Peer-to-Peer Networking and Applications
https://doi.org/10.1007/s12083-024-01727-6

An improved Harris Hawks optimizer based feature selection technique with effective two-staged classifier for network intrusion detection system

U Nandhini[1] · Santhosh Kumar SVN[1]

Received: 15 October 2023 / Accepted: 4 May 2024

Abstract
Due to the increase in network attacks, maintaining network security is significantly difficult, to overcome security vulnerabilities Intrusion Detection System (IDS) is utilized. IDS is a software application that monitors the network traffic and detects the malicious activity in the network. Network Intrusion Detection System (NIDS) identifies the suspicious behaviour of nodes in the network by analysing the network traffic. Most of the existing IDS suffer from achieving better feature selection with high classification accuracy with reduced false alarm rate. In the proposed system, the Principal Component Analysis (PCA) technique is utilized to reduce the dimensionality of the dataset. Improved Harris Hawks Optimizer (IHHO) is employed for effective feature selection which provides powerful global search capability. For classification, two-staged classifier is proposed which employs Support Vector Machine (SVM) for stage-1 and K-Nearest Neighbors (KNN) for stage-2. The main goal of the proposed system is to combine the advantages of SVM and KNN to enhance classification accuracy with a reduced false alarm rate. The performance of the proposed system is evaluated by using the NSL- KDD dataset and it has achieved an overall classification accuracy of 95.01%, a False alarm rate of 0.01%, and an overall detection rate of 92.01%.

Keywords Network-based Intrusion Detection System (NIDS) · Principal Component Analysis (PCA) · Improved Harris Hawks Optimizer (IHHO) · Support Vector Machine (SVM) · K-Nearest Neighbors (KNN)

1 Introduction

An Intrusion Detection System (IDS) is an application software that monitors and analyses networks, services, and user information by analysing the traffic which effectively manages the networks and identifies security attacks. IDS is prominently used to protect the system's integrity, confidentiality, and availability. IDS consists of three stages namely the monitoring stage, analysis stage, and detection stage. In the monitoring stage, it identifies whether the sensors are host-based or network-based. In the analysis stage, attribute extraction or model identification method is selected. In the detection stage, the type of intrusion is analysed to detect the nature of the attack whether it is anomaly or misuse. IDS is categorized into anomaly-based and signature-based IDS [1]. Signature-based IDS detects known attacks whose pattern is already stored in the database whereas anomaly-based IDS detects only unknown attacks. Based upon network deployment IDS is further categorized into host-based or network-based. Host-based Intrusion Detection System (HIDS) monitors and analyses to detect any anomalies in the internal behaviour of the system. NIDS monitors the log of network traffic and it is positioned at a planned point within the network. When a malicious activity is detected, an alert is sent to the network administrator for further action.

The security of the data stored in computer systems is threatened by the complexity and frequency of attacks in the networks. Enterprises use NIDS to safeguard prominent network data and infrastructure. NIDS frequently detects intrusions by analysing network traffic in the form of packet captures. NIDS is categorized into two primary groups: signature-based and behaviour-based. Network traffic is categorized and addressed by signature-based NIDS [2] using a predetermined set of rules, metrics, or calculations. Behavioural NIDS is dependent on complex operations which involve Machine Learning (ML) algorithms to identify sophisticated and constantly changing threats.

✉ Santhosh Kumar SVN
santhoshkumar.svn@vit.ac.in

U Nandhini
nandhini.u@vit.ac.in

[1] School of Computer Science Engineering and Information Systems, Vellore Institute of Technology, Vellore, India

Published online: 08 June 2024

In this paper, a two-layer classification and detection technique-based NIDS is proposed. Data pre-processing is considered an important aspect in enhancing accuracy and improving the quality of a dataset. In data pre-processing, one-hot encoding [3] technique is used for converting categorical data into a new column, and label values are converted into numerical values. Further, the data is normalized and PCA is used for dimensionality reduction. PCA is used for reducing the number of dimensions in large datasets by condensing a large collection of variables into a smaller set that retains most of the large set's information [4]. ML algorithms can analyse data points considerably more quickly and easily with smaller datasets because there are fewer irrelevant variables to process.

Feature selection is one of the most prominent techniques for choosing the most significant properties or features from the entire dataset. The main advantage of feature selection is, that it removes unnecessary and irrelevant features, and saves the time and complexity of the system in terms of execution. Optimization techniques reduce the computational complexity and help in identifying an optimal feature. Harris Hawks Optimizer (HHO) [5], a swarm intelligence-based meta-heuristic algorithm, is proposed to imitate the coordinated foraging and multiple strategy encircling of prey by Harris hawks. HHO comprises two phases: exploitation and exploration, which are alternated by the prey's energy of escape. Enhanced HHO is a combination of Opposition-Based Learning (OBL), a self-adaptive approach, and Chaotic Local Search (CLS). The three strategies are combined with HHO to enhance its functionality and quicken the convergence curve.

The optimized features are further classified into anomaly and misuse using classification algorithms. For identifying anomalies, a Support Vector Machine (SVM) is utilized which classifies the data into normal traffic and malicious traffic. SVM is the best learning algorithm and pattern classifier which is based on statistical learning techniques for classification and regression with a range of kernel functions. Due to its high generalization capabilities and ability to overcome dimensions, SVM [6] is considered an important technique for anomaly intrusion detection. Further, the attack traffic is analysed for misuse detection using the K- Nearest Neighbors algorithm. KNN is a lazy learning algorithm which is simplest when compared to other algorithms. It is instance-based learning and it does not provide any information regarding non-parametric data. KNN [7] is effective in attaining better accuracy. The combination of both algorithms produces a better accuracy and reduced false alarm rate in the NIDS. Table 1 gives the abbreviations and acronyms which are used in this work.

The major contribution of the proposed work is as follows:

1. In data pre-processing, a one-hot encoding technique is utilized to convert categorical data into numerical data.
2. The data is normalized and PCA is used for dimensionality reduction.
3. Further, in feature selection the pre-processed data is optimized using IHHO which enhances the performance of the system, and execution time is reduced.
4. A two–layer classifier is proposed. In stage-1 anomaly is detected using SVM and in stage-2 misuse is detected by using KNN which improves the classification accuracy of the system.

1.1 Motivation

Due to the evolution of communication technology, the Internet is witnessing a growing number of connected devices. Network attacks are simultaneously controlling the network devices by employing various methods by the intruder. The security of ubiquitous IoT systems is crucial, so it is critical to detect IoT security risks and identify existing security mechanisms. Conventional security measures against known attacks have different uses and they might be effective only during certain circumstances, but they may have vulnerabilities. IoT networks have been exposed to network security breaches despite the presence of conventional security measures like, secure data transformation, user authentication, authorization control, and data privacy. In such a scenario, the relevance of the Intrusion Detection System (IDS) for IoT is significant. Therefore, introducing NIDS to identify malignant activity is specifically essential for network security.

1.2 Research Gap

Many researchers have made contributions to the development of efficient IDS and to achieve metrics such as increased detection rate, decreased false alarm rate, class accuracy, and F-score. Several researchers have focused on employing classification approaches such as decision trees, KNN, SVM, NB, or Meta classifiers as a single classifier. However, they do not achieve the expected detection accuracy due to drawbacks such as overfitting, computational cost, sensitivity to parameter tuning, imbalanced data, and instability. In recent works, the HHO algorithm is utilized but it suffers from dimensionality issues. In this paper, IHHO is employed which handles complex problems, enhances the convergence, and provides better exploitation of solutions. To overcome the drawback of dimensional reduction in IHHO, the PCA method is employed in the data pre-processing stage. In the proposed system, a two-staged classifier is employed. In stage-1 anomaly attacks are detected utilizing SVM and in stage-2 misuse attacks

Table 1 Acronym and Abbreviation

Acronym	Abbreviation
IDS	Intrusion Detection System
IoT	Internet of Things
DoS	Denial of Service
R2L	Root to Local
U2R	User to Root
ML	Machine Learning
NIDS	Network Intrusion Detection System
HIDS	Host Intrusion Detection System
PCA	Principal Component Analysis
HHO	Harris Hawks Optimizer
IHHO	Improved Harris Hawks Optimizer
SVM	Support Vector Machine
KNN	K-Nearest Neighbors
OBL	Opposition-Based Learning
CLS	Chaotic Local Search
LSTM	Long Short-Term Memory
BiLSTM	Bidirectional Long Short-Term Memory
CNN	Convolutional Neural Network
DNN	Deep Neural Network
ANN	Artificial Neural Network
DCNNBiLSTM	Deep Convolutional Neural Network with Bidirectional Long Short-Term Memory
ELM	Extreme Learning Machines
BEHHO	Binary Enhanced Harris Hawks Optimizer
GWO	Grey Wolf Optimizer
GWO-SVM	Grey Wolf Optimizer with Support Vector Machine
PSO	Particle Swarm Optimization
GA	Genetic Algorithm
AOA	Arithmetic Optimization Algorithm
RLMPSO	Reinforcement Learning based Memetic Particle Swarm Optimization
DT	Decision Tree
LR	Logistic Regression
RF	Random Forest
CIA	Confidentiality, Integrity, Availability
SIEM	Security Information and Event Management
SVR	Support Vector Regression
SRM	Structural Risk Minimization

are detected utilizing KNN. To increase the classification accuracy and the overall performance of the system, a two-staged classifier approach is employed.

1.3 Objectives

1. The objective of the proposed system is to reduce the dimensionality of the dataset and provide effective features for feature selection by employing PCA.
2. The IHHO deals with the dimensionality problem the objective of the proposed work is to reduce it by utilizing PCA so that it enhances the performance of the system and execution time can be reduced.
3. The main goal of utilizing Improved Harris Hawks Optimization (IHHO) is to attain good accuracy with a limited number of features with better execution.
4. The main objective of the proposed system is to detect anomaly and misuse attacks by using two-staged classifier with better classification accuracy and false alarm rate.

1.4 Paper Organization

The rest of the paper is structured as follows: Section 2 gives a review of the related works. Section 3 provides a theoretical framework of the related concepts. Section 4 provides the architecture and flowchart of the proposed system. Section 5 presents the proposed methodology. The performance evaluation and results are given in Section 8. In Section 7 conclusion of this paper is presented. Finally, Section 8 includes Limitations, and Future work is described.

2 Related Works

Various researchers have proposed many mechanisms for securing the network by using NIDS. Among them, Binbusayyis et al. [8] have proposed an effective ensemble feature selection technique to minimize the false alarm rate and detection time. Their system combines four filter-based feature selection measures such as distance, information, correlation, and consistency. Initially, feature encoding is performed by using the one-hot encoding method and feature scaling is carried out by using min–max scaling. The selected features using the ensemble feature selection approach are given as input to the random forest classifier. Their feature selection approach enhances the performance with a minimal set of features. However, in their scheme, there is a scope for improvement in intrusion detection accuracy.

Mushtaq et al. [9] have proposed an embedded classifier utilizing Long Short-Term Memory (LSTM) and autoencoder to elect optimal features for classification which identifies anomaly and normal attacks effectively. Their system employs one-hot encoding and a standard scaling technique for pre-processing the data. The pre-processed data consists of high dimensionality which is minimized using autoencoder and the features are selected for classification. When optimal features are selected, LSTM is used for classification. A combination of Autoencoder and LSTM provides better classification accuracy with a reduced false alarm rate.

Hnamte et al. [10] have proposed a combination of deep learning techniques which includes Convolutional Neural Network (CNN) and Bidirectional Long Short-Term Memory (BiLSTM) which effectively identifies the intrusion in the network. In their scheme, the data pre-processing is carried out by using a one-hot encoding method which converts categorical data into numerical data. Then it is normalized using the standard scalar method. The performance of the system is analysed and computed using Deep Neural Network (DNN), Convolutional Neural Network (CNN), Auto Encoder, LSTM, Deep Convolutional Neural Network with Bidirectional Long Short-Term Memory (DCNNBiLSTM). The results of their scheme show that the performance of multi-class classification achieves better accuracy in training and testing datasets when their system is compared with other existing schemes.

Choudhary et al. [11] have proposed a Deep neural network-based IDS for detecting intrusion in IoT. Their system identifies intrusion based on the patterns. Their system utilizes three datasets for training and testing. The developed DNN model requires a huge amount of data to achieve better accuracy. DNN with 20 hidden layers have been utilized in their system. In their scheme, Training data is given as input for feature extraction when the features are extracted the data is trained. The performance of the trained data is evaluated using different metrics in the testing phase. The developed DNN model outperforms the accuracy attained by other proposed systems.

Salo et al. [12] have proposed an information gain with PCA for feature selection and an ensemble classifier for IDS. Their system is proposed to address issues like redundant handling and to remove irrelevant features. A hybrid technique to reduce the dimensionality is proposed and various classifiers which include SVM, Instance-Based Learning algorithm, and Multilayer perceptron are combined. The average of probabilities algorithm is utilized to achieve the final decision based on the base learners. The performance of the Information Gain – Principal Component Analysis (IG-PCA) ensemble method is better in terms of accuracy and false alarm rate.

Pajouh et al. [13] have proposed a hybrid dimensionality reduction and classification model for IDS. Their model detects User to Root (U2R) and Remote Local (R2L) attacks in the IoT environment. Dimensionality is reduced using PCA and linear discriminate analysis. During this process, higher dimensional datasets are converted to lower dimensions with minimal features. The diminished features are further classified using Naïve Bayes and KNN which identifies the malicious behaviour effectively. The advantage of the proposed system is that it identifies U2R and R2L attacks more accurately.

Peng et al. [14] have proposed a mini-batch K-means method to address issues related to IDS. Their system utilizes a clustering approach in combination with PCA. The dataset is pre-processed and normalized to enhance the efficiency of the clustering. Further, the dimensionality of the pre-processed data is reduced using PCA and data clustering occurs. PCA converts high-dimensional data into low-dimensional data with the same amount of information. The overall performance of their system is more efficient in terms of intrusion detection accuracy when compared with K means, mini-batch K means and K means with PCA.

Alzaqebah et al. [15] have proposed a nature-inspired algorithm for classification and detection of attacks. The intrusion attacks are rapidly increasing so the proposed hierarchical IDS identifies the network attacks effectively. In their system, Harris Hawks Optimization with Extreme

Learning Machines (ELM) is utilized as a base classifier. Moreover, the optimizer generates a better feature set with the weight of ELM. The selected features are split into binary classification problems and results are combined as predicted labels. The advantage of their system is that it has a better detection rate than another multi-class classifier.

Peng et al. [16] have proposed an Enhanced Harris Hawks Optimizer for selecting optimal features. HHO is a swarm-based intelligence algorithm that has global searching ability. Their proposed system is developed to overcome issues related to feature selection and complex problems. In their scheme, the selected features using Binary Enhanced Harris Hawks Optimizer (BEHHO) provide classification by utilizing the KNN classifier. The advantage of their system is that it can deal with high-dimensional data. The advantages are it provides better intrusion detection accuracy with limited features.

Hussian et al. [17] have proposed a flexible HHO by combining OBL, self-adaptive learning, and CLS for obtaining effective feature selection and global optimization. Feature Selection plays a prominent role in achieving classification accuracy. The dimension of the selected features is reduced using the optimization technique. CLS, OBL, and self-adaptive learning are combined with HHO to enhance the performance and speed up the convergence curve. The performance of their system is analysed by removing one or more components from Enhanced Harris Hawks Optimizer. Their system provides better performance in terms of improving intrusion detection accuracy and reducing false alarm rates.

Zhang et al. [18] have proposed an Enhanced Harris Hawks Optimizer hybridized with external optimization to enhance the performance of HHO. Their system is proposed to address three major issues such as flaws of insufficient information utilization and extreme randomization in the exploration phase. Another issue is to properly balance between the exploration and exploitation stages. The final issue is to combine HHO with refracted OBL to increase the convergence speed and quality of the solution. Their system carries out external optimization operations with excellent local search capabilities which improves the exploitation potential. The advantage of the proposed system is that it has better accuracy and reliability.

Wisanwanichthan et al. [19] have combined Naïve Bayes and SVM to develop an embedded approach for NIDS. Their system is organized into two groups in which data preparation, feature selection, and validation occur individually. In the Data transformation stage normalization, one-hot encoding and PCA techniques are utilized. The features are selected using the intersectional correlated method. Further Naïve Bayes and SVM classifiers are used for training and validation. Naïve Bayes classifier identifies Denial of Service (DoS) and Probe attack whereas SVM detects R2L and U2R. The advantage is execution time of the proposed system is improved.

Gu et al. [20] have proposed an SVM-based framework with feature embedded Naïve Bayes for developing a reliable IDS. Their system utilizes Naïve Bayes feature transformation technique which is used to generate new features from the original features. Further, the transformed features are trained using an SVM classifier. Their system effectively detects whether the traffic is normal or intrusion. Their method has robust performance which is evaluated using five benchmark datasets. Additionally, the proposed method has significant advantages in terms of false alarm rate, detection rate, and accuracy.

Chen et al. [21] have proposed a combination of SVM and Artificial Neural Network (ANN) for intrusion detection. A simple frequency-based scheme and the term frequency-inverse document frequency scheme are used as encoding methods. Their system analyses each technique and produces a result stating which methodology is more efficient in terms of performance for intrusion detection. SVM has superior performance than ANN because it reduces the generalization error whereas ANN increases the generalization error. Term frequency-inverse document frequency scheme encoding is better than a simple frequency-based method because of system calls uniqueness. The result of the proposed system indicates that the performance is enhanced in terms of accuracy and false alarm rate.

