**Journal of Intelligent Systems and Internet of Things**
**Vol. 14, No. 01, PP. 01-15, 2025**
DOI: https://doi.org/10.54216/JISIoT.140101

# A Hybrid Heuristic AI Technique for Enhancing Intrusion Detection Systems in IoT Environments

**Yousra Abdul Alsahib S. Aldeen[1,*], Fadhel K. Jabor[2], Ghufran A. Omran[2], Mohammed Hamid Kassem[3] Raghad Hamid Kassem[4], Ali Naseer Abood[3]**

[1]Department of Computer Science, College of Science for Women, University of Baghdad, Iraq

[2]Office of the Vice President for Scientific, University of Baghdad, Iraq

[3]Department of Computer Science, University of Technology, Iraq

[4]Department of Computer Science, University of Information Technology & Communications, Iraq

Emails: yousraaa_comp@csw.uobaghdad.edu.iq; fadhel.k.jabor@uobaghdad.edu.iq; ghufran@uobaghdad.edu.iq; mh2618108@gmail.com; raghedhamid@yahoo.com; alinaseer443gg@gmail.com

**Abstract**

In the evolving landscape of the Internet of Things (IoT), effective intrusion detection is paramount for maintaining security and data integrity. This study introduces a hybrid heuristic technique utilizing artificial intelligence for enhancing intrusion detection systems (IDS) in IoT environments. By integrating various machine learning models, the research focuses on training, tuning, and validating a sequential neural network to predict intrusion occurrences based on extensive data analysis. The methodology involves modelling, which starts with training machine learning algorithms to predict labels from features, tuning the models to meet organizational requirements, and validating them using holdout data. Key machine learning techniques explored include logistic regression, k-nearest neighbors (KNN), naive Bayes, support vector machines (SVM), decision trees, random forests, and neural networks. Each technique's applicability to classification tasks, particularly binary and multivariate scenarios, is discussed in the context of enhancing IDS capabilities. A sequential neural network model, comprising multiple dense and dropout layers, was developed and trained with 148,033 parameters to achieve high accuracy and robustness. The architecture's effectiveness in learning intricate patterns associated with malicious activities while avoiding overfitting is emphasized. The study demonstrates the model's proficiency in binary classification tasks, which is critical for distinguishing between normal and anomalous behaviors in IoT systems. The results indicate that the neural network, optimized using the hybrid heuristic approach, shows a significant reduction in validation loss and a steady improvement in accuracy over multiple epochs. Despite initial overfitting signs, the model maintains high performance on unseen data, underscoring the importance of ongoing model assessment and tuning.

**Received: January 22, 2024 Revised: April 15, 2024 Accepted: June 20, 2024**

**Keywords:** Intrusion Detection System (IDS); Internet of Things (IoT); Hybrid Heuristic Technique; Machine Learning; Neural Network

## 1. Introduction

The exponential rise in the number of applications depending on computer systems has driven to a comparing rise in the significance of arrange security measures [1]. Concurrently, there are security gaps in each framework, which might welcome assist ambushes and hose financial development. Since of this, distinguishing vulnerabilities in the framework that is portion of the arrange has gotten to be more imperative, and it is fundamental to do so in genuine time with as much exactness as conceivable [2]. Inside the limits of this scratch pad, a demonstrate will be created and prepared with the utilize of an SVM classifier in arrange to decide whether or not the organize bundle contains an assault [3].The fundamental objective of an Interruption Location Framework (IDS) is to keep

an eye on all of the information passing over a organize and inform the right individuals if anything fishy is seen [4]. It is a piece of computer program that checks a framework or organize for potential threats like arrangement infringement or antagonistic performing artists [5]. Security data and occasion administration (SIEM) frameworks frequently caution directors or centralise all pertinent information in the occasion of a breach or pernicious action [6]. In arrange to separate between authentic and malevolent exercises, a security data and occasion administration framework (SIEM) coordinating information from a few sources and employments caution-sifting calculations [7].

It is conceivable to guard a particular endpoint from both inner and outside dangers by sending a host-based interruption discovery framework (IDS) on that endpoint [8]. Such an interruption discovery framework (IDS) may be able to monitor information streaming into and out of the computer, as well as the programs running in the foundation and the framework logs [9]. The perceivability of a host-based interruption discovery framework (IDS) is limited to the machine that it is introduced on, which diminishes the sum of setting that is available for decision-making. Nevertheless, this kind of IDS has broad understanding into the inside workings of the have computer [10]. A network-based intrusion detection system (IDS) may be used to continuously monitor a completely protected network. It is able to monitor all data packets as they go across the network and make decisions based on the contents and metadata included inside them [11]. This broader view provides more context and the capacity to spot broad hazards; yet these systems cannot see into the inner workings of the endpoints they are supposed to protect [12]. The main contributions of this paper are the following:

1- To assess the effectiveness of a sequential neural network model within the framework of " A Hybrid Heuristic AI Technique for Enhancing Intrusion Detection Systems in IoT Environments"
2- To amalgamate diverse algorithms and methods to discern the most productive artificial intelligence models.
3- To demonstrate the use of a sequential neural network, which consists of layers with different numbers of units (64, 128, 512, 128), and is designed for binary classification.
4- To guarantee that the chosen model not only exhibits high performance on the training data but also effectively applies to new, unseen data, which is crucial for dependable intrusion detection.
5- To focus on the significance of choosing AI models that strike a balance between learning from training data and generalizing to new data.

## 2. Immune System Detection Method

**Signature-based Method**: Bytes, ones, and zeros are only a few examples of the predetermined patterns in network traffic that signature-based intrusion detection systems may use to identify assaults [13]. It goes beyond that by looking for patterns of instructions that have been associated with malicious software in the past, which allows it to identify hazardous applications. The intrusion detection system's (IDS) signatures are the patterns it finds [14]. Attacks may be easily detected by signature-based intrusion detection systems if the pattern (signature) of the attack is already present in the system. Since the pattern (signature) of new malware assaults is unknown, it is very tough to detect them [15].

**Anomaly-based Method:** The rapid development of new malware necessitated the development of intrusion detection systems that could spot attacks from previously unseen threats. Machine learning is used in anomaly-based intrusion detection systems (IDS) to construct a trustworthy activity model. When fresh data comes in, it is cross-referenced with this model; anything that does not fit is considered suspect. The machine learning approach offers a more generic quality than signature-based intrusion detection systems (IDS). This is because it is possible to train models that are uniquely suited to a certain set of use cases and hardware specifications [16].

