Journal of Optimization of Soft Computing (JOSC)

Vol. 2, Issue 3, pp: (7-21), Autumn-2024
Journal homepage: http://josc.rimag.ir/en

Paper Type (Research paper)

A Novel Approach for Intrusion Detection System in IoT Using Correlation-Based Hybrid Feature Selection and Harris Hawk Optimization Algorithm

Yashar Salami[1], Yaser Ebazadeh[2], Mehdi Hamrang[2], Nooshin Allahbakhshi[3]

[1]*Department of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran*
[2]*Department of[2]Computer Engineering, Germi Branch, Islamic Azad University, Germi. Iran*
[3]*Department of Computer and Information Technology Engineering, Khoy Branch, Islamic Azad University, Khoy, Iran*

Article Info

Article History:
Received: 2024/11/04
Revised: 2024/12/06
Accepted: 2024/12/16

DOI:

Keywords:
Data Mining, Security, Machine Learning, Anomaly Detection, Cybersecurity

**Corresponding Author's Email Address: yaser_ebazadeh@yahoo.com*

Abstract

With the rapid growth of the IoT, the number of devices connected to various networks has significantly increased. These devices generate vast amounts of data and are often deployed in open and unsecured environments, making them vulnerable to cyber-attacks. Therefore, ensuring the security of IoT networks has become a primary concern for researchers. One of the most effective methods for maintaining network security is using IDS. Intrusion detection monitors and analyzes incoming data to detect suspicious activities and potential attacks. Given the resource constraints of IoT devices and the complexity of the networks, improving the accuracy and efficiency of IDS is crucial. The primary goal of this research is to present a novel and optimized IDS for IoT networks. A hybrid feature selection method has been employed to enhance accuracy and reduce computational complexity, combining correlation-based filtering and wrapper methods using the (HHO) algorithm. In this approach, unnecessary features are removed, and essential features for classification are selected. Simulation results indicate that this method has achieved a 96.46% accuracy, outperforming traditional methods such as DT and SVM while improving false positive and false negative rates.

1. Introduction

With the rapid advancement of technology and the integration of the IoT into daily life, IoT has emerged as one of the most significant and transformative innovations of the past decade[1], [2]. By 2030, around 30 billion devices will be connected to the IoT, being utilized in various aspects of life, from smart homes to smart cities[3], [4]. These devices, equipped with the ability to sense their surroundings, collect data, and perform automated actions, have revolutionized our lives[5], [6]. The connection of these devices to cloud technologies has dramatically enhanced data storage and analysis processes. The potential of IoT to transform our lives is fascinating[7], [8]. However, along with these advantages, significant security challenges have also emerged.

The heterogeneity of IoT devices, the use of various communication protocols, and the large volume of data processed by these devices make them attractive targets for cyber-attacks[6], [9]. Considering these challenges, IoT devices, especially in sensitive environments such as military sectors, manufacturing industries, and smart grids, are exposed to numerous risks[10], [11]. Cybercriminals and hackers can easily exploit these vulnerabilities to access sensitive data and critical information. This security concern demands attention, as well as the development of novel approaches to counter IoT threats and vulnerabilities[12], [13].

Security studies have shown that traditional methods for combating these threats in IoT devices are not effective due to resource limitations and the complexity of communications[14][15]. Particularly, with cloud and fog computing that leads to vast amounts of data exchange between various devices and cloud servers, traditional security approaches cannot effectively protect these devices[16][17]. Therefore, the need for lighter and more optimized security methods for detecting and preventing cyber-attacks is strongly felt[10].

IDS has been introduced as one of the most effective security tools in this context[18]. These systems analyze and monitor the behavior patterns of IoT devices and networks to detect any abnormal activities or potential cyber-attacks. IDS can identify complex and unknown attacks using machine learning techniques and optimization algorithms[19]. By analyzing data and detecting anomalous patterns in device behavior, IDS alerts system administrators and allows them to take preventive measures before an attack occurs[20].

In environments such as smart homes and smart cities, where IoT devices are widely used, IDS has become one of the key tools for preventing cyber-attacks. The importance of sensitive information in these environments has made high accuracy and efficiency in detecting intrusions and cyber-attacks a necessity[21]. In this regard, using attack modeling and DL techniques can help organizations better understand the nature of attacks and effectively manage security risks[22].

DL techniques, especially given their high computational power and ability to train complex models, are essential in detecting hidden and intricate patterns in IoT data[23]. Utilizing the vast and diverse data generated by IoT devices, these techniques can effectively identify threats and attacks and enhance network security[24]Overall, transitioning from traditional methods to modern intrusion detection approaches and relying on advanced security techniques is essential to protecting sensitive information and ensuring the security of IoT devices and networks.

Thus, developing machine learning and DL-based security systems, particularly in the IoT domain, can significantly enhance security and reduce these devices' vulnerabilities. The future of security in the IoT world depends on advancements in intelligent security techniques that can keep pace with the growing number of devices and the complexity of their communications while predicting and mitigating cyber threats.

1.1. Contribution

Ultimately, DL techniques play a significant role in detecting attacks and enhancing network security with their high computational power and ability to train complex models. These techniques allow organizations to identify hidden and intricate patterns within IoT data and use them to prevent cyber-attacks. The transition from traditional methods to modern intrusion detection approaches and the adoption of advanced security techniques, particularly in IoT environments, is essential for safeguarding sensitive information and ensuring network security.

1.2. Paper organization

This article‘s general structure will be described in detail so that the reader can understand the arrangement and contents of each section well. The second part examines and analyzes related works. In this section, the previous research on the subject will be reviewed, and the strengths and weaknesses of each will be carefully examined. This literature review will help the reader to gain a better understanding of the current situation and the need for further research. In the third part, the research proposal is presented. This section describes in detail the methods and techniques chosen to solve the problem in question. Also, the advantages of using these methods compared to the previous methods and the innovations used in this plan are explained in detail. The fourth part deals with simulation and results. Here, the performed simulations are described and their results are analyzed. Also, the graphs and tables used in this section help to show the research findings visually and provide a deeper analysis. Finally, the fifth section is dedicated to summarizing and concluding. In this section, the key results of the research are reviewed with an emphasis on their importance, and suggestions for future research are also presented. This section, as a summary of the discussed topics, allows the reader to form their own opinions related to the findings and conclusions.

1.3. Symbols

The symbols used in this article are listed in Table 1.

Table 1: symbols used in the article.

