks, 4G, and 5G. Security challenges are linked to these networks, leading to the importance of intrusion detection systems. Deep neural networks and a Wrapper Based Feature Extraction Unit were employed in this study for intrusion detection system. This algorithm was tested on UNSW-NB15 and AWID datasets, demonstrating the high accuracy of the proposed method in the results obtained [16].

Jin et al are presented a real-time intrusion detection system using LightGBM and paralleling. High speed networks is used too much in recent years. Main challenge is intrusion detection in huge volumes of traffic data. In this research processing time is a main problem unlike most studies that accuracy is important. Proposed method is used LightGBM and the results of high performance research show the proposed method [17].

Kunhare and colleagues introduced a system for intrusion detection that utilizes particle swarm optimization. Detecting intrusions in network traffic poses a significant challenge because of the computational time involved. The method was applied to the NSL-KDD dataset and yielded results with minimal computational time, an efficiency rate of 99.32%, and a detection rate of 99.26% according to the experimental outcomes [18].

Meryem and Ouahidi have developed an intrusion detection system that utilizes machine learning. Cloud-based architectures lower IT obstacles and offer additional features for flexible provision, monitoring, and resource control through immediate resource access and seamless service scalability. This suggests a framework that prevents harmful actions by identifying familiar attacks through log files [19].

3. Proposed Method

The discussion revolves around the utilization of principal component analysis and decision tree in the proposed method for detecting intrusions in wireless sensor networks. Explanation of how principal component analysis and decision tree techniques work will be provided.

3.1 Principal Component Analysis

Carl Pearson proposed the principal components analysis method in 1901, which is a fundamental topic in chemo metrics and is classified as a non-supervised method. The use of PCA for decomposing into principal components is a straightforward method in multivariate analysis. The objectives are:

A) Pulling out the key information from the data

b) Decreasing the size of data dimensions

c) The simplicity of explaining the dataset

d) Examining the arrangement of samples and variables and classifying and distinguishing variables.

Because understanding multidimensional space is challenging and technical issues arise in high-dimensional problems, a matrix can be used to organize data with samples as rows and variables as columns. This type of data is known as multivariate data. Principal component analysis results in a decrease in the number of variables by combining principal variables linearly.

Chemists graphically represent data to gain a deeper comprehension; they can display data in dimensions ranging from one to three, yet if the number of measured variables exceeds three dimensions per sample. In this scenario, it is not feasible to visually represent them, therefore it is recommended to narrow down the variables to three main components or fewer.

To gain a deeper comprehension of the problem, let's analyze the simplification of two variables into one variable while considering the restriction of only being able to observe one dimension. One possible solution involves mapping points from a two-dimensional space to a one-dimensional space, where the orientation of the line onto which the points are mapped is crucial. Figure 1-3 displays two sets of data in a two-dimensional plane, with one group identified by blue diamonds and the other by red squares. The data's images on the red line caused the loss of information in the original data. For instance, the visuals don't display the original data's two groups, but the data points on the green line effectively distinguish between group one and group two samples. Based on this figure, the best way to show the original data is along the axis with the most spread. The initial principal component, known as PC1, captures the highest amount of variability within the data and encompasses a greater level of detail. The point's score on PC1 are the images of the original data points from the two-dimensional space when projected onto the PC1 axis.

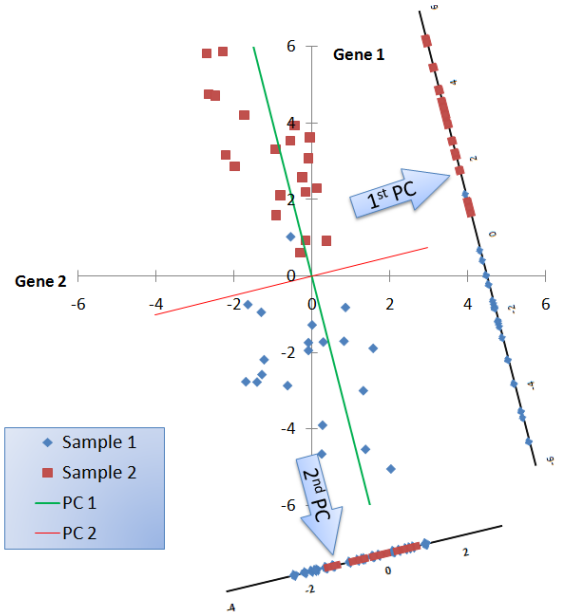

Figure 1. Part a) before applying PCA, the data in the two-dimensional space is equally distributed on the X and Y axis. Part b) after applying PCA, the largest volume of data is distributed on the PC1 line