## 3. Related Work

[17] When it comes to the Internet-based services that cloud-computing environments provide to customers, security is a major problem [17]. Because of this, implementing an Intrusion Detection System (IDS) is crucial in these types of situations. When assessing intrusion detection systems (IDSs), several factors are considered; nevertheless, the feature selection method utilised to distinguish between legitimate and malicious actions is the most crucial. Our study's overarching objective is to find a feature selection technique that will help classifiers become more accurate at detecting intrusions. To transform the challenge of feature selection into an optimisation problem, a technique known as Hybrid Ant-Bee Colony Optimisation (HABCO) has been presented. With the use of BHSVM, IDSML, DLIDS, HCRNNIDS, SVMTHIDS, ANNIDS, and GAPSAIDS, we investigated the accuracy of HABCO. Compared to the approaches that were stated, it has been shown that HABCO has a better level of accuracy. [18] A large amount of data has been generated due to the growth of applications that use the Integrating the Internet of Things (IoT) into every facet of your life. There have been major security issues with Internet of Things applications since they sometimes need a number of technologies, such as cloud and fog

computing [18]. The inadequacy of the current security measures has contributed to a rise in cyberattacks, which in turn has increased the use of these technologies. Many AI-based security solutions have come out in the last few years. A system for detecting intrusions is one of these options. The development of smart analytical tools, which include data pretreatment and the enhancement of machine learning algorithm performance, is not possible without feature selection (FS) methods. By reducing the number of criteria, FS aims to increase classification accuracy. This study demonstrates how to enhance the Gorilla Troops Optimizer (GTO), an FS approach that is based on the BSA algorithm. This particular BSA is used by the newly produced GTO-BSA to enhance the performance exploitation of GTO. Reason being, it is quite good at finding real-world domains with ideal solutions. Both convergence and the final product's quality will improve because of this. A battery of performance measurements was applied to four unique IoT-IDS datasets to evaluate GTO-BSA. Datasets used were NSL-KDD, CICIDS-2017, UNSW-NB15, and BoT-IoT. The findings were contrasted with those of previous cutting-edge methods, such as the initial GTO and BSA, among many others. Research consistently shows that GTO-BSA achieves better results than rival algorithms when it comes to convergence rate and solution quality. [19] A deluge of data is being produced because of the fast integration of internet-of-things (IoT) applications into several parts of our daily life [19]. Cloud computing and fog computing, two of the most popular technologies used in IoT applications, have resulted in significant security breaches. Using these technologies has led to an increase in cyberattacks, which is a direct result of the present state of inadequate security measures. Intrusion detection systems (IDS) are only one of several AI-based security solutions that have been presented over the last few years. Intelligent technologies that depend on data preparation and machine learning algorithm performance improvement employ feature selection (FS) approaches to increase classification accuracy while minimising the quantity of features picked. This is accomplished by avoiding the selection of an excessive number of attributes. The utilize of metaheuristic enhancement methodologies in include determination, on the other hand, has expanded amid the final a few decades. For the reason of selecting highlights for interruption location frameworks, this ponder recommends a crossbreed advancement approach. GWDTO is the acronym for "dark wolf" and "scoop throated enhancement," the names of two calculations upon which the proposed strategy is based. The proposed approach may give more noteworthy execution since it more equitably disseminates the time went through investigating and misusing all through the advancement handle. To test how well the proposed GWDTO strategy worked on the IoT-IDS dataset, a battery of evaluation measurements was utilized. We did this to demonstrate that our calculation is superior to all the other unveiled advancement strategies. Along with this, we run the proposed procedure through a measurable assessment to see how solid and fruitful it is. The test comes about appear that the recommended strategy is superior at moving forward the interruption classification precision in IoT-based systems. [20] In order to identify potential dangers, breaches, and unauthorised access, an intrusion detection system (IDS) is considered crucial. Intruder detection systems keep an eye on massive amounts of data from networks, some of which can include unnecessary or excessive components [20]. Despite their importance for sound decision-making, these characteristics have a major influence on the system's efficiency since they slow down the categorization process. Multiple methods are used to assess its efficacy; all of them need massive amounts of data and substantial network traffic. Several methods have been used in the development of intrusion detection systems (IDSs) that are both sufficient and enhanced. This class of methods includes data mining, machine learning, ANNs, and swarm intelligence. By using a genetic algorithm (GA) to determine the optimal feature subsets, this study intends to provide a fresh approach to feature selection using the NSL-KDD dataset. Furthermore, a hybrid classification method that incorporates decision trees (DT) and logistic regression (LR) has been used to provide a higher DR and ACC. By executing and assessing a plethora of meta-heuristic algorithms, this study optimised the optimisation of the chosen optimum features. The results of the suggested study are compared to those of the existing feature selection methods in order to confirm that the performance has been enhanced. In terms of feature reduction (=20 out of =41), the testing findings demonstrate that the grey wolf optimisation (GWO) technique achieves the best levels of accuracy (99.44%) and detection (99.36%). [21] Cybersecurity faces a major challenge with these smart devices because of their increased susceptibility to malicious activities that may be detected in network data. Quite a few systems and end-users feel the pinch when this happens [21]. On the other hand, intrusion detection systems (IDS) are often used to avoid cyberattacks. Although intrusion detection systems (IDS) are vital in identifying and preventing cyberattacks in networks enabled for the Internet of Things (IoT), building an efficient and fast IDS to detect cyberattacks is a challenging research area. Removing unnecessary or duplicate data from massive IDS datasets is a crucial first step in creating effective and efficient IDS. Include determination (FS) is moreover a significant method for consolidating shifted traits into IDS datasets. In this manner, this ponder centres on a cross breed include choice approach. Combining measurable test-based channel procedures such as Chi-Square ($\chi$^2), Pearson’s Relationship Coefficient (PCC), and Shared Data (MI) with a metaheuristic methodology based on the Non-Dominated Sorting Hereditary Calculation (NSGA-II) permits this framework to enhance characteristics. As portion of the proposed procedure, NSGA-II's guided populace initialization employments filter-based calculations to rank highlights. The objective is to get to an arrangement quicker; hence, we do this. Utilizing the ToN-IoT dataset, we assess the proposed scheme's execution, taking into account the exactness and number of chosen highlights. A few state-of-the-art

innovative approaches are differentiated with the exploratory discoveries. When looking at the information, it is clear that the proposed plot performs much better. Out of 43 characteristics, as it were, 13 were advanced, and 99.48% of those highlights were redress. [22] Cybercriminals looking for to abuse user-specific security data are being pulled into open systems with open get to and the usage of Web Convention adaptation 6 (IPv6) [22]. This is the primary reason why security blemishes in IoT gadgets have recently taken middle arrange in the media. This study's yield incorporates a Web of Things (IoT) device-specific RNN-IDS, which stands for arbitrary neural arrange based heuristic interruption location framework. Preparing and testing neurons utilizing the NSL-KDD dataset at diverse learning rates takes after including recognizable proof. Two approaches were utilized to evaluate the recommended framework, which moved forward RNN-IDS's precision from 85.5% to 95.25. Moreover, the comes about illustrate that when compared to current machine learning calculations, the displayed shrewdly interruption discovery strategy outflanks them when it comes to recognizing between commonplace and unusual activity designs. [23] There is a part of zero-day security vulnerabilities in the Web of Things (IoT) since of all the open remote sensor systems (WSNs) [23]. A lightweight machine learning-based interruption discovery framework that performs well inside resource-limited IoT remote systems is the objective of this ponder. The framework is called a Web of Things (IoT) interruption discovery framework (IoTIDS). Interruption location is a vital security arrangement. The hereditary calculation (GA) and the dim wolf optimizer (GWO) were hybridized to shape IoTIDS, and this combination is known as GA-GWO. The essential objective of the crossbreed calculation for the Web of Things Interruption Discovery Framework (IoTIDS) is to decrease the dimensionality of the gigantic remote arrange activity by intentioned choosing the most valuable activity characteristics. By combining the best highlights of GA and GWO, we trust to compensate for their shortcomings by means of hybridization. In arrange to assess how well GA-GWO performs on IoTIDS, it is tried on AWID, a as of late created real-world remote interruption dataset. Taking after the dataset's preprocessing beneath different conditions, this assessment is performed. The experimental results demonstrated that the suggested GA-GWO enhanced the IoTIDS's performance in terms of processing costs, helped the IoTIDS detect with a high degree of precision, and reduced the number of false alarms. Among the existing approaches, GA-GWO has shown superior performance compared to FWP-SVM-GA (FS, weight, and parameter optimisation of SVM based on GA) and BGWO (binary GWO). [24] Intrusion detection systems, or IDSs, scan network traffic for abnormalities with the goal of strengthening network security [24]. The challenge of anomaly detection in networks is to identify genuine from suspicious incoming data. In order to identify incoming anomalous traffic patterns, automated detection systems generally employ widely known methods like machine learning. The Information Gain-based technique is one of the algorithms utilised in this article. The programme chooses the features with the optimal number of features from the NSL-KDD dataset. Support Vector Machine (SVM) is a machine learning approach that we have included into the feature selection process. By combining the Optimization-Cuckoo Search Algorithm with the Artificial Bee Colony Algorithm, we were able to optimise the SVM hyper parameters for successful dataset classification. Using the state-of-the-art intrusion dataset NSLKDD, we tested how well the suggested technique worked. In contrast to other modern NSLKDD algorithms, the proposed method not only outperforms them but also achieves a high degree of accuracy, as shown in the trials. [25] Every day, more and more individuals join the ranks of those who use the internet, driving up both data and network traffic [25]. Due to several reasons, including networking protocols and open broadcast transmission, the Internet of Things ecosystem's energy-limited sensor node resources are vulnerable to attacks. It does not take long for these hackers to get access to the system and launch a barrage of attacks, lowering service quality and overall performance. Firewalls miss certain kinds of attacks, but intrusion detection systems can catch them all. The purpose of these systems is to identify these kinds of assaults. Based on the attributes, the intrusion detection system can distinguish between normal and abnormal system characteristics. Up until now, every intrusion detection system has relied on some kind of machine learning model. Conversely, if we want to boost classification performance, the process of feature selection is crucial. To assist you in selecting the most appropriate features, this article presents a method for feature selection that is based on deep learning. Applying the decision tree algorithm as a classifier enables the detection of attacks in the IoT network as well as the classification of deep features. On the benchmark NSL-KDD dataset, we compared the proposed model to the state-of-the-art models using key performance indicators like as accuracy, precision, recall, and f1-score to evaluate its improved performance. In comparison to conventional intrusion detection systems, the newly created hybrid model has a maximum accuracy of 99.49%.