IoT	*Internet of Things*
IDS	*Intrusion Detection Systems*
HHO	*Harris Hawk Optimization*
SVM	*Support Vector Machine*
DT	*Decision Tree*
PCA	*Principal Component Analysis*
RBM	*Restricted Boltzmann Machines*
DL	*deep learning*
DDOS	*Distributed Denial of Service*
CFS	*Correlation-Based Feature Selection*
KNN	*K-Nearest Neighbors*
TP	*True Positive*
FP	*False Positive*
TN	*True Negative*
FN	*False Negative*

2.Related work

In [25], a novel IoT intrusion detection approach is presented based on a migration DL model. This method utilizes a feature extraction algorithm to enhance system performance by combining DL and intrusion detection technology. Experiments conducted using the KDD CUP 99 dataset have shown that the proposed algorithm outperforms other algorithms with just 10% of the training data. Empirical results indicate that this approach reduces detection time while providing higher accuracy in identifying attacks. These findings demonstrate that the migration DL model and feature extraction algorithms can significantly improve security in IoT networks and smart cities.

In [26], a framework for intrusion detection based on RBM is proposed. This framework, utilizing RBM—an artificial neural network—can learn high-level features from raw data without supervision. Experimental results from real data collected from a smart water distribution factory showed that this framework has remarkable performance in detecting attacks and has significantly improved system efficiency. This framework indicates that RBM can be an effective tool for enhancing security in the IoT.

In [27], PCA is employed for feature dimensionality reduction and combined with ensemble-based classifiers to predict intrusion attacks. This research, focusing on intelligent hospitals and medical devices, demonstrated that using PCA and classifiers could enhance the security of smart networks and prevent intrusions in critical systems. The KDDCup'99 dataset was utilized to test this method, yielding significant improvements in attack prediction and enhanced Internet of Medical Things security.

In [28], a group intrusion strategy based on cyber intelligence is introduced, utilizing a combination of machine learning algorithms such as Random Forest, Bayesian Network, C5.0, and CART for detecting botnet attacks in IoT networks. This strategy, employing a cyber intelligence framework and group learning, successfully identified various attacks with high accuracy. Results indicated that applying this framework in smart cities significantly improved detection rates and reduced false favorable rates, contributing to enhanced security in IoT networks.

In [29], methods for improving the performance of IDS against malicious attacks are examined. This method enhances the IDS performance by utilizing adversarial retraining to cope with attacks. Experimental results showed that adversarial retraining could increase the detection accuracy of the IDS to over 99% against malicious attacks, thereby strengthening security in smart cities.

In [30], artificial intelligence techniques for enhancing security in smart cities are explored. Security and privacy issues have gained more attention as the use of information technologies in data management increases in intelligent cities. This study revealed that employing sophisticated artificial intelligence techniques could help improve monitoring, increase security, and protect data in intelligent city networks.

In [31], a method for detecting DDoS attacks in IoT networks is presented. This method utilizes feature selection to reduce data dimensionality and improve IDS performance. This method selects appropriate features for attack detection using multi-objective optimization and employs extreme learning machines to enhance detection accuracy. Results indicate that this method can effectively increase security in IoT networks.

In [32], a DL--based intrusion detection system for IoT devices is introduced. This system employs a four-layer deep network to identify malicious traffic and has demonstrated satisfactory performance in attack detection with an accuracy of 93.74%. This system can function independently of communication protocols and enhance security in IoT networks.

3. Proposed Scheme

This research proposes a combined filter-wrapper approach for feature selection aimed at intrusion detection in IoT networks. This approach consists of two main stages. The first stage is the filter stage, where features related to the class label are identified using correlation measures between the features and the class labels. In this stage, features that correlate with a certain threshold are selected as important and influential features. This stage aims to reduce the dimensionality of the data and select prominent features using statistical criteria.

In the next stage, the wrapper stage, the HHO algorithm is used to select a quasi-optimal subset of the selected features. Inspired by the group hunting behavior of Harris hawks, the HHO algorithm finds the best combination of features. This algorithm acts as a population-based optimization method, aiming to improve feature selection through optimization.

The combined filter-wrapper approach in this study enhances the accuracy and speed of intrusion detection in IoT. In this approach, the filter stage helps reduce computational complexity by identifying effective features. Then, the wrapper stage utilizes a machine learning model and the HHO algorithm to refine the selected features. The objective of these two stages is to improve the efficiency and accuracy of the final model simultaneously.

This combined approach's advantage is that it simultaneously employs filter and wrapper methods, reducing data dimensionality and increasing detection accuracy. This combination can be suitable for problems with complex datasets and numerous features, as it simultaneously reduces computational complexity and enhances model accuracy.

Feature selection algorithms generally select a subset of essential features from large datasets. Their primary goals are to reduce data dimensions, improve the performance of machine learning models, and decrease computational complexity. Eliminating unnecessary or redundant features results in faster model training and increased accuracy. Moreover, appropriate feature selection helps reduce the risk of overfitting and improves the model's generalization on new data.

3.1. Filter-Based Feature Selection

In the feature selection process, there are usually two main stages:

1. Initial Feature Selection: In this stage, the goal is to find the smallest subset of features that results in the most minor classification error. Various methods are available, including filter, wrapper, and hybrid methods.

2. Model Evaluation: In this stage, the model is trained using the selected features, and its performance is evaluated using test data. This stage aims to reduce the training samples' classification error and improve the test samples' prediction accuracy.

Selecting essential and useful features can significantly improve the performance and accuracy of machine learning models. Therefore, feature selection is considered a fundamental step in the machine learning process.

Feature selection methods fundamentally improve the performance of machine learning models by removing unnecessary or redundant features, reducing data dimensions, or increasing the models' generalization capability. These methods help models better identify important patterns in the data and prevent overfitting to the training data.

In supervised feature selection, a machine learning model is trained using labeled data, and important features for accurately predicting labels are determined. These methods are typically based on criteria such as information gain, feature importance, or prediction error.

In unsupervised feature selection, data is used without labels, and only essential and separable features are validated. These methods are generally performed based on dimensional analysis or pattern recognition capabilities.

In semi-supervised feature selection, labeled and unlabeled data are used to select important features for class separation. These methods are often suitable for addressing issues like the scarcity of labeled samples.

3.2. Feature Selection Based on Correlation

The CFS algorithm is one of the most powerful and popular methods for feature selection in classification problems. Due to its simplicity and high efficiency, this algorithm is widely used in many machine-learning issues. The main objective of CFS is to identify a subset of features that not only have a strong correlation with target classes but also exhibit a low correlation with one another. This reduces redundant information among features and helps selected features better differentiate between classes.

