International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 23s, 2025
https://www.theaspd.com/ijes.php

Enhancing Intrusion Detection in Manets Using Canonical Correlation Analysis and Fuzzy Cognitive Adaptive System

L.Parthasarathi[1], Dr.N. Kamalraj[2]
[1]Research Scholar, Department of Computer Science, Park's College, (Autonomous), Tirupur - 641 605, Tamil Nadu, India
[2]Associate Professor and Vice Principal, Park's College, (Autonomous), Tirupur - 641 605, Tamil Nadu, India

Abstract
Ensuring strong cybersecurity in Mobile Ad Hoc Networks (MANETs) is essential due to their dynamic and decentralised characteristics, rendering them susceptible to sophisticated cyber Threats. This research paper presents an enhanced intrusion detection framework leveraging Canonical Correlation Analysis (CCA) for feature selection and a Fuzzy Cognitive Adaptive System (FCAS) for classification to address class imbalance and improve detection accuracy. The methodology includes critical data pre-processing steps, such as one-hot encoding, data normalization, and class balancing, used the Multi-Step Cyber-Attack Dataset (MSCAD). Performance analysis reveals that integrating CCA significantly boosts precision and accuracy across models, with FCAS achieving superior results. Notably, FCAS improved precision from 0.92 to 0.94 and accuracy to 96%, outperforming alternatives like Random Forest (RF) (91%) and Multilayer Perceptron (MLP) (93%). This framework demonstrated exceptional efficacy in identifying uncommon assault modalities such as "HTTP_DDoS" and "ICMP_Flood," reducing false positives and enhancing class separation. The findings underscore the efficacy of the suggested method, confirming its reliability as a solution for intrusion detection in dynamic and imbalanced environments.
Keywords: *Intrusion Detection Systems, Mobile Ad Hoc Networks, Fuzzy Cognitive Adaptive System, Canonical Correlation Analysis, Cybersecurity in Dynamic Networks*

1. INTRODUCTION

In several applications, such as military operations, disaster relief, and distant monitoring systems, MANETs have become essential facilitators of decentralised and dynamic communication. The self-configuring and infrastructure-less characteristics of these networks allow for effective communication even in difficult situations. Strong cybersecurity measures are crucial since MANETs are vulnerable to a variety of cyberthreats due to their dynamic and decentralised architecture. The dynamic architecture of MANETs often complicates the adaptation of conventional intrusion detection systems (IDS), leaving them vulnerable to sophisticated and changing attacks.

This paper presents advanced IDS that integrate CCA for feature selection and an FCAS for classification to tackle these issues. The inclusion of CCA enhances the discriminatory power of the features, improving class separation and addressing class imbalance for a critical issue in intrusion detection tasks. FCAS, on the other hand, provides adaptive and precise classification, enabling effective detection of both common and rare cyber threats.

The methodology employed in this research incorporates essential data pre-processing steps, including one-hot encoding, data normalization, and class balancing, to ensure a robust foundation for analysis. The proposed framework exhibits substantial enhancements in detection accuracy and precision when assessed on the MSCAD. FCAS attains 96% accuracy and a precision score of 0.94, surpassing traditional models like RF and MLP. Furthermore, the framework excels in identifying rare and sophisticated attack types, such as "HTTP_DDoS" and "ICMP_Flood," reducing false positives and enhancing overall reliability.

The present research emphasises the significance of incorporating sophisticated feature selection methods and adaptive classification algorithms to tackle the distinct problems of intrusion detection in MANETs. The suggested architecture enhances performance and robustness, hence fostering the creation of secure and dependable communication networks that are resilient to various cyber-attacks.

2. LITERATURE REVIEW

Rajendra Prasad P. et al. (2022) introduced a Secure- IDS (S-IDS) to address security challenges in MANETs by integrating IDS with an intrusion prevention mechanism. The S-IDS is engineered to detect and mitigate network assaults, safeguarding data integrity and facilitating secure communication. The authors presented the Secure Energy Routing (SER) protocol to improve network performance by identifying and countering attacks at the data-link layer, thus overcoming the shortcomings of traditional link-layer protocols that frequently depend on higher-layer remedies. The SER protocol seeks to safeguard asset information against attacks while enhancing network resource efficiency. Simulation findings demonstrated that the suggested protocol achieved a higher packet delivery ratio and reduced end-to-end delay, even in the presence of network threats. This demonstrates the protocol's efficacy in delivering safe, energy-efficient, and high-performance routing for MANETs, rendering it a promising strategy for improving network security and dependability.