## 4. Proposed Model

### 4.1 Data Description

This dataset is indicative of network traffic and is usually used by network intrusion detection systems (NIDS). The dataset has 125,972 objects organised into 43 columns. There is a unique attribute of the network traffic or an outcome related to network security represented by each column in this table. Here we will go over each column and the data type that goes along with it in detail:

1. Duration (int64): Time it takes for the link to complete.
2. Protocol type (object): Whether it be TCP, UDP, or ICMP, the protocol type is important.
3. Service (object): Internet access at the final destination (for example, FTP, Telnet, or HTTP).
4. Flag (object): The connection's normal or bad state.
5. src_bytes (int64): Number of bytes sent from source to destination.
6. dst_bytes (int64): Number of bytes sent from one location to another.
7. Land (int64): This binary flag will be set to 1 (true) or 0 (false) depending on whether the connection is going to or from the same host and port.
8. wrong_fragment (int64): An incorrect number of pieces in this conjunction.
9. urgent (int64): This connection has a high volume of critical packets..
10. hot (int64): The quantity of "hot" indications included in the connection's content.
11. num_failed_logins (int64): Failures in login attempts.
12. logged_in (int64): A true or false signal that shows whether the login was successful or not.
13. num_compromised (int64): amount of worsening conditions.
14. root_shell (int64): To indicate if the root shell was acquired or not, a binary flag was toggled from 0 to 1.
15. su_attempted (int64): How many times the "su root" command was tried to be executed.
16. num_root (int64): how many times root has been accessed.
17. num_file_creations (int64): all procedures performed in order to generate a file.
18. num_shells (int64): How many shell commands were executed.
19. num_access_files (int64): The amount of action involving ACLs.
20. num_outbound_cmds (int64): Maximum number of instructions that may be sent across a File Transfer Protocol connection.
21. is_host_login (int64(): A binary flag (1 for true, 0 for false) indicates whether the login is linked to the host login.
22. is_guest_login (int64): If this binary flag is set to 1, it means that the login is for a visitor. In such case, it will be initialised to 0.
23. Count (int64): The sum of all connections established to the same host in the last two seconds, including this one.
24. srv_count (int64): How many times has a user accessed the same service in the last two seconds?
25. serror_rate (float64): the proportion of connections experiencing issues with "SYN".
26. srv_serror_rate (float64): The fraction of service connections that result in a "SYN" failure per 1,000.
27. rerror_rate (float64): ratio of failed connections with "REJ" messages.
28. srv_rerror_rate (float64): The number of "REJ" failures seen for every 1,000 connections to a certain service.
29. same_srv_rate (float64): The percentage of customers that utilise the same cloud service.
30. diff_srv_rate (float64): Quantity of service connections expressed as a percentage.
31. srv_diff_host_rate (float64): Tell me what percentage of service connections travel to different hosts.
32. dst_host_count (int64): Quantity of connections to a certain host.

33. dst_host_srv_count (int64): The number of additional connections that have been set up to the same host and service as the one being used now.
34. dst_host_same_srv_rate (float64): The proportion of connections that utilise the same service when all hosts are using it.
35. dst_host_diff_srv_rate (float64): Time spent connecting to certain services as a percentage of all connections to a single host.
36. dst_host_same_src_port_rate (float64): What percentage of all connections to a host utilise the same port as the current one.
37. dst_host_srv_diff_host_rate (float64): Detailed comparison of all connections made by a certain service to a specific host, organised by hostname.
38. dst_host_serror_rate (float64): Count of instances when connections to the same host encountered "SYN" errors.
39. dst_host_srv_serror_rate (float64): This metric tracks the frequency with which connections to the same host provider experience "SYN" errors.
40. dst_host_rerror_rate (float64): The sum of all "REJ" failures encountered by connections heading to the same host.
41. dst_host_srv_rerror_rate (float64): The fraction of connections that encounter "REJ" issues while connecting to the same service leading to the same host.
42. Outcome (object): The outcome of the link (for example, typical, unusual).
43. Level (int64): The gravity of the situation posed by the link.

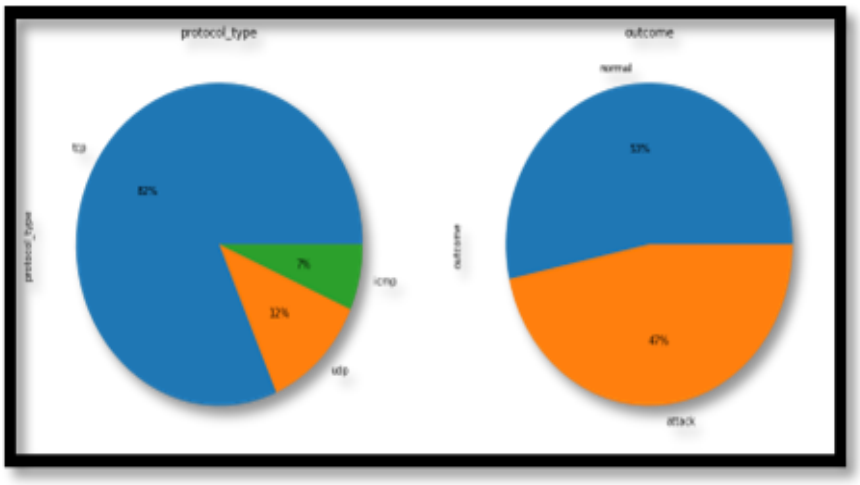


**Figure 1.** Dataset Description

### 4.2 Principal Component Analysis

If you have data that is too high dimensional, you may reduce its dimensions using principal component analysis (PCA), a statistical technique. To do this, we need to determine which features of the dataset are most important for collecting data. It is the variations in output that the traits' existence causes that determine their selection. The feature that accounts for the bulk of the variance is the first main component. Along the hierarchy, the trait responsible for the second-biggest variation is considered the second-primary component, and the process continues thereafter in the same manner. What has to be highlighted is the fact that the main components are completely unrelated to each other.