The algorithm initially ranks all features based on their correlation with the classes. After that, a subset of features with a high correlation to the classes and a low correlation with each other is selected. These features are recognized as a set of features with "best generalization" and play a key role in enhancing the performance of classification models. These features make the models more accurate and increase their ability to detect and classify data.

The advantage of using the CFS algorithm as a filter for feature selection lies in its ability to identify and eliminate irrelevant features or those with a high correlation to other features. This process reduces model complexity and increases efficiency, as only features related to the classes remain independent.

The equation (1) used in this algorithm to evaluate the quality of a subset of features is as follows:

$$Merit_s = \frac{kr_{cf}}{\sqrt{k+(k+1)r_{ff}}} \quad (1)$$

Where Merit is the exploratory "merit" of a feature subset (S) containing (k) features, (rcf) is the average class correlation of the feature where (rff) is the average feature-feature correlation. The numerator of this equation acts as a class prediction index using the selected features, and the denominator indicates redundancy among the features.

CFS employs Pearson correlation to evaluate the features. This method standardizes features and ranks them based on their correlation with target classes. Features that have little correlation with the classes typically provide useless or inefficient information for classification and are, therefore, eliminated. Additionally, redundant features that correlate highly with one or more other features are discarded from the model to enhance its efficiency and accuracy. This process increases accuracy and makes classification models more efficient in processing data and correctly predicting classes.

3.3 Wrapper-Based Feature Selection

Wrapper-based methods for feature selection utilize classification models to evaluate and select a subset of features that yield the best performance for the model. These wrapper methods employ a search strategy to find the optimal subset of features. Such strategies may include forward sequential search, backward sequential search, genetic algorithms, etc. A subset of features is selected at each step of the search, and a learning model is trained on this subset. The model's performance is assessed based on accuracy, sensitivity, specificity, etc. The subset of features that provides the best model performance is then selected as the final subset.

These methods directly leverage the learning model for evaluation, enabling them to choose features that contribute optimally to the model's performance. The model becomes more straightforward and interpretable by eliminating unnecessary features and retaining important ones. Selecting fewer features reduces the time and resources required for training and using the model. However, these methods require repeated execution of learning algorithms on various subsets of features, which can be very time-consuming and costly due to the significant resources needed for searching and evaluating.

Wrapper-based methods can improve the performance of machine learning models. Still, they require a thorough and meticulous evaluation of feature subsets and the selection of appropriate modeling and assessment techniques. These methods can help reduce model complexity, improve interpretability, and enhance prediction accuracy.

3.4 Feature Selection Based on HHO

This research employs the HHO algorithm to select optimal features that correlate highly with class labels. The primary objective of this approach is to identify significant features from datasets related to the IoT. The HHO algorithm aims to optimize the feature selection process by considering two fundamental goals: reducing the number of features and enhancing the accuracy of intrusion detection.

The process begins with the algorithm analyzing the existing features, attempting to identify key and important features using heuristic methods. Subsequently, through computationally implemented learning and optimization techniques, the algorithm moves towards finding a subset of features that exhibits the most minor complexity while maintaining the highest accuracy in intrusion detection.

Given that the feature selection problem is classified as NP-Hard, the HHO utilizes specific optimization methods to find a genuinely optimal solution. This method is inspired by the natural behaviors of Harris hawks, which employ a combination of random and guided search strategies to discover the best features.

3.5 Fitness Function

The fitness function plays a critical role in evolutionary and metaheuristic algorithms, serving as a key stage for assessing the quality and performance of each solution within a population. This function evaluates the performance of each solution and determines its success in addressing the problem at hand. In other words, the value of the fitness function is determined based on the selected features from the datasets, indicating how closely each solution aligns with the desired objective.

This function automatically computes the target value based on criteria such as the prediction error of other defined metrics. Solutions yielding a higher fitness value are recognized as superior candidates and are selected for subsequent iterations of the evolutionary algorithm. Ultimately, solutions that achieve the best fitness values are designated as the final and optimal results and utilized to solve the problem effectively. This iterative process contributes to creating an evolutionary cycle, facilitating continuous improvement of solutions and optimizing the resolution of complex issues.

In the next phase, each solution is evaluated based on defined objectives and the fitness function. This assessment employs a proposed fitness function that combines the feature selection rate with the prediction error of intrusion in the IoT context. The fitness function shows question 2 is defined as follows:

$$Minimize\ F(x) = \begin{cases} f_1(x) = \frac{L}{A} & , L \in A, A \in \mathbb{R}^+ \\ f_2(x) = \frac{FP+FN}{P+N} & , (P+N) \in \mathbb{R}^+ \end{cases} \qquad (2)$$

In this equation (2), A represents the total number of features, while L indicates the number of selected features. These calculations intertwine with issues related to security and safety within systems. The variables TP, FP, TN, and FN denote the actual number of healthy nodes, nodes erroneously identified as intrusions, nodes correctly identified as intrusions, and nodes mistakenly recognized as healthy, respectively.

The set P encompasses the sum of TP and TN, while the set N consists of the sum of FP and FN. These variables are utilized to evaluate the accuracy and quality of a model or security system, ensuring that the feature selection process effectively enhances the model's performance while minimizing the risk of misclassification.

4 Simulation

This paper presents an innovative IoT intrusion detection and prediction method. This method is based on a combination of feature selection using a filter-wrapper approach based on correlation and using HHO to classify the data. This section introduces the standard dataset used for intrusion detection in IoT. This dataset, named BotNeTIoT, is extracted from the UCI database and contains information related to over 50,000 nodes. Some nodes are classified as healthy, while others are classified as intrusive.

The main objective of using this dataset is to compare the proposed method for intrusion detection in IoT with other existing methods. This comparison aims to evaluate the improvements the proposed method can bring in terms of the accuracy and performance of intrusion detection models. During the implementation of this method, challenges such as data imbalance and the high volume of existing features in the dataset arise. Therefore, selecting key and effective features can significantly improve the accuracy and efficiency of classification algorithms and machine learning, assisting in more precise intrusion detection in IoT. Table 2 shows details of the features available in this dataset.

Table 2: Features the dataset.

