Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 44 No. 6 (2023)

An Effective Clustering and CSRvNN Model for Intelligent Intrusion Detection for the Internet of Things

[1*]R.Anushiya and [2]Dr.V.S.Lavanya

[1*]*Assistant Professor, Department of Computer Science, P.K.R. Arts College for Women, Gobichettipalayam.*

[2]*Associate Professor, Department of Computer Science, P.K.R. Arts College for Women, Gobichettipalayam,*

***Abstract*:** The identification of network intrusions is the primary function of an Intelligent Intrusion Detection System (IIDS), and it is essential to maintaining the protection of the Internet of Things (IoT). Deep learning (DL) was quite successful in identifying intrusions. Although the actual deployment of DL-based high-complexity frameworks is hampered by the limited processing power and data storage of IoT equipment. Moreover, DL-based anomaly identification is frequently linked to high false alarms with reduce precision and detection rates if it is struggling to precisely identify a variety of threats. This work proposes a hybrid learning method through combination of Adaptive K-Means (AKM) clustering and Context Sensitive Recursive Neural Network (CSRvNN) for the recognition and sorting of IoT threads. The suggested method entails grouping all of the information into a relevant cluster before utilizing a classifier to do categorization. The training instances are initially divided into clusters utilizing the Manhattan distance and AKM. The network dataset's dimensionality was decreased, and in order to enhance the suggested approach, pertinent features were chosen from the dataset using the Assimilated Artificial Fish Swarm Optimization (AAFSO) method. The generated clusters, which show a density zone of normal or abnormality cases, used as the foundation for an FFO-CRvNN classification algorithm. This is useful in assessing how well the clustering finds hidden attack patterns in the data. The outcomes of the experiments demonstrate that the suggested model is more successful than the conventional categorization method in identifying assault patterns.

***Keywords**: Intelligent Intrusion Detection System, Internet of Things, feature selection and clustering, deep learning,*

Introduction

The IoT is a rapidly developing field that has gained traction in the past decade. It allows objects to communicate and interact with one another through a network, which is driving the development of new business processing technologies. The rapid evolution of cybersecurity threats has consequently brought to light a number of difficulties in various domains, including financial, credibility-building, enforcement, and commercial operations. Typically, cloud computing is utilized as an Internet of Things data storage solution, which is designed to provide customers with a range of services and resources. In general, cloud computing reduces the amount of human interaction amongst sources and users. Organizations and users have taken notice of it seriously because of its remarkable qualities [1]. Still may be a number of difficult security and operational platform transition challenges when switching from the present system to a cloud-based system. The precious information that resides remotely on servers is connected to cloud computing's insecurity.

Numerous individuals are prevented from choosing or migrating to a cloud-based because of this security risk, resulting in it appealing for many scammers and hackers. The frequency of these attacks has significantly increased for a number of factors. The primary causes are the availability of simple-to-use hacking devices, which

enable unsuspecting hackers to swiftly target cloud storage without the need for advanced training or specialization. Enormous advancements in communication systems and the creation of the IoT are the result of remarkable increases in regular utilization of technological apps and services. Devices function as units or "things" with the capacity to detect their surroundings, connect with one another, and exchange data via the Internet under the terms of the IoT. [1]. One trillion IP addresses linked to the Internet via IoT networks by 2022 [2].

Smart homes and cities, with a range of usage fields and associated services, were developed lately utilizing the IoT model. Through the resolution of problems with energy use, and industrial requirements are intended to improve output [3]. The notable extension of IoT-based apps and services accessible through numerous networks is a direct result of this objective. IoT devices are vulnerable to a range of security threats, including distributed denial-of-service (DDoS) and denial-of-service (DoS) attacks. In an IoT network, these kinds of assaults have the possibility of significant harm smart environment systems and IoT services [4]. Subex, a Bengaluru-based company, has amassed over 4,000 IoT sensors in its "honeypot" network, which is a ruseful computer system utilized to trap hackers. 33,450 high-grade attacks were tracked by Subex researchers throughout the June quarter, 500 of which were classified as "very high sophistication" [5].

An IoT system's network layer is where an IDS operates. When implemented for an IoT system, an IDS should be capable of real-time analysis of data packets and response generation, analysis of data packets in various IoT network levels with varying protocol stacks, and adaptability to various IoT solutions [6]. For IoT-based smart settings, an IDS must function in harsh situations with limited processing power, quick reaction times, and large volumes of data processing. For this reason, traditional IDSs might not be entirely appropriate for IoT setups. IoT security is a persistent and significant problem, so it's important to stay current on knowledge about IoT system security flaws and to find solutions for problems [7].

A great deal of neglect was shown in recent decades to many problems in the field of cyberattacks, including IDS [7]. In addition, a number of machine learning (ML) approaches, with the decision tree algorithm (DT), support vector machine (SVM) frameworks in, k-means, k-nearest neighbor (kNN), and numerous ML methods, were applied to solve the cybersecurity concerns [7]. Its main objective is to present a new and effective IIDS framework that combines clustering with classification and the best feature selection technique. In order to accomplish these goals, this work presents the following contributions:

- To begin with, the UNSW Bot-IoT data set was determined to be the most pertinent IIDS for IoT data set. Subsequently, pre-processing transforms the data-set into an executable form for the method.
- Create an optimal feature selector framework utilizing an AAFSO framework to identify the best feature from the preceding IoT datasets.
- The training set as a whole is broken into smaller, more manageable subsets by AKM clustering, allowing the CSRvNN to learn each subset more rapidly, reliably, and accurately.
- Lastly, the network traffic is classified as Attack or Normal employing the CSRvNN technique, which takes the features of the data set as inputs.