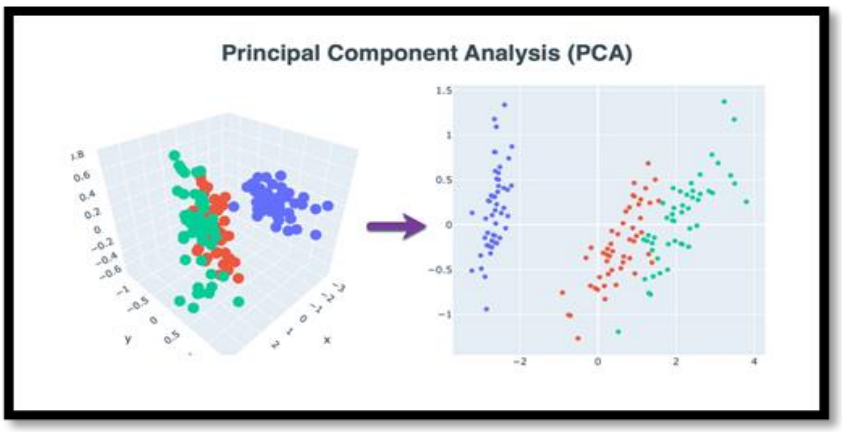


**Figure 2.** Principal Component Analysis

### 4.3 Advantages of PCA

There are two main advantages of reducing dimensionality using principal component analysis (PCA).

- When the number of features is reduced, the amount of time required for training the algorithms is greatly reduced.
- In certain cases, it may not be feasible to do data analysis in high dimensions. Take, for example, a dataset that has one hundred different characteristics. For visualising the data, the total number of scatter plots that would be needed would be 100(100-1)2 = 4950. In a practical sense, it is not practicable to do data analysis in this manner.

## 5. Data Analysis

### 5.1 Modelling

In order to model anything, you must first train a machine-learning algorithm to predict labels based on features, and then tune the algorithm to meet the requirements of the company, and then validate the method with holdout data. When modelling is completed, the result is a trained model that can be used for inference, which is the process of creating predictions based on fresh data points.

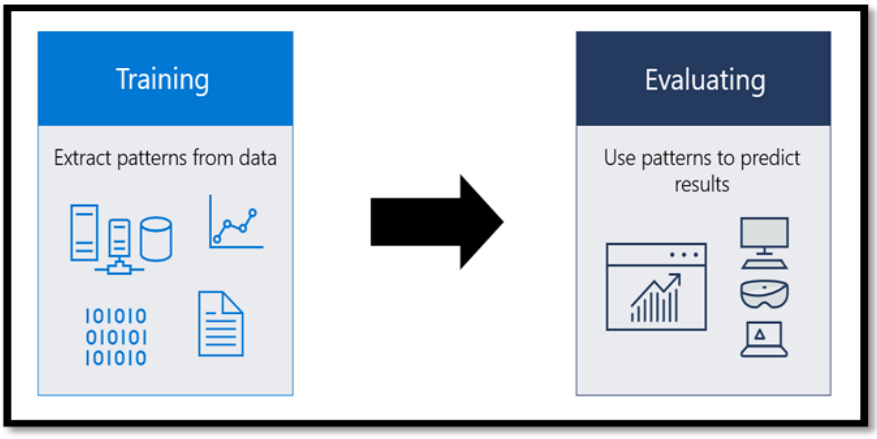


**Figure 3.** Data Processing

A file that has been taught to recognise certain sorts of patterns is what constitutes a machine-learning model in and of itself. The process of training a model involves providing it with an algorithm and a set of data so that it may learn and reason based on the data. After the model is trained, it may use previously unseen data to reason and generate predictions. Consider, for example, the situation where you want to create an application that can determine a user's emotional state from their facial expressions. When you provide a model image of faces with specific emotion labels on them, the model may be trained. Subsequently, you may use that model to an application that can identify any user's emotions.

### 5.2 Regression using Logistic

Classification and prescient analytics regularly utilize this specific kind of measurable show, which is too known as the logit demonstrate. By combining a number of autonomous factors, calculated relapse permits us to find out the likelihood of an event, like voting or not voting. This likelihood is evaluated utilizing the dataset. Based on the probability of the result, the subordinate variable can as it were take on values between zero and one. When calculating the chances in calculated relapse, a calculated change is utilized. The chances are calculated by separating the probability of victory by the likelihood of disappointment. This is moreover known as the log chances or the characteristic logarithm of chances. You can express this calculated work utilizing these equations:

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta T x}}$$

$$g(z) = \frac{1}{1 + e^{-z}}$$

The subordinate or reaction variable, h, and the autonomous variable, x, are both included in this calculated relapse condition. It is common hone to utilize the most extreme probability estimation (MLE) approach to gauge the beta parameter, moreover known as the coefficient, in this show. This strategy looks for the ideal fit of log chances by more than once testing different beta values. Finding the ideal parameter appraise in calculated relapse is as straightforward as boosting the log probability work, which is created after each cycle. It is conceivable to calculate and record the conditional probabilities for each perception and at that point include them together to produce a anticipated likelihood after finding the ideal coefficient (or, in the case of numerous autonomous factors, the coefficients). To discover out how likely it is that something will happen, this may be done. If the likelihood is less than 0.5, then the value will be predicted to be 0, and if it is more than 0, then the value will be predicted to be 1. After building the model, it is advised to assess its "goodness of fit," or its capacity to correctly predict the dependent variable**.**

### 5.3 Binary logistic regression

For this technique to work, the dependent variable or response must be binary. Responses or dependent variables may only take on two values, like 0 or 1, in this case. Among the most prevalent uses of this technology are the screening of emails for spam and the diagnosis of cancer via tumour analysis. This is the standard procedure for logistic regression and, more generally, for binary classification.

### 5.4 Multivariate logistic regression equation

The dependant variable in this logistic regression model may take on three or more values, but the sequence in which they appear is left unspecified. To better target their audiences, movie studios, for example, would benefit from having a better idea of the kind of films that would appeal to moviegoers. The studio may use a multinomial logistic regression model to determine how much a person's age, gender, and relationship status influence their preferred film genre. As a result, the studio may target the exact people most likely to see its flick in an advertising campaign.

## 6. K-Nearest Neighbours

A supervised learning classifier does not rely on parameters but instead uses the idea of proximity to provide predictions or classifications on the grouping of individual data points. The K-nearest neighbour’s algorithm (KNN) and k-NN are a couple of other names for this technique. Classification is where it really shines, although it can handle regression and other classification-related problems as well. The reason for this is that it is based on the premise that points with similar characteristics tend to be found near one other.