Feature	Feature Number	Feature Title	Feature Number
HH_L0.1_pcc	13	MI_dir_L0.1_weight	1
HH_jit_L0.1_weight	14	MI_dir_L0.1_mean	2
HH_jit_L0.1_mean	15	MI_dir_L0.1_variance	3
HH_jit_L0.1_variance	16	H_L0.1_weight	4
HpHp_L0.1_weight	17	H_L0.1_mean	5
HpHp_L0.1_mean	18	H_L0.1_variance	6
HpHp_L0.1_std	19	HH_L0.1_weight	7
HpHp_L0.1_magnitude	20	HH_L0.1_mean	8
HpHp_L0.1_radius	21	HH_L0.1_std	9
HpHp_L0.1_covariance	22	HH_L0.1_magnitude	10
HpHp_L0.1_pcc	23	HH_L0.1_radius	11
Label	24	HH_L0.1_covariance	12

4.1. Dimensionality Reduction Based on High Correlation

In this research, a dataset available in the UCI repository that contains 24 different features is utilized. These features are listed in Table 2. Since not all features may be equally effective in intrusion detection and using all of them could increase model complexity, selecting a subset of useful features is essential. In this study, a CFS approach is employed to optimize the selection of features.

Pearson correlation is a statistical method that examines the linear relationship between two variables. This correlation can take values ranging from -1 to 1, where values close to 1 indicate a direct and positive linear relationship, values close to -1 indicate a reverse relationship and values close to 0 indicate no linear relationship between the variables. In this study, Pearson correlation analysis is used between the features and class labels to identify and select features that have the highest correlation with the class label. This approach improves intrusion detection accuracy and helps reduce data dimensions.

Table 2 presents the results of the Pearson correlation analysis. These results illustrate how selecting more relevant features can enhance the performance of classification models and intrusion prediction in IoT networks. Table 3 discusses the importance of the correlation between features and class labels in an analytical model. The correlation between features and class labels falls from -1 to 1. A positive correlation indicates a direct relationship between a feature and the class label; as the feature's value increases, the class label's value also increases. Conversely, a negative correlation indicates a reverse relationship, where an increase in the feature value leads to a decrease in the class label value.

Bar charts like Figure 1 are utilized to visualize these relationships better. Based on the calculated correlation, these charts graphically display the relationship between features and the class label. These charts help researchers identify important features that correlate significantly with the class label. By selecting these key features significantly related to the class label, the analytical model can provide better accuracy and performance, ultimately enhancing the model's effectiveness.

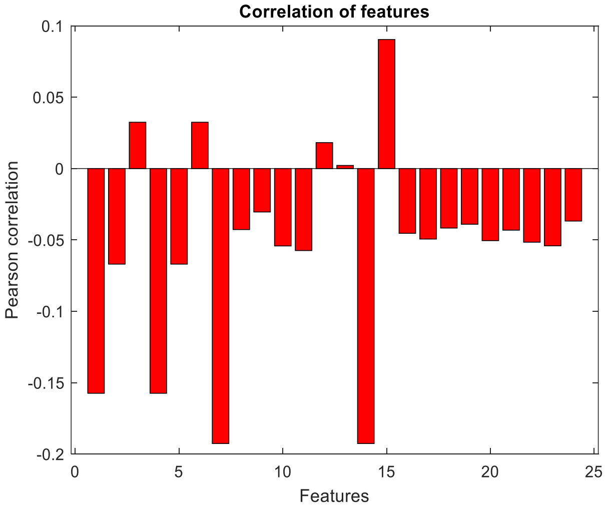

Fig 1. *Correlation between features and class label.*

In Figure 1, various features exhibit either a positive or negative correlation with the class label and the degree of this correlation varies across different features. This diversity in correlation can play a crucial role in selecting the most suitable features for analytical models. For features with positive correlation, the closer the correlation value is to 1, the stronger the relationship between the feature and the class label. This indicates that if a feature's correlation with the class label approaches 1, it is considered an effective and valuable element for model inclusion.

Conversely, features with negative correlation are also significant. The closer a negative correlation value is to -1, the stronger the inverse relationship between the feature and the class label. In other words, if the negative correlation of a feature with the class label approaches -1, that feature may also improve the model's performance. The proposed method establishes a specific threshold to identify valuable features. Features that have a positive correlation and whose correlation values exceed the average positive correlation of other features are selected as valuable. Similarly, features with negative correlation and whose correlation values fall below the average negative correlation of other features are also considered useful. Selecting these key features can enhance the accuracy and efficiency of statistical and predictive models. Table 4 presents the valuable features and their correlation values with the class label. The selection of 12 features with the highest correlation with the class labels from the 24 available features in the dataset can help improve the performance and efficiency of the statistical and predictive models. These selected high-correlation features will act as useful and impactful inputs for the models, and by eliminating the less important features, the accuracy and speed of the models will increase. This careful feature selection can lead to an overall enhancement in the performance of the prediction and decision-making systems.

As shown in Table 4, the features that meet the defined threshold criteria are identified as key and valuable features and selected for use in statistical and predictive models. This intelligent and precise feature selection significantly improves model accuracy and efficiency, enhancing predictive systems' performance. Additionally,

The show Figure 2 depicts a **correlation matrix**, which is one of the key tools in data analysis and is used to examine the relationships among variables in a dataset. This matrix, displayed as a square table, provides the Pearson correlation coefficients for each pair of variables. These coefficients help researchers understand how changes in one variable might relate to changes in another. The matrix is presented both numerically and visually, allowing for a quick and intuitive understanding of patterns and relationships between variables.

The correlation coefficients, which range from [-1, +1], describe the **strength and direction of the linear relationship** between two variables:

- **Positive values (close to +1):** Indicate a strong direct correlation, meaning that as the value of one variable increases, the value of the other variable also increases proportionally. For example, if two economic variables such as "income" and "consumer spending" have a correlation close to +1, an increase in income is likely to result in an increase in spending.
- **Negative values (close to -1):** Indicate a strong inverse correlation, meaning that as the value of one variable increases, the value of the other variable decreases. For instance, in environmental studies, there might be a strong negative correlation between "rainfall levels" and "drought percentage."
- **Values close to zero:** Suggest no linear relationship or a very weak relationship between two variables. This means that changes in one variable have no significant impact on the other. Such relationships might be random or influenced by unrelated factors.