Safaldin et al. [22] have proposed a modified Binary Grey Wolf Optimizer with SVM (GWOSVM – IDS) as part of an improved IDS. In the feature selection stage, Grey Wolf Optimizer (GWO) is utilized in which fitness is calculated and the convergence curve is updated. Further, in classification the selected features are scaled, features are vectorized, the model is selected, cross-validation is done and the SVM model is created. Their techniques intend to decrease the false alarm rate, and the number of features produced by IDS and enhance the intrusion detection accuracy while decreasing processing time.

Saif et al. [23] have proposed an embedded IDS utilizing metaheuristic algorithms and ML algorithms. The proposed system is developed to detect security attacks on cloud systems. Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Differential Evaluation, and other metaheuristic algorithms are used to select the best features, and supervised learning algorithms like KNN and Decision tree which is utilized to accurately classify the normal and attack classes based on the features. Additionally, a hybrid strategy for feature selection and classification is proposed. The performance of their system has better memory usage, CPU utilization, accuracy, and execution time.

Ding et al. [24] have proposed a KNN and procreative adversarial networks-based embedded method for intrusion detection. Their system addresses the unbalanced learning problem for which a tabular data sampling approach is utilized to balance between attack samples and normal samples.

The KNN method is employed for efficient under-sampling of normal samples to minimize the loss of sample information. Then, for attack sample oversampling, a tabular auxiliary classifier procreative adversarial method is utilized. The data is balanced by combining normal data after undersampling and the attack data after oversampling. The advantage of their scheme is that it has better F-measure, AUC, Recall, and accuracy.

Zameer et al. [25] have proposed a group stacked IDS that employs five classifiers for obtaining an ideal solution in feature selection. Their system addresses issues related to malware. A robust IDS is proposed to defend the computing infrastructure which protects data confidentiality. The proposed system utilizes five classifiers as base learners and MLP as meta learners. The output of base learners is given as input to the meta learner from which the final output is achieved. Their system is evaluated using ten separate runs which produces reduced standard deviation errors and enhanced generalizability. The advantage of the proposed system is that its computational cost is minimized with minimal features and performance is enhanced.

Lahasan et al. [26] have proposed a lightweight deep auto-encoder model for detecting the intruder. Their system achieves advantages like lowering latency, reducing communication energy, and protecting data security by simultaneously selecting the input characteristics, the training instances, and the number of hidden neurons using an effective two–layer optimizer. The accuracy of a KNN classifier and the autoencoder model's complexity are considered as a building block for the optimized deep model. The proposed system has outperformed many other optimizers such as Arithmetic Optimization Algorithm (AOA), PSO, and Reinforcement Learning based Memetic Particle Swarm Optimization (RLMPSO).

Mansoor [27] has proposed a blockchain collaborated with clustering-based IDS for the Industrial Internet of Things network. Their system is developed to address security issues in the network. In their work, HHO is employed to identify cluster head and chicken swarm optimization with unit-based is utilized to identify the intrusion. Accuracy, precision, f-score, and recall are enhanced in the proposed system. Limitations are multipath route planning is not performed in their system.

Kurni et al. [28] have proposed a Deep max-out network optimized by manta Ray political optimization for detecting the intrusion in the network. In their system, features are selected using the Fisher score and wrapper method based on Hellinger distance. Features dimensionality is increased using data augmentation and further deep maxout network is utilized to detect misuse and anomaly behaviour. Further, the proposed system needs to enhance the performance and reduce the computational cost.

Narengbam et al. [5] have proposed an artificial neuron-based HHO algorithm for detecting intrusion in the Wi-Fi network. Their system addresses issues related to attacks in the network. Artificial neurons are trained with a bio-inspired algorithm for a maximum number of iterations and an attack is identified. Utilizing the HHO algorithm avoids early convergence, diversity, and inequality between exploitation and exploration. However, it suffers from premature convergence and sub-optimal solutions.

Shitharth et al. [29] have proposed a rapid stochastic correlated optimization integrated with neural network technique to classify and detect attacks in the system. Their system comprises four stages namely data pre-processing, grouping, attribute selection, and classification. The data is normalized using a graph-based clustering algorithm and optimal features are selected using rapid stochastic correlated optimization technique. Further, a neural network mechanism is utilized to categorize the predicted label. Their system has enhanced the detection efficiency, and performance and minimized the computational time.

Amanullah et al. [30] have proposed a CNN for predictive modelling with optimistic multi-faceted feature attraction for preventing phishing attacks. Their system identifies phishing attacks by utilizing URL functions and weight is calculated for the phishing index. Further, the weighted features are examined using an optimistic multi-faceted feature selection technique, which is employed to lower the dimension of log variation and further, it is trained using CNN. Their method transforms URLs into regularized size scales and categorizes the attribute as a risk. The performance, accuracy, and sensitivity of their proposed system are outperformed in comparison with other methods.

From the overall observation of the literature survey drawbacks and research gaps are identified. The main limitation of IDS is detection accuracy and false alarm rate. During the feature selection phase, a prominent set of features is not selected which majorly affects the classification phase. Because of this the classification accuracy gets reduced and vulnerable attacks are not identified. Apart from detection accuracy, most of the proposed schemes have detection delays, increased false alarm rates, and computational and communicational overhead costs. Motivated from these observations in this proposed work an Improved Harris Hawks Optimizer has been proposed to effectively identify the features and enhance the performance of classification accuracy. The optimal features are selected which automatically reduces the training and testing time. The proposed system employs a two-staged classifier which effectively increases the classification accuracy and reduces the false alarm rate by identifying malicious traffic accurately. The advantage of the proposed system is that it has a reduction in computational and communicational overhead costs. Moreover, the proposed system enhances the detection of DoS, Probe, R2L, and U2R attacks. Table 2 gives a comparative analysis of existing works.

Table 2 Comparative analysis of existing works

Author	Methodology	Advantages	Disadvantages
Binbusayyis et al. [8]	Ensemble feature selection technique	Feature selection approach enhances the performance with minimal set of features	There is a scope in improvement of intrusion detection accuracy
Mushtaq et al. [9]	Embedded classifier utilizing Long Short-Term Memory (LSTM) and auto encoder	Combination of Auto encoder and LSTM provides better classification accuracy. with reduced false alarm rate	Increased training time for AE-LSTM and AE-BiLSTM
Hnamte et al. [10]	Deep learning techniques which include Convolutional Neural Network (CNN) and Bidirectional Long Short-Term Memory (BiLSTM)	Performance of multi class classification achieves better accuracy in training and testing datasets	The proposed system has increased training time when compared to other Deep Learning
Choudhary et al. [11]	Deep neural network-based IDS	DNN model outperforms the accuracy	Time complexity of the proposed system is high
Salo et al. [12]	Information gain with PCA for feature selection and ensemble classifier for IDS	Performance of the Information Gain – Principal Component Analysis (IG-PCA) ensemble method is better in terms of accuracy and false alarm rate	The proposed system is not able to handle huge amount of data
Pajouh et al. [13]	Hybrid dimensionality reduction and classification model	It identifies U2R and R2L attacks more accurately	False alarm rate is increased
Peng et al. [14]	Mini batch K-means method	Performance of their system is more efficient in terms of intrusion detection accuracy	Increase in time complexity
Alzaqebah et al. [15]	Harris Hawks Optimization with Extreme Learning Machines (ELM)	Better detection rate than other multi class classifier	Dimensionality reduction is still a drawback
Peng et al. [16]	Enhanced Harris Hawks Optimizer	Better intrusion detection accuracy with limited features	Run time and accuracy need to be reduced
Hussian et al. [17]	Flexible HHO by combining OBL, self-adaptive learning and CLS	Better performance in terms of improving intrusion detection accuracy and reducing false alarm rate	High dimensional problem
Zhang et al. [18]	Enhanced Harris Hawks Optimizer hybridized with external optimization	The proposed system has better accuracy, and reliability	Convergence Speed is slow for selective optimization problem
Wisanwanichthan et al. [19]	Naïve Bayes and SVM	Execution time of the proposed system is improved	Accuracy needs to be improved
Gu et al. [20]	SVM based framework with feature embedded Naïve Bayes	Has significant advantages in terms of false alarm rate, detection rate and accuracy	Different attack types are not considered
Chen et al. [21]	SVM and Artificial Neural Network (ANN)	Performance is enhanced in terms of accuracy and false alarm rate	Overfitting problem
Safaldin et al. [22]	Modified Binary Grey Wolf Optimizer with SVM (GWOSVM – IDS)	Intrusion detection accuracy is enhanced while decreasing processing time	To improve classification accuracy other classifiers can be utilized
Saif et al. [23]	Metaheuristic algorithms and ML algorithms	The performance of the system has better memory usage, CPU utilization, accuracy, and execution time	False positive rate needs to be enhanced
Ding et al. [24]	KNN and procreative adversarial networks based embedded method	The advantage of their scheme is that it has better F-measure, AUC, Recall and accuracy	Impact of class overlap need to be improved
Zameer et al. [25]	XGBoost, Decision tree, random forest, bagging classifier and extra tree with multilayer perceptron	The advantage of the proposed system is that its computational cost is minimized with minimal features and performance is enhanced	Accuracy and Detection rate need to be enhanced

Table 2 (continued)

Author	Methodology	Advantages	Disadvantages
Lahasan et al. [26]	Deep auto encoder model	The proposed system has outperformed many other optimizers	IoT input features need to be minimized
Mansoor [27]	Blockchain collaborated with clustering-based IDS	The proposed system accuracy, precision, f-score, and recall are enhanced	The disadvantage of the proposed system is multipath route planning is not performed in the work
Kurni et al. [28]	Deep maxout network optimized by manta Ray political optimization	It has achieved a maximum accuracy during testing	The proposed system needs to enhance the performance and reduce the computational cost
Narengbam et al. [5]	Artificial neurons based HHO algorithm	The proposed system has attainable solution utilizing the bio-inspired algorithm	But it also suffers from premature convergence and sub-optimal solutions
Shitharth et al. [29]	Rapid stochastic correlated optimization integrated with neural network	The proposed system has enhanced the detection efficiency, performance and minimized the computational time	Parameter tuning for identifying attacks is not performed
Amanullah et al. [30]	CNN for predictive modelling with optimistic multi-faceted feature attraction	The performance, accuracy, and sensitivity of the proposed system	Detection rate can be enhanced

3 Theoretical Framework

3.1 Intrusion Detection System

Internet connectivity is essential for communicating and transferring data. Computers are exposed to various threats which need to be continuously observed and identified. The various computer security threats include unauthorized disclosure of data, denial of service, and data corruption. Confidentiality, Integrity, and Availability (CIA) are compromised by the intrusions into the network. Intrusion detection monitors and detects illegitimate malignant behaviour of a system or a network for detecting intrusions [31]. Any malicious attack is monitored and reported to the administrator or collected using a Security Information and Event Management system (SIEM). This system integrates output from multiple sources and uses alarm filtering techniques to differentiate malicious activity from false alarms.

IDS works based on three stages: The first stage is the monitoring stage which identifies whether it is network-based or host-based, the second stage is the analysis stage which identifies feature extraction or pattern identification technique and the third stage is the detection stage which detects anomaly or misuse. IDS is broadly classified based on methodology and deployment. Based on methodology it is classified into signature-based detection and anomaly detection. In signature-based IDS it detects the attacks whose pattern is already stored in the system but it is quite difficult to detect the new malware attack as their pattern is not known. In anomaly-based IDS it detects the unknown malicious attack in the network.

Based on deployment IDS is classified into NIDS and HIDS. NIDS analyses network traffic to identify malicious activity, illegal access, or violations of security policies. The main aim of NIDS is to detect and notify the network administrators of any possible or ongoing attacks. The data packets are examined with distinct patterns or actions to notify the existence of an attack. NIDS is an important element of network security strategy. Threats need to be recognized and neutralized before they cause serious damage or jeopardize vulnerable data. HIDS is utilized to analyse unusual activity in a network. Both internal and external intrusions are identified by HIDS. The main goal of HIDS is to analyse suspicious patterns that might indicate a system breach. The security group can identify the type of threat which they are dealing with and take necessary action to mitigate the threat. Network activity is additionally monitored by HIDS [32].

IDS suffers from two main common problems. The unknown attacks in the network are not identified in an efficient manner. Hence, machine learning algorithms are utilized to improve the detection. False alarm rate is a major concern where a normal attack is also assumed to be

a violation. The main aim of IDS is to monitor the network or host for suspicious activity, generate alerts when intrusion is detected, and respond to malicious activities.

3.2 Principal Component Analysis

The amount of data acquired to produce a statistically significant result grows exponentially with the number of attributes in the dataset. This may result in problems like overfitting, longer computation times and decreased machine-learning model accuracy. When dealing with high dimensional data, issues known as the curse of dimensionality can occur. The combination of features is increased exponentially with the number of dimensions and it increases the computational complexity during classification and clustering. Furthermore, the number of dimensions can affect some ML algorithms, demanding additional data to reach the same accuracy as reduced dimensional data [4].

Mathematician Karl Pearson was the first researcher to propose the PCA technique in 1901. It functions under the requirement that the lower dimensional mapping should maximize the variance of the data. A set of correlated variables is transformed into a set of uncorrelated variables using an orthogonal transformation in the statistical process known as PCA. The most popular tool in ML for prediction models and exploratory data analysis is PCA.

In unsupervised learning algorithms, PCA is utilized to analyse the interdependencies among variables. This is referred to as general factor analysis in which regression establishes the optimal fit line. The primary goal of PCA is to reduce the dataset's dimensionality ensuring the retention of crucial patterns or relationships between variables without any prior knowledge of the target variables. The purpose of PCA is to diminish the dimensionality of a dataset, involving the identification of a smaller set of variables than the original and it is applicable for regression and data classification.

Principal components are formed as linear combinations of the original dataset variables, arranged in decreasing order of significance. The total variance encompassed by all the principal components is identical to the total variance in the initial dataset [33]. The primary principal component captures the most variability in the data, whereas the second principal component captures the maximum variance orthogonal to the first and this trend persists. PCA is utilized for feature selection, data compression and data visualization. The main aim of PCA is to handle complex datasets and make them efficient.

3.3 Harris Hawks Optimization

Machine learning classification methods are considered as a core for IDS. Similarly, the feature selection technique significantly influences the system's overall performance. Therefore, by selecting robust features with ease and incorporating them into the classification process, the effectiveness of IDS can be significantly increased. Meta-heuristic algorithms are methods of feature selection. Bio-inspired meta-heuristic algorithms are influenced by the behaviour of living organisms under specific circumstances, such as actions made when hunting and pursuing prey [34]. These algorithms work well with data that has various dimensions. Additionally, these algorithms have improved performance in resolving optimization issues. Bio–inspired ML and DL methods for IDS are easily adapted to different kinds of threats and attacks. Moreover, an effective IDS can be created to handle large amounts of data and identify online intrusions.

HHO is an evolutionary optimization algorithm [35] utilized for solving global search problems. HHO emulates the astute hunting instincts in nature. Both the HHO and optimization algorithms in general rely on the optimized solutions successive building, which is based on the best solutions built relatively. To handle the intricacies of the optimization process, the initial solutions may involve considering some of the least favourable options, and these choices might develop into the most effective solution.

The HHO's hunting behaviour enables it to function in a realistic and dynamic environment. The feature selection for IDS is viewed as a dynamic environment since network traffic varies. HHO is an excellent algorithm in terms of simplicity of use, computation speed and search space traversal efficiency. But like other meta-heuristic algorithms, it experiences delayed convergence and can enter local optima under certain conditions. To ensure both local and global optimization avoid trapping into local optima during optimization [36]. HHO is split into two stages: exploration and exploitation phase transitions are based on the prey's energy of escape. Harris hawks' ability to cooperate as they age and to exhibit a variety of attack patterns in response to environmental changes and prey escape strategies is their most notable trait. The Harris hawk individuals are candidate solutions in the Harris hawk's optimization algorithm, and the individual with increased fitness is considered prey. HHO has several advantages such as high convergence speed, solving optimization problems, versatility, fewer tunable parameters, fewer assumptions, and dynamic adaptation.

3.4 Support Vector Machine

Support Vector Machine (SVM) is a supervised machine learning technique which is used for classification and regression problems. SVM is the best classification method due to its generalization ability and theoretical principle when compared to other classification techniques. The SVM technique is classified into two types namely linear and nonlinear. Different kinds of kernels can be fixed in the SVM model. Linear dataset is used for linear model whereas 'rbf'

and 'polynomial' kernels can be used for non-linear model. The main objective of the SVM model is to locate a hyperplane in the best degree to divide the data points from one class and another [37]. The hyperplane with the biggest margin between the two classes is considered as a "best" degree. SVM does not inherently allow multiclass classification in its most basic form. It facilitates the division of data points into two classes and binary categorization. To solve multiclass classification, the multiclass problem is decomposed into several binary classification problems to apply the same principle to solve multiclass classification. SVM works well and is more effective in class margin of separation of classes and in high dimensional spaces. SVM have the advantage of converting the optimization issue into dual convex quadratic programs, which eliminates the challenge of employing linear functions in the high-dimensional feature space. The main aim of the SVM is to maximize the margin by separating the several classes in the given training dataset. SVM functions based on the Structural Risk Minimization principle (SRM) which minimizes the generalization error instead of minimizing the mean squared error on the training dataset [38]. This principle is used for the empirical risk minimization method. SVM is good at handling small sets of input data used for classification and regression methods.