The following is the arrangement of the following sections of this work: The relevant IDS structure study is reviewed in Section 2. The suggested IoT security model is then explained in Section 3. In Section 4, the findings and a commentary are provided. In Section 5, the study's conclusion is finally expressed, and potential further research is suggested.

Related work

Keserwani et al., [8] proposed an IDS to recognize different types of assaults on IoT networks. To extract pertinent IoT network properties, GWO and PSO are integrated. To attain high assault detection accuracy, a RF classifier is given the retrieved data. Nevertheless, it makes use of a huge quantity of data and attributes that are unneeded, irrelevant, and unsuitable, which results in a long recognition time and mediocre accuracy.

In [9], The study uses a recently developed method called Fuzzy OPF, which depends on fuzzy logic and graphs, to identify threats that evade the normal traffic on an IoT network. However, there is still a gap in the literature

on legitimate uses of attack recognition in IoT environments, which often represents a difficult task made up of several assault types.

Alweshah et al., [10] proposed a wrapper FS framework that solves FS for IoT problems using a K-nearest neighbor classifier and the EPC approach to investigate the problematic space. The efficacy of the suggested EPC framework was assessed in simulations utilizing nine popular IoT datasets. The algorithm achieved 98% rate of classification, which was undoubtedly superior to that of the MOPSO and MOPSO-Lévy approaches with regard to accuracy and FS size. To attain the best level of protection, the transmitted data filtered and chosen according to the type of the problem addressed.

Alazzam et al., [11] introduced a thin-client IDS with a low false alarm rate and a high detection rate. The suggested NIDS combines two primary subsystems that operate concurrently. OCSVM is used to teach each subsystem. One system develops utilizing regular packets, and the other uses attack packets. Each packet that travels across the network is given an adequate estimate by combining the output of the two subsystems. These notifications become uncontrollable and have a high false alarm rate in large networks.

Al Shorman et al., [12] suggested an unsupervised evolving system for detecting IoT botnets. The primary goal of the suggested approach is to identify IoT botnet attacks that originate from undermined IoT hardware. It takes advantage of the effectiveness of a recently developed GWO, which allows for the optimization of the OCSVM's hyperparameters while recognizing the characteristics and characterize the IoT botnet issue. The effectiveness of the recommended approach is assessed utilizing standard anomaly recognition assessment metrics over an updated version of an actual benchmark dataset in order to demonstrate its efficacy. Additionally, it minimizes the amount of selected features while achieving the lowest detection time. However, it retains certain drawbacks not being confirmed with more recent botnets.

Davahli et al., [13] presented a high-performing, portable ML-based IDS for IoTIDS. The hybridization of the GA with the GWO, known as GA–GWO, is the basis of IoTIDS. The primary goal of the mixed approach for the IoTIDS is to perceptively select the relevant traffic attributes to minimalize the dimensionality of the enormous amount of Wi-fi network data. All the same, the given findings can be regarded as statistically insignificant because relatively little of the data was utilized for training.

Zeeshan et al., [14] suggested a PB-DID design, compared elements from the UNSWNB15 and Bot-IoT data-sets according to flow and TCP to build a data-set of packets. It achieves the distinctive classification of non-anomalous, DDoS, DoS traffic by addressing problems such as unbalanced and over-fitting. Utilizing the DL approach 96.3% classification accuracy obtained; the packet payload and attribute extraction remain opaque.

Liu et al., [15] proposed a PSO-LightGBM for the reason of detecting intrusions. This approach employs OCSVM to locate and identify destructive data after PSO-LightGBM is employed to extract the features of the data. The IDS is validated using the UNSW-NB15 dataset. Because IoT terminals and applications are becoming more diverse and integrated, they are more susceptible to different types of intrusion challenges.

Alsaedi et al., [16] proposed an innovative data-driven IoT/IIoT dataset with ground truth that includes sub-classes of threats, as well as a type feature specifies normal and attack classes for multi-classification issues. The suggested dataset, called TON_IoT, was gathered from a realistic simulation of a medium-scale network at the Cyber Range and IoT Labs at UNSW Canberra. It contains Telemetry data of IoT/IIoT services, Os logs, and Network traffic of IoT network. The suggested dataset of IoT/IIoT service telemetry data and its features are also described in this work; however, the real data derived from sensor measurements is missing.

Summary: Many research has developed a framework utilizing one data set and then tested or validated it utilizing another data set; however, very few have examined the properties of two or more data sets and then integrated data to produce a novel data set. Furthermore, this is the first time that feature selection for two data sets was done using the network layer. This study's contribution consists of classifying and clustering harmful and benign packets.

Proposed Methodology

Although IoT gadgets are low-power and rarely need a lot of processing power, maintaining data security is a key task. In general, an IIDS is utilized to detect and prevent unwanted packets from accessing the network. Next, employing features from the BoT-IoT and UNSW-NB15 data sets, utilize AAFSO for feature selection to identify the best elements. After that, suggest feature clusters in accordance with Flow, Message Queuing Telemetry Transport (MQTT), and Transmission Control Protocol (TCP). After using AKM clustering to remove issues with the data set such as imbalance, over-fitting, and the expletive of dimensionality, the clusters are subjected to the CSRvNN method. RF on Flow & MQTT functions, TCP structures, and top characteristics from both clusters were employed in cluster-based approaches to obtain excellent classification accuracy, displayed in Fig. 1. In addition, the suggested featured clusters outperform other cutting-edge DL-based methods as a result of accuracy and training time.