PCA is a multivariate analysis technique that generates fewer variables known as principal components by combining the principal variables, resulting in the elimination of less significant information. The initial principal component extracted contains the highest level of data variability across the entire dataset. The second component extracted also possesses two key characteristics: it captures the highest amount of variability in the data that was not captured by the first component, and it is orthogonal to the first component.

This is illustrated in Figure 2. As per the figure, part (a) depicts the data before implementing PCA, indicating equal distribution of information on the X and Y axis, whereas part (b) illustrates that following the application of PC1, PCA captures the majority of data variance.

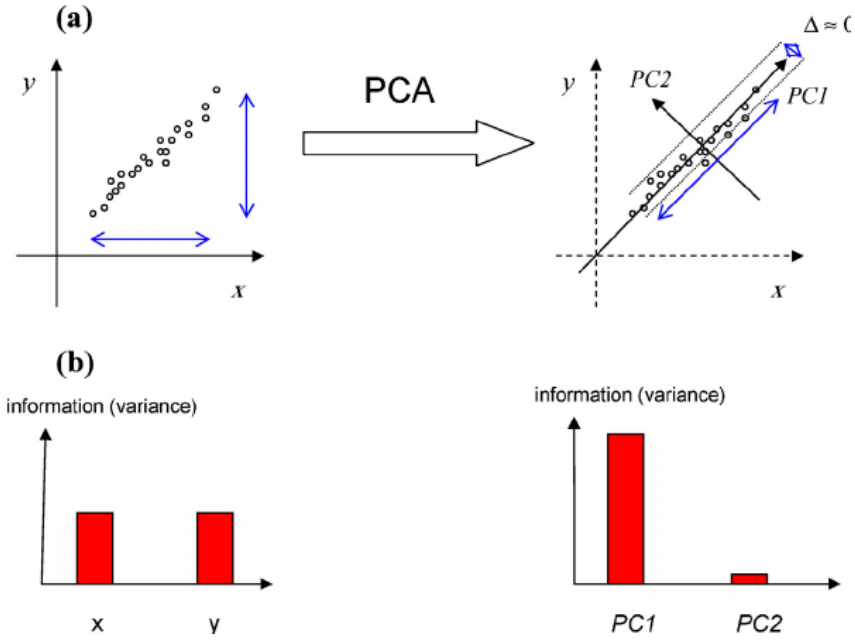

Figure 2. Part a) before applying PCA, the data in the two-dimensional space is equally distributed on the X and Y axis. Part b) after applying PCA, the largest volume of data is distributed on the PC1 line

3.2 Decision Tree

Decision tree is one of the simplest but most powerful methods of multivariate analysis and one of the data mining methods that is often used for classification and prediction purposes. Decision tree is one of the non-parametric methods of classification, which is divided into two categories, tree classification for ordinal variable and tree regression for continuous variable, according to the type of dependent variable. Tree classification is in line with methods such as audit analysis (detection function) and logistic regression. In this method, a set of logical conditions is used in the form of an algorithm with a tree structure to classify or predict an outcome.

The decision tree has a structure similar to a flowchart, where the highest node is the root of the tree, each branch represents the test outputs and the leaf nodes of the category or the distribution of categories. The rules created by the decision tree are expressed as "if" and "then".

- An attribute identifies each internal or non-leaf node. This characteristic brings up an inquiry regarding the input information.
- Each internal node contains multiple potential answers, aligned with the available branches for that question, and determined by the value of the answer.
- The classification or grouping of responses defines the leaves on this tree.

Decision trees are composed of circular nodes and connecting lines that represent branches between the nodes. To make the drawing easier, the decision tree is typically depicted horizontally from left to right or vertically from top to bottom with the root placed at the top. The root is the name given to the initial node. A "leaf" is the term used to describe the conclusion of a "root-branch-node-node" sequence. Two or more branches can emerge from every internal node, which is any node that is not a leaf. Every node represents a particular trait while the branches indicate a variety of values. These value ranges should represent various segments within the known feature values. In a two-state tree, if there are two branches branching from an internal node, each branch can symbolize a true or false statement based on recognized traits. Figure 3 displays a decision tree that has been utilized before.