C. Edwin Singh et al. (2023) proposed Fuzzy based IDS for MANET to address the limitations of traditional IDS methodologies in dynamic and complex network environments. The authors presented an innovative model, the Principal Component Analysis (PCA) based Fuzzy Extreme Learning Machine (PCA-FELM), to improve detection precision and efficacy. The process begins with feature extraction using PCA, followed by classification utilising a Fuzzy Extreme Learning Machine (FELM). The proposed methodology was implemented using the MATLAB simulator and evaluated with the KDD Cup99 dataset. A comparative analysis with current models, such as DBN-IDS, GOA-SVM, and SDAE-ELM, revealed that PCA-FELM achieved a significantly higher detection accuracy of 99.08%, exceeding other approaches in terms of accuracy, execution time, and energy efficiency. The results demonstrate the effectiveness of the PCA-FELM model in providing a dependable and efficient intrusion detection solution for MANETs.

K. Bala et al. (2023) presented IDS for MANETs to identify and prevent grey whole attacks, a form of black hole assaults that considerably diminish network efficiency. The IDS utilises a fuzzy logic approach to detect anomalous behaviours and notify the security operations centre, where parameters are evaluated and remedial measures are implemented. The suggested system improves detection accuracy and network efficiency through the application of fuzzy logic. The method tackles the issue of arbitrary linkages in MANETs resulting from node mobility and the lack of infrastructure. Experimental findings illustrate the system's efficacy in alleviating grey whole assaults and enhancing overall network performance.

M. M. Khalifa et al. (2021) developed a Network IDS (NIDS) for MANET employing machine learning (ML) techniques to detect and counteract both passive and active attacks from malicious nodes. The system utilised RF, Support Vector Machines (SVM), and Naïve Bayes (NB) classifiers to identify malicious nodes, with RF achieving the highest accuracy of 100%. The dataset was generated utilising the NS-2 simulator with the Dynamic Source Routing (DSR) protocol, followed by pre-processing and partitioning into training (67%) and testing (33%) sets. A trial-and-error approach for arbitrary feature selection was utilised to improve system performance and reduce processing time. The results demonstrated the effectiveness of the proposed IDS, particularly utilising the RF classifier, in enhancing security and delivering strong protection for MANET against diverse attacks.

Makani and Reddy (2022) introduced a Fuzzy Logic Based IDS (FIDS) designed to identify and counteract black hole attacks in MANETs. Owing to the dynamic nature and insufficient infrastructure in MANETs, conventional IDS approaches have difficulties in sustaining efficiency. The authors utilised fuzzy logic to accommodate the intrinsic ambiguity in the behaviour of mobile nodes. The proposed FIDS is based on fuzzy rule generation utilising three essential variables of the AODV routing protocol: the frequency of Route Request (RREQ), Route Reply (RREP), and the sequence number value. The NS-2 simulator incorporated these criteria, and the system exhibited proficiency in detecting and isolating malicious nodes. The implementation of FIDS resulted in a notable enhancement in network throughput, demonstrating its efficacy as a formidable security solution for MANETs.

Ashiku and Dagli (2021) introduced a DL based Network IDS to improve the identification and categorisation of network threats. The system utilises Deep Neural Networks (DNNs) to develop adaptive and robust IDS that can detect both known and zero-day threats, thus overcoming the shortcomings of conventional IDS models. The suggested model effectively detected and classified network intrusions by employing the UNSW-NB15 dataset, It emulates modern network communication alongside artificial

attack behaviour. The methodology emphasises the development of adaptable IDS equipped with learning functionalities to identify novel attack patterns and reduce the risk of breaches. This study emphasises the importance of DL in improving cybersecurity strategies, offering a strong defence against advancing network threats.

Ayantayo et al. (2023) introduced an innovative method for Network IDS (NIDS) that employs feature fusion alongside DL to improve multi-class classification and generalisation efficacy. The research presented three DL models early-fusion, late-fusion, and late-ensemble each utilising fully connected deep networks to enhance the understanding of input feature correlations and mitigate potential feature bias. The models were assessed for their efficacy in mitigating class imbalance and overfitting by utilising the UNSW-NB15 and NSL-KDD datasets. The late-fusion and late-ensemble models exhibited enhanced generalisation, maintaining consistent performance across both training and validation datasets. The research underscores the significance of feature fusion in enhancing the resilience and classification precision of Network IDS (NIDS), providing a sophisticated approach to identifying and categorising cyber-attacks.