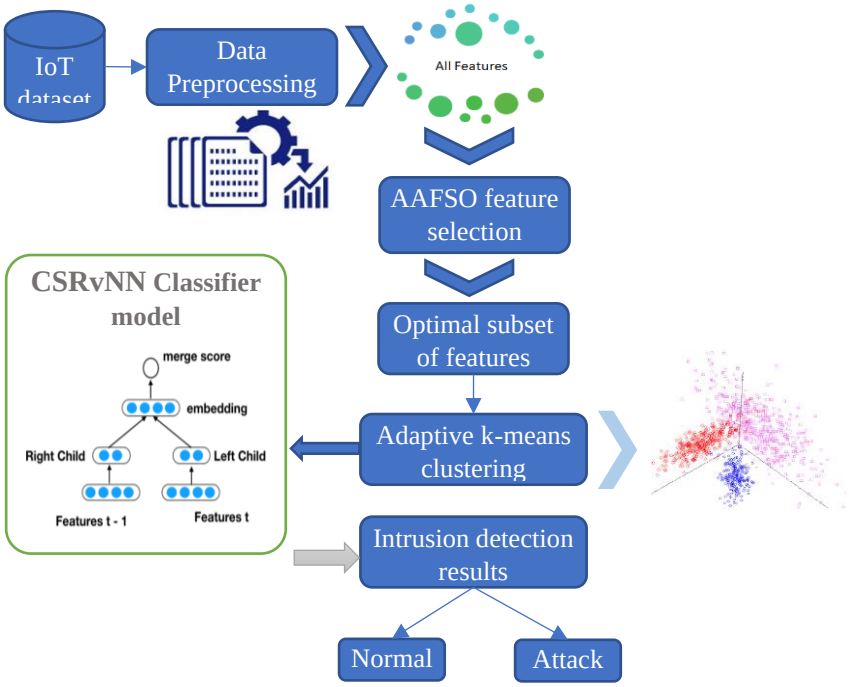

Fig.1. The general framework diagram of proposed IIDS based on clustering and classification

1.1. Input Data collection and feature selection

The raw network traffic must be gathered in the first stage. Utilize the BoTNeTIoT-L01 [19] publicly accessible dataset and UNSW-NB15 [20] as the sources of raw traffic in this study. Attacks are denoted by 0 in the dataset class label and normal samples by 1.

Feature extraction and Description: Obtaining packet-level labels and extracting pertinent fields, each field refers to a feature from these packets in the unprocessed network traffic constitutes the second stage. primarily utilizing aggregated packet data, the features employed in this research depend on header field data from individual packets. Broad traffic characteristics could be captured, as opposed to producing attack-specific data that would only be useful for identifying certain attack kinds. After that, the 344 PCAP files were analyzed using the TShark application to extract the header fields of individual packets. The packets were then labeled and saved as CSV files. thereby, the PCAP files were limited to IP packets, and a 29 packet header fields were obtained. Since ARP packets are needed to convert IP addresses to MAC addresses and are unrelated to the attacks that are suggested in the dataset, they were eliminated.

AAFSO based feature selection step: Each parameter in the dataset contributes to prediction accuracy, thus choosing the most essential characteristics to include in the model is a difficult issue. For the purpose of choosing important data, a feature selection method that utilizes metaheuristic techniques is employed. This paper proposes

the AAFSO metaheuristic method, which was motivated by the behavior of fish underwater. The objective space is given in Fig.2 for 20 iterations and the implementation of the ensuing subset of features:

- kind of service (e.g., web, ftp, smtp,. etc) (service)
- Quantity of bytes from source to end point (sbytes)
- time to live from source to end point (sttl)
- communicated packet size mean by the source (smean)
- The quantity of rows in 100 rows that are the same sport and dstip (ct_dst_sport_ltm)

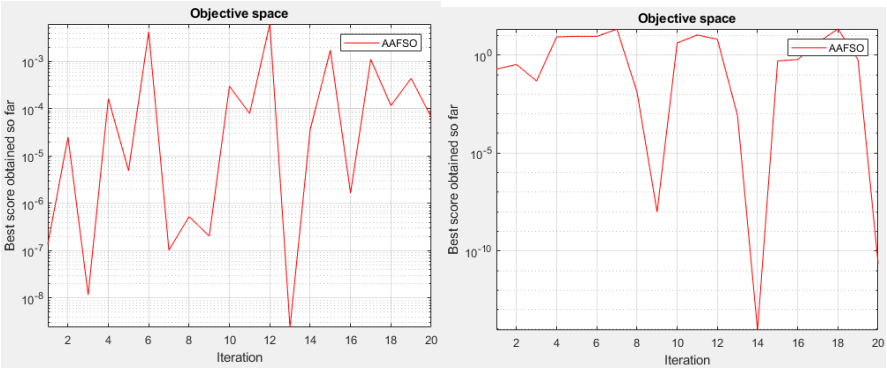

Fig.2. Objective space analysis of feature selection method for UNSW-NB15 and BoTNeTIoT-L01

Adaptive K-means based feature clustering

AKM algorithm is a wholly automated means of clustering an input feature employing k-means concept, however, hereit is not necessary to mention the number of clusters or any initial feature value to begin the iteration, this algorithm automatically gets the number of clusters and cluster center by iterative means. It is an extremely rapid realization of clustering of an image with no knowledge of the number of clusters. Initially, cluster the features as in k-means, herein AKM based method not necessary to mention the number of clusters for clustering and thus eextremely rapid implementation and also ssimple to interpret and the flow diagram of AKM based clustering is shown in Fig.2. The finest characteristics are grouped in AKM clustering, which is performed by using a single layer service or on features relying on packet and flow. Six clusters were created in this instance: DNS, FTP, HTTP, MQTT, TCP, and flow. As a simple message protocol for devices with limited bandwidth, MQTT has been designed. For IoT apps, it is hence perfect. MQTT tracks efficiency and enables instructions to be transmitted via sensor nodes.