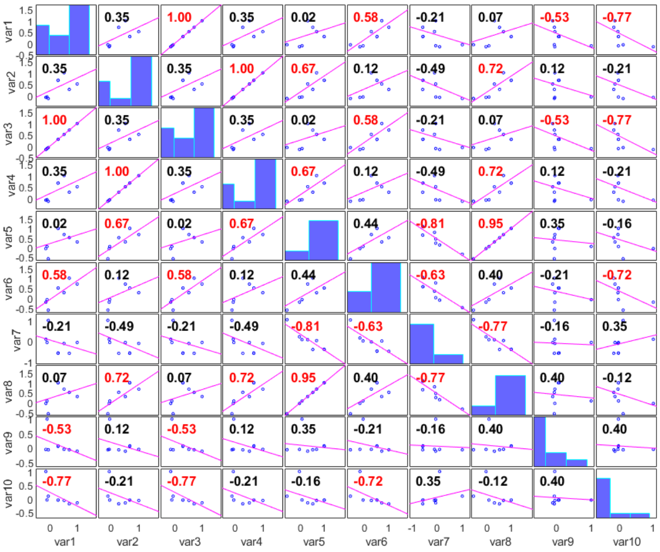

Fig 2. *Correlation of selected features.*

4.2 Feature Subset Selection Based on HHO

The HHO algorithm is primarily used to find optimal solutions across various problems. As an advanced search method, it explores different solutions to provide an optimal combination of features for classification and prediction tasks. This study evaluates the HHO algorithm based on classification error, which serves as a criterion for assessing solutions. Utilizing this algorithm, an optimal feature set is identified for classifying and predicting intrusions in the IoT. The HHO offers the most efficient combination of intrusion detection features and IoT prediction based on the available characteristics. This method can significantly enhance the performance and efficiency of predictive and classification models while determining the best feature combinations for various issues. Table 5 illustrates the optimal solution selected by the Harris Hawk.

Table 5: Optimal Solutions in HHO

Feature Number	***Feature Name***
2	*H_L0.1_weight*
4	*H_L0.1_mean*
5	*HH_jit_L0.1_mean*
10	*HH_L0.1_magnitude*
15	*MI_dir_L0.1_mean*

According to **Table** 5, the HHO significantly improved intrusion detection within IoT by selecting seven essential features for use in classifying training samples and predicting test samples. The algorithm evaluates its performance using classification error as the assessment criterion, calculated based on a fitness function. The convergence chart presented in **Figure 3** illustrates the reduction in the fitness function and the classification error for intrusions achieved through the solutions provided by the HHO algorithm. As the progresses, both the fitness function value and classification error improve, indicating the effectiveness and efficient performance of this algorithm in detecting intrusions within the IoT environment employing the optimal features selected by the HHO and accurately calculating the classification error as the primary evaluation criterion can enhance classification and intrusion prediction performance in IoT, facilitating more accurate and reliable predictions.

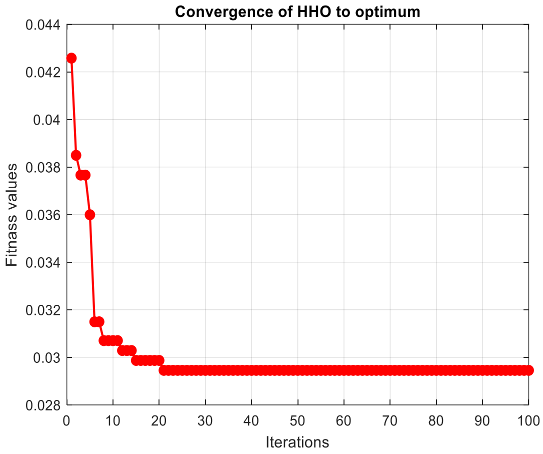

Fig 3. Convergence of the fit function.

As depicted in **Figure 3**, the HHO effectively solves feature subset selection issues for intrusion detection in the IoT. With increased iterations and improved performance, the algorithm approaches the optimal fitness function, yielding high accuracy. For instance, after 100 iterations, the algorithm achieves a fitness function value of **0.029**, demonstrating improved performance and high accuracy. These results indicate that the HHO reaches the best possible solution for the feature subset selection problem and minimizes the classification error to the lowest feasible extent. The convergence chart also shows that the algorithm moves towards the optimal point, reducing the classification error, which reflects better performance and higher accuracy in detecting intrusions within the IoT. This algorithm can substantially improve intrusion detection and prediction, enhancing preventive and security measures. This algorithm leverages the combined strength of the HHO technique and the computational power of s to achieve optimal and efficient solutions for complex issues. This makes the HHO an effective and efficient tool for detecting and preventing intrusions in the IoT.

4.3 Evaluation of the Proposed Method

Various assessment metrics are utilized to evaluate the proposed method for analyzing healthy and intrusion nodes. These metrics are derived from the confrontation between the predicted class labels for healthy and intrusion nodes and their actual class labels, as delineated in the confusion matrix. The primary evaluation metrics applicable for binary classification methods include:

1. **Accuracy**: Equation (3) represents the ratio of correctly classified nodes to the total number of nodes[33].

$$Accurace = \frac{TP + TN}{TP+TN+FP+FN} \tag{3}$$

2. **Sensitivity or Recall**: Equation (4) indicates the proportion of actual positive nodes that are correctly classified[34].

$$Recall = \frac{TP}{TP+FN} \tag{4}$$

3. **Specificity**: Equation (5) metric shows the proportion of actual negative nodes that are accurately classified[35].

$$\text{Precision} = \frac{TP}{TP+FP} \tag{5}$$

F1-Score: Equation (6) The F1-score is the geometric mean of precision and sensitivity, serving as a combined measure of the model's accuracy and information content[36].

$$F - measure = \frac{2*Precision *Recall}{Precision+Recall} \tag{6}$$

These metrics assist in evaluating the classification method's performance on the test data, allowing us to quantify its accuracy and effectiveness. The confusion matrix, along with the parameters TP, FP, TN and FN, provides a foundational framework for calculating these metrics and comprehensively understanding the classification method's performance. Figure 4 show the pseudocode of the proposed method.

According to Table 6, the proposed method, which integrates features based on Pearson correlation and Harris Hawk optimization, achieves commendable results for the evaluation metrics during testing with new data and nodes. These results indicate that the method performs well in detecting and classifying the targeted cases. Furthermore, the KNN classification method demonstrates improved performance compared to the other examined methods. This suggests that KNN can be a robust and effective data classification and pattern recognition approach, yielding better results than alternative methods.