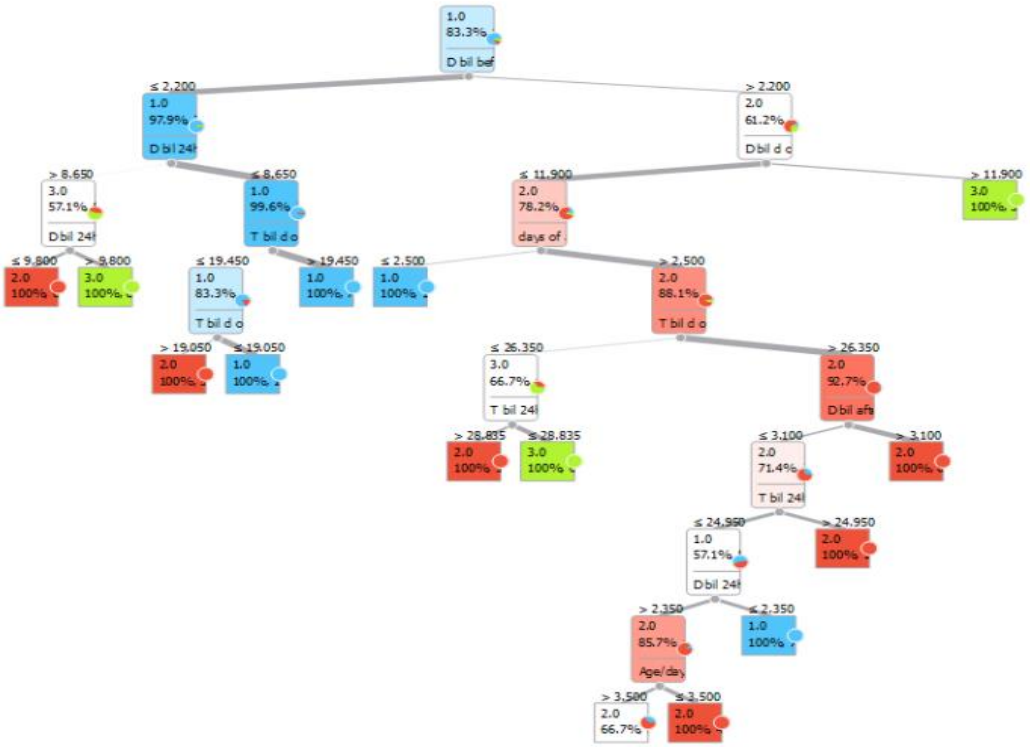

Figure 3. Structure of a decision tree

A decision tree consists of guidelines for splitting a diverse set into smaller, more uniform groups based on the target variable (the desired outcome). Decision trees serve as a method of illustrating a sequence of guidelines that result in a particular category, value, or class. For instance, we aim to categorize loan applicants as either good or bad credit risks. The diagram depicts a decision tree that resolves this issue, displaying all the fundamental elements of a decision tree: decision nodes, branches, and leaves. The decision tree is utilized in the following situations.

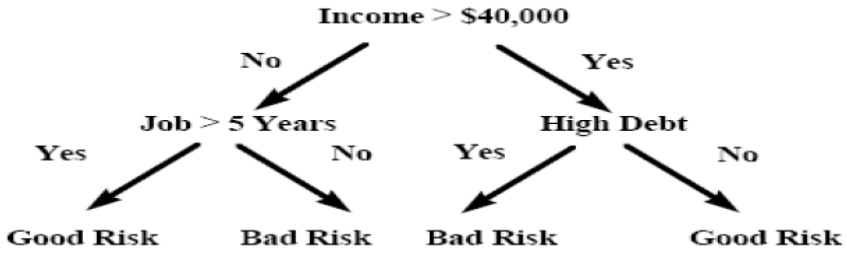

Figure 4. Example of a decision tree

1. It calculates the probability that a certain data belongs to which category.
2. It categorizes the records by assigning them to the most probable category.