SVM is more suitable for high dimensional spaces and it is versatile in nature which is applicable for both linear and non-linear problems. But SVM also suffers from issues such as vulnerability to the selection of kernel and parameters, incurs high computational costs for large datasets and interpretation could be complex. It is used in various applications such as visual recognition, text classification, biological informatics, handwriting analysis, and financial prediction. SVM is further extended into Support Vector Regression (SVR) and Nu-SVM for natural regularization factor.

3.5 K – Nearest Neighbors

It is a famous machine learning algorithm which is also used for classification and regression techniques. It is based on the idea that the same data or dataset has the same values or labels. KNN is used to store the whole training dataset as references in the training module and computes the Euclidean equation between the input datasets. The Euclidean distance helps to identify the k-nearest neighbour among the input dataset. The KNN classifier algorithm assigns predicted labels for the k-neighbours and the most common class labels among the input dataset. For the regression algorithm, the average weight of the target value is calculated for the k-neighbours to predict the value among the input dataset. To categorize the data points, the KNN classifier finds its k-nearest neighbours. The data point is then classified by a majority vote. To avoid overfitting or underfitting the model, it is important to carefully select the value of k. To choose the ideal value of k for the KNN algorithm, which enhances the performance and guards against overfitting or underfitting, one might employ cross-validation [39]. Prior to using the KNN technique, the outliers are also found using cross-validation. KNN is commonly used for its low computational time ease of interpretation and predictive power. Even though KNN is the simplest technique in nature, it can provide highly competitive results as well. The KNN model is most frequently used for disease prediction, which estimates the probability of a disease based on symptoms and accessible data. In handwriting recognition, KNN helps to identify the characters written by hand. In image classification techniques, KNN helps to identify the images in computer vision. KNN is simple and intuitive it does not involve any complicated mathematical equations regarding data distribution. This model does not require a training phase since it memorizes the data and makes it suitable for active datasets. Like SVM this method is also versatile because it gets adapted to various kinds of problems. It is a non-parametric model which does not make any expectations about data distribution [40]. KNN is more effective for small datasets and computational cost is high. But KNN also suffers from issues such as memory requirement, curse of dimensionality and imbalance data issues.

4 Proposed System Architecture

Figure 1 illustrates the architecture model of the proposed system. In this model, the network intrusion detection system which utilizes both misuse and anomaly methods is proposed. The proposed architecture comprises of data preprocessing module, feature selection module and classification and detection module.

For data pre-processing module network traffic (NSL-KDD) is given as input. Data is pre-processed using one-hot encoding method which deals with categorical data. The encoded data is further processed using PCA for the reduction of dimension. These data are further optimized using Improved Harris Hawks Optimizer in the feature selection module. This algorithm has an effective global search capability which enhances the accuracy. The optimized features are given as input for the classification module. This module comprises of two-staged classifier for stage -1 SVM is utilized to identify the intrusion and it classifies the traffic as normal or attack. In stage -2 KNN is utilized which detects whether the attack is still present or not. And finally, the activity of intrusion is reported to the end-user or administrator. In the proposed scheme decision manager controls and coordinates the other function modules in the system. The main role of the decision manager is to make decisions during feature selection, classification, and detection of network-based attacks in the network. Decision manager is

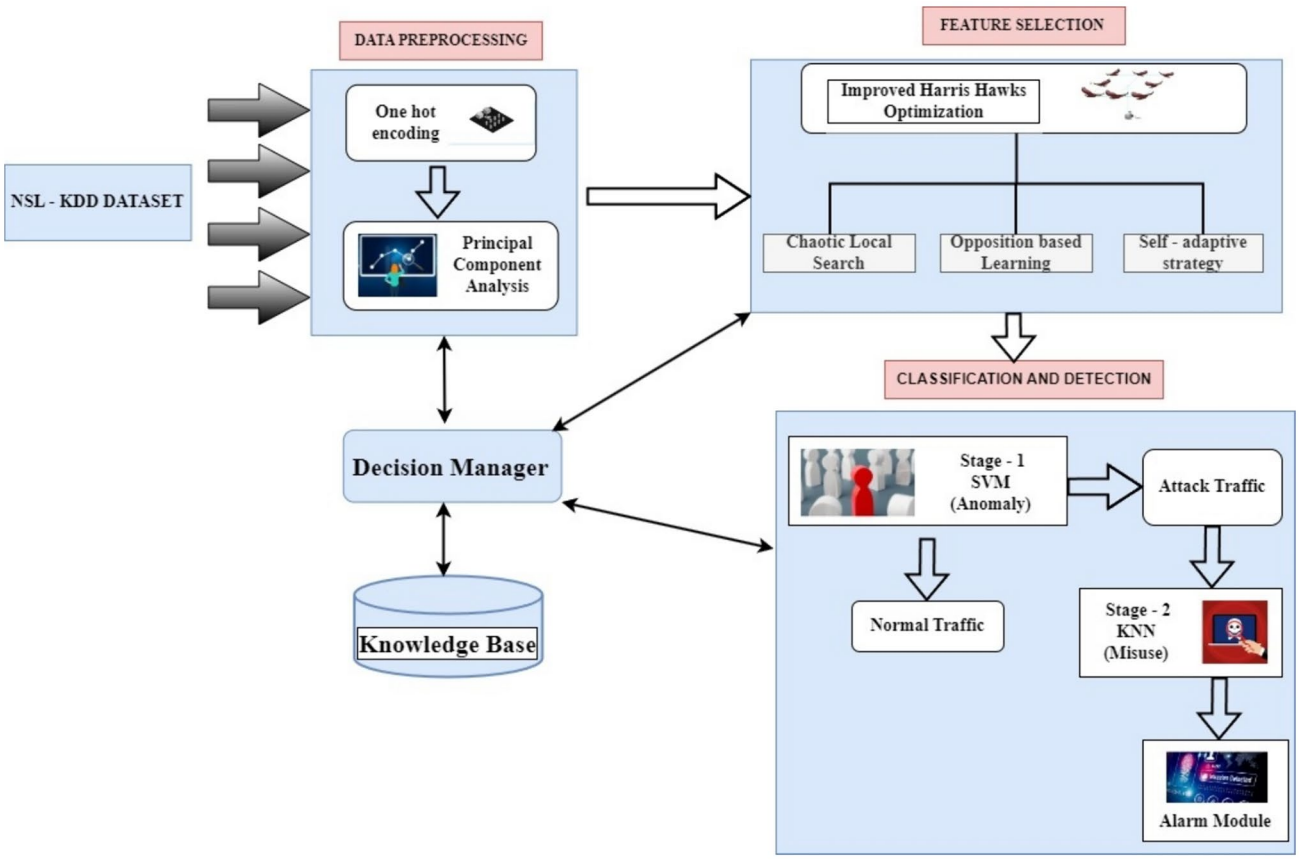

Fig. 1 Proposed System Architecture

supported by the knowledge base for making efficient decisions. The knowledge base is the repository where all the attack patterns are stored.

Figure 2 illustrates the overall flow of the proposed system. It comprises three stages Data Pre-processing, Feature selection, Classification and Detection. In the Data pre-processing stage

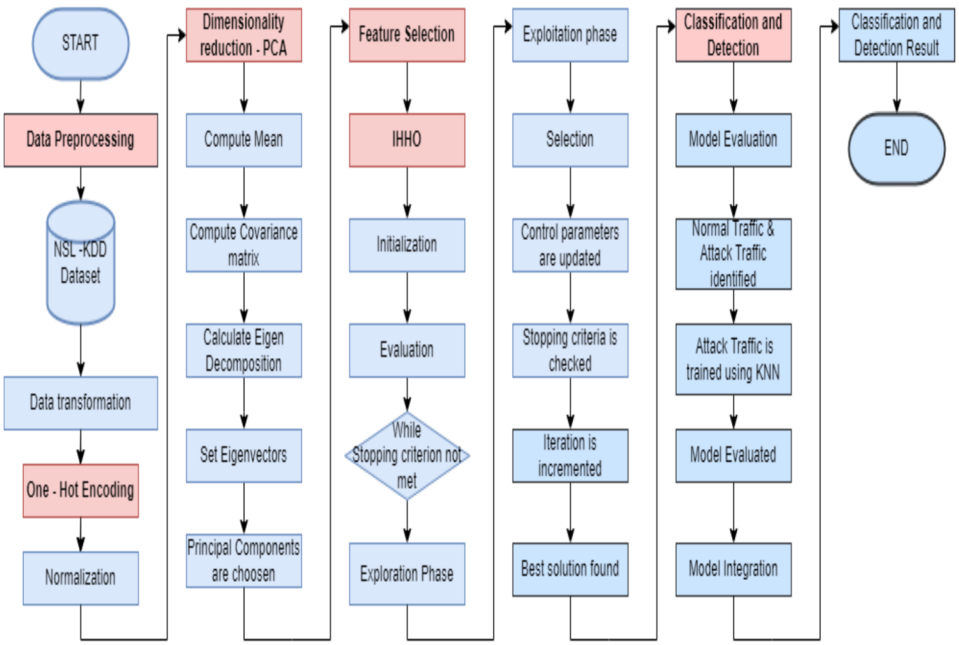

Fig. 2 Flowchart for proposed system

NSL-KDD dataset is pre-processed by utilizing one-hot encoding for categorical features and normalization is performed for numerical features. Further, the features are combined and given as input of PCA. In PCA the dimensionality of the features is reduced by computing mean and covariance matrix. The eigen decomposition is calculated and eigenvectors are set. The principal components are chosen because of dimensionality reduction. The features are optimized in the feature selection phase employing the IHHO algorithm. The random population is initialized and the fitness function is computed. The current position is updated using the exploitation and exploration phase. The test solution is obtained after multiple iterations. In the classification and detection phase, the SVM and KNN model is employed. The SVM model detects normal traffic and the KNN model detects attack traffic. The results of both models are integrated to provide a better classification accuracy. Table 3 gives the list of notations and their descriptions utilized in the proposed system.

5 Proposed Methodology

In this work, an intelligent network-based intrusion detection system has been proposed for detecting anomalies in the NIDS [41]. The proposed system consists of 3 major modules namely the Data Pre-processing module, feature selection module, classification, and detection module.

5.1 Data Pre-processing Techniques

The first module of the proposed system is the data pre-processing module. In the data pre-processing module one-hot encoding is employed to convert categorical data into numerical data. PCA is employed for reducing the dimensionality.

5.1.1 One-hot encoding

The process of converting non-numeric attributes to numeric values is known as feature encoding. In which continual, distinct, and symbolic features are present in the data set. Since most of the ML algorithms are designed in a way to deal with numeric values, they cannot be used with symbolic characteristics. Hence, the encoding strategy is primarily utilized. Categorical data can be encoded in two ways label and one-hot encoding.

One-hot encoding is a widely used technique to handle categorical data. The encoding technique in context is widely preferred due to its efficiency in processing the individual small blocks making it attractive over other techniques. Though the label encoding is simpler it is not preferred as it misinterprets some numeric values because of problems with ordering and this issue is efficiently addressed by one-hot encoding [42]. The label values are changed to a numeric value of 0 or 1 and each categorial value is translated into a new column.

Table 3 List of Notation and their Description

Notations	Description
μ	Mean
Σ	Standard deviation
$C_V\ M$	Covariance Matrix
M	Present iteration
r, q1, q2, q3, and q4	Arbitrary numbers
Y_{rand}	Randomly chosen hawk
Y_{bunny}	Position of the prey
Y_e	Hawks average position
$Y_j(m)$	Hawk position
I_o	First energy state
E	Escaping energy of prey
LF	Levy Flight
M_s	Master solution
TP	Position of target
Y_{best}	Best solution
$Z_{j+1}(m)$	Updated solution
S_R	Jumping rate
v	Vector of weights
y	Input vector
c	Bias
c_i	Respective class
s	Support vector
$\alpha_{m,n}$	Non -negative parameter
$\varphi(y)$	Mapping function
$\varphi(y)\varphi(y_i)$	Numerical optimizations
λ_{nm}	Normal class
λ_{ic}	Intrusion Class
K	Observation length
K	Number of Observations
$F_{1\times M}$	M variables feature set
P	Sample length
L	Window length
ir	Number of iterations
u, v	Two points of Euclidean n-space
u_m–v_m	Euclidean vectors

Further, the encoded data are normalized using a standard scaler which will have a standard deviation and mean value of 1 and 0 respectively.

$$z = \frac{y - \mu}{\sigma} \tag{1}$$

z-score is calculated using Eq. (1). y is the value for which z-score needs to be calculated and μ is the mean, σ is the standard deviation.

5.1.2 Reduction of Dimensionality

One-hot encoding method increases the dimensionality which hampers the speed of the training drastically and it becomes more complex. To overcome this challenge PCA method is

utilized. PCA is an unsupervised machine learning algorithm formally using huge datasets and improving data interpretation by retaining the information. PCA is a statistical method that minimizes the dimensionality of the dataset. To achieve this the data are linearly transformed into a new coordinate system in which a change in the data is expressed using fewer dimensions.

Algorithm 1 Pseudocode for PCA to reduce dimensionality

Input: Dataset with instances and attributes.

Output: reduced dimensional data matrix.

Begin

Step 1: Entire dataset is selected. Let us consider that the feature space consists of K – dimensional samples without any class labels. Matrix of size a × b is assumed and it must be transformed into D dimensional vector.

$$Input\ Data\ a \times b = \begin{pmatrix} Y11 & Y12 & \ldots\ldots\ldots & Y1B \\ Y21 & Y22 & \ldots\ldots\ldots & Y2B \\ . & . & . & . \\ . & . & . & . \\ YA1 & YA2 & \ldots\ldots\ldots & YAB \end{pmatrix} = [Y1, Y2, \ldots\ldots\ldots YD] \ldots\ldots\ldots (2)$$

The matrix $a \times b$ is flattened by listing the elements of the matrix in row wise for generating one-dimensional vector. Dimensional vector D which is equal to the product of number of horizontal rows (a) and number of vertical columns (b) in the matrix. The elements are ordered according to the matrix and the matrix $a \times b$ is transformed into D – dimensional vector using Equation (2).

Step 2: D-dimensional mean vector is computed using given Equation (3):

$$Y_{\text{mean}} = \left(\frac{1}{D}\right) \sum_{J=1}^{D} Y_J \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3)$$

Mean is calculated by summing up all the values (Y_1……Y_D) and dividing the result by total number of values (D). Y_{mean} is the mean of the values Y_1, Y_2, D is the cardinality of the set, $\sum$ is the summation symbol, to add all the Y values.

Step 3: Covariance matrix is computed using given Equation (4):

$$C\vee M = \begin{pmatrix} C\vee M11 & C\vee M12 & \ldots\ldots\ldots & C\vee M1N \\ C\vee M21 & C\vee M22 & \ldots\ldots\ldots & C\vee M2N \\ . & . & . & . \\ . & . & . & . \\ C\vee MD1 & C\vee MD2 & \ldots\ldots\ldots & C\vee MDN \end{pmatrix} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4)$$

Covariance matrix is calculated to interpret the association among the original data and the mean. The $C\vee M$ is arranged in a grid form.

Step 4: Eigenvectors and Eigenvalues is calculated using Equation (5 – 6).

$$(E - \lambda iJ)e_i = 0 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5)$$

e_i eigenvector of matrix E is computed). λi which is the eigenvalue and multiplied with the identity matrix J which results in the zero vector.

$$|E - \lambda J| = 0 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (6)$$

Eigenvalue of matrix E is computed using the matrix determinant $E - \lambda J = 0$

Step 5: Selecting components and formation of feature vector: The resulting eigenvectors are organised in descending order according to eigenvalues. Matrix Z with dimension u × v is formed using l-eigenvectors from sorted eigenvectors. Each column in matrix represents different eigenvector.

Transforming features into principal components:

$$\begin{bmatrix} u11 & u12 & u13 & u14 \\ u21 & u22 & u23 & u24 \\ u31 & u32 & u33 & u34 \\ u41 & u42 & u43 & u44 \end{bmatrix} [v1\ v2\ v3\ v4] = [z1 \quad z2 \quad z3 \quad z4] \ldots\ldots\ldots\ldots\ldots\ldots\ldots (7)$$

In Equation (7) the 4 × 4 matrix components are represented as u_{ij} and it is multiplied with another 4 ×1 matrix eigenvectors. The result obtained is 4 ×1 principal components matrix.

Transforming principal component to features:

$$\begin{bmatrix} u11 & u12 & u13 & u14 \\ u21 & u22 & u23 & u24 \\ u31 & u32 & u33 & u34 \\ u41 & u42 & u43 & u44 \end{bmatrix} [z1 \quad z2 \quad z3 \quad z4] = [v1\ v2\ v3\ v4] \ldots\ldots\ldots\ldots\ldots\ldots\ldots (8)$$

In Equation (8) the 4 × 4 eigenvectors are represented as u_{ij} and it is multiplied with another 4 ×1 principal component. The result obtained is 4 ×1 original features.