4.4 Comparison of the Proposed

In this section, after assessing the performance of the proposed method against the test datasets, the results obtained from this method—which comprises a combination of CFS and the Harris Hawks alongside classification methods—are compared with previous approaches under identical conditions on a specific dataset concerning intrusion detection in IoT. The significance of this comparison stems from the challenges associated with heart disease detection, tracking, and identifying heart patients amidst heterogeneous datasets, primarily due to issues like imbalanced class distributions. Consequently, accuracy metrics that illustrate the relationship between actual samples and those identified by machine learning models serve as the optimal criterion for evaluating the proposed method. Figure 5 presents the bar chart corresponding to the accuracy metric in the proposed method juxtaposed with various classification methods and prior approaches. This method enhances the performance and accuracy of data classification by selecting essential and practical features based on their correlations. Utilizing the Harris Hawks empowers this approach to optimize and refine the performance of the KNN classification model. The selection of significant features through correlation analysis aids in improving the accuracy and precision of data classification, resulting in noteworthy improvements compared to previous methods. The results yielded by this proposed method with prior classification techniques indicate that CFS and the Harris Hawks facilitate superior performance and evident enhancements in the accuracy and effectiveness of data classification. These comparisons demonstrate that the proposed method can deliver more optimal and accurate data classification results while improving classification techniques' performance.

5. Conclusion

The IoT encompasses systems composed of various devices connected to the internet, facilitating information exchange between them. These devices can extensively and adaptively change based on user needs. Security and privacy challenges have markedly increased with the rising utilization of IoT devices. Connected IoT devices access sensitive information, thereby heightening the likelihood of cyberattacks. Thus, developing IDS to secure these devices and their communications is critical. IDS in IoT are designed to identify and prevent intrusions and attacks targeting devices and networks. Typically, these systems utilize various analytical algorithms and models to monitor activities at either local or cloud levels. They play a vital role in enhancing security and protecting devices and data linked to the IoT, assisting organizations in mitigating security threats and preventing attacks. Given the substantial volume of data and the diverse features of nodes within IoT networks, employing feature selection methods to increase the accuracy of IDS is essential. In this research, an intrusion detection system based on a combined filter-wrapper feature selection method, utilizing correlation analysis and the Harris Hawks, has been proposed. Results indicate that this method has optimized the performance of the KNN classification model, achieving an accuracy of 96.46%. The selection of significant features based on their correlations has improved accuracy in data classification, yielding substantial results compared to previous approaches. The comparison of this method with other classification techniques has revealed that the integration of CFS and Harris Hawks optimization results in significant improvements in the accuracy and performance of data classification systems, showcasing the ability to deliver more precise and optimized results in data classification.

References

[1] M. Wazid, P. Bagga, A. K. Das, S. Shetty, J. J. P. C. Rodrigues, and Y. Park, “AKM-IoV: Authenticated Key Management Protocol in Fog Computing-Based Internet of Vehicles Deployment,” *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8804–8817, 2019.

[2] Y. Salami, V. Khajehvand, and E. Zeinali, “SOS-FCI: a secure offloading scheme in fog–cloud-based IoT,” *J. Supercomput.*, vol. 80, no. 1, pp. 570–600, 2024, doi: 10.1007/s11227-023-05499-3.

[3] X. Mu and M. F. Antwi-Afari, “The applications of Internet of Things (IoT) in industrial management: a science mapping review,” *Int. J. Prod. Res.*, vol. 62, no. 5, pp. 1928–1952, 2024.

[4] M. Sugar and I. H. Berkovitz, “Fog Computing Conceptual Model,” *Adolesc. Psychiatry (Hilversum).*, vol. 1, no. 2, pp. 169–178, 2011, doi: 10.2174/2210677411101020169.

[5] M. Nassereddine and A. Khang, “Applications of Internet of Things (IoT) in smart cities,” in *Advanced IoT technologies and applications in the industry 4.0 digital economy*, CRC Press, 2024, pp. 109–136.

[6] Y. Salami, V. Khajehvand, and E. Zeinali, “SAIFC: A Secure Authentication Scheme for IOV Based on Fog-Cloud Federation,” *Secur. Commun. Networks*, vol. 1, pp. 1–19, 2023.

[7] A. Rajagopalan *et al.*, “Empowering power distribution: Unleashing the synergy of IoT and cloud computing for sustainable and efficient energy systems,” *Results Eng.*, p. 101949, 2024.

[8] Y. Salami, Y. Ebazadeh, and V. Khajehvand, “CE-SKE: cost-effective secure key exchange scheme in Fog Federation,” *Iran J. Comput. Sci.*, vol. 4, no. 3, pp. 1–13, 2021.

[9] A. Souri, M. Norouzi, and Y. Alsenani, “A new cloud-based cyber-attack detection architecture for hyper-automation process in industrial internet of things,” *Cluster Comput.*, vol. 27, no. 3, pp. 3639–3655, 2024.

[10] Y. Salami and S. Hosseini, “BSAMS: Blockchain-Based Secure Authentication Scheme in Meteorological Systems,” *Nivar*, vol. 47, no. 120–121, pp. 181–197, 2023.

[11] Y. Salami, F. Taherkhani, Y. Ebazadeh, M. Nemati, V. Khajehvand, and E. Zeinali, “Blockchain-Based Internet of Vehicles in Green Smart City: Applications and Challenges and Solutions,” *Anthropog. Pollut.*, vol. 7, no. 1, pp. 87–96, 2023.

[12] S. C. Vetrivel, R. Maheswari, and T. P. Saravanan, “Industrial IOT: Security Threats and Counter Measures,” in *Communication Technologies and Security Challenges in IoT: Present and Future*, Springer, 2024, pp. 403–425.

[13] Y. Salami, V. Khajevand, and E. Zeinali, “Cryptographic Algorithms: A Review of the Literature, Weaknesses and Open Challenges,” *J. Comput. Robot.*, vol. 16, no. 2, pp. 46–56, 2023.

[14] U. D. Maiwada, S. A. Imran, K. U. Danyaro, A. A. Janisar, A. Salameh, and A. B. Sarlan, “Security Concerns of IoT Against DDoS in 5G Systems,” *Int. J. Electr. Eng. Comput. Sci.*, vol. 6, pp. 98–105, 2024.

[15] Y. Salami, V. Khajehvand, and E. Zeinali, “A new secure offloading approach for internet of vehicles in fog-cloud federation,” *Sci. Rep.*, vol. 14, no. 1, p. 5576, 2024.