The top five and top six attributes from flow / MQTT and TCP were retained, after feature importance was determined. The AKM clustering algorithm begins with choosing the K components from the input data set. These K components make the seeds of clusters and are chosen in random. The features of every component also make the attributes of the cluster, which is defined by the component. The algorithm depends on the capability of computing the distance amongst a certain element and cluster. This function is also used for computing the distance between two components. One more significant aspect for this function is that it must be capable of measuring the distance in accordance with attributes, which have been normalized such that the one property dominates the distance or some attributed is not neglected when the distance is computed. In several scenarios, the Euclidean distance may be enough. For instance, if spectral data definite by ‘n’ dimensions is considered, the distance amongst two data elements $\mathbb{E}_1$ and $\mathbb{E}_2$, is equivalent to

$$Dist(\mathbb{E}_1, \mathbb{E}_2) = \sqrt{(\mathbb{E}_{11} - \mathbb{E}_{12})^2 + (\mathbb{E}_{12} - \mathbb{E}_{22})^2 + (\mathbb{E}_{1n} - \mathbb{E}_{2n})^2},$$

Where $\mathbb{E}_1 = \{\mathbb{E}_{11}, \mathbb{E}_{12}, \ldots, \mathbb{E}_{1n}\}$ and $\mathbb{E}_2 = \{\mathbb{E}_{21}, \mathbb{E}_{22}, \ldots, \mathbb{E}_{2n}\}$. It must be observed that owing to superior performance, the square root function may be ignored. In other scenarios, it is required to change the distance function. These cases can be depicted by data in which the scaling of one dimension is done differently in

comparison with other dimensions or it may be necessary that the attributes may uniquely weights while comparing. Using the distance function, the algorithm is as given:

- Compute the distance of every cluster from all the other clusters and stored in the form of a triangular matrix in a 2D array. Minimum distance $Dist_{min}$ amongst any two clusters Cl_{m1} and Cl_{m2} and also recognition of these two nearest clusters is taken into account.
- For every non clustered element $\mathbb{E}_i$, distance of $\mathbb{E}_i$ from each cluster must be computed. To assign this element to a cluster, there may be three criteria listed as follows:

A. If the distance element and the cluster is 0, get the element assigned to that cluster, and start operating on the next element.

B. In instance the distance amongst the element and cluster is lesser than the distance $Dist_{min}$, get this element assigned to its closest cluster. Consequent to this allocation, the cluster format, or centroid, may vary. The centroid is recomputed in the form of an average of characteristics of all the components present in the cluster. Moreover, the distance of the impacted cluster from all the other clusters, in addition to the minimum distance amongst any two clusters which are nearest to one another is recalculated.

C. If the distance $Dist_{min}$ is less compared to the distance of the component from the adjoining cluster, choose the two nearest clusters C_{m1} and C_{m2}, and merge v into C_{m1}. Eliminate the cluster C_{m2} by demolishing every element in the cluster and by removing its representation and include the new element into this void cluster, thereby creating a novel cluster with success. The distances amongst all the clusters are recalculated and the two nearest clusters are found once more.

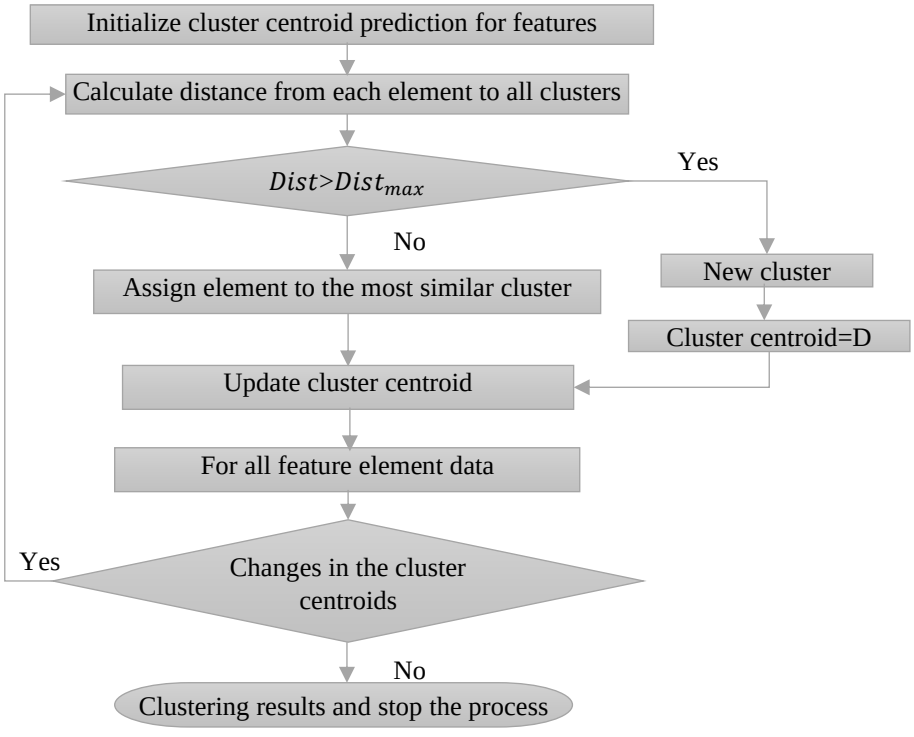

Fig.2. The flow diagram of proposed AKM clustering

IIDS Model Based on CSRvNN Model

An overall learning framework is proposed in this part with the goal of identifying organisation in feature-length inputs. Two steps are taken with a sequence of features as the input: To capture specific context statistics, the framework first gives a continuous form to each feature in an n-dimensional embedding space. These representations can be pretrained on unlabeled corpora or started arbitrarily. Second, to estimate the probability that these two traits will be integrated by assigning a score to each pair of surrounding words utilizing a neural

network. The modeled versions of the two features will serve as the only input to this network, and it will collapse the two features into an n-dimensional representation of the end result as a byproduct of calculating the syntactic score. The features in the sequence are then replaced by this new feature embedding, which may eventually become a child of another feature that spans additional attributes. Until all of the input features are mapped to the embedding space, the procedure is repeated.