Step 6: Principal Components Formation: u × v eigenvector matrix Z is used to create a new subspace from samples.

$$\begin{bmatrix} u11 & u12 & u13 & u14 \\ u21 & u22 & u23 & u24 \end{bmatrix} [v1\ v2\ v3\ v4] = [z1 \quad z2 \quad z3 \quad z4] \ldots\ldots\ldots\ldots\ldots\ldots\ldots (9)$$

In Equation (9) the 4 × 2 matrix components are represented as u_{ij} and it is multiplied with 4 ×1 matrix. The result obtained is 4 ×1 matrix. Using feature extraction technique dimensionality is reduced. Most important principal components are selected which explains about the relationship of the features.

End

In algorithm 1, the first step is to standardize the data in which all mean value is assigned to 0 and standard deviation value is assigned to 1 using the given Eq. (3). In the next step covariance matrix is computed using the given Eq. (4). The covariance matrix's eigenvalues and eigenvectors are determined using the Eq. (5–6). The most significant path in which the data is varied is represented by eigenvectors, and along each eigenvector, the degree of variation is represented by eigenvalues using Eq. (7–8). The highest eigenvalues are considered as principal components. The data which varies the most are selected for transformation using Eq. (9). The high-dimensional original data is transformed into lower-dimensional space [43].

5.2 Feature selection

The process of selecting the most informative features while minimizing the presence of redundant and irrelevant features is known as feature selection. Filter, wrapper, and embedded methods are prominent feature selection techniques. The wrapper technique is forced to employ a swarm intelligence algorithm which improves the performance of the feature selection method. HHO is a swarm-based intelligence optimization technique that produces an ideal solution by initiating the predation approach. Harris hawk is a well-known bird which is known for its unusual cooperative foraging behaviours. The hawks employ a variety of hunting techniques which include trailing surrounding and directly approaching and attacking [44]. The "Surprise pounce" is a skilled hunting technique used by hawks to pursue flying prey. The mathematical model comprises three phases: expedition, transformation between expedition and exploitation, and exploitation.

Even though HHO is simple to use and effective in searching the search space and has high computation speed like all metaheuristic algorithms, it has drawbacks such as settling into local optima and delayed convergence [45]. To overcome these challenges a modification is performed in the HHO algorithm to enhance the performance. Hawks position can be identified by using Eq. (10–11).

$$Y(m+1) = \left\{ Y_{rand}(m) - q1 \left| Y_{rand}(m) - 2q2Y(m) \right| r \geq 0.5 \right. \quad (10)$$

$Y(m+1)$ is the position of subsequent iteration and $Y(m)$ is the position of the present iteration. m is the present iteration number. The position of subsequent iteration $Y(m+1)$ is calculated using $Y_{rand}(m)$ which is randomly chosen hawk, $q1$ and $q2$ are the random values which lies between [0,1]. r is employed to choose the strategy randomly. If the r is greater than or equal to 0.5 then set $Y(m+1) = Y_{rand}(m) - z\left|Y_{rand}(m) - 2q2(m)\right|$.

$$Y(m+1) = \left\{ \left(Y_{bunny}(m) - Y_e(m) \right) - q3(BB + q4(TB - BB))r < 0.5 \right. \tag{11}$$

If r is less than 0.5 then the position of subsequent iteration $Y(m+1)$ is computed using $Y_{bunny}(m)$ which is the target position, $Y_e(m)$ is the mean location of all the individuals in the iteration m. $q3, q4$ are arbitrary numbers in the interval [0,1]. *BB* and *TB* refers to the position of bottom bound and top bound features. The subsequent iteration is identified using Eq. (10–11).

$$Y_e(m) = \frac{1}{M} \sum_{O}^{M} Y_j(m) \tag{12}$$

$Y_e(m)$ is the average position which is computed using Eq. (12). M is the maximum iteration count, $Y_j(m)$ is the individual hawk position. The average position $Y_e(m)$ is calculated by adding up all the values from (j = 0 to j = M) and it is divided by the sum of values M. The next phase is transforming from expedition to exploitation.

$$E = 2I_o\left(1 - \frac{m}{M}\right) \tag{13}$$

The prey energy calculated using Eq. (13) where I_o is the first energy state and E is the escaping energy of the prey. I_o ranges between [-1,1]. m refers to present iteration number and M is the maximum iteration count. The prey energy is computed by dividing present iteration number by maximum iteration count and reducing it by 1 and further it is multiplied twice the first energy state.

Soft Assault: when |E| and q ≥ 0.5. It is defined by following Eq. (14–15).

$$Y(m+1) = \Delta Y(m) - E\left|J\ Y_{bunny}(m) - Y(m)\right| \tag{14}$$

The soft assault technique to identify the position of the subsequent iteration $Y(m+1)$ is computed using Eq. (14) where $\Delta Y(m)$ is the current locations distance is multiplied with random number J which lies between [0,2] and minuses from the position of present iteration and multiplied with E which refers to prey's energy. Further, it is subtracted from $\Delta Y(m)$ which is the current location distance from the prey position and the position of subsequent iteration is found.

$$\Delta Y(m) = Y_{bunny}(m) - Y(m) \tag{15}$$

In Eq. (15), current locations distance from the prey's position is determined by subtracting $Y_{bunny}(m)$ which is the target position by $Y(m)$ which is the subsequent iterations position.

Soft assault with quick dive: when q < 0.5 and |E| ≥ 0.5. The prey has necessary energy to flee, mathematical pattern of levy flight (LF) is described using given Eq. (16):

$$LF(y) = \frac{c \times \sigma}{|u|} \times 0.01 \tag{16}$$

c and u are random values between (0,1) and σ is a default constant. $LF(y)$ is equal to the product of $c \times \sigma$ and it is divided by $|u|$ and then it is multiplied by 0.01.

So,

$$Y(m+1) = \left\{ \left(Y_{bunny}(m) - E\left|J\ Y_{bunny}(m) - Y(m)\right|, W = X + R \times LF(D) \right. \right. \tag{17}$$

The position of the subsequent iteration $Y(m+1)$ for soft assault with quick dive scenario is computed using Eq. (17). The target position $Y_{bunny}(m)$ is subtracted by prey energy E. $Y_{bunny}(m) - Y(m)$ where present iteration number is subtracted by position of present iteration and multiplied with random variable J and Y_{bunny} which is the target position.

Hard assault with quick dive: when |E| and q ≤ 0.5. The prey lacks the energy necessary to flee. It is defined by following Eq. (18):

So,

$$Y(m+1) = \left\{ \left(Y_{bunny}(m) - E\left|J\ Y_{bunny}(m) - Y_e(m)\right|.., W = X + R \times LF(D) \right. \right. \tag{18}$$

The position of the subsequent iteration $Y(m+1)$ for hard assault with quick dive scenario is computed using Eq. (18). The target position $Y_{bunny}(m)$ is subtracted by prey energy E. $Y_{bunny}(m) - Y_e(m)$ where present iteration number is subtracted by average position and multiplied with random variable J and Y_{bunny} which is the target position.

Hard assault: when q ≥ 0.5 and |E| < 0.5. The behaviour is defined using Eq. (19):

$$Y(m+1) = Y_{bunny}(m) - E|\Delta Y(m)| \tag{19}$$

In hard assault scenario, the $Y(m+1)$ is the position of subsequent iteration is computed using Eq. (19) where current location distance $\Delta Y(m)$ is multiplied with energy E and subtracted from the target position $Y_{bunny}(m)$. Opposition based learning is utilized to compare the fitness of an individual with its equivalent reverse number so that the best one is taken into consideration.

$$\overline{y} = tb + bb - y \tag{20}$$

The top bound (tb) and bottom bound (bb) value are added and subtracted from a real number y to obtain the reverse number $\bar{y}$ using Eq. (20).

$$\bar{y}_j = tb_j + bb_j - y_j \tag{21}$$

The current solution value is assigned to $\bar{y}_j$. The top bound (tb) and bottom bound (bb) value are added and subtracted from a real number $\bar{y}_j$ to obtain the reverse number $\bar{y}_j$ using Eq. (21).

CLS: chaos is a phenomenon that appears to be random but occurs in non – linear and deterministic systems. Chaotic sequence is generated using logistic map [43].

$$h^{o+1} = Mh^o(1 - h^o) \tag{22}$$

$h^o$is the random value [0,1]. M is the chaotic sequence. The features of chaotic system are considered to create a search operator and it is combined with meta heuristic algorithm; the solution produced by CLS is obtained by Eq. (23):

$$M_s = (1 - \mu) \times TP + \mu M_j \tag{23}$$

M_s is the master solution; TP is the position of target. The master solution is computed using Eq. (23) in which μ is the random variable subtracted by 1 and multiplied by position of the target TP and added with the chaotic sequence.

$$\mu = \frac{MaximumIteration - presentIteration}{MaxIteration} + 1 \tag{24}$$

The μ value for obtaining master solution is calculated using Eq. (24) where maximum iteration is subtracted by present iteration and divided by maximum iteration and added by 1.

$$\overline{M}_j = BB + MJ \times (TB - BB) - 1 \tag{25}$$

The reverse of chaotic sequence is computed by using Eq. (25) where top bound TB is subtracted by bottom bound BB and multiplied with chaotic sequence and added with bottom bound value and further it is subtracted by 1.

$$Z_j(m+1) = Y_j(m+1) + S_R\left(Y_{best} - Y_j\right)(m) \tag{26}$$

Equation (26) is used to achieve the updated solution $Z_j(m+1)$, the social component $Y_j(m+1)$ is added with the cognitive component $S_R.\left(Y_{best} - Y_j\right)(m)$ .Y_{best} is the best solution and S_R is the jumping rate. The $Y_j(m+1)$ is the output of HHO algorithm which enhances the ability to exploit regions surrounding the optimal solutions. The cognitive component $S_R\left(Y_{best} - Y_j\right)(m)$ is incorporated as a local search operator. In algorithm 2 pseudocode for Improved Harris Hawks Optimization is given.

Algorithm 2 Pseudocode for Improved Harris Hawks Optimization Algorithm

Input: Population _Size, max_generations.
Output: Best_Solution
Begin
Step 1: Generate the parameters (Population size (P), Maximum Iteration (M), TB, BB, and Dimension)
Step 2: Random population is initialized
Step 3: For each hawk Yp fitness function is computed
Step 4: Compute $\bar{Y}$
Step 5: $Y \cup \bar{Y}$ determines the solution
Step 6: The current positions are updated as best position $[y_s, best]_{j=1}^{p}$
While (iteration ≤ MaximumIteration) do
For each hawk y_s fitness function is computed
Y_{bunny} – Finest hunt agent
For each hawk (y_s) do
Step 7: Initial energy is updated, jump strength is updated using equation (13)
$$E = 2I_0\left(1 - \frac{m}{M}\right) \ldots\ldots\ldots\ldots\ldots\ldots (13)$$
Step 8: Utilize operator to enhance exploration and exploitation.
if(|E|≥1) then
Hawk position is updated using equation (11):
$$Y(m+1) = \{(Y_{bunny}(m) - Y_e(m)) - q3(BB + q4(TB - BB))$$
$$r < 0.5 \ldots\ldots\ldots\ldots (11)$$
end if
Step 9: Recompute the fitness criteria.
if(|E|≤1) then
if (q ≥0.5 and |E|≥0.5) then
Update position of hawk using equation (14):
$$Y(m+1) = \Delta Y(m) - E\left|J\, Y_{bunny}(m) - Y(m)\right| \ldots\ldots\ldots\ldots (14)$$
elseif (q≥0.5 and |E|<0.5) then
Update position of hawk using equation (19):
$$Y(m+1) = Y_{bunny}(m) - E\left|\Delta Y(m)\right| \ldots\ldots\ldots\ldots\ldots (19)$$
elseif (q<0.5 and |E|≥0.5) then
Update position of hawk using equation (17):
$$Y(m+1) = \{(Y_{bunny}(m) - E\left|J\, Y_{bunny}(m) - Y(m)\right|,$$
$$W = X + R \times LF(D) \ldots\ldots\ldots\ldots\ldots (17)$$
else
Update position of hawk using equation (18):
$$Y(m+1) = \{(Y_{bunny}(m) - E\left|J\, Y_{bunny}(m) - Ye(m)\right|$$
$$W = X + R \times LF(D) \ldots\ldots\ldots\ldots\ldots (18)$$
end if
end if
Update position of hawk using equation (26):
$$Z_j(m+1) = Y_j(m+1) + S_R(Y_{best} - Y_j)\ (m) \ldots\ldots\ldots\ldots\ldots\ldots (26)$$
Step 10: Determine the performance of a solution.
if (random < output) then
Compute $\bar{y}_{j+1}$ and its fitness
$\bar{y}_{j+1} - \bar{y}_{j+1}$ if $f(\bar{y}_{j+1}) < f(y_{j+1})$
end if
end for
Update Y_{bunny}
Step 11: Consider focusing on local optimization, guided searches, or alternative methods of exploitation.
Local search is performed using equations (22,23,24,25):
$h^{o+1} = Mh^o(1 - h^o) \ldots\ldots\ldots\ldots\ldots\ldots (22)$
$M_s = (1 - \mu) \times TP + \mu M_j \ldots\ldots\ldots\ldots\ldots\ldots (23)$
$$\mu = \frac{MaximumIteration - presentIteration}{MaxIteration} + 1 \ldots\ldots\ldots\ldots\ldots\ldots (24)$$
$\overline{M}_j = BB + Mj \times (TB - BB) - 1 \ldots\ldots\ldots\ldots\ldots\ldots (25)$
End
Step 12: Provide the optimal solution identified.
Return Y_{bunny}
End

5.3 Classification and detection

In this paper, a two-staged classifier for network intrusion detection which employs SVM as an anomaly detection at stage-1 and KNN as a misuse detection at stage-2 is proposed. The NSL-KDD dataset with 41 features is considered for experimenting with the dominance of the proposed system. Later, 10 prominent features are selected and analysed to compare the classification accuracy, detection rate F-measure, and false alarm rate. Network traffic is a combination of attack and normal traffic that flows through stage-1(SVM) which distinguishes normal and attack classes. Stage 2(KNN) compromises attack traffic which is further classified into DOS, probe, U2R, and R2L attacks. The two-staged classifier minimizes computing complexity while employing selected 10 features, resulting in greater accuracy with reduced false alarm rate.

5.3.1 Stage-1 Anomaly (SVM)

The multiclass—SVM (Stage-1) anomaly classifier was first modelled using the radial basis kernel function on the training set which consists of both attack and normal traffic. The test datasets with unknown normal and attack are used to validate the anomaly module. SVM [46] is generally used to solve two-class classification issues. A hyperplane or linear line is built as a decision boundary between two classes of datasets for classification. Support vectors are the data points closest to the hyperplane that contribute to its formation. The hyperplane is expressed as:

$$v^w y + c = 0 \tag{27}$$

$v^w$is the vector of weights; y is an input vector and c are the bias. The hyperplane value is set to 0 in Eq. (27).

$$v^w y + c = +1 \; for \; c_i = +1 \tag{28}$$

$$v^w y + c = -1 \; for \; c_i = +1 \tag{29}$$

Based on the respective classes, values of the hyperplane are represented as -1 and + 1 in the Eq. (28–29). c_i is the respective class, $c_i = +1$ for class A,$c_i = -1$ for class B.

$$min\phi(v) = \frac{1}{2} v^w v \tag{30}$$

The quadratic form $\phi(v)$ is minimized by using vector v and vector weightv^w in the Eq. (30).

The final output function:

$$f(y) = sign\left(\sum_{i=1}^{s} a_{m,n} \left(y^w . y_1 \right) + c \right) \tag{31}$$

In the Eq. (31) function of input vector y which need to be classified is termed as$f(y)$, s is the support vector, $\alpha_{m,n}$ is the non -negative parameter which is used to differentiate support vector among input vector, y^w is the vector weight of y and y_i is the respective class of y,c is bias. The modified output function is:

$$f(y) = sign\left(\sum_{i=1}^{s} \alpha_{m,n} \left(\varphi(y) \varphi\left(y_i \right) \right) + c \right) \tag{32}$$

The modified output function for$f(y)$, is computed using Eq. (32) where $\alpha_{m,n}$ is the non -negative parameter, s is the support vector, $\varphi(y)$ mapping function of vector y and $\varphi(y_i)$ is the respective class of vector y which is used to convert linearly non separable pattern into higher dimensional feature space, c is bias.

$$f(y) = sign\left(\sum_{i=1}^{s} a_{m,n} K\left(y, y_i \right) + c \right) \tag{33}$$

Further the numerical optimization complexity of $\varphi(y)\varphi(y_i)$ is reduced using Eq. (33). The vector y,vectors representative class y_i ,$\alpha_{m,n}$ is the non -negative parameter, s is the support vector and bias is c are computed to reduce the optimization complexity.

For classifying non-linear patterns SVM employs several kernel functions which includes linear, sigmoid, polynomial, and radial basis function. In this paper three functions of kernel are utilized. The method creates k different classifiers for k-class classification. In k^ht classifier the data which belongs to k^ht class are considered as true values whereas the k-1 classes are considered as false values.

Algorithm 3 is the classifiers training phase λ_{nm} is the normal class, λ_{ic} is the intrusion class, k is the sample length, K is the number of samples, F_{IXM} is the feature set of M variables are the parameters of the training phase. Kernel scale, kernel function and cross-validation techniques are given as input for the training phase. model_svm is the output of the trained model. λ_{nm} is generated with Poisson distribution using K signals of k-dimensions. The extracted features from each sample are termed as normal class. λ_{ic} is generated with Poisson distribution using K signals of k-dimensions. The extracted features from each sample are termed as intrusion class. From the observation labels and vectors are integrated vertically and classifier is trained.