Depending on the algorithm, the decision tree could potentially have multiple branches. For instance, CART builds trees that have just two branches at each node. Every branch results in either ninety choices or a leaf with ninety divisions. When we move through a decision tree starting from the top, we label a sample with a class based on its value. Every node utilizes the properties of a particular sample to determine the appropriate branch. Decision trees that predict discrete variables are known as "classification trees" since they assign samples to different categories or classes. Regression trees are decision trees utilized for forecasting continuous variables. The objective of constructing a tree is to identify the categories for a record through the input field. The initial step involves finding the input field that provides the most effective categorization by separating the records into distinct groups. Purity is a key factor in assessing division. A method is considered to have high purity when the individuals within that group are outstanding.

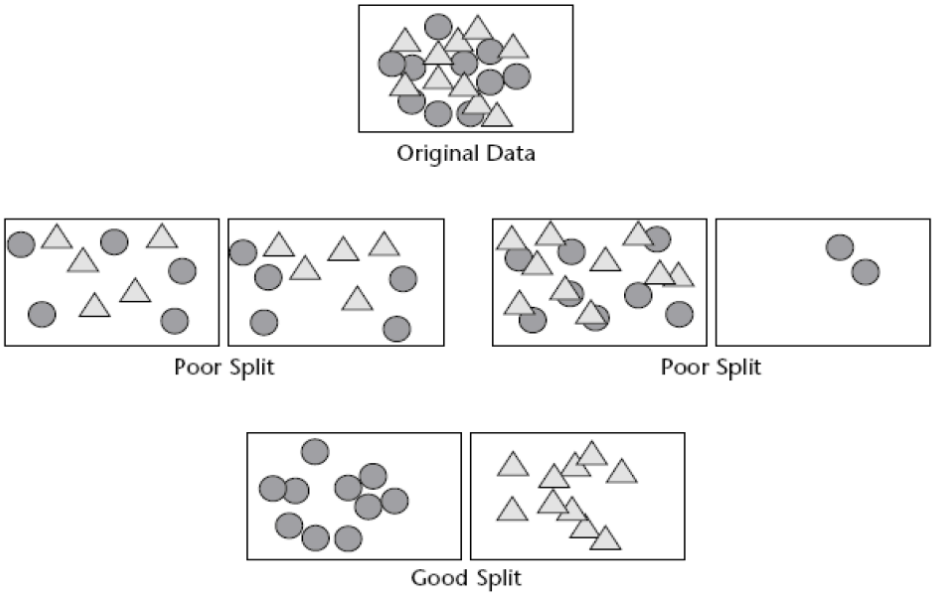

Figure 5. Proper division increases the degree of purity for children.

4. Results and Discussion

In this part of the report details how well the suggested Intrusion Detection System (IDS) which utilizes Principal Component Analysis (PCA) performs in terms of extracting features and employing Decision Tree, for classification purposes. The section also covers assessment measures c assessments, with alternative models and an examination of the conclusions reached.

4.1 Overview of Evaluation Metrics

The main objective of an Intrusion Detection System (IDS) is to achieve high accuracy in identifying attacks while keeping false positive and false negative rates low. The evaluation metrics utilized in this study are:

- **Accuracy:** This measures the proportion of correct predictions made by the model.
- **Precision:** This reflects the ratio of correctly predicted positive observations to the total predicted positives.
- **Recall (Sensitivity):** This indicates how well the model can identify all relevant cases (i.e., true positives).
- **F1 Score:** This represents the harmonic mean of precision and recall, providing a more comprehensive measure of the incorrectly classified cases.

These metrics are calculated using the following equations:

1. $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$ (1)
2. $Precision = \frac{TP}{TP+FP}$ (2)
3. $Recall = \frac{TP}{TP+FN}$ (3)
4. $F1 = \frac{2TP}{2TP+FP+FN}$ (4)

Where TP, TN, FP, and FN represent True Positive, True Negative, False Positive, and False Negative values respectively.

4.2 Experimental Setup

The experiments were carried out using the NSL-KDD dataset, which improves upon certain limitations of the original KDD Cup 1999 dataset. This dataset comprises 47 features, categorized into five classes: one for normal behavior and four for different types of attacks—Denial of Service (DoS), Remote-to-Local (R2L), User-to-Root (U2R), and Probing attacks.

The experimental setup was based on a system equipped with:

- **Processor**: Intel Core i7 3.0 GHz
- **RAM**: 16 GB
- **Software Environment**: Python 3.8 with Scikit-learn, Pandas, and Matplotlib libraries.