RvNN: Recursive neural networks (RvNNs) interpret structured inputs differently from normal neural networks since they apply the same neural network twice at each node of a directed acyclic graph (DAG) [21]. They were previously limited to applications where directed acyclic graphs were initially generated by another often symbolic element. These DAGs were then sent into the RvNN as input. In this setup, the DAG's non-leaf nodes are all connected to the same neural network. Stated differently, all network replications have identical weights. All of these replicated feedforward networks receive their input characteristics from either the children's already computed representation through the use of their labels to find the corresponding representation. Although n-arry DAGS can theoretically be employed as the RvNN's input, for reasons of clarity, presently concentrate on binary trees. Let's say that the input consists of feature vectors $(x_1, \ldots, x_n)$, all of which have dimensions $x_i \in \mathbb{D}^n$. These types of trees indicate that a parent node $\mathfrak{p}$, has two offspring, and that each cv_k may be a non-terminal node in the tree or an input x_i. These were one-on-one vectors in the initial formulation [22]. Notice that the parent requires the same dimensionality as each of its offspring for them to recreate the neural network and generate node representations in a bottom-up manner. Calculate the activations of each node starting at the bottom and working your way up the tree structure.

$$\mathfrak{p} = \tan h(W[cv_1 \parallel cv_2] + b)$$

where cv_1, cv_2, $\mathfrak{p} \in \mathbb{D}^{n\times 1}$, the phrase $[cv_1 \parallel cv_2]$ signifies the concatenation of the two child column vectors ensuing in a $\mathbb{D}^{2n\times 1}$ vector and hence weight $W \in \mathbb{D}^{n\times 2n}$. The regression layer is able to be stacked at the top.

Proposed CSRvNN: One of CSRvNN's primary benefits is that each phrase has a distributed feature representation associated with its content. This representation can be leveraged by adding a simple softmax layer to each CSRvNN parent node, which is added after the scoring layer sc is removed, to predict class labels like normal or attack. This additional layer might take context into account, just like the contextualized collapsing choice did. The mistake will backpropagate and affect the CSRvNN settings along with the feature representations of CSRvNN in binary tree form, as shown in Fig. 3, when the cross-entropy error of this softmax layer is minimized.

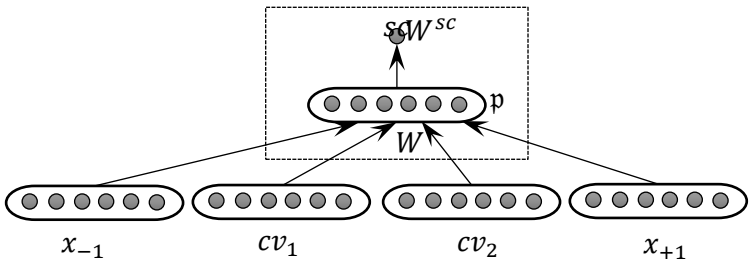

Fig.3: The diagram of CSRvNN Model

The context-sensitive recursive neural network, that is repeated for every pair of potential input feature vectors, is depicted in Figure 3. The surrounding words' context units are indicated by dashed lines. Compared to the original RvNN structure, this network forecasts a score sc for being a correct feature constituent, utilizes context feature, and permits several layers at each node. A parent node $\mathfrak{p}$ with two children is indicated by each such triplet, and each cv_k might be either an input x_i or a non-terminal node in the tree. Calculate the activations of each node starting at the bottom and working your way up the tree structure.

$$\mathfrak{p} = \tan h(W[concat(x_{-1}; c_1; c_2; x_{+1})] + b)$$

Where the term $concat$ denotes the concatenation. Next, add up the scores of all the decisions that collapsed to get the overall score sc of every tree:

$$SC = W^{sc} \cdot \mathfrak{p}$$

Ideally, CSRvNN merely needs to recurse over the tree depth to attempt to learn and analyse every child concurrently. This implies that in order to determine whether the induced tree-depth has been reached and whether it can stop, requires a mechanism. All existential probabilities, with the exception of the final, should be very near to 0 when the optimal moment for stopping approaches (0 signifies that it is no longer in need of more processing). Nothing can be constructed to the right of the final position. Therefore, make sure that the final position always has a composition probability of 0 and an existential probability of 1. In general, we just check to see if all existential possibilities, save the final one, are smaller than a tiny threshold th that indicates when to stop.

Experimental results and discussion

The Bot-IoT and UNSW-NB15 data sets are combined to form a single customized data set, which can be utilized by the suggested IIDS framework to train the CSRvNN utilizing 26 features. Both binary and multi-class classifications have an average classification accuracy that is higher than 96%. The findings demonstrate the applicability of the suggested method in real-world Internet of Things settings and its improved accuracy in detecting various forms of intrusions through the use of flow and TCP features. Each object in the study is defined by 41 attributes that come together to form a vector. Keep in mind that certain features are nominal and some are continuous. These nominal values will first be transformed to continuous values while the clustering and classification techniques need continuous values. The training data should be divided into subsets for the fuzzy clustering component utilizing the AKM clustering component, and CSRvNN should be utilized for intrusion detection.