Algorithm 3 Pseudocode for Training phase of SVM

Input: Normal class: λ_{nm}, Intrusion Class: λ_{ic}, Observation length: k, Number of Observations: K, M variables feature set: $F_{l \times M}$, Cross_validation, Kernel_function, Kernel_scale

Output: Model_svm: Classification module trained

Begin

Step 1: Training data is loaded.

Step 2: Using poisson distribution parameter λ_{nm} generate K signals of k-dimensions $\longrightarrow Y_{A\times b}^{nm}$

Step 3: Features are extracted from $Y_{A\times b}^{nm} \longrightarrow Y_{A\times M}^{Fnm}$

Step 4: Initialize a SVM classifier utilizing kernel.

Step 5: Normal class is labelled $\longrightarrow X_{A\times 1}^{nm}$

Step 6: Using poisson distribution parameter λ_{ic} generate K signals of k-dimensions $\longrightarrow Y_{A\times b}^{ic}$

Step 7: Features are extracted from $Y_{A\times b}^{ic} \longrightarrow Y_{A\times b}^{Fic}$

Step 8: Train the SVM model utilizing the training data.

Step 9: Execute predictions using the trained SVM classifier.

Step 10: Intrusion class is labelled $\longrightarrow Y_{A\times 1}^{ic}$

Step 11: Two vectors are vertically integrated $X_{2A\times 1} \longrightarrow Y_{A\times 1}^{nm}$

Step 12: SVM model is training using cross_validation, kernel function with observation $X_{2A\times M}$ and $Y_{2A\times M} \longrightarrow$ Model_svm.

End

Algorithm 4 is the classifier testing phase λ_{nm} is the normal class, λ_{ic} is the intrusion class, P is the test signal length, windows length is l and F_{lXM} is the feature set of M variables are the parameters of the testing phase. model_svm is given as the input to the testing phase. Random number x is generated between P and l. Poisson distribution parameters λ_{nm} generates x-dimensional normal signal. Similarly, the Poisson distribution parameter λ_{ic} generates (p-x) dimensional intrusion signal. From the observation vectors and labels are integrated horizontally to achieve a single P dimensions signal. For the first element windows length L is extracted from P dimensions signal and given as input to the classifier for classification output is stored in z vector [47]. Similarly, for all (P-L+1) element signals are tested and the output is stored in z vector.

Algorithm 4 Pseudocode for Testing phase of SVM

Input: Normal class: λ_{nm}, Intrusion Class: λ_{ic}, Sample length: P, window length: l, M variables feature set: $F_{1 \times M}$, ir – number of iterations

Output: Label predicted: $X_{ir} \times (P - p+1)$

Begin

Step 1: Test the SVM model using training data

Step 2: for r in ir do

Step 2.1: Features extracted from data samples

Step 2.2: Class label is predicted using SVM model.

Step 2.3: True class label is identified

Step 2.4: Predictions should match with class label.

Step 3: y ⟶ random number $\leq P$

Step 4: $Y^{nm}_{1\times(y-1)}$ ⟶ Using poisson distribution parameter λ_{nm} generate (y-1)

Step 5: $Y^{ic}_{1\times(P-y+1)}$ ⟶ Using poisson distribution parameter λ_{ic} generate (P-y+1)

Step 6: $Y_{1\times P}$ ⟶ $[X^{nm}_{1\times(y-1)}, Y^{ic}_{1\times(P-y+1)}]$ ⟶ Integrate vectors horizontally

Step 7: For i in P-l+1 do

Step 8: Y^{O} ⟶ $Y(i{:}\, i+m)$

Step 9: Y^{OF} ⟶ Features are extracted

Step 10: $X_{r,i}$ ⟶ model_svm (Y^{OF}), Test Y^{OF} using trained model_svm.

Step 11: Determine the test data model accuracy.

end for

End

5.3.2 Stage -2 Misuse (KNN)

KNN Classifier is employed for misuse detection the attack traffic from stage-1 is analysed in stage-2 classifier and it is further classified into 4 classes: DoS, R2L, U2R and Probe. KNN is a supervised, non-parametric ML technique for sample categorization and regression. KNN is based upon the similarity between existing data and the new data. It keeps the data while working in the training phase and, when the new dataset appears, it categorises the new data in a category that is most comparable to the previously existing dataset category [48]. The test or validation datasets k parameter displays the set of cases that are closest to a certain set of cases. Algorithm 5 explains the KNN classifier for misuse detection.

Algorithm 5 Pseudocode for KNN classifier for misuse detection

Input: Training and testing data.

Output: Identified attacks.

Step 1: initialize the parameter fitness (fness) =0

selected_features = {f_1,f_2,f_3,f_4 ……f_{10}}

total_no_of_features = {f_1,f_2,f_3 ……f_{41}}

Step 2: Training and testing data are loaded.

Step 3: Train the KNN classifier with the provided training data.

train = select f_{set} from the features randomly

Step 4: Trained KNN model is saved and utilized to test data.

test = select same features selected for training and test

for m=1 to k_test

for n=1 to k_train

Step 4.1: Euclidian distance between y_m and y_n are calculated using given equation (34):

$$d(u,v) = (\sum_{m=1}^{n}(u_m - v_m)^2)^2 \quad \ldots\ldots\ldots\ldots\ldots\ldots (34)$$

End for

Step 4.2: k_train is sorted in ascending order based upon distance

Step 4.3: top 3 neighbors are chosen and identify their class

Step 4.4: Determine the class of the test point by considering the predominant class in the selected points.

if (class of y_m = major class of y_n)

count =1

else

count =0

endif

fness = count+fness

end for

return fness

Step 5: Determine the result using the KNN model.

Step 6: Analyse the model's performance.

Table 4 Dataset Details

Traffic	Training	Testing
Attack	58630	12833
Normal	67343	9711
Total	125973	22544

Table 5 Selected Features

Type of Attacks	Selected Features
DoS	41,40,27,23,13,12,10,6,5
Probe	41,40,33,31,28,27,6,5,3,2
R2L	41,40,28,25,24,22,13,7,6,5
U2R	41,40,33,28,27,25,15,6,5,3

Consider a set of observation and targets, where observation $u_m \epsilon R^d$and targets $v_m \in \{0,1\}$ .Among the training samples, neighbors of a test sequence are rated by KNN and the nearest neighbors class label is utilized to identify the test class. As a result, KNN classifies the new points based on the k-nearest points in the training data that has majority of votes. The Euclidean distance is frequently employed in KNN as the distance metric to assess the similarity of two vectors [49]. The number of neighbors in a set of training observations that are closest to an observation in a validation or testing data set is represented by the k parameter of KNN classifiers. This classifier can be used to address multiclass problems unlike SVM. Since, SVM and KNN are similar they are merged to achieve better accuracy. Once the attacks are identified by the classification module, the decision manager sends a notification to the alarm manager. The alarm manager sends the alarm signals to the network administrator that attack has been detected in network.

6 Performance Evaluation

The performance of the proposed IDS is analysed using NSL-KDD dataset. The experiment is implemented in Python using the operating system Windows 10, the processor is AMD Ryzen 5 3500U with Radeon Vega Mobile GFX 2.10 GHz, the RAM is 8 GB and the software platform is Jupyter Notebook (Anaconda3). NSL-KDD dataset detects malicious traffic and classifies the attacks. It is the updated version of the KDD dataset which has no duplicate records in the training set. NSL-KDD dataset consists of 125,973 training instances. Among them 52 are U2R attacks, 995 are R2L attacks, 11,656 are probe attacks, 67,343 are normal and 45,927 are DoS attacks. NSL-KDD dataset consists of 22,544 testing instances. Among them 200 are U2R attacks, 2756 are R2L attacks, 242 are probe attacks, 7456 are DoS attacks and 9711 are normal. NSL-KDD dataset consists of 41 features among them 10 features are selected for performance evaluation. NSL-KDD is the standard dataset for IDS which is significantly utilized in Intrusion detection and ML. The dataset details are provided in Table 4. The selected 10 features for each attack are shown in Table 5.

Accuracy, Detection rate, F-measure, False Alarm Rate (FAR), Precision, and Recall are performance metrics that are employed to assess the performance of the proposed system. Accuracy is a prominent performance metric. Additionally, the confusion matrix is also used as a performance indicator. In the confusion matrix [50], TP represents the proportion of attack records that were correctly classified as such, TN represents the proportion of normal records that were similarly correctly classified, FP represents the proportion of normal records that were incorrectly classified as attacks, and FN represents the proportion of attack records that were similarly incorrectly classified as normal. The confusion matrix for existing classifiers and the proposed classifier is shown in Table 6, 7, 8, 9, 10. The following can be used to compute the performance metrics:

Accuracy: Accuracy is the proportion of records that were correctly identified among all records. Accuracy is computed using Eq. (35):

Table 6 Confusion Matrix for Existing classifier SVM + NB

Attacks	TP	TN	FP	FN
DoS	43321	10312	0	0
Probe	6651	4613	1	0
R2L	1200	762	0	0
U2R	81	17	3	1
Normal	51382	21016	176	2

Table 7 Confusion Matrix for Existing classifier SVM + RF

Attacks	TP	TN	FP	FN
DoS	38421	13467	1	0
Probe	5541	3142	0	0
R2L	1600	814	134	8
U2R	68	79	81	0
Normal	41657	19657	154	0

Table 8 Confusion Matrix for Existing classifier KNN + DT

Attacks	TP	TN	FP	FN
DoS	37617	5751	0	0
Probe	5178	4136	0	0
R2L	1341	567	18	1
U2R	75	85	132	8
Normal	56178	17854	175	0

Table 9 Confusion Matrix for Existing classifier KNN + LR

Attacks	TP	TN	FP	FN
DoS	35812	8856	0	0
Probe	5651	5651	0	1
R2L	1285	1100	27	0
U2R	85	95	0	0
Normal	61852	9651	158	2

Table 10 Confusion Matrix for Proposed classifier KNN + SVM

Attacks	TP	TN	FP	FN
DoS	45084	8016	0	0
Probe	7814	4013	0	0
R2L	2189	1200	0	0
U2R	152	100	1	1
Normal	64854	10,381	121	0

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{35}$$

Precision: Precision is the proportion of accurately identified attack records among all attack records that have been identified. Precision is computed using Eq. (36):

$$Precision = \frac{TP}{TP + FP} \tag{36}$$

Recall: Recall is the proportion of accurately identified attack records among all attack records. Recall is computed using Eq. (37):

$$Recall = \frac{TP}{TP + FN} \tag{37}$$

F – Measure: Harmonic mean of Precision and Recall. F-Measure is computed using Eq. (38):

$$F - Measure = \frac{2(Recall \times Precision)}{Recall + Precison} \tag{38}$$

Figures 3–4 shows the detection rate of the proposed system with selected 10 features and 41 features when compared with other existing classifiers. From the graph analysis, it is understood that the proposed classifier has a better detection rate with selected 10 features. The proposed classifier detection rate is enhanced as it employs an IHHO for effective feature selection.

A comparison of the detection rate with selected 10 features and 41 features are given in Table 11 is are carried out for existing classifiers. The proposed classifiers detect DoS attacks with a detection rate of 98.75% for selected 10 features and 98.04% for 41 features. Whereas for probe attack the detection rate is 90.31% for selected 10 features and 89.38% for 41 features. For R2L attack the detection rate is 98.99% for selected 10 features and 96.72% for 41 features. And, for U2R attack the detection rate is 79.96% for selected 10 features and 74.21% for 41 features.

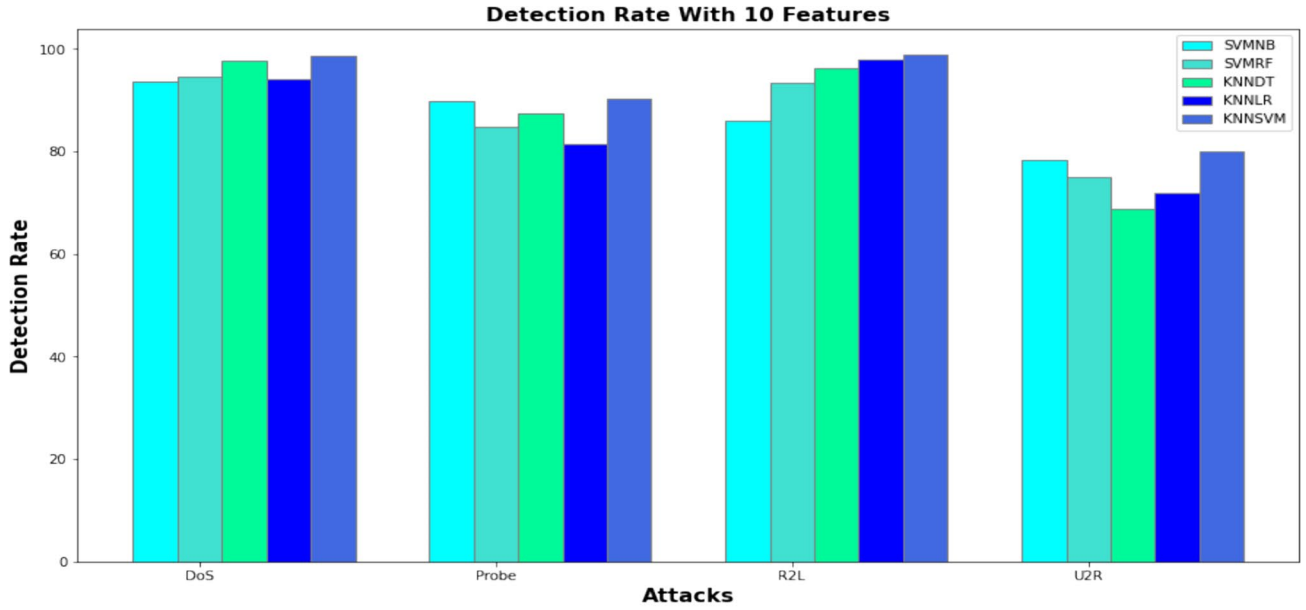

Fig. 3 Detection Rate for 10 features

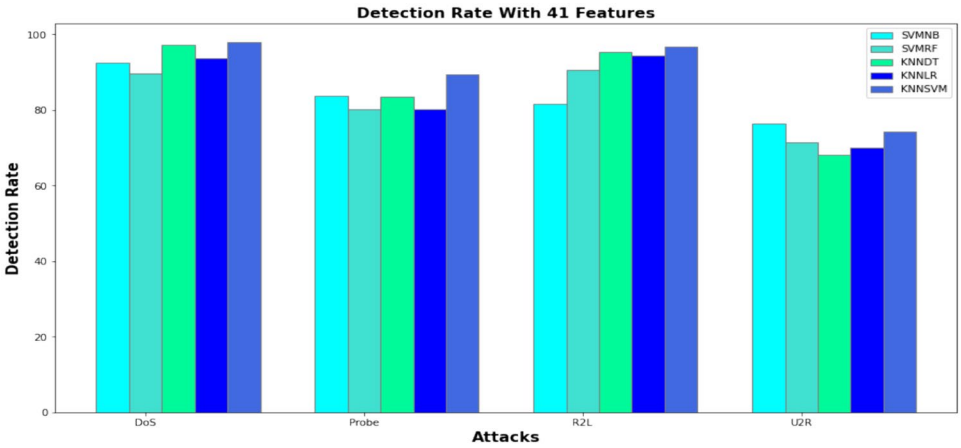

Fig. 4 Detection Rate for 41 features

Table 11 Comparison of attacks detection rate with different classifiers

Attacks	Classifiers	Detection rate	
		With 10 features	With 41 features
DoS	**SVM+NB**	93.65	92.48
	SVM+RF	94.56	89.76
	KNN+DT	97.78	97.31
	KNN+LR	94.19	93.61
	KNN+SVM	98.75	98.04
Probe	**SVM+NB**	89.90	83.84
	SVM+RF	84.78	80.23
	KNN+DT	87.42	83.62
	KNN+LR	81.52	80.31
	KNN+SVM	90.31	89.38
R2L	**SVM+NB**	85.90	81.65
	SVM+RF	93.45	90.65
	KNN+DT	96.34	95.43
	KNN+LR	97.86	94.32
	KNN+SVM	98.99	96.72
U2R	**SVM+NB**	78.24	76.33
	SVM+RF	74.98	71.34
	KNN+DT	68.76	68.04
	KNN+LR	71.92	70.12
	KNN+SVM	79.96	74.21

Figures 5–6 shows the false alarm rate of the proposed system with selected 10 features and 41 features when compared with other existing classifiers. From the graph analysis, it is understood that the proposed classifier has reduced the false alarm rate with the selected 10 features. The proposed classifier has reduced FAR since it employs two-staged classifier which enhances the probability of detecting attacks effectively.

A comparison of false alarm rate with selected 10 features and 41 features are given in Table 12 with different classifiers. DoS attack, Probe attack, R2L attack, and U2R attack analysis are carried out for existing classifiers. The proposed classifiers detect DoS attacks with a false alarm rate of 0.01% for selected 10 features and 0.10% for 41 features. Whereas for probe attack the false alarm rate is 0.01% for selected 10 features and 0.02% for 41 features. For R2L attack the false alarm rate is 0.01% for selected 10 features and 0.03% for 41 features. And, for U2R attack the false alarm rate is 0.02% for selected 10 features and 0.06% for 41 features.