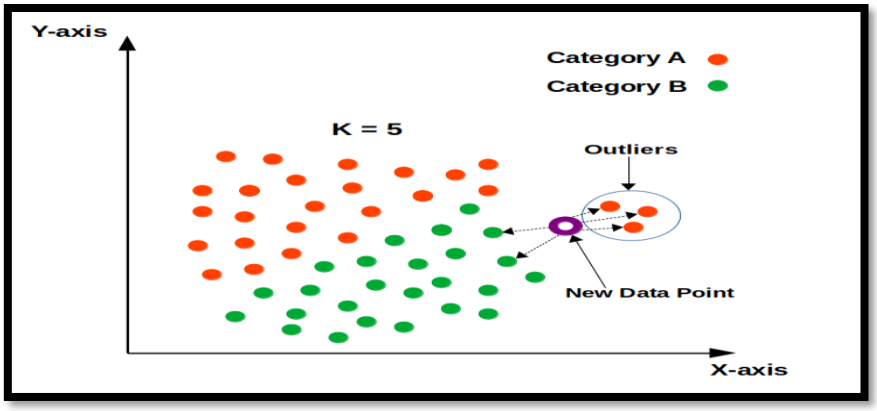


**Figure 4.** K-Nearest Neighbors

It is necessary to determine the distance between each data point and the query point in order to get the data points that are closest to the query point. The inquiry points are divided into several areas by decision borders, which are in part determined by these distance measurements. It is standard practice to use a Voronoi diagram to illustrate decision constraints.

## 7. Naive Bayes

Bayes' Theorem serves as the foundation for the Naive Bayes classifiers, which are a collection of other classification techniques. As opposed to being a single algorithm, it is really a family of algorithms, all of which adhere to the same fundamental idea. There is no correlation between any of the pairs of characteristics that are being categorised. When applied to events that occur in the real world, the assumptions that Naive Bayes makes are not always accurate. In actuality, the assumption of independence is never accurate, yet it often functions well in practice. A thorough familiarity with Bayes' theorem is now required.

### 7.1 Theorem of Bayes

The Bayes' Theorem is a mathematical tool for determining the likelihood of an occurrence based on the likelihood of another event that has already taken place. Here is a mathematical expression of Bayes' theorem:

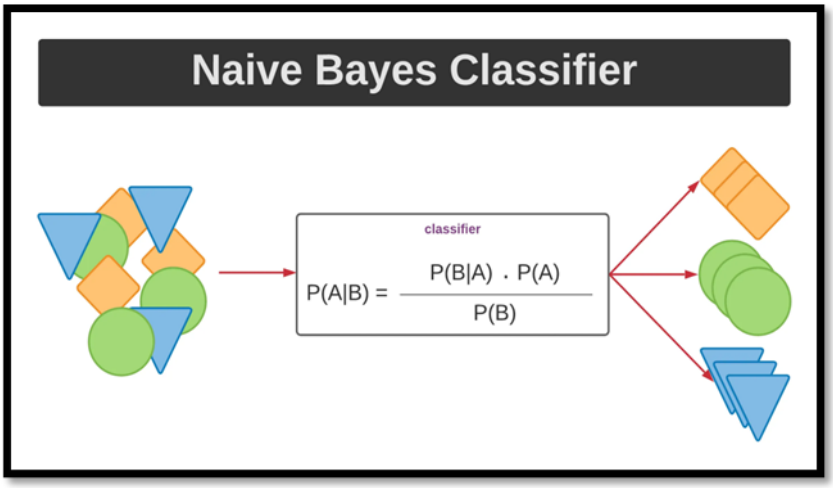


**Figure 5.** Naive Bayes

If the probability of event B is not zero, then A and B are events.

- Essentially, we are trying to figure out how likely it is that event A will happen if event B is true. This piece of evidence may alternatively be called Evidence B.
- The a priori, or prior probability, is P (A), which stands for the likelihood of an event happening before evidence is seen. In particular, event B serves as the proof, which is a value linked to an unknown occurrence.
- A posteriori probability, denoted as P (A|B), indicates the likelihood of an event occurring after evidence has been seen.

### 7.2 Vector Machine Support

Support Vector Machine, or SVM for short, is one easy supervised machine learning technique. Its applications include regression and classification. While its primary usage is in classification, there are cases when it might be useful for regression as well. In its most basic form, the SVM finds a hyperplane that divides the data into several groups. Looking at this hyper-plane in two-dimensional space, all it is a line. The support vector machine (SVM) method involves plotting a dataset with N attributes or characteristics in a space with N dimensions. Next, we need to locate the hyperplane that divides the data most efficiently. This should have made it quite clear that SVMs are limited to binary classification, meaning they can only initially choose between two classes. In contrast, multi-class scenarios lend itself to several different ways. In Order to Resolve Issues Involving Multiple Classes, Using Support Vector Machine It is possible to create a binary classifier for every data class to use support vector machines (SVM) on problems with many classes. Here are the two results that each classifier will produce:
There are two possible outcomes: either the data point is a member of that class, or it is not a member of that class.

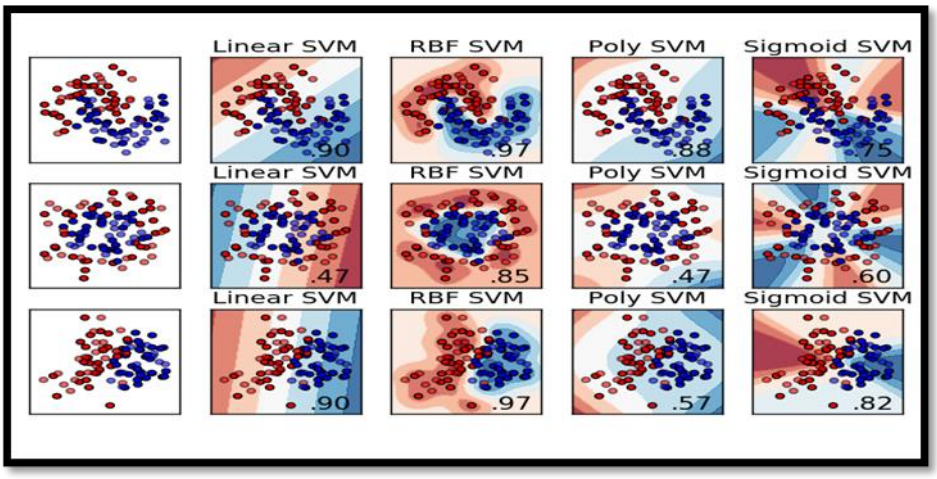


**Figure 6.** Support Vector Machines

To implement multi-class classification, for instance, we may create a binary classifier for every fruit in each class. Whether you want, to know whether something is a mango or not, a binary classifier can do it. A binary classifier, for instance, will be present in the "mango" class. It is determined that the output of the SVM will be the classifier that received the greatest score. SVM when dealing with complex data that is not linearly separable when applied to linearly separable data, SVM works well out of the box. If a data set can be visually represented and partitioned into categories using a straight line, we say that it is linearly separable.
We use kernelized support vector machines when working with data that is not linearly separable. Assume for a moment that we have one-dimensional data that may be non-linearly segregated. The data may be transformed into two dimensions and back into two dimensions to make it linearly separable in two dimensions. A two-dimensional ordered pair is found for every one-dimensional data point in order to do this. It is possible to make data that is not separable along any dimension linearly separable by translating it to a higher dimension. This programme may be used with any kind of data. There is a great deal of ground to cover in this shift. A kernel is just a metric for determining how similar two sets of data are. One way to find out how similar two data points are in the original feature space and the newly converted feature space is by looking at the kernel function of a kernelized support vector machine (SVM). Two of the numerous kernel functions available are quite popular, and they are given here:
It is clear from the graph below that the degree of similarity between any two points in the transformed feature space quickly declines as the distance between the vectors and the initial input space grows. The term "radial basis function kernel" is an abbreviation for just that. The RBF kernel is the standard option for support vector machines. The polynomial kernel stands out due to its required extra parameter "degree," which regulates the computing cost of the transformation as well as the complexity of the model.