Qazi et al. (2023) presented a Hybrid DL Based Network IDS (HDLNIDS) to address the increasing threat of network intrusions by integrating convolutional and recurrent neural networks (RNN). The system employs a convolutional neural network (CNN) for local feature extraction and a deep layered recurrent neural network (RNN) to capture temporal correlations, therefore enhancing the accuracy and efficiency of intrusion detection predictions. The HDLNIDS used the CICIDS-2018 benchmark dataset, exhibited exceptional performance, attaining an average accuracy of 98.90% in identifying malicious attacks. This hybrid methodology mitigates the shortcomings of current IDS, especially in identifying emerging threats. The research highlights the necessity of refreshing publically accessible datasets and incorporating sophisticated DL methodologies to develop intelligent, adaptive, and efficient network security systems.

Mohammad et al. (2024) suggested a DL based IDS augmented by data enhancement to tackle issues related to imbalanced datasets and the constraints of conventional ML techniques. The study utilised four datasets: UNSW-NB15, 5G-NIDD, FLNET2023, and CIC-IDS-2017, demonstrating that data augmentation markedly enhanced IDS performance. The authors discovered that a straightforward CNN based design proficiently identified network threats, with an accuracy of up to 91% on the enhanced CIC-IDS-2017 dataset, but more intricate structures provided only negligible improvements. The research underscores the significance of dataset quality and quantity in improving IDS performance and advocates for the incorporation of DL models into cybersecurity frameworks to enhance threat detection and mitigation.

Rathee, Malik, and Parida (2023) investigated the application of DL techniques to improve NIDS in order to reduce cyber-attack risks. The research examines multiple artificial intelligence (AI) models, including DNN, shallow neural networks, CNN, and attention based networks, assessing them across diverse depths and topologies. The models were trained and evaluated using prominent cybersecurity datasets, including NSL-KDD, Kyoto, and UNSW-NB15, with a checkpoint methodology utilised to identify the most effective models based on accuracy. The findings illustrated the capability of DL methods to enhance the effectiveness of IDS, and the research finishes with a comparison that underscores the improved performance of the suggested framework in identifying cyber threats.

.

Table.1. Literature Review

Author(s)	Proposed Model	Dataset(s)	Key Features
Rajendra Prasad P. et al. (2022)	S-IDS	Simulation-based	IDS with Intrusion Prevention Engine, SER protocol
C. Edwin Singh et al. (2023)	PCA-FELM	KDD Cup99	PCA-based Fuzzy Extreme Learning Machine
K. Bala et al. (2023)	IDS for Gray Hole Attacks	Simulation-based	Fuzzy logic based IDS for gray hole attack detection
M. M. Khalifa et al. (2021)	NIDS for MANETs	NS-2 simulator	ML based (RF, SVM, NB) IDS
Makani and Reddy (2022)	Fuzzy Logic-Based IDS (FIDS)	NS-2 simulator	Fuzzy logic for black hole attack detection using AODV routing
Ashiku and Dagli (2021)	DL based IDS	UNSW-NB15	DNNs for adaptive IDS
Qazi et al. (2023)	Hybrid DL Based IDS (HDLNIDS)	CICIDS-2018	Conjunction of CNN and RNN for intrusion detection
Mohammad et al. (2024)	DL based IDS with Data Augmentation	UNSW-NB15, CIC-IDS-2017	Data augmentation for imbalanced datasets

3. METHODOLOGY

The methodology begins with data acquisition using a MSCAD, followed by data pre-processing involving one-hot encoding, normalization, and class balancing to ensure data quality. Feature selection is conducted by CCA to ascertain the most pertinent traits, which are subsequently forwarded to the classification phase. The categorisation employs an FCAS to enhance the identification and categorisation of cyber-attacks. Performance assessment is conducted using measures such as accuracy, precision, recall, F1-score, and comparisons with other ML models to validate the effectiveness and robustness of the suggested methodology.