Figures 7–8 shows the classification accuracy of the proposed system with selected 10 features and 41 features when compared with other existing classifiers. For classification two machine learning algorithms namely SVM and KNN is employed. SVM detects the anomaly whereas KNN detects whether the malicious attack is still present. From the graph analysis, it is understood that the proposed

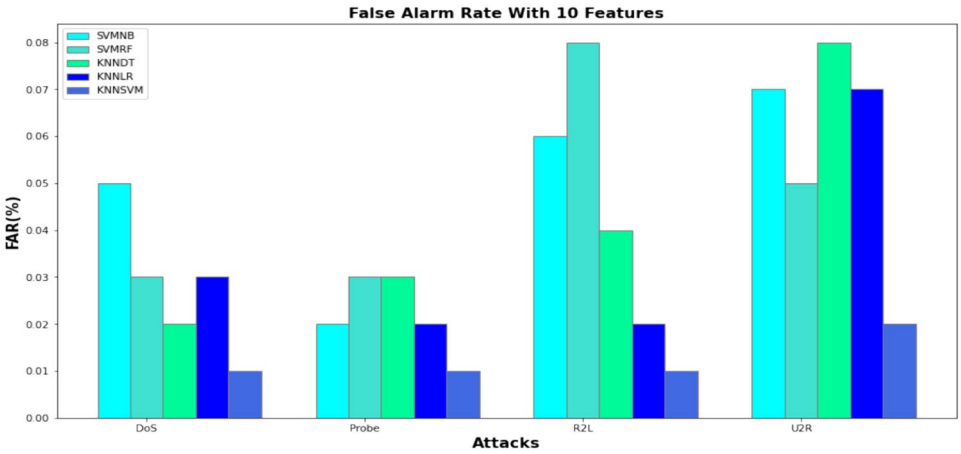

Fig. 5 False Alarm Rate for 10 features

classifier has achieved better classification accuracy with the selected 10 features.

A comparison of classification accuracy with selected 10 features and 41 features are given in Table 13 with different classifiers. DoS attack, Probe attack, R2L attack, and U2R attack analysis are carried out for existing classifiers. The proposed classifiers detect DoS attacks with a classification accuracy of 92.38% for selected 10 features and 90.68% for 41 features. Whereas for probe attack the classification accuracy is 96.90% for selected 10 features

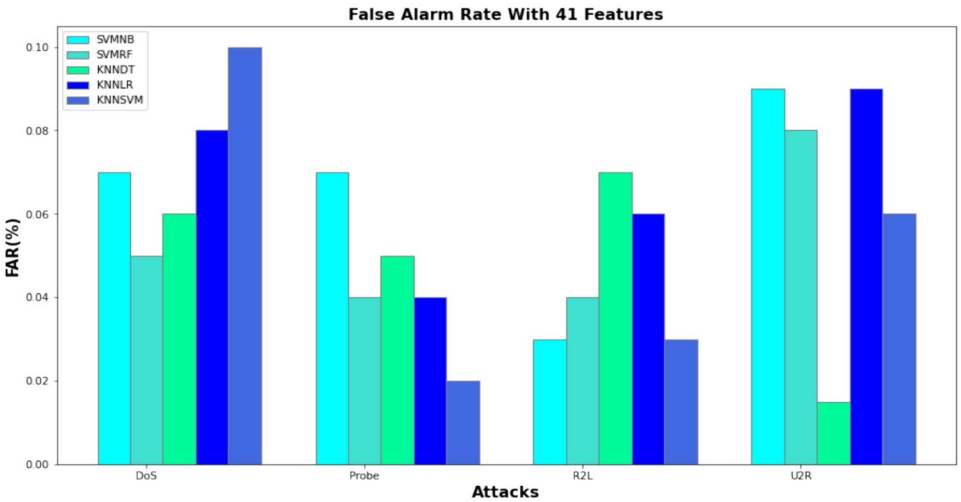

Fig. 6 False Alarm Rate for 41 features

Table 12 Comparison of attacks False alarm rate with different classifiers

Attacks	Classifiers	Detection rate	
		With 10 features	With 41 features
DoS	**SVM+NB**	0.05	0.07
	SVM+RF	0.03	0.05
	KNN+DT	0.02	0.06
	KNN+LR	0.03	0.08
	KNN+SVM	0.01	0.10
Probe	**SVM+NB**	0.02	0.07
	SVM+RF	0.03	0.04
	KNN+DT	0.03	0.05
	KNN+LR	0.02	0.04
	KNN+SVM	0.01	0.02
R2L	**SVM+NB**	0.06	0.03
	SVM+RF	0.08	0.04
	KNN+DT	0.04	0.07
	KNN+LR	0.02	0.06
	KNN+SVM	0.01	0.03
U2R	**SVM+NB**	0.07	0.09
	SVM+RF	0.05	0.08
	KNN+DT	0.08	0.015
	KNN+LR	0.07	0.09
	KNN+SVM	0.02	0.06

and 93.06% for 41 features. For R2L attack the classification accuracy is 96.06% for selected 10 features and 93.06% for 41 features. And, for the U2R attack, the classification accuracy is 94.73% for selected 10 features and 91.83% for 41 features.

Figures 9–10 shows the precision of the proposed system with selected 10 features and 41 features when compared with other existing classifiers. Precision indicates the accuracy of the proposed system in detecting normal and anomaly attacks effectively. Increased precision value ensures that the false positive rate has been minimized and an effective IDS is proposed. From the graph analysis, it is understood that the proposed classifier has achieved better precision with the selected 10 features.

Comparison of Precision with selected 10 features and 41 features are given in Table 14 with different classifiers. DoS attack, Probe attack, R2L attack, and U2R attack analysis are carried out for existing classifiers. The proposed classifiers detect DoS attacks with a precision of 0.92% for selected 10 features and 0.89% for 41 features. Whereas for probe attack the precision is 0.91% for selected 10 features and 0.87% for 41 features. For R2L attack the precision is 0.93% for selected 10 features and 0.90% for 41 features. And, for U2R attack the precision is 0.90% for selected 10 features and 0.88% for 41 features.

Figures 11–12 shows the recall of the proposed system with selected 10 features and 41 features when compared with other existing classifiers. Recall identifies the actual threat in the proposed system effectively. Increased recall value ensures that the false negative rate has been minimized and an effective IDS is proposed. From the graph analysis, it is understood that the proposed classifier has achieved better recall with selected 10 features.

A comparison of Recall with selected 10 features and 41 features are given in Table 15 with different classifiers. DoS attack, Probe attack, R2L attack, and U2R attack analysis are carried out for existing classifiers. The proposed classifiers

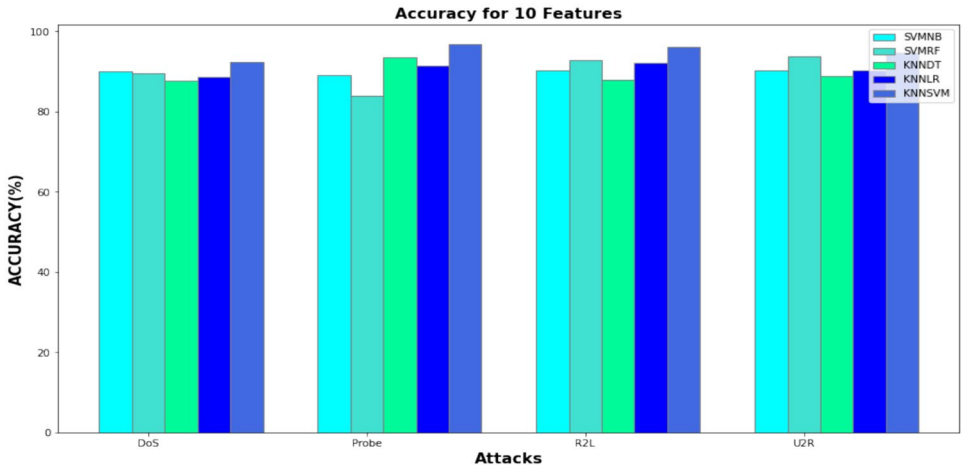

Fig. 7 Accuracy for 10 features

Fig. 8 Accuracy for 41 features

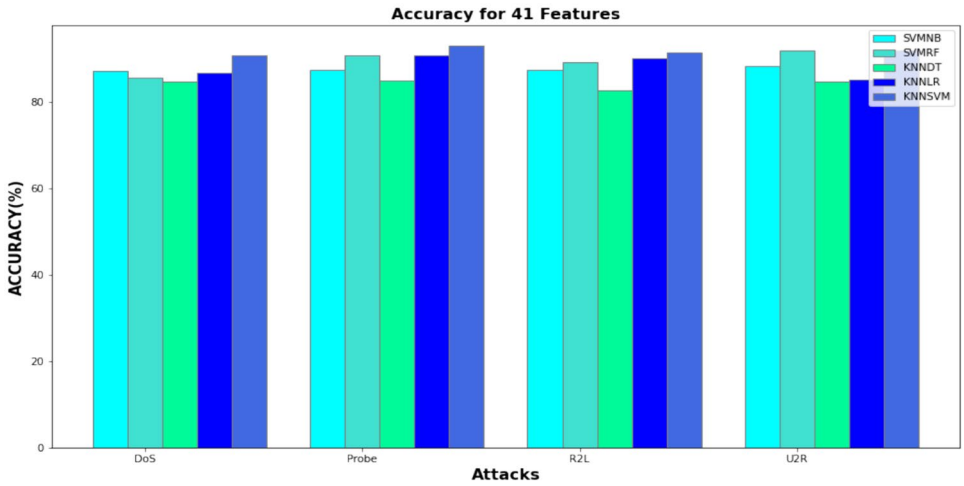

Table 13 Comparison of attacks Classification Accuracy with different classifiers

Attacks	Classifiers	Detection rate	
		With 10 features	With 41 features
DoS	**SVM+NB**	90.12	87.12
	SVM+RF	89.65	85.65
	KNN+DT	87.65	84.57
	KNN+LR	88.53	86.58
	KNN+SVM	92.38	90.68
Probe	**SVM+NB**	89.12	87.44
	SVM+RF	84.02	90.67
	KNN+DT	93.65	84.93
	KNN+LR	91.52	90.63
	KNN+SVM	96.90	93.63
R2L	**SVM+NB**	90.34	87.32
	SVM+RF	92.87	89.17
	KNN+DT	87.93	82.56
	KNN+LR	92.12	90.13
	KNN+SVM	96.06	91.35
U2R	**SVM+NB**	90.32	88.32
	SVM+RF	93.72	91.92
	KNN+DT	88.75	84.75
	KNN+LR	90.17	85.17
	KNN+SVM	94.73	91.83

detect DoS attacks with a recall of 0.91% for selected 10 features and 0.90% for 41 features. Whereas for probe attack the recall is 0.89% for selected 10 features and 0.86% for 41 features. For R2L attack the recall is 0.91% for selected 10 features and 0.89% for 41 features. And, for U2R attack the recall is 0.89% for selected 10 features and 0.85% for 41 features.

Figures 13–14 shows the F-Measure of the proposed system with selected 10 features and 41 features when compared with other existing classifiers. The classification module detects the attacks more effectively as it utilizes a two-staged classifier. The true positive rate is enhanced which produces better results in terms of F-Measure. From the graph analysis, it is understood that the proposed classifier has achieved a better F- measure with selected 10 features.

Comparison of F-Measure with selected 10 features and 41 features are given in Table 16 with different classifiers. DoS attack, Probe attack, R2L attack, and U2R attack analysis are carried out for existing classifiers. The proposed classifiers detect DoS attacks with an F-Measure of 0.99% for selected 10 features and 0.93% for 41 features. Whereas for probe attack the F-Measure is 0.94% for selected 10 features and 0.88% for 41 features. For R2L attack the F-Measure is 0.85% for selected 10 features and 0.71% for 41 features. And, for U2R attack the F-Measure is 0.61% for selected 10 features and 0.51% for 41 features.

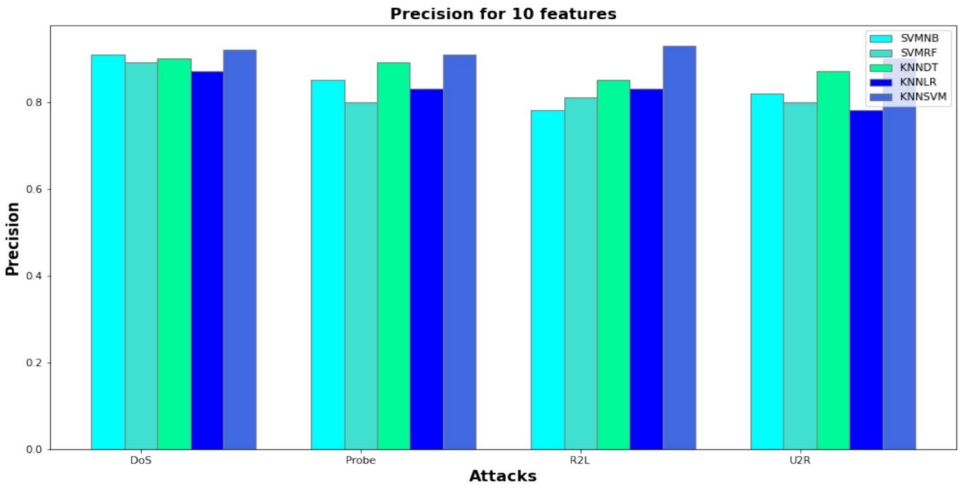

Fig. 9 Precision for 10 features

Figures 15–16 shows the Specificity of the proposed system with selected 10 features and 41 features when compared with other existing classifiers. Specificity identifies the abnormal activity or pattern of known attack accurately. Increased Specificity value ensures that the false positive rate has been minimized and an effective IDS is proposed. From the graph analysis, it is understood that the proposed classifier has achieved better Specificity with selected 10 features.

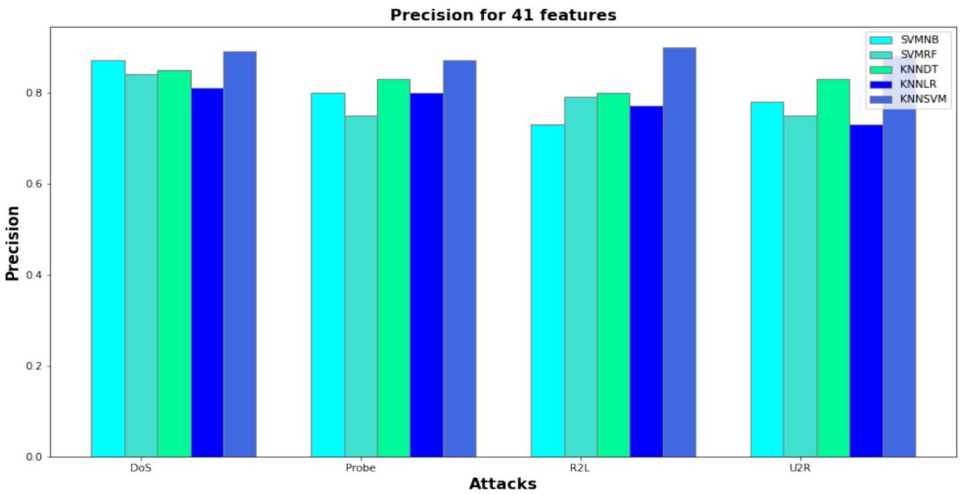

Fig. 10 Precision for 41 features

Table 14 Comparison of attacks Precision with different classifiers

Attacks	Classifiers	Detection rate	
		With 10 features	With 41 features
DoS	**SVM+NB**	0.91	0.87
	SVM+RF	0.89	0.84
	KNN+DT	0.90	0.85
	KNN+LR	0.87	0.81
	KNN+SVM	0.92	0.89
Probe	**SVM+NB**	0.85	0.80
	SVM+RF	0.80	0.75
	KNN+DT	0.89	0.83
	KNN+LR	0.83	0.81
	KNN+SVM	0.91	0.87
R2L	**SVM+NB**	0.78	0.73
	SVM+RF	0.81	0.79
	KNN+DT	0.85	0.80
	KNN+LR	0.83	0.77
	KNN+SVM	0.93	0.90
U2R	**SVM+NB**	0.82	0.78
	SVM+RF	0.80	0.75
	KNN+DT	0.87	0.83
	KNN+LR	0.78	0.73
	KNN+SVM	0.90	0.88