Evaluation Metrics: The efficacy of ML techniques was assessed using a number of measures on the suggested IIoT dataset. Specifically, F-score, recall, accuracy, and precision were utilized to assess the efficacy of the techniques chosen quantitatively, including the suggested CSRvNN, GWO-PSO-RF [17], MOPSO-Lévy-KNN [18], and AAFSA with GA-FR-CNN. The fraction of correctly determined normal and attack data is the accuracy statistic, which indicates a model's overall efficacy. The recall measure displays the ratio of successfully identified assaults to all attack observations in the test dataset. The precision measure displays the proportion of accurately identified attack notes relative to all attacks that were detected. The harmonic (equally-weighted) mean of precision and recall is determined by the F-score.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100$$

$$\text{F} - \text{measure} = 2 * \left(\frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}\right)$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \times 100$$

the number of actual attack records that are properly recognized as attacks is identified as True Positive (TP), the quantity of actual normal data that has been recognized as normal is identified as True Negative (TN), the number of actual attack cases that are incorrectly categorized as normal is identified as False Negative (FN), and the quantity of actual usual events that are mistakenly detected as attacks is identified as False Positive (FP). The numerical results of four parameters for existing and proposed based on UNSW-NB15 and BoTNeTIoT dataset is given in Table 1 and 2 individually.

Table 1. The numerical results of four parameters for four methods- UNSW-NB15 dataset

	GWO-PSO-RF	MOPSO-Lévy-KNN	AAFSA with GA-FR-...	CSRvNN
Accuracy	84.4600	87.1250	94.4880	96.2611
Precision	80.8560	85.7890	94.2942	97.1640
Recall	79.4320	86.6540	94.5631	96.2888
F- Measure	79.0100	87.3450	94.4284	96.7244
Error	15.5400	14.8750	5.5120	3.7389

Table 2. The numerical results of four parameters for four methods- BoTNeTIoT dataset

	GWO-PSO-RF	MOPSO-Lévy-KNN	AAFSA with GA-FR-...	CSRvNN
Accuracy	83.7800	89.8500	93.7756	95.2483
Precision	81.5600	83.9000	86.6687	86.6694
Recall	82.3200	86.5400	95.8740	96.8494
F- Measure	80.9100	85.4500	91.0393	91.4770
Error	16.2200	10.1500	6.2244	4.7517

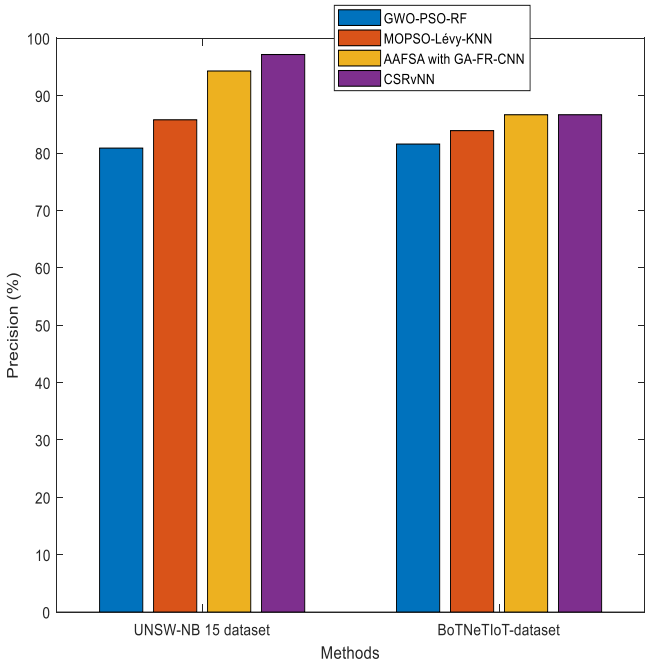

Fig.4. Result of Precision

Figure 4 indicates the accuracy of current and suggested algorithms for the quantity of characteristics in the provided databases. The precision is maximized in combination with the growing quantity of features. Utilizing UNSW-NB15 and BoTNeTIoT, the CSRvNN offers a precision of 97.1640% and 86.6694%, accordingly, when contrasted with MOPSO-Lévy-KNN, GWO-PSO-RF, and AAFSA with GA-FR-CNN. This is so that the AAFSA with GA-FR-CNN can find a slightly greater organized collection of input within a given time frame without requiring high-dimensional features or derived factors. In general, it was observed that employing the AKM clustering method, the suggested classification produced superior outcomes and increased the CSRvNN's detection precision.

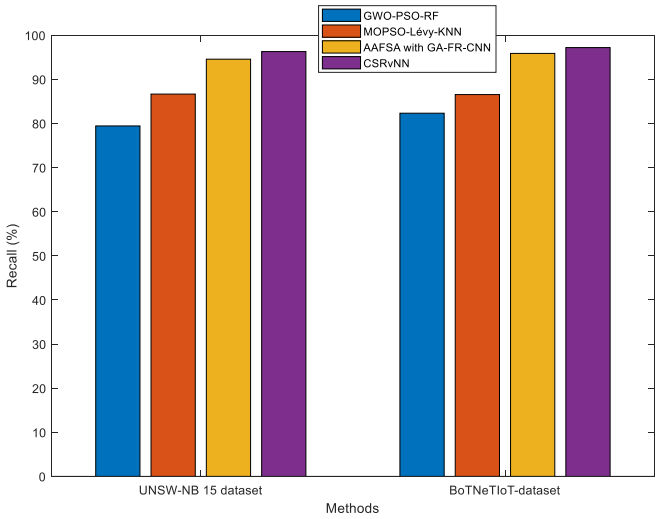

Fig.5. Result of Recall

The recall of optional and present estimations for features measure in a specified database is revealed in Fig. 5. Maximizing the quantity of characteristics also maximizes the recall. Utilizing UNSW-NB15 and BoTNeTIoT, for example, the CSRvNN offers a precision of 96.2888% and 96.8494%, accordingly, when contrasted with MOPSO-Lévy-KNN, GWO-PSO-RF, and AAFSA with GA-FR-CNN. This is so that CSRvNN fine-tuning can be done as easily as possible because the AAFSA lessens the computation time of the produced features. In general, the deep learning methods produced the best outcomes when it came to identifying intrusions from IoT network data. Note that the recall of CSRvNN is more because AKM clustering reduce the training time will increase the recall measure.