[16] Y. Salami and V. Khajehvand, “SMAK-IOV: Secure Mutual Authentication Scheme and Key Exchange Protocol in Fog Based IoV,” *J. Comput. Robot.*, vol. 13, no. 1, pp. 11–20, 2020.

[17] Y. Salami, V. Khajehvand, and E. Zeinali, “LSMAK-IOV: Lightweight Secure Mutual AKE Scheme in

Fog-Based IoV," in *2024 10th International Conference on Artificial Intelligence and Robotics (QICAR)*, IEEE, 2024, pp. 1–5.

[18] Z. Wang, J. Li, S. Yang, X. Luo, D. Li, and S. Mahmoodi, "A lightweight IoT intrusion detection model based on improved BERT-of-Theseus," *Expert Syst. Appl.*, vol. 238, p. 122045, 2024.

[19] O. B. J. Rabie, S. Selvarajan, T. Hasanin, A. M. Alshareef, C. K. Yogesh, and M. Uddin, "A novel IoT intrusion detection framework using Decisive Red Fox optimization and descriptive back propagated radial basis function models," *Sci. Rep.*, vol. 14, no. 1, p. 386, 2024.

[20] E. Altulaihan, M. A. Almaiah, and A. Aljughaiman, "Anomaly Detection IDS for Detecting DoS Attacks in IoT Networks Based on Machine Learning Algorithms," *Sensors*, vol. 24, no. 2, p. 713, 2024.

[21] M. M. Inuwa and R. Das, "A comparative analysis of various machine learning methods for anomaly detection in cyber attacks on IoT networks," *Internet of Things*, vol. 26, p. 101162, 2024.

[22] A. Aldhaheri, F. Alwahedi, M. A. Ferrag, and A. Battah, "Deep learning for cyber threat detection in IoT networks: A review," *Internet Things cyber-physical Syst.*, vol. 4, pp. 110–128, 2024.

[23] N. O. Aljehane *et al.*, "Golden jackal optimization algorithm with deep learning assisted intrusion detection system for network security," *Alexandria Eng. J.*, vol. 86, pp. 415–424, 2024.

[24] C. Hazman, A. Guezzaz, S. Benkirane, and M. Azrour, "Enhanced ids with deep learning for iot-based smart cities security," *Tsinghua Sci. Technol.*, vol. 29, no. 4, pp. 929–947, 2024.

[25] D. Li, L. Deng, M. Lee, and H. Wang, "IoT data feature extraction and intrusion detection system for smart cities based on deep migration learning," *Int. J. Inf. Manage.*, vol. 49, pp. 533–545, 2019.

[26] A. Elsaeidy, K. S. Munasinghe, D. Sharma, and A. Jamalipour, "Intrusion detection in smart cities using Restricted Boltzmann Machines," *J. Netw. Comput. Appl.*, vol. 135, pp. 76–83, 2019.

[27] T. Saba, "Intrusion detection in smart city hospitals using ensemble classifiers," in *2020 13th International Conference on Developments in eSystems Engineering (DeSE)*, IEEE, 2020, pp. 418–422.

[28] E. M. Onyema, S. Dalal, C. A. T. Romero, B. Seth, P. Young, and M. A. Wajid, "Design of intrusion detection system based on cyborg intelligence for security of cloud network traffic of smart cities," *J. Cloud Comput.*, vol. 11, no. 1, p. 26, 2022.

[29] M. M. Rashid *et al.*, "Adversarial training for deep learning-based cyberattack detection in IoT-based smart city applications," *Comput. Secur.*, vol. 120, p. 102783, 2022.

[30] M. Abdedaime, A. Qafas, M. Jerry, and A. Guezzaz, "A KNN-based intrusion detection model for smart cities security," in *International Conference on Innovative Computing and Communications: Proceedings of ICICC 2022, Volume 3*, Springer, 2022, pp. 265–272.

[31] M. Roopak, G. Y. Tian, and J. Chambers, "Multi-objective-based feature selection for DDoS attack detection in IoT networks," *IET Networks*, vol. 9, no. 3, pp. 120–127, 2020.

[32] A. Awajan, "A novel deep learning-based intrusion detection system for IOT networks," *Computers*, vol. 12, no. 2, p. 34, 2023.

[33] J. Li, M. Gao, and R. D'Agostino, "Evaluating classification accuracy for modern learning approaches," *Stat. Med.*, vol. 38, no. 13, pp. 2477–2503, 2019.

[34] J. Miao and W. Zhu, "Precision–recall curve (PRC) classification trees," *Evol. Intell.*, vol. 15, no. 3, pp. 1545–1569, 2022.

[35] Ž. Vujović, "Classification model evaluation metrics," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 6, pp. 599–606, 2021.

[36] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, pp. 1–13, 2020.

Table 3: Correlation Values between Features and Class Labels

Correlation	***Feature Name***	***Feature Number***	***Correlation***	***Feature Name***	***Feature Number***
0.0022	*HH_L0.1_pcc*	*13*	*-0.1574*	*MI_dir_L0.1_weight*	*1*
-0.1927	*HH_jit_L0.1_weight*	*14*	*-0.0670*	*MI_dir_L0.1_mean*	*2*
0.0904	*HH_jit_L0.1_mean*	*15*	*0.0324*	*MI_dir_L0.1_variance*	*3*
-0.0454	*HH_jit_L0.1_variance*	*16*	*-0.1574*	*H_L0.1_weight*	*4*
-0.0494	*HpHp_L0.1_weight*	*17*	*-0.0670*	*H_L0.1_mean*	*5*
-0.0417	*HpHp_L0.1_mean*	*18*	*0.0324*	*H_L0.1_variance*	*6*
-0.0390	*HpHp_L0.1_std*	*19*	*-0.1927*	*HH_L0.1_weight*	*7*
-0.0505	*HpHp_L0.1_magnitude*	*20*	*-0.0428*	*HH_L0.1_mean*	*8*
-0.0432	*HpHp_L0.1_radius*	*21*	*-0.0304*	*HH_L0.1_std*	*9*
-0.0516	*HpHp_L0.1_covariance*	*22*	*-0.0542*	*HH_L0.1_magnitude*	*10*
-0.0541	*HpHp_L0.1_pcc*	*23*	*-0.0575*	*HH_L0.1_radius*	*11*
-0.0368	*TnBPSrcIP*	*24*	*0.0182*	*HH_L0.1_covariance*	*12*

Table 4: Selected Features.