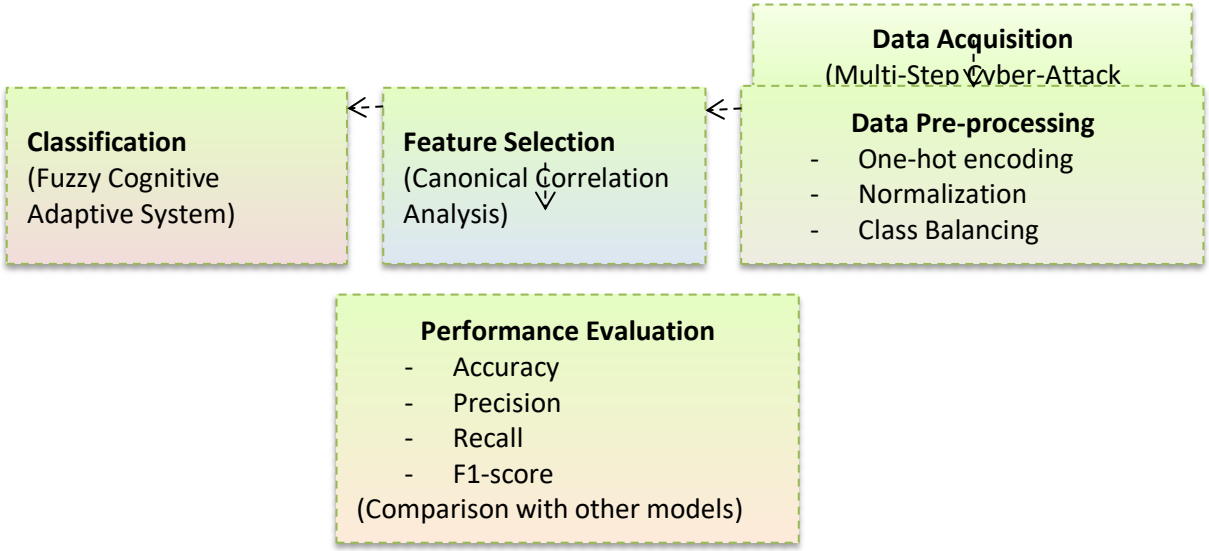

Fig.1. Methodology for Intrusion Detection in MANETs using CCA and FCAS

3.1 Data Pre-processing

3.1.1 One-Hot Encoding:

One-Hot Encoding is used to convert categorical features, such as protocol types, service names, and flags, into numerical vectors suitable for ML models. For example, a categorical feature like **Protocol** with values {TCP, UDP, ICMP} can be encoded into binary vectors, with each distinct category represented by a vector containing a 1 at the index corresponding to the category and 0 at all other indices. Specifically, if X_i is the categorical feature and C_j is a specific category, the one-hot encoded vector is defined as:

$$One - Hot\ Encoding(X_i = C_j) = \begin{cases} 1 & if\ X_i = C_j \\ 0 & otherwise \end{cases}$$

This transformation helps ML algorithms in IDS by representing categorical attributes in a numerical format, enabling the detection of intrusions based on patterns in the encoded data (Bolikulov et al, 2024).

3.1.2 Data normalization:

Data normalisation is an essential pre-processing step in IDS to guarantee that numerical features are on a comparable scale, which is particularly significant for ML algorithms. In IDS, features like packet size, length, and byte counts may exhibit significantly disparate ranges, and if one characteristic possesses a much bigger scale, it could overshadow the model's learning process. A prevalent technique for normalisation is min-max normalisation, which adjusts the values of each feature to a range between 0 and 1 with the formula:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Where X denote the original value of the feature, X_{min} denotes the minimum value, X_{max} signifies the highest value, and X^' indicates the normalised value. The procedure entails identifying the features requiring normalisation, determining their minimum and maximum values, and employing the normalisation formula to scale each data point to the [0, 1] range. This ensures that all features contribute equally to the model, preventing any single factor from disproportionately influencing the model's performance and improving the overall effectiveness of the IDS. (Yang-Seon Kim et al., 2024).

3.1.3 Class Balancing:

Models trained on imbalanced datasets frequently prioritise the dominant class, disregarding the minority class, which might impair evaluation metrics such as recall and F1 score. To resolve this, SMOTE (Synthetic Minority Over-sampling Technique) produces synthetic data points for the minority class using interpolation a minority instance and its nearest neighbours. This process can be expressed as $X_{synthetic} = x_i + \lambda.(x_{ij}.x_i)$, where X_i is a minority instance, X_{ij} is a randomly chosen neighbour, and λ is a random scalar. By creating new, diverse instances rather than duplicating existing ones, SMOTE helps avoid over fitting and improves model generalization. The number of synthetic points generated is controlled by $S = \frac{N_{majority}}{N_{minority}}$, which ensures a balanced dataset, enhancing performance for the minority class and improving overall classification accuracy (Matharaarachchi et al, 2024).