## 8. Decision Tree

Decision trees are the most common and effective tool for a wide range of classification and prediction applications. Every node in a decision tree stands for an attribute that has been tested, every branch indicates the outcome of that test, and every leaf node includes the label of the corresponding class. The decision tree is a kind of hierarchical data visualisation that is similar to a flowchart.

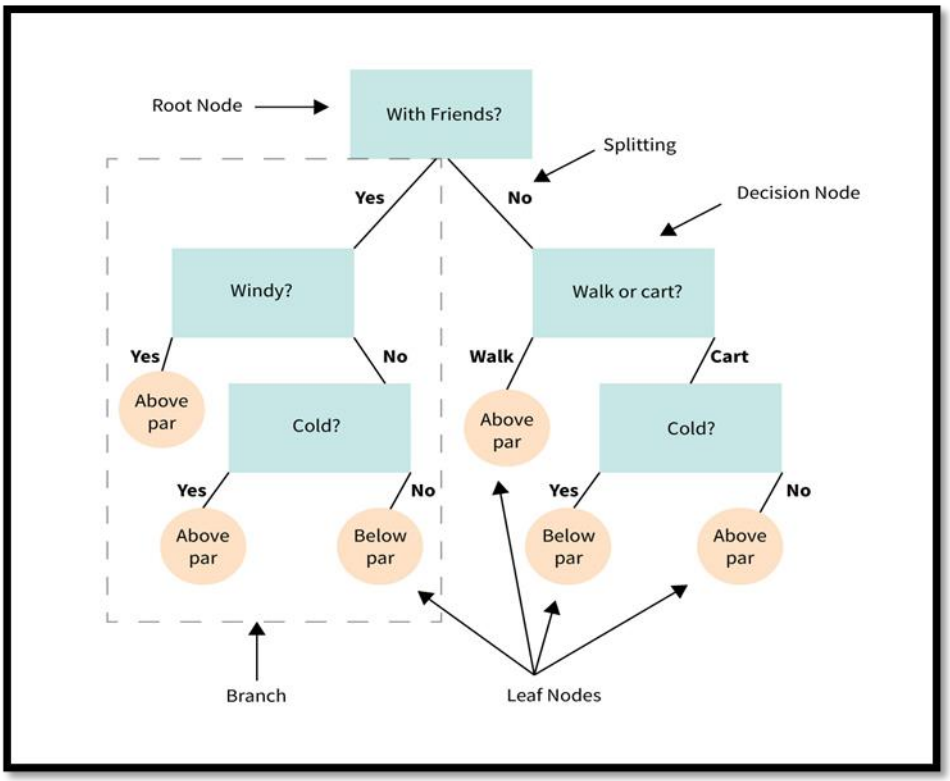


**Figure 7.** Decision Tree

Subsets of the source set may be created during the "learning" phase of a tree based on the outcomes of attribute value tests. This process is applied iteratively to each derived subset using a technique called recursive partitioning. When the target variable's value is equal across all node subsets or when splitting yields no more useful predictions, the recursion terminates. It is thought that both of these things might happen. Starting at the root node and finishing at a leaf node, decision trees arrange instances down the tree to provide the categorization of the instance. Starting at the root node of the tree and inspecting the property it provides, we may classify an instance by following the branch of the tree that corresponds to the value of the attribute, as shown in the diagram above. Subtrees rooted at the new node go through the same process thereafter. As shown in the image above, a decision tree is used to classify mornings based on whether or not they are suitable for tennis. The tree then provides the associated classification for each leaf. (Here, it comes down to a simple yes or no).

## 9. Random Forest

To illustrate the concept of supervised learning, consider the random forest method. It builds what seems to be a "forest" but is really just a collection of decision trees trained, on average, using the "bagging" method. A number of distinct learning models might overlap in the bagging strategy, which is predicated on the idea that the result can be improved. Among random forest's many advantages is its applicability to the two main categories of machine learning problems—classification and regression—that the majority of these systems now tackle. It also does not become over fit, which is a problem with decision trees.

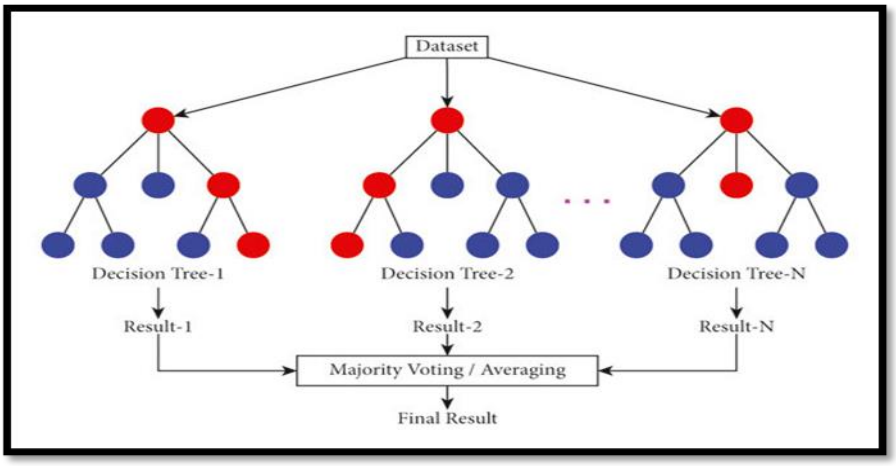


**Figure 8.** Random Forest

## 10. Neural Networks

The backbone of deep learning methods are neural networks, which go by a few different names: artificial neural networks (ANNs) and simulated neural networks (SNNs). The other names for neural networks are neural networks. Their design and moniker are inspired by the human brain, which exemplifies how biological neurons interact with one another. An input layer, one or more hidden layers, and an output layer are the building blocks of an ANN's structure. They have an output layer as well. There is a weight and a threshold associated with every artificial neuron, or node. Nodes are also linked to other nodes. Data transmission to the next layer of the network will begin at any node whose output exceeds the predetermined threshold. Data will not be sent to the next layer of the network until this condition is satisfied.