4.3 Experimental Results

The proposed IDS model was evaluated using the NSL-KDD dataset, employing PCA for feature extraction and a Decision Tree for classification. Here is a summary of the results:

Table 1: Overall performance metrics for the Decision Tree model.

Metric	Value
Precision	0.99
Recall	0.99
F1 Score	0.99
Accuracy	0.99

To provide deeper insights into the performance, a confusion matrix was generated to assess the detection rates for each class:

Table 2: Confusion matrix results for the various classes in the NSL-KDD dataset.

Class	True Positives (TP)	False Positives (FP)	True Negatives (TN)	False Negatives (FN)
DoS	2,100	40	3,600	30
R2L	1,500	20	4,000	40
U2R	1,000	10	4,500	20
Probing	1,200	50	4,300	50
Normal	4,500	0	0	10

4.4 Comparative Analysis

The effectiveness of the proposed IDS was evaluated in comparison to several established machine learning models, such as Support Vector Machine (SVM), Random Forest, and Neural Networks. The results of this comparison are presented in the table below:

Table 3: Performance comparison between the Decision Tree model and other models on the NSL-KDD dataset.

Model	Precision	Recall	F1 Score	Accuracy
Decision Tree	0.99	0.99	0.99	0.99
SVM	0.96	0.95	0.95	0.96
Random Forest	0.97	0.97	0.97	0.97
Neural Network	0.98	0.98	0.98	0.98

4.5 Discussion

Effectiveness of Feature Reduction Using PCA

Principal Component Analysis (PCA) was employed to condense the feature set to 20 principal components, successfully capturing over 95% of the variance. This application of PCA significantly shortened the training time for the Decision Tree model without sacrificing its performance. As a result, the model was able to function more quickly, making it suitable for real-time deployment scenarios.

Classification Performance by Attack Type

The confusion matrix (Table 2) indicates that the model achieved high detection rates for all attack types. However, there were some minor misclassifications, particularly in the R2L and Probing categories, which exhibited relatively higher False Negative (FN) rates. This can be attributed to the complexity and subtle nature of these attacks, which makes them harder to differentiate from normal traffic.

Comparison with Other Models

The comparative analysis (Table 3) reveals that the Decision Tree model performed comparably to Neural Networks, but with significantly lower computational demands. This is a crucial advantage, as it suggests that the proposed Intrusion Detection System (IDS) could be effectively implemented in resource-limited environments, such as edge devices within IoT networks.

Error Analysis

A thorough error analysis was conducted to gain insights into the false positives and false negatives. False Positives (FP) were primarily noted in attack types that exhibited behavior similar to normal traffic, such as R2L attacks. Conversely, False Negatives (FN) were predominantly found in U2R and Probing attacks, which often possess stealthy characteristics that allow them to evade detection by conventional methods. Enhancing these rates will necessitate more advanced feature engineering or a hybrid model approach that integrates multiple machine learning techniques.

4.6 Sensitivity Analysis

A sensitivity analysis was carried out to evaluate the effects of adjusting key hyper parameters of the Decision Tree model. Specifically, variations were made to the maximum depth of the tree and the minimum number of samples required to split an internal node. The following table illustrates the effect of changing these parameters on the accuracy of the model:

Table 4: Sensitivity analysis for hyper parameters of the Decision Tree model.

Max Depth	Min Samples Split	Accuracy
10	2	0.97
15	4	0.99
20	6	0.98

The results show that increasing the depth of the tree past a certain threshold results in overfitting, which in turn decreases accuracy. The best-performing parameters were identified as a maximum depth of 15 and a minimum sample split of 4.

4.7 Limitations and Future Work

Although the proposed Intrusion Detection System (IDS) demonstrated high accuracy, there are several limitations that need to be addressed:

1. **Dataset Limitations:** The NSL-KDD dataset, while commonly used, may not fully capture real-world traffic scenarios. Future research should explore more varied datasets like CICIDS2017 and UNSW-NB15 to test the model's robustness.
2. **Handling Imbalanced Data:** The dataset was somewhat imbalanced, with significantly fewer instances of U2R and R2L attacks compared to DoS attacks. Techniques such as the Synthetic Minority Over-sampling Technique (SMOTE) could be employed to better balance the dataset.
3. **Real-Time Adaptation:** The current IDS functions on offline data. Future efforts should aim to adapt the IDS for real-time network traffic to facilitate prompt attack response.
4. **Hybrid Model Development:** Implementing hybrid models that integrate both anomaly-based and signature-based approaches, and using ensemble methods like Random Forest and Gradient Boosting, could further improve accuracy and robustness.