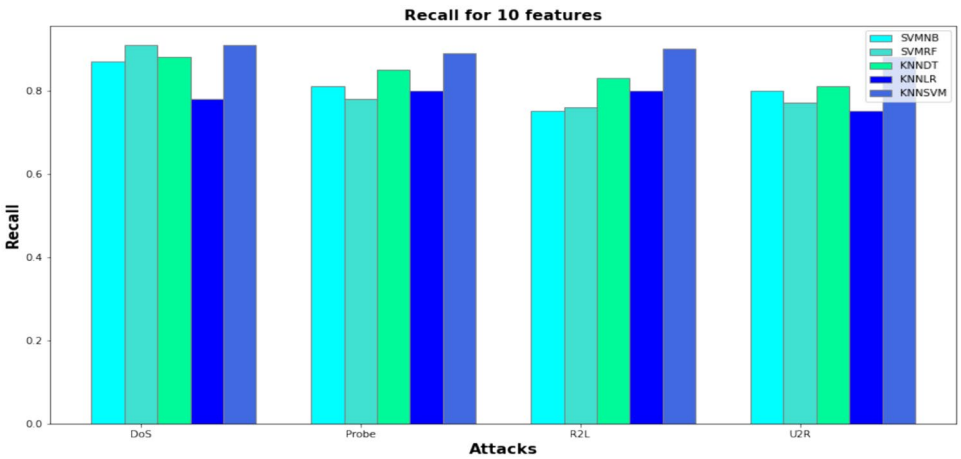

Fig. 11 Recall for 10 features

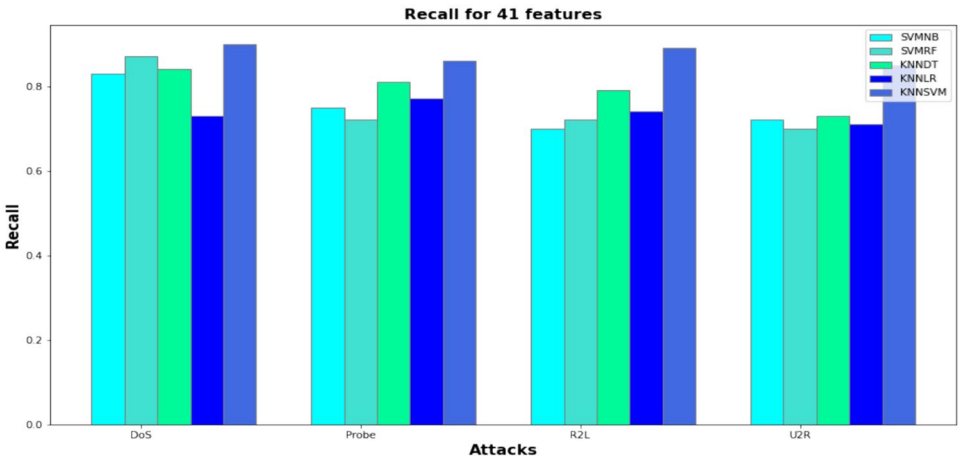

Fig. 12 Recall for 41 features

Table 15 Comparison of attacks Recall with different classifiers

Attacks	Classifiers	Detection rate	
		With 10 features	With 41 features
DoS	**SVM+NB**	0.87	0.83
	SVM+RF	0.91	0.87
	KNN+DT	0.88	0.84
	KNN+LR	0.78	0.83
	KNN+SVM	0.91	0.90
Probe	**SVM+NB**	0.81	0.75
	SVM+RF	0.78	0.72
	KNN+DT	0.85	0.81
	KNN+LR	0.80	0.77
	KNN+SVM	0.89	0.86
R2L	**SVM+NB**	0.75	0.70
	SVM+RF	0.78	0.72
	KNN+DT	0.83	0.79
	KNN+LR	0.80	0.74
	KNN+SVM	0.91	0.89
U2R	**SVM+NB**	0.80	0.72
	SVM+RF	0.77	0.70
	KNN+DT	0.81	0.73
	KNN+LR	0.75	0.71
	KNN+SVM	0.89	0.85

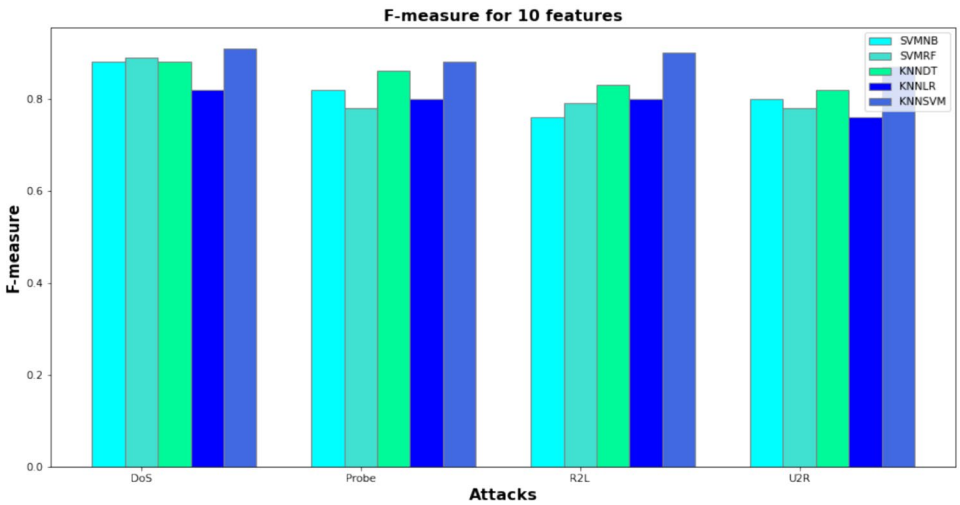

Fig. 13 F-Measure for 10 features

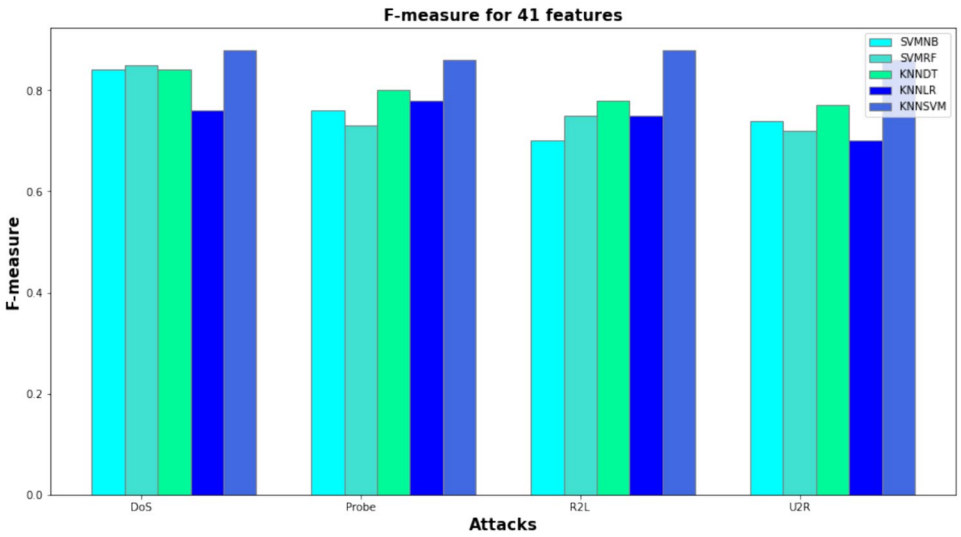

Fig. 14 F-Measure for 41 features

Table 16 Comparison of attacks F- measure with different classifiers

Attacks	Classifiers	Detection rate	
		With 10 features	With 41 features
DoS	**SVM+NB**	0.88	0.84
	SVM+RF	0.89	0.85
	KNN+DT	0.88	0.84
	KNN+LR	0.82	0.76
	KNN+SVM	0.90	0.88
Probe	**SVM+NB**	0.82	0.76
	SVM+RF	0.78	0.73
	KNN+DT	0.86	0.80
	KNN+LR	0.80	0.78
	KNN+SVM	0.88	0.86
R2L	**SVM+NB**	0.76	0.70
	SVM+RF	0.79	0.75
	KNN+DT	0.83	0.78
	KNN+LR	0.80	0.75
	KNN+SVM	0.90	0.88
U2R	**SVM+NB**	0.80	0.74
	SVM+RF	0.78	0.72
	KNN+DT	0.82	0.77
	KNN+LR	0.76	0.70
	KNN+SVM	0.88	0.86

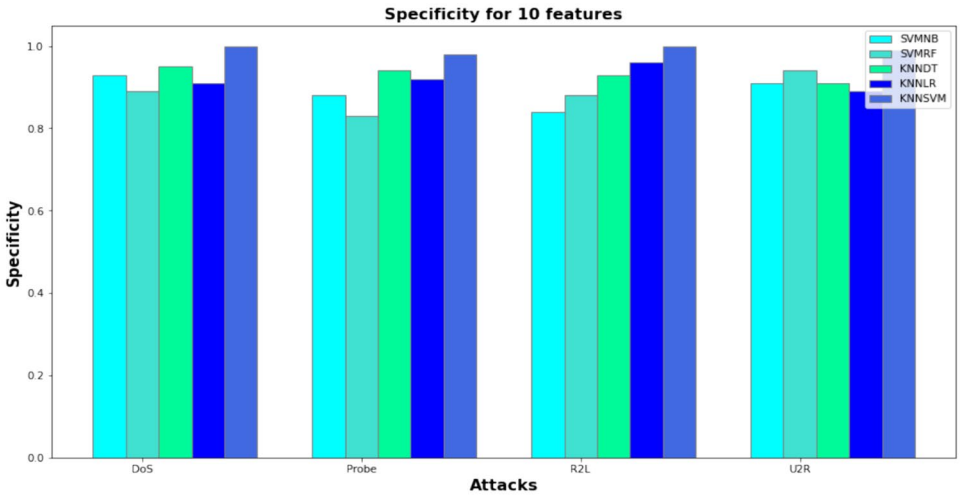

Fig. 15 Specificity for 10 features

Comparison of Specificity with selected 10 features and 41 features are given in Table 17 with different classifiers. DoS attack, Probe attack, R2L attack, and U2R attack analysis are carried out for existing classifiers. The proposed classifiers detect DoS attacks with a Specificity of 1.0% for selected 10 features and 0.98% for 41 features. Whereas for probe attack the Specificity is 0.98% for selected 10 features and 0.96% for 41 features. For R2L attack the Specificity is 1.0% for selected 10 features and 0.99% for 41 features. And, for U2R attack the Specificity is 0.99% for selected 10 features and 0.95% for 41 features.

A comparison of training time and testing time for attacks using different classifiers for the selected 10 features and 41 features is given in Table 18. From the experiment, it is understood that it takes less time to develop a model with 10 features than 41 features. Because of the feature selection technique, the time taken for training and testing is automatically reduced.

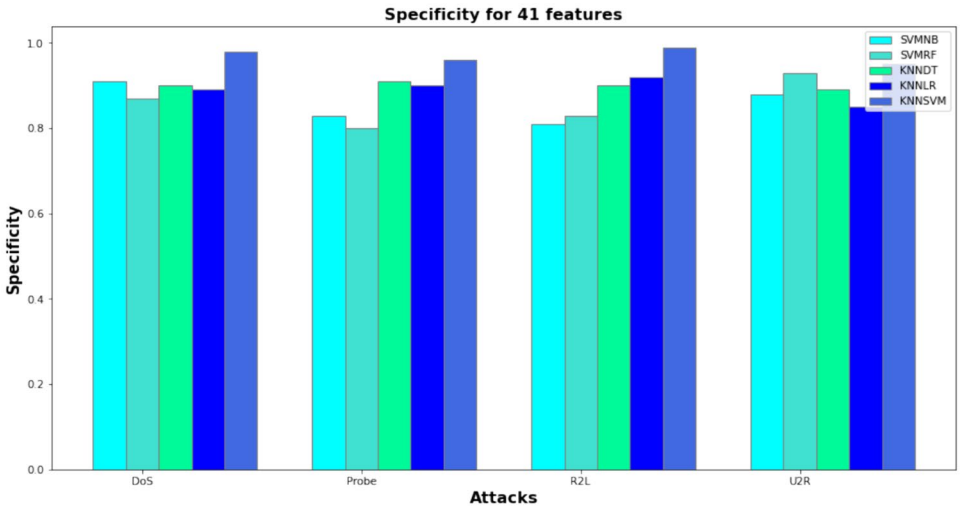

Fig. 16 Specificity for 41 features

Table 17 Comparison of attacks Specificity with different classifiers

Attacks	Classifiers	Detection rate	
		With 10 features	With 41 features
DoS	**SVM+NB**	0.93	0.91
	SVM+RF	0.89	0.57
	KNN+DT	0.95	0.90
	KNN+LR	0.91	0.89
	KNN+SVM	1	0.98
Probe	**SVM+NB**	0.88	0.83
	SVM+RF	0.83	0.80
	KNN+DT	0.94	0.91
	KNN+LR	0.92	0.90
	KNN+SVM	0.98	0.96
R2L	**SVM+NB**	0.84	0.81
	SVM+RF	0.88	0.83
	KNN+DT	0.93	0.30
	KNN+LR	0.96	0.92
	KNN+SVM	1	0.99
U2R	**SVM+NB**	0.91	0.88
	SVM+RF	0.94	0.93
	KNN+DT	0.91	0.89
	KNN+LR	0.89	0.85
	KNN+SVM	0.99	0.95

Table 18 Comparison of Training and Testing time for different classifiers

Classifiers	CPU TIME(SECONDS)		
	Training Time	Testing Time	Total Time
SVM+NB	109.8378	0.0783	109.9161
SVM+RF	65.9373	0.0954	66.0327
KNN+DT	78.0863	0.0698	78.1561
KNN+LR	152.3456	0.0768	152.4224
KNN+SVM	17.4237	0.0547	17.4784

6.1 Computational Complexity analysis

In the proposed system, assuming that the problem complexity is P, the Iteration count is I, maximum population size is M. The Improved Harris Hawks Optimization (IHHO) algorithm comprises three main components: population initialization, fitness estimation, and location update for individuals. The time complexity of initializing a population is determined by the problem complexity and maximum population size and the time complexity for initializing is O ($M \times P$). Fitness is estimated at each iteration and the time complexity of fitness estimation is O ($M \times P \times I$). At each iteration, the individuals will upgrade their position beginning with the first three persons and the time complexity of location update is O ($3 \times P \times I$). The other person's position update is the initial HHO position update and the time complexity is O ($(M - 3) \times P \times I$). As a result, the time complexity that IHHO demands for updating each location is O ($M \times P \times I$). The overall time complexity of IHHO is O ($2 \times M \times P \times I + M \times P$).

6.2 Mathematical Justification of the proposed approach

Tables 19 and 20 provide the mathematical justification of the proposed approach when it is compared with HHO in terms of population size and dimension. For the mathematical analysis, the 10 features are considered from the dataset to compute the average and standard deviation for the proposed approach with HHO. From the analysis, the proposed approach has a better average and standard deviation for all 10 features in terms of population size and dimensionality.

Table 19 Mathematical justification of IHHO and HHO for population size

F1	Average		Standard Deviation		F6	Average		Standard Deviation	
	IHHO	HHO	IHHO	HHO		IHHO	HHO	IHHO	HHO
10	3.07E+7	1.78E+6	1.03E+8	5.68E+6	10	1.43E+3	2.16E+2	2.61+4	3.91E+2
20	1.06E+4	6.69E+3	1.58E+6	2.56E+7	20	1.67E+3	3.17E+2	2.42+6	3.36E+3
40	1.54E+2	1.33E+2	7.03E+4	2.89E+5	40	1.53E+4	4.16E+2	2.08+2	4.18E+4
F2	**Average**		**Standard Deviation**		**F7**	**Average**		**Standard Deviation**	
	IHHO	HHO	IHHO	HHO		IHHO	HHO	IHHO	HHO
10	8.67E+3	7.68E+5	4.38E+7	5.41E+4	10	4.56E+2	2.14E+2	1.12E+2	8.41E+3
20	2.01E+3	1.20E+5	1.29E+5	3.14E+3	20	2.31E+3	1.18E+2	2.21E+1	9.31E+4
40	1.48E+4	3.14E+4	1.17E+6	5.09E+5	40	3.81E+2	1.31E+3	2.42E+2	6.81E+2
F3	**Average**		**Standard Deviation**		**F8**	**Average**		**Standard Deviation**	
	IHHO	HHO	IHHO	HHO		IHHO	HHO	IHHO	HHO
10	1.06E+2	1.41E+2	2.31E+4	3.16E+2	10	5.63E+3	2.21+2	1.63E+2	5.18E+2
20	3.03E+3	1.82+2	2.42E+3	1.97E+2	20	4.48E+4	4.81E+6	2.72E+3	2.96E+2
40	4.01E+3	3.76E+3	1.32E+3	4.81E+2	40	5.31E+2	3.16E+4	3.61E+2	3.14E+3
F4	**Average**		**Standard Deviation**		**F9**	**Average**		**Standard Deviation**	
	IHHO	HHO	IHHO	HHO		IHHO	HHO	IHHO	HHO
10	2.76E+2	1.31E+4	8.09E+2	2.81E+2	10	1.49E+3	3.61E+2	5.63E+3	3.54E+2
20	3.18E+3	1.01E+5	7.61E+3	3.61E+2	20	2.56E+2	2.78E+3	2.81E+3	3.62E+2
40	1.63E+3	6.73E+2	6.54E+2	4.22E+2	40	4.31E+2	3.75E+2	2.05E+3	1.08E+2
F5	**Average**		**Standard Deviation**		**F10**	**Average**		**Standard Deviation**	
	IHHO	HHO	IHHO	HHO		IHHO	HHO	IHHO	HHO
10	1.98E+3	2.02E+2	4.16E+4	3.41E+3	10	2.61E+2	1.78E+3	7.65E+4	8.91E+2
20	4.51E+4	3.16E+2	2.14E+5	1.94E+3	20	3.79E+2	1.54E+2	5.18E+3	4.62E+2
40	5.63E+3	4.52E+2	2.82E+4	3.6E+2	40	2.31E+2	1.64E+2	6.76E+2	3.71E+2