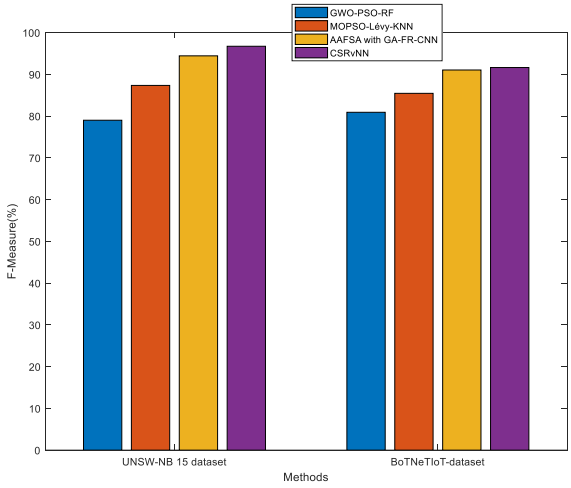

Fig.6. Result of F-measure

The f-measure for current and suggested frameworks for features value in provided databases is revealed in Fig. 6. The f-measure is optimized in tandem with feature count maximization. Utilizing UNSW-NB15 and BoTNeTIoT, CSRvNN offers a precision of 96.7244% and 91.4770%, accordingly, when contrasted with MOPSO-Lévy-KNN, GWO-PSO-RF, and AAFSA with GA-FR-CNN. Furthermore, a reduction in training time can be attained if the CSRvNN module is able to function in parallel. The AKM clustering element, which divides a heterogeneous training set into many homogeneous subsets and affects the detection F-measure, is primarily responsible for this enhancement.

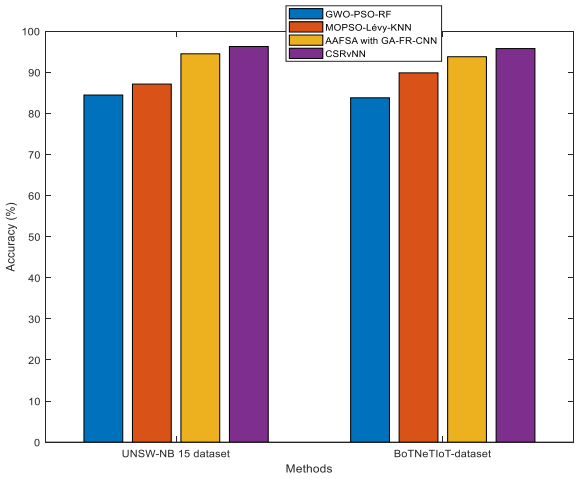

Fig.7. Result of Accuracy

The accuracy of suggested and present frameworks for features value in stated databases can be found in Fig. 7. The AAFSA with GA-FR-CNN shortens processing times while improving accuracy. In comparison to MOPSO-Lévy-KNN, GWO-PSO-RF, and AAFSA with GA-FR-CNN, CSRvNN offers an accuracy of 96.2611 and 95.2483% for UNSW-NB15 and BoTNeTIoT, accordingly. For this reason, it doesn't need an excessive quantity of resulting factors. Recall, precision, and F-value are steady for high-frequency attacks (DoS, PRB) and increase with increasing k for low-frequency attacks (R2L, U2R). This indicates that CSRvNN can achieve greater reliability and detection accuracy.

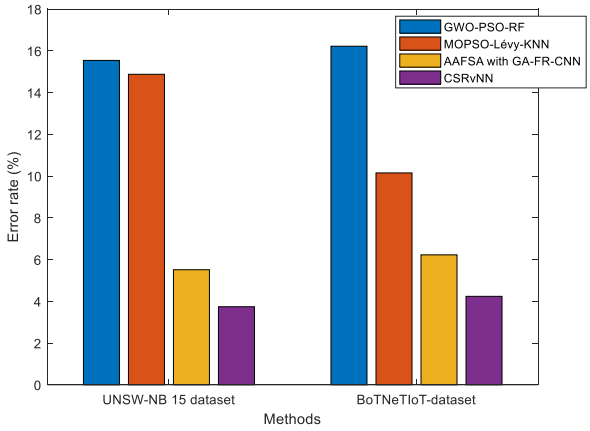

Fig.8. Error rate comparison results

The f-measure for current and suggested frameworks for features value in provided databases is revealed in Fig. 8. The f-measure is optimized in tandem with feature count maximization. For example, when equated to the MOPSO-Lévy-KNN, GWO-PSO-RF, and AAFSA with GA-FR-CNN, CSRvNN offers a precision of 3.7389 and 4.7517 for UNSW-NB15 and BoTNeTIoT, accordingly. Furthermore, with improved feature selection and clustering techniques, less training time can be obtained if the CSRvNN module can function in parallel. The AKM clustering component, which divides a heterogeneous training set into many homogeneous subsets and affects the detection F-measure, is primarily responsible for this enhancement.

Conclusion and future work

It is impractical to expect to prevent security breaches entirely using the security technologies that are now available. Monitoring for intrusions is therefore a crucial part of network security. IDS has the ability to reduce the number of people required for evaluation, increase detection efficacy, provide data that would not otherwise

be available, aid in the discovery novel vulnerabilities by the information security community, and provide proof for legal purposes. This study suggests a novel AKM clustering-based IDS named CSRvNN. The heterogeneous training set is separated into many homogenous subsets utilizing the fuzzy clustering method. As a result, each subtraining set's complexity is decreased, improving detection efficiency as a result. By deleting a small number of features and shrinking the data set, this study eliminated some problems with the entire data set, like over-fitting, dimensionality curse, and imbalanced data. Through the use of CSRvNN methods to evaluate the outcomes, the accuracy differences between employing clusters and the other features are demonstrated. In general, CSRvNN utilizing AKM oriented clustering produced the highest accuracy score of 96% in binary categorization on the entire data set. All things considered; these findings might open up possibilities for the future creation of reliable detection techniques for the particular datasets.