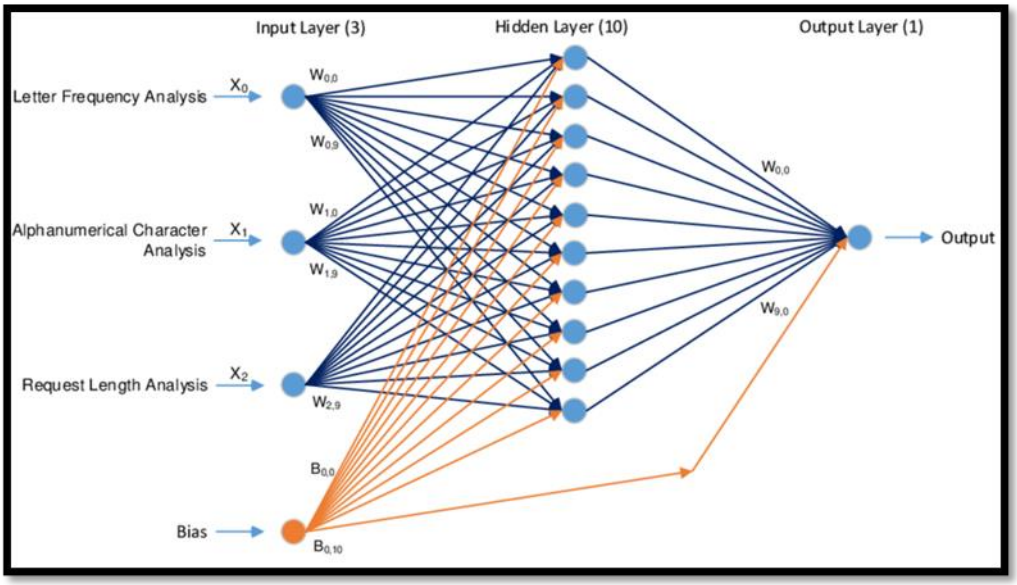


**Figure 9.** Neural Networks

Training data is essential for neural networks so that they may learn and become more accurate over time. On the other hand, if they have been adjusted for precision, these learning algorithms may be valuable tools in AI and computer science. Thanks to them, we can quickly group and label data. It may take minutes, instead of hours, to do jobs requiring speech recognition or image recognition, as compared to the manual identification technique performed by human experts. One of the best-known neural networks powers Google's search engine algorithms.

## 11. Results

The results of the sequential neural network model can be related to the title "A Hybrid Heuristic Technique for Artificial Intelligence Selection in Intrusion Detection Systems for IoT" in the following way:

The sequential neural network described, with its 148,033 trainable parameters, is a sophisticated AI model suitable for complex tasks like intrusion detection in IoT systems. The model's architecture, featuring multiple dense layers interspersed with dropout layers, is indicative of a design optimized for high accuracy and robustness, essential characteristics for intrusion detection. In the context of a hybrid heuristic technique for AI selection, this model could be one of several candidate architectures evaluated. The hybrid heuristic approach might involve combining various algorithms and techniques to select the most effective AI model for intrusion detection. The dense layers provide the capacity to learn intricate patterns associated with malicious activity, while the dropout layers prevent overfitting, ensuring the model generalizes well to new, and unseen data. The specific layer configuration, with progressively increasing and then decreasing units (64, 128, 512, 128), suggests a design that captures both detailed local features and broader patterns, crucial for distinguishing between normal and anomalous behaviour in IoT environments. The final single-unit output layer indicates a binary classification task, aligning perfectly with the requirement to identify the presence or absence of intrusions. Overall, this sequential neural network exemplifies the type of advanced AI model that would be selected and optimized using a hybrid heuristic technique to enhance the effectiveness and reliability of intrusion detection systems in IoT.

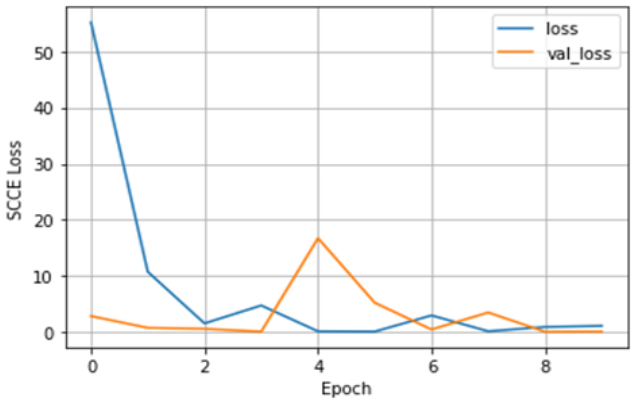


**Figure 10.** Validation Loss

The graph depicting the training loss of a machine-learning model over a series of epochs shows how the model's performance evolves during training. The x-axis, labeled "Epoch," represents the number of complete passes through the dataset, while the y-axis, labelled "val_loss," indicates the validation loss, a measure of the model's performance on a separate dataset not used for training. Initially, the loss is high, but it decreases over time, indicating that the model is learning and improving its performance. After about six epochs, the loss levels out, suggesting the onset of overfitting, where the model learns the training data too well and fails to generalize to unseen data.

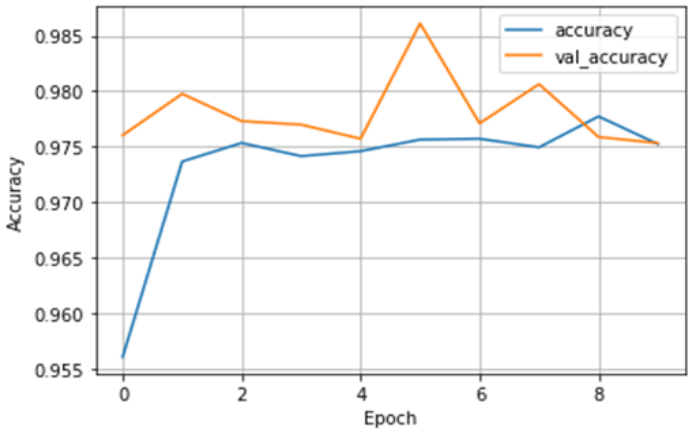


**Figure 11.** Accuracy Graph

The graph illustrates the accuracy of a measuring device over time, with the x-axis labelled "Epoch," representing the number of times the device was measured, and the y-axis labeled "Accuracy," ranging from 0.955 to 0.985. The blue line shows the device's accuracy, while the red line represents validation accuracy. As the number of epoch's increases, the accuracy of the device, indicated by the blue line, shows a slight improvement, suggesting that the device is becoming more accurate over time. However, the validation accuracy, shown by the red line, is generally lower than the device's accuracy but follows a similar trend. This indicates that while the device performs

well on the data it has been calibrated with, its accuracy on unseen data is slightly less. The observed gap between the training and validation accuracy suggests that the device might be overfitting to the calibration data, performing marginally less accurately on new data. This highlights the importance of ensuring that the device not only improves in accuracy with repeated measurements but also maintains high accuracy on new, unseen data to ensure reliable performance in real-world applications.

## 12. Conclusion

The study aimed to assess the effectiveness of a sequential neural network model within the framework of "A Hybrid Heuristic Technique for Artificial Intelligence Selection in Intrusion Detection Systems for IoT." The model contains 148,033 trainable parameters and includes several dense layers that are alternated with dropout layers, resulting in a high level of accuracy and robustness. This architecture is very suitable for complicated tasks such as intrusion detection in IoT systems. It can learn subtle patterns that are linked with malicious behaviours, while also avoiding the problem of overfitting to the training data. The hybrid heuristic technique entails the amalgamation of diverse algorithms and techniques to discern the most efficacious artificial intelligence models. The study demonstrates the use of a sequential neural network, which consists of layers with different numbers of units (64, 128, 512, 128), and is designed for binary classification. This advanced AI model has the potential to be further optimised utilising the mentioned technique. This approach guarantees that the chosen model not only exhibits high performance on the training data but also effectively applies to new, unseen data, which is crucial for dependable intrusion detection.