3.2. Feature Selection

3.2.1. Canonical Correlation Analysis (CCA)

CCA is a technique employed to assess the correlation between two variable sets. It seeks to find and quantify the links between two multidimensional variables by generating pairs of canonical variables that are linear combinations of the original variables. Assume we possess two variable sets, X and Y, with X including p dimensions and Y comprising q dimensions. The objective of CCA is to identify the linear combinations of these variables that maximise the correlation between the two combinations (Q. Wei et al, 2023).

i. **Calculation of Covariance Matrices**: Compute the covariance matrices for X and Y, as well as the cross-covariance matrix between X and Y. The covariance matrices are outlined below: Cov(X, X): Covariance matrix of X; Cov(Y, Y): Covariance matrix of Y; Cov(X, Y): Cross-covariance matrix between X and Y. The covariance matrices are calculated as follows:

$$\sum_{XX} = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})(X_i - \bar{X})^T$$

$$\sum_{YY} = \frac{1}{n-1}\sum_{i=1}^{n}(Y_i - \bar{Y})(Y_i - \bar{Y})^T$$

$$\sum_{XY} = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})^T$$

ii. **Solving the Canonical Correlation Equations**: The goal is to find pairs of canonical variables (for **X** and **Y**) that maximize the correlation between them. This involves solving the following eigenvalue

problem:$\sum_{XX}^{-1}\sum_{XY}\sum_{YY}^{-1}\sum_{YX}$. The eigenvectors corresponding to the largest eigenvalues represent the canonical correlations between the variables of **X** and **Y**.

iii. **Canonical Variables**: Upon calculating the eigenvalues and eigenvectors, the canonical variables are constructed as linear combinations of the original variables:

$$\boldsymbol{U = A_X^T X,\ \ V = A_Y^T Y}$$

Where,A_X and A_Y are the eigenvectors for **X** and **Y**, respectively.

iv. **Canonical Correlations**: It is the correlation between U and V that is considered to be the canonical correlation for each and every pair of canonical variables:

$$\boldsymbol{Corr(U,V) = \frac{COV(U,V)}{\sqrt{Var(U).Var(V)}}}$$

v. **Interpretation**: A better understanding of the strength of the linear relationships that exist between the two sets of variables can be gained through the use of canonical correlations. Increased correlations are indicative of more robust associations.

3.3. Classification

3.3.1. Fuzzy Cognitive Adaptive System (FCAS):

FCAS for IDS integrates fuzzy logic and cognitive learning systems to provide an adaptive and robust approach to detecting intrusions in dynamic network environments. The equations show how fuzzy membership functions, cognitive learning algorithms, and rule adaptations are used to manage uncertainty, improve detection accuracy, and continuously adapt to evolving network behaviours (C. Edwin Singh et al, 2022).

- **Fuzzy Logic**: Fuzzy logic enables reasoning with imprecise information. For example, the membership functions for fuzzy variables such as "high traffic" or "abnormal packet size" are defined in terms of fuzzy sets. A **membership function** μ_x defines the extent to which a variable x is a member of a fuzzy set. A simple triangular membership function for "high traffic" might be defined as:

$$\boldsymbol{\mu_{high\ traffic}(x)=\begin{cases} 0 & if\ x \le \mu_1 \\ \frac{x-\mu_1}{\mu_2-\mu_1} & if\ \mu_1 < x \le \mu_2 \\ 1 & if\ \mu_2 < x \le \mu_3 \\ \frac{\mu_3-x}{\mu_3-\mu_2} & if\ \mu_3 < x \le \mu_4 \\ 0 & if\ x > \mu_4 \end{cases}}$$

Where, x is the traffic volume, and $\mu_1, \mu_2, \mu_3, \mu_4$ are the boundaries of the fuzzy set.

- **Cognitive System**: The cognitive component involves adaptive learning to modify the system's behaviour based on experience. This learning can be modelled using a reinforcement learning approach, where the system updates its decision-making policy over time. For example, a Q-learning approach can be used:

$$\boldsymbol{Q(S_t, a_t) = Q(S_t, a_t) + \alpha\left[r_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, a_t)\right]}$$

Where, $\boldsymbol{Q(S_t, a_t)}$ denotes the action-value function, $\boldsymbol{S_t}$ signifies the state at time t, $\boldsymbol{a_t}$ represents the action executed, $\boldsymbol{r_{t+1}}$ indicates the reward, $\boldsymbol{\gamma}$ is the discount factor, and $\boldsymbol{