F-number	***F_Name***	***Correlation***	***F_number***	***F_Name***	***Correlation***
1	*MI_dir_L0.1_weight*	*-0.1574*	*11*	*HH_L0.1_radius*	*-0.0575*
2	*MI_dir_L0.1_mean*	*-0.0670*	*14*	*HH_jit_L0.1_weight*	*-0.1927*
4	*H_L0.1_weight*	*-0.1574*	*15*	*HH_jit_L0.1_mean*	*0.0904*
5	*H_L0.1_mean*	*-0.0670*	*20*	*HpHp_L0.1_magnitude*	*-0.0505*
7	*HH_L0.1_weight*	*-0.1927*	*22*	*HpHp_L0.1_covariance*	*-0.0516*
10	*HH_L0.1_magnitude*	*-0.0542*	*23*	*HpHp_L0.1_pcc*	*-0.0541*

Table 6. Average values of the evaluation metrics for various classification methods

Classification Method	***F1-Score***	***Accuracy***	***Sensitivity***	***Specificity***
KNN	*98.27*	*97.29*	*99.25*	*96.46*
Neural Networks	*63.57*	*53.47*	*91.48*	*86.18*
Decision Tree	*97.14*	*97.22*	*97.07*	*95.60*
Naive Bayes	*93.60*	*95.98*	*91.48*	*91.93*
Support Vector Machine	*85.30*	*94.32*	*77.93*	*79.33*

```
#define MAX_FEATURES Numbers
// Step 1: Feature Extraction
// This function takes sensor data and extracts relevant features.
void extract_features(float data[MAX_FEATURES][MAX_FEATURES], int n) {
   // Collect data and extract features
}
// Step 2: Calculate Correlation
// This function computes the correlation matrix based on the extracted features.
void calculate_correlation(float data[MAX_FEATURES][MAX_FEATURES], int n, float corr[MAX_FEATURES][MAX_FEATURES]) {
   // Calculate the correlation matrix
}
// Step 3: Identify Causal Relationships
// This function identifies causal relationships among the features using the correlation matrix.
void identify_causal_relationships(float corr[MAX_FEATURES][MAX_FEATURES], int n, int causal[MAX_FEATURES][MAX_FEATURES]) {
   // Identify causal relationships
}

// Step 4: Analyze Mutual Influence
// This function analyzes the mutual influence of features upon each other.
void analyze_mutual_influence(int causal[MAX_FEATURES][MAX_FEATURES], int n, float influence[MAX_FEATURES][MAX_FEATURES]) {
   // Analyze mutual influences
}

// Step 5: Optimize Causal Model
// This function optimizes the causal model based on the mutual influences.
void optimize_causal_model(int causal[MAX_FEATURES][MAX_FEATURES], int n, float influence[MAX_FEATURES][MAX_FEATURES], int optimized[MAX_FEATURES][MAX_FEATURES]) {
   // Optimize the causal model
}

// Step 6: Evaluate Model
// This function evaluates the optimized model and compares the results with other methods.
void evaluate_model(int optimized[MAX_FEATURES][MAX_FEATURES], int n) {
   // Evaluate and compare results of the optimized model
}
int main() {
   float sensor_data[MAX_FEATURES][MAX_FEATURES]; // Sensor data
   float correlation_matrix[MAX_FEATURES][MAX_FEATURES]; // Correlation matrix
   int causal_relationships[MAX_FEATURES][MAX_FEATURES]; // Causal relationships
   float mutual_influence[MAX_FEATURES][MAX_FEATURES]; // Mutual influences
   int optimized_causal_relationships[MAX_FEATURES][MAX_FEATURES]; // Optimized causal relationships
   int num_features =; // Number of features

   // Execute research steps
   extract_features(sensor_data, num_features);
   calculate_correlation(sensor_data, num_features, correlation_matrix);
   identify_causal_relationships(correlation_matrix, num_features, causal_relationships);
   analyze_mutual_influence(causal_relationships, num_features, mutual_influence);
   optimize_causal_model(causal_relationships, num_features, mutual_influence, optimized_causal_relationships);
   evaluate_model(optimized_causal_relationships, num_features); // Evaluate the model
   return 0;
}
```

Fig 4. pseudocode of the proposed method.

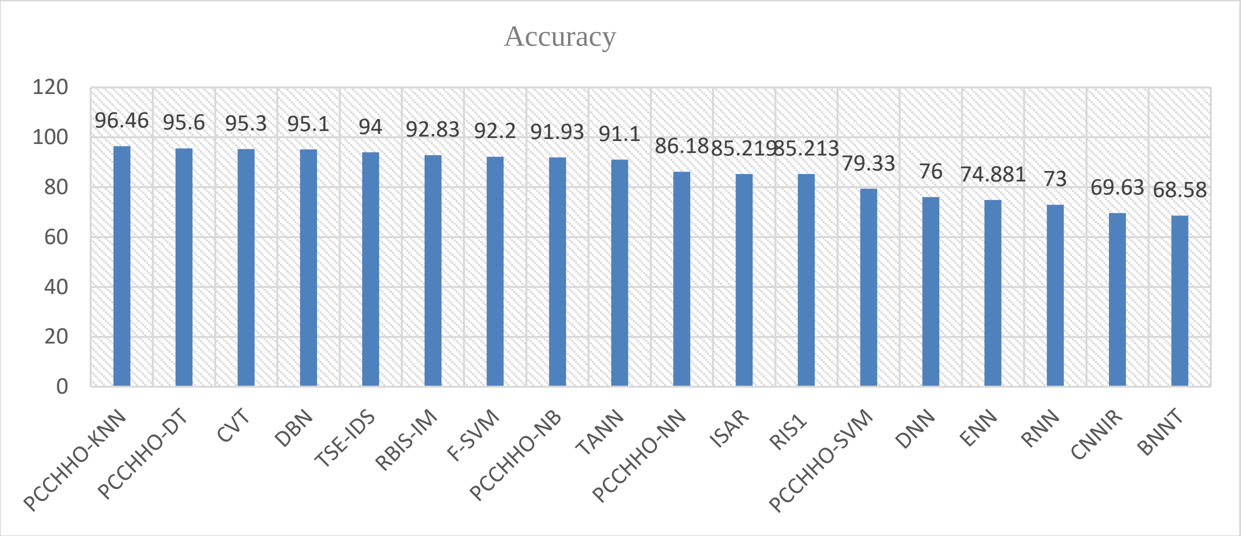

Fig 5. Comparing the accuracy of the proposed method with other methods.