Further research might be conducted in the future to enhance the baseline techniques' effectiveness on the suggested datasets. To improve outcomes, one might employ sophisticated parameter optimization techniques like evolutionary algorithms and Bayesian optimisation to maximize the framework's hyperparameters. The issue concerning how to calculate the right number of clusters for further study is yet unanswered. Other problems could be worked on in the future; for instance, the A's convergence speed needs to be accelerated. To address this issue, the AAFSA utilized with additional artificial searches. Additionally, in subsequent research, the AAFSA utilized for training DL models that will improve the method of classification for other applications, such as IDS.

References

[1] Wheelus, C., & Zhu, X. (2020). IoT network security: threats, risks, and a data-driven defense framework. *IoT*, *1*(2), 259-285.

[2] Vermesan, O., & Friess, P. (Eds.). (2022). *Internet of things applications-from research and innovation to market deployment*. CRC Press.

[3] Kabalci, Y., Kabalci, E., Padmanaban, S., Holm-Nielsen, J. B., & Blaabjerg, F. (2019). Internet of things applications as energy internet in smart grids and smart environments. *Electronics*, *8*(9), 972.

[4] Salim, M. M., Rathore, S., & Park, J. H. (2020). Distributed denial of service attacks and its defenses in IoT: a survey. *The Journal of Supercomputing*, *76*(7), 5320-5363.

[5] The times of India, Available at: https://sectrio.com/media-coverage/india-sees-most-iot-attacks-in-apr-jun/

[6] Alhowaide, A., Alsmadi, I., & Tang, J. (2021). Ensemble detection model for IoT IDS. *Internet of Things*, *16*, 100435.

[7] Zarpelão, B. B., Miani, R. S., Kawakani, C. T., & de Alvarenga, S. C. (2017). A survey of intrusion detection in Internet of Things. *Journal of Network and Computer Applications*, *84*, 25-37.

[8] Keserwani, P. K., Govil, M. C., Pilli, E. S., & Govil, P. (2021). A smart anomaly-based intrusion detection system for the Internet of Things (IoT) network using GWO–PSO–RF model. *Journal of Reliable Intelligent Environments*, *7*(1), 3-21.

[9] Xu, Y., de Souza, R. W., Medeiros, E. P., Jain, N., Zhang, L., Passos, L. A., & de Albuquerque, V. H. C. (2022). Intelligent IoT security monitoring based on fuzzy optimum-path forest classifier. *Soft Computing*, 1-10.

[10] Alweshah, M., Hammouri, A., Alkhalaileh, S., & Alzubi, O. (2022). Intrusion detection for the internet of things (IoT) based on the emperor penguin colony optimization algorithm. *Journal of Ambient Intelligence and Humanized Computing*, 1-18.

[11] Alazzam, H., Sharieh, A., & Sabri, K. E. (2022). A lightweight intelligent network intrusion detection system using OCSVM and Pigeon inspired optimizer. *Applied Intelligence*, *52*(4), 3527-3544.

[12] Al Shorman, A., Faris, H., & Aljarah, I. (2020). Unsupervised intelligent system based on one class support vector machine and Grey Wolf optimization for IoT botnet detection. *Journal of Ambient Intelligence and Humanized Computing*, *11*(7), 2809-2825.

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

[14] Zeeshan, M., Riaz, Q., Bilal, M. A., Shahzad, M. K., Jabeen, H., Haider, S. A., & Rahim, A. (2021). Protocol-Based Deep Intrusion Detection for DoS and DDoS Attacks Using UNSW-NB15 and Bot-IoT Data-Sets. *IEEE Access*, *10*, 2269-2283.

[15] Liu, J., Yang, D., Lian, M., & Li, M. (2021). Research on intrusion detection based on particle swarm optimization in IoT. *IEEE Access*, *9*, 38254-38268.

[16] Alsaedi, A., Moustafa, N., Tari, Z., Mahmood, A., & Anwar, A. (2020). TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems. *IEEE Access*, *8*, 165130-165150.

[17] Keserwani, P. K., Govil, M. C., Pilli, E. S., & Govil, P. (2021). A smart anomaly-based intrusion detection system for the Internet of Things (IoT) network using GWO–PSO–RF model. *Journal of Reliable Intelligent Environments*, *7*(1), 3-21.

[18] Habib, M., Aljarah, I., & Faris, H. (2020). A modified multi-objective particle swarm optimizer-based Lévy flight: An approach toward intrusion detection in Internet of Things. *Arabian Journal for Science and Engineering*, *45*(8), 6081-6108.

[19] BoTNeTIoT-L01, link: https://www.kaggle.com/datasets/azalhowaide/iot-dataset-for-intrusion-detection-systems-ids.

[20] UNSW-NB 15 dataset: link: https://www.kaggle.com/datasets/mrwellsdavid/unsw-nb15?select=UNSW-NB15_1.csv

[21] Bowman, S. R., Potts, C., & Manning, C. D. (2014). Recursive neural networks can learn logical semantics. *arXiv preprint arXiv:1406.1827*.

[22] Chen, X., Zhou, Y., Zhu, C., Qiu, X., & Huang, X. J. (2015, September). Transition-based dependency parsing using two heterogeneous gated recursive neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 1879-1889).