The training loss graph illustrates the model's progressive improvement over time, with an initial high loss that gradually decreases during training. However, after around six epochs, the loss stabilises, suggesting the possibility of overfitting. This implies that although the model efficiently acquires knowledge from the training data, there is a potential for it to not perform well when applied to new, unseen data. In addition, the accuracy graph depicted the progressive enhancement of the measuring device's precision over time, with a minor improvement in accuracy observed in each epoch. Nevertheless, the validation accuracy regularly remained below the training accuracy, exhibiting a parallel pattern but emphasising a disparity caused by overfitting. This highlights the importance of having models that consistently achieve high levels of accuracy when dealing with new data. This is a crucial factor to consider during the hybrid heuristic selection process. To summarise, the study highlights the significance of choosing AI models that strike a balance between learning from training data and generalising to new data. The utilisation of the hybrid heuristic technique is essential in optimising models such as the sequential neural network, guaranteeing its efficacy and dependability in detecting intrusions in IoT systems. The observed patterns in the metrics for training and validation highlight the importance of ongoing assessment and modification to avoid overfitting and sustain optimal performance in practical scenarios.

**Funding:** “This research received no external funding”

**Conflicts of Interest:** “The authors declare no conflict of interest.”

## References

**[1]** Darch Abed Dawar, A. (2024). Enhancing Wireless Security and Privacy: A 2-Way Identity Authentication Method for 5G Networks. International Journal of Mathematics, Statistics, and Computer Science, 2, 183–198. https://doi.org/10.59543/ijmscs.v2i.9073

**[2]** Savanović, N., Toskovic, A., Petrovic, A., Zivkovic, M., Damaševičius, R., Jovanovic, L., ... & Nikolic, B. (2023). Intrusion detection in healthcare 4.0 internet of things systems via metaheuristics optimized machine learning. Sustainability, 15(16), 12563.

**[3]** Kang, D. W., Ye, S. Q., Ahmad, S. Z. R. S., Mo, L. P., Qin, F., & Zhou, P. (2024). An Adaptive Harmony Search Part-of-Speech tagger for Square Hmong Corpus. Baghdad Science Journal, 21(2 (SI)), 0622-0622.

**[4]** Kareem, S. S., Mostafa, R. R., Hashim, F. A., & El-Bakry, H. M. (2022). An effective feature selection model using hybrid metaheuristic algorithms for iot intrusion detection. Sensors, 22(4), 1396.

**[5]** Dankolo, N. M. D., Radzi, N. H. M., Mustaffa, N. H., Talib, M. S., Yunos, Z. M., & Gabi, D. (2024). Efficient Task Scheduling Approach in Edge-Cloud Continuum based on Flower Pollination and Improved Shuffled Frog Leaping Algorithm. Baghdad Science Journal, 21(2 (SI)), 0740-0740.

**[6]** Salih, N., Ksantini, M., Hussein, N., Halima, D. B., Razzaq, A. A., & Ahmed, S. (2023). Deep learning models and fusion classification technique for accurate diagnosis of retinopathy of prematurity in preterm newborn. Baghdad Science Journal. Published online October, 20.

**[7]** Hu, W., Cao, Q., Darbandi, M., & Jafari Navimipour, N. (2024). A deep analysis of nature-inspired and meta-heuristic algorithms for designing intrusion detection systems in cloud/edge and IoT: state-of-the-art techniques, challenges, and future directions. Cluster Computing, 1-27.

**[8]** Saadouni, R., Gherbi, C., Aliouat, Z., Harbi, Y., & Khacha, A. (2024). Intrusion detection systems for IoT based on bio-inspired and machine learning techniques: a systematic review of the literature. Cluster Computing, 1-27.

**[9]** Muneer, S., Farooq, U., Athar, A., Ahsan Raza, M., Ghazal, T. M., & Sakib, S. (2024). A Critical Review of Artificial Intelligence Based Approaches in Intrusion Detection: A Comprehensive Analysis. Journal of Engineering, 2024(1), 3909173.

**[10]** Saied, M., Guirguis, S., & Madbouly, M. (2024). Review of artificial intelligence for enhancing intrusion detection in the internet of things. Engineering Applications of Artificial Intelligence, 127, 107231.

**[11]** Samir, N. M., Musni, M., Hanapi, Z. M., & Radzuan, M. R. (2021). Impact of Denial-of-Service Attack on Directional Compact Geographic Forwarding Routing Protocol in Wireless Sensor Networks. Baghdad Science Journal, 18(4 (Suppl.)), 1371-1371.

**[12]** Ghasemi, H., & Babaie, S. (2024). A new intrusion detection system based on SVM–GWO algorithms for Internet of Things. Wireless Networks, 1-13.

**[13]** Heidari, A., & Jabraeil Jamali, M. A. (2023). Internet of Things intrusion detection systems: a comprehensive review and future directions. Cluster Computing, 26(6), 3753-3780.

**[14]** Issa, M. M., Aljanabi, M., & Muhialdeen, H. M. (2024). Systematic literature review on intrusion detection systems: Research trends, algorithms, methods, datasets, and limitations. Journal of Intelligent Systems, 33(1), 20230248.

**[15]** Ashour, M. A. H. (2022). Optimized Artificial Neural network models to time series. Baghdad Science Journal, 19(4), 0899-0899.

**[16]** Gaber, T., Awotunde, J. B., Folorunso, S. O., Ajagbe, S. A., & Eldesouky, E. (2023). Industrial internet of things intrusion detection method using machine learning and optimization techniques. Wireless Communications and Mobile Computing, 2023(1), 3939895.

**[17]** Alkanhel, R., El-kenawy, E. S. M., Abdelhamid, A. A., Ibrahim, A., Alohali, M. A., Abotaleb, M., & Khafaga, D. S. (2023). Network Intrusion Detection Based on Feature Selection and Hybrid Metaheuristic Optimization. Computers, Materials & Continua, 74(2).

**[18]** Sangaiah, A. K., Javadpour, A., Ja'fari, F., Pinto, P., Zhang, W., & Balasubramanian, S. (2023). A hybrid heuristics artificial intelligence feature selection for intrusion detection classifiers in cloud of things. Cluster Computing, 26(1), 599-612.

**[19]** Kunhare, N., Tiwari, R., & Dhar, J. (2022). Intrusion detection system using hybrid classifiers with meta-heuristic algorithms for the optimization and feature selection by genetic algorithm. Computers and Electrical Engineering, 103, 108383.

**[20]** Dey, A. K., Gupta, G. P., & Sahu, S. P. (2023). Hybrid Meta-Heuristic based feature selection mechanism for cyber-attack detection in IoT-enabled networks. Procedia Computer Science, 218, 318-327.)

**[21]** Alrashidi, M., Ibrahim, R., & Selamat, A. (2024). Hybrid CNN-based Recommendation System. Baghdad Science Journal, 21(2 (SI)), 0592-0592.

**[22]** Qureshi, A. U. H., Larijani, H., Ahmad, J., & Mtetwa, N. (2019). A heuristic intrusion detection system for Internet-of-Things (IoT). In Intelligent Computing: Proceedings of the 2019 Computing Conference, Volume 1 (pp. 86-98). Springer International Publishing.)

**[23]** Davahli, A., Shamsi, M., & Abaei, G. (2020). Hybridizing genetic algorithm and grey wolf optimizer to advance an intelligent and lightweight intrusion detection system for IoT wireless networks. Journal of Ambient Intelligence and Humanized Computing, 11(11), 5581-5609.)

**[24]** Al-Safi, A. H. S., Hani, Z. I. R., & Zahra, M. M. A. (2021). Using a hybrid algorithm and feature selection for network anomaly intrusion detection. J Mech Eng Res Dev, 44(4), 253-262.)

**[25]** Simon, J., Kapileswar, N., Polasi, P. K., & Elaveini, M. A. (2022). Hybrid intrusion detection system for wireless IoT networks using deep learning algorithm. Computers and Electrical Engineering, 102, 108190)