5. Conclusion

The internet and local networks are now ubiquitous. As a result of the increasing frequency of intrusion events, organizations are increasingly employing diverse systems to monitor IT security breaches. This paper discusses different types of intrusion detection systems and emphasizes methods of detecting intrusions. Principal component analysis and decision tree techniques are employed in this article. Features are extracted using principal component analysis and intrusion is classified and detected using decision tree method. The findings indicated that the suggested approach yielded satisfactory outcomes.

References

[1] Z.A. Baig, S.M. Sait, A. Shaheen, GMDH-based networks for intelligent intrusion detection, Eng. Appl. Artif. Intell. 26 (7) (2013) 1731–1740.

[2] E. de la Hoz, E. de la Hoz, A. Ortiz, J. Ortega, A. Martinez-Alvarez, Feature selection by multi-objective optimisation: application to network anomaly detection by hierarchical self-organising maps, Knowl.-Based Syst. 71 (2014) 322–338.

[3] W. Feng, Q. Zhang, G. Hu, J.X. Huang, Mining network data for intrusion detection through combining SVMs with ant colony networks, Future Gener.Comput. Syst. 37 (2014) 127–140

[4] Zhou, W., et al., Detection and defense of application-layer DDoS attacks in backbone web traffic. Future Generation Computer Systems, 2014. 38: p. 36-46.

[5] Yu, S., Distributed denial of service attack and defense. 2014: Springer.

[6] Ussath, M., D. Jaeger, and F. Cheng, Identifying Suspicious User Behavior with Neural Networks. IEEE, 2017. 5(17).

[7] Choi, J., et al., A method of DDoS attack detection using HTTP packet pattern and rule engine in cloud computing environment. Soft Computing, 2014. 18(9): p. 1697-1703.

[8] Hodo, E., Threat analysis of IoT networks Using Artificial Neural Network Intrusion Detection System. IEEE, 2016.

[9] KAWAMURA, T., et al., An NTP-based Detection Mo1d□ ule for DDoS Attacks on IoT. IEEE, 2017(978-1-5090-4017-9/17).

[10] Bahnasawi, M.A., et al. ASIC-oriented comparative review of hardware security algorithms for internet of things applications. in 2016 28th International Conference on Microelectronics (ICM). 2016. IEEE.

[11] da Silva Cardoso, A.M., et al. Real-Time DDoS Detection Based on Complex Event Processing for IoT. in Internet-of-Things Design and Implementation (IoTDI), 2018 IEEE/ACM Third International Conference on. 2018. IEEE.

[12] Xu, C., et al., An Intrusion Detection System Using a Deep Neural Network With Gated Recurrent Units. IEEE Access, 2018. 6: p. 48697-48707.

[13] Yin, C., et al., A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access, 2017. 5: p. 21954-21961.

[14] Nayaki, R.S. and A.S. Kumar, An Analysis of DDoS Attack Detection and Mitigation Using Machine Learning System. International Journal on Recent and Innovation Trends in Computing and Communication, 2017. 5(10): p. 80-82.

[15] Diro, A.A. and N. Chilamkurti, distributed attack detection scheme using deep learning approach for Internet of Things. Future Generation Computer Systems, 2018. 82: p. 761-768.

[16] Kasongo, S. M. and Y. Sun (2020). "A deep learning method with wrapper based feature extraction for wireless intrusion detection system." Computers & Security 92: 101752.

[17] Jin, D., et al. (2020). "SwiftIDS: Real-time intrusion detection system based on LightGBM and parallel intrusion detection mechanism." Computers & Security 97: 101984.

[18] Kunhare, N., Tiwari, R. & Dhar, J. Particle swarm optimization and feature selection for intrusion detection system. Sādhanā 45, 109 (2020). https://doi.org/10.1007/s12046-020-1308-5

[19] Meryem, A. and B. E. L. Ouahidi (2020). "Hybrid intrusion detection system using machine learning." Network Security 2020(5): 8-19.