Table 20 Mathematical justification of IHHO and HHO for dimension

F1	Average		Standard Deviation		F6	Average		Standard Deviation	
	IHHO	HHO	IHHO	HHO		IHHO	HHO	IHHO	HHO
10	7.53E+2	6.54E+5	2.16E+6	3.75E+4	10	4.13E+2	3.62E+6	5.72E+3	3.61E+2
20	6.21E+7	1.52E+8	3.78E+2	2.61E+3	20	5.61E+3	1.78E+2	4.18E+7	2.24E+2
40	4.31E+6	1.63E+7	6.92E+3	2.93+2	40	7.78E+4	1.23E+4	2.67E+8	2.78E+3
F2	**Average**		**Standard Deviation**		**F7**	**Average**		**Standard Deviation**	
	IHHO	HHO	IHHO	HHO		IHHO	HHO	IHHO	HHO
10	1.65E+2	1.41E+3	1.48E+5	2.93E+9	10	1.69E+3	1.23E+4	2.73E+2	1.69E+3
20	2.79E+4	5.18E+3	2.15E+4	7.02E+2	20	3.72E+2	2.69E+2	4.87E+3	2.17E+4
40	7.65E+4	1.62E+9	2.56E+4	2.36E+4	40	4.81E+4	3.78E+3	7.81E+2	1.77E+5
F3	**Average**		**Standard Deviation**		**F8**	**Average**		**Standard Deviation**	
	IHHO	HHO	IHHO	HHO		IHHO	HHO	IHHO	HHO
10	2.18E+4	5.25E+4	2.15E+4	4.52E+4	10	3.17E+2	8.22E+4	6.64E+2	1.48E+4
20	1.02E+6	3.21E+5	5.19E+3	7.96E+5	20	4.27E+3	5.79E+4	2.13E+2	5.80E+5
40	2.91E+6	6.14E+7	2.79E+4	1.93E+4	40	6.52E+4	4.16E+3	2.79E+4	4.69E+5
F4	**Average**		**Standard Deviation**		**F9**	**Average**		**Standard Deviation**	
	IHHO	HHO	IHHO	HHO		IHHO	HHO	IHHO	HHO
10	1.02E+6	1.66E+8	1.33E+2	9.67E+3	10	1.25E+3	2.17E+3	2.16E+2	1.98E+3
20	7.65E+4	4.52E+7	1.62E+9	3.09E+2	20	2.67E+2	4.26E+4	5.18E+3	2.67E+2
40	3.52E+5	6.91E+3	2.56E+3	2.15E+3	40	3.78E+4	5.18E+3	8.76E+2	1.89E+2
F5	**Average**		**Standard Deviation**		**F10**	**Average**		**Standard Deviation**	
	IHHO	HHO	IHHO	HHO		IHHO	HHO	IHHO	HHO
10	3.23E+6	1.67E+3	2.96E+3	2.15E+2	10	1.76E+3	5.81E+4	1.92E+2	3.78E+3
20	2.17E+8	3.72E+2	1.18E+4	3.06E+4	20	2.81E+4	2.70E+3	3.69E+2	4.68E+4
40	5.19E+4	4.18E+3	1.75E+3	7.02E+2	40	5.91E+2	3.18E+2	2.87E+3	5.79E+2

7 Conclusion

In this paper, a system with two–layer classifier is proposed to effectively detect the intrusion from the network traffic. For data pre-processing the one-hot encoding method was employed which handles categorical data and further the dimensionality is reduced using PCA. Efficient features are selected by using IHHO. For classification two classifiers are utilized they are SVM and KNN. In stage -1 SVM is used to identify anomalies that can be attacked and in stage -2 KNN is utilized which identifies whether attacks still exist. Furthermore, a comparative analysis of the SVM + KNN-based classifier with another machine learning-based classifier was performed. Improving the training and testing time automatically increases the accuracy and detection rate. The main aim of the proposed system is to utilize the advantage of both misuse and anomaly classification techniques which also helps in reducing computational complexity and resulting in better accuracy. The future work of the proposed system is to improve the intrusion detection accuracy and reduce the false alarm rate. Moreover, the future work of this system aims to reduce the communication and computation overhead.

8 Limitation and Future Work

Even though the proposed system has an optimistic performance, there is a scope for improvement to handle the massive data flows in real time. The extended future work using the proposed system will be to detect attacks in the other layers of IoT architecture which includes support and application layers. It also aims to further improve intrusion detection accuracy and reduce the false alarm rate by utilizing hybrid feature selection algorithms. In the proposed system a standard benchmark dataset is used, and in the future real-time data traffic may be considered. In future work, Deep learning and reinforcement learning techniques can be utilized to identify the unknown attacks and the proposed IDS can be compared to multiple standard benchmark datasets and analyse their detection accuracy. Future development in IDS includes utilizing blockchain technology to enhance the integrity and visibility of Intrusion detection logs and data.

Author Contributions All the authors have contributed in equal manner.

Funding No funding.

Data Availability Data is available based on the request.

Declarations

Ethics Approval Authors Provide the Ethics Approval for the given manuscript.

Consent to Publish All the authors gave permission to Consent to publish

Competing interests The authors declare no competing interests.

References

1. Khraisat A, Gondal I, Vamplew P, Kamruzzaman J (2019) Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity 2(1). https://doi.org/10.1186/s42400-019-0038-7
2. Chaabouni N, Mosbah M, Zemmari A, Sauvignac C, Faruki P (2019) Network Intrusion Detection for IoT Security Based on Learning Techniques. IEEE Commun Surv Tutorials 21(3):2671–2701. https://doi.org/10.1109/COMST.2019.2896380
3. Assy AT et al (2023) Anomaly-based intrusion detection system using one-dimensional convolutional neural network. Procedia Computer Science 220:78–85. https://doi.org/10.1016/j.procs.2023.03.013
4. Udas PB, Karim MdE, Roy KS (2022) Spider: A shallow PCA based network intrusion detection system with enhanced recurrent neural networks. Journal of King Saud University - Computer and Information Sciences 34(10):10246–10272. https://doi.org/10.1016/j.jksuci.2022.10.019
5. Narengbam L, Dey S (2023) WIFI intrusion detection using artificial neurons with bio-inspired optimization algorithm. Procedia Computer Science 218:1238–1246. https://doi.org/10.1016/j.procs.2023.01.102
6. Mohammadi M et al (2021) A comprehensive survey and taxonomy of the SVM-based Intrusion Detection Systems. J Netw Comput Appl 178:102983. https://doi.org/10.1016/j.jnca.2021.102983
7. Gao X et al (2019) An adaptive ensemble machine learning model for intrusion detection. IEEE Access 7:82512–82521. https://doi.org/10.1109/access.2019.2923640
8. Binbusayyis A, Vaiyapuri T (2019) Identifying and benchmarking key features for cyber intrusion detection: An ensemble approach. IEEE Access 7:106495–106513. https://doi.org/10.1109/access.2019.2929487
9. Mushtaq E et al (2022) A two-stage intrusion detection system with auto-encoder and lstms. Appl Soft Comput 121:108768. https://doi.org/10.1016/j.asoc.2022.108768
10. Hnamte V, Hussain J (2023) DCNNBILSTM: An efficient hybrid deep learning-based Intrusion Detection System. Telematics and Informatics Reports 10:100052. https://doi.org/10.1016/j.teler.2023.100053
11. Choudhary S, Kesswani N (2020) Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 datasets using Deep Learning in IOT. Procedia Comput Sci 167:1561–1573. https://doi.org/10.1016/j.procs.2020.03.367
12. Salo F, Nassif AB, Essex A (2019) Dimensionality reduction with Ig-PCA and ensemble classifier for network intrusion detection. Computer Networks 148:164–175. https://doi.org/10.1016/j.comnet.2018.11.010
13. Pajouh HH et al (2019) A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in IOT backbone networks,. IEEE Trans Emerg Top Comput 7(2):314–323. https://doi.org/10.1109/tetc.2016.2633228
14. Peng K, Leung VC, Huang Q (2018) Clustering approach based on mini batch Kmeans for intrusion detection system over Big Data,. IEEE Access 6:11897–11906. https://doi.org/10.1109/access.2018.2810267

15. Alzaqebah A, Al-jarah I, Al-Kadi O (2021) A hierarchical intrusion detection system based on Extreme Learning Machine and nature-inspired optimization. SSRN Electronic Journal [Preprint]. Available at: https://doi.org/10.2139/ssrn.3996054
16. Peng L et al (2023) Hierarchical Harris Hawks optimizer for feature selection. J Adv Re [Preprint]. Available at: https://doi.org/10.1016/j.jare.2023.01.014
17. Hussien AG, Amin M (2021) A self-adaptive Harris Hawks optimization algorithm with opposition-based learning and chaotic local search strategy for global optimization and feature selection. International Journal of Machine Learning and Cybernetics 13(2):309–336. https://doi.org/10.1007/s13042-021-01326-4
18. Zhang HL (2022) An improved Harris Hawks optimizer combined with extremal optimization. Int J Mach Learn Cybern 14(3):655–682. https://doi.org/10.1007/s13042-022-01656-x
19. Wisanwanichthan T, Thammawichai M (2021) A double-layered hybrid approach for network intrusion detection system using combined naive Bayes and SVM. IEEE Access 9:138432–138450. https://doi.org/10.1109/access.2021.3118573
20. Gu J, Lu S (2021) An effective intrusion detection approach using SVM with naïve Bayes feature embedding. Comput Secur 103:102158
21. Chen WH, Hsu SH, Shen HP (2005) Application of SVM and ann for intrusion detection. Comput Oper Res 32(10):2617–2634. https://doi.org/10.1016/j.cor.2004.03.019
22. Safaldin M, Otair M, Abualigah L (2020) Improved binary gray wolf optimizer and SVM for Intrusion Detection System in wireless sensor networks. J Ambient Intell Humaniz Comput 12(2):1559–1576. https://doi.org/10.1007/s12652-020-02228-z
23. Saif S et al (2022) HIIDS: Hybrid intelligent intrusion detection system empowered with machine learning and metaheuristic algorithms for application in IOT based healthcare. Microprocess and Microsyst p. 104622. Available at: https://doi.org/10.1016/j.micpro.2022.104622
24. Ding H et al (2022) Imbalanced Data Classification: A KNN and generative adversarial networks-based hybrid approach for intrusion detection. Future Gener Comput Syst 131:240–254. https://doi.org/10.1016/j.future.2022.01.026
25. Mushtaq E, Zameer A, Khan A (2022) A two-stage stacked ensemble intrusion detection system using five base classifiers and MLP with Optimal Feature Selection. Microprocess Microsyst 94:104660. https://doi.org/10.1016/j.micpro.2022.104660
26. Lahasan B, Samma H (2022) Optimized Deep Autoencoder model for internet of things intruder detection. IEEE Access 10:8434–8448. https://doi.org/10.1109/access.2022.3144208
27. Mansour RF (2022) Blockchain assisted clustering with intrusion detection system for Industrial Internet of Things Environment. Expert Syst Appl 207:117995. https://doi.org/10.1016/j.eswa.2022.117995
28. Kurni M et al (2022) MRPO-Deep Maxout: Manta Ray political optimization based deep maxout network for big data intrusion detection using Spark Architecture. Adv Eng Softw 174:103324. https://doi.org/10.1016/j.advengsoft.2022.103324
29. Shitharth S et al (2021) An enriched RPCO-BCNN mechanisms for attack detection and classification in SCADA systems. IEEE Access 9:156297–156312. https://doi.org/10.1109/access.2021.3129053
30. Amanullah M et al (2022) CNN based prediction analysis for web phishing prevention. 2022 International Conference on Edge Computing and Applications (ICECAA) [Preprint]. https://doi.org/10.1109/icecaa55415.2022.9936112
31. Si-Ahmed A, Al-Garadi MA, Boustia N (2023) Survey of machine learning based intrusion detection methods for internet of medical things. Appl Soft Comput 140:110227. https://doi.org/10.1016/j.asoc.2023.110227
32. Jamalipour A, Murali S (2022) A taxonomy of machine-learning-based intrusion detection systems for the internet of things: A survey. IEEE Internet Things J 9(12):9444–9466. https://doi.org/10.1109/jiot.2021.3126811
33. Abdelmoumin G, Rawat DB, Rahman A (2022) On the performance of machine learning models for anomaly-based intelligent intrusion detection systems for the internet of things. IEEE Internet Things J 9(6):4280–4290. https://doi.org/10.1109/jiot.2021.3103829
34. Fraihat S et al (2023) Intrusion detection system for large-scale IOT NetFlow networks using machine learning with modified arithmetic optimization algorithm. Internet of Things 22:100819. https://doi.org/10.1016/j.iot.2023.100819
35. Prashanth SK, Shitharth S, Praveen Kumar B et al (2022) Optimal Feature Selection Based on Evolutionary Algorithm for Intrusion Detection. SN COMPUT SCI 3:439. https://doi.org/10.1007/s42979-022-01325-4
36. Gharehchopogh FS et al (2023) A multi-objective mutation-based dynamic Harris Hawks optimization for botnet detection in IOT. Internet of Things 24:100952. https://doi.org/10.1016/j.iot.2023.100952
37. Li S et al (2023) CRSF: An intrusion detection framework for industrial internet of things based on pretrained CNN2D-RNN and SVM. IEEE Access 11:92041–92054. https://doi.org/10.1109/access.2023.3307429
38. Boukraa L et al (2023) Intelligent intrusion detection in software-defined networking: A Comparative Study of SVM and Ann Models. Procedia Computer Science 224:26–33. https://doi.org/10.1016/j.procs.2023.09.007
39. Bukhari O et al (2023) Anomaly detection using ensemble techniques for boosting the security of Intrusion Detection System. Procedia Computer Science 218:1003–1013. https://doi.org/10.1016/j.procs.2023.01.080
40. Ayo FE et al (2023) A genomic rule-based KNN model for fast flux botnet detection. Egyptian Informatics Journal 24(2):313–325. https://doi.org/10.1016/j.eij.2023.05.002
41. Shitharth S, Kshirsagar PR, Balachandran PK, Alyoubi KH, Khadidos AO (2022) An Innovative Perceptual Pigeon Galvanized Optimization (PPGO) Based Likelihood Naïve Bayes (LNB) Classification Approach for Network Intrusion Detection System. IEEE Access 10:46424–46441. https://doi.org/10.1109/ACCESS.2022.3171660
42. Srikanth Yadav M, Kalpana R (2019) Data preprocessing for intrusion detection system using encoding and normalization approaches. 2019 11th International Conference on Advanced Computing (ICoAC) [Preprint]. https://doi.org/10.1109/icoac48765.2019.246851
43. Liu L et al (2018) An intrusion detection method for internet of things based on suppressed fuzzy clustering.EURASIP J Wirel Commun Netw 2018(1). https://doi.org/10.1186/s13638-018-1128-z
44. Zivkovic M et al (2022) Novel Harris hawks optimization and deep neural network approach for intrusion detection. Algorithms for Intelligent Systems 239–250. https://doi.org/10.1007/978-981-19-0332-8_17
45. Piri J, Mohapatra P (2021) An analytical study of modified multi-objective Harris Hawk optimizer towards Medical Data Feature Selection. Comput Biol Med 135:104558. https://doi.org/10.1016/j.compbiomed.2021.104558
46. Borkar GM et al (2019) A novel clustering approach and adaptive SVM classifier for intrusion detection in WSN: A data mining concept. Sustainable Computing: Informatics and Systems 23:120–135. https://doi.org/10.1016/j.suscom.2019.06.002
47. Rajendran R et al (2018) Detection of DOS attacks in cloud networks using intelligent rule-based classification system. Clust Comput 22(S1):423–434. https://doi.org/10.1007/s10586-018-2181-4
48. Aburomman AA, IbneReaz MB (2016) A novel SVM-kNN-PSO Ensemble Method for Intrusion Detection System. Appl

Soft Comput 38:360–372. https://doi.org/10.1016/j.asoc.2015.10.011
49. Saif S et al (2022) HIIDS: Hybrid intelligent intrusion detection system empowered with machine learning and metaheuristic algorithms for application in IOT based healthcare. Microprocess Microsyst 104622. https://doi.org/10.1016/j.micpro.2022.104622
50. Rose T et al (2020) A hybrid anomaly-based intrusion detection system to improve time complexity in the internet of energy environment. Journal of Parallel and Distributed Computing 145:124–139. https://doi.org/10.1016/j.jpdc.2020.06.012

Nandhini U is currently pursuing her PhD in the School of Computer Science Engineering and Information Systems, Vellore Institute of Technology, Vellore. She is working in the area of wireless sensor networks, IOT and network security. She has completed her M.Tech in Software Engineering from Vellore Institute of Technology, Vellore, India in the year 2020. Her areas of interest include computer networks, cryptography and artificial intelligence.

Dr. Santhosh Kumar S.V.N is working as an Associate Professor in VIT-Vellore Campus, India. He works in the areas of security and data dissemination in wireless sensor networks. His areas of interests include Wireless Sensor Networks, Internet of Things, Mobile Computing. He has received his B.E. degree in Computer Science and Engineering and M.E. degree in Software Engineering from Anna University, Chennai, India in the years 2011 and 2013 respectively. He has completed Ph.D from college of Engineering, Guindy Anna University, Chennai in the year 2018. He has published more than 65 research papers in reputed journals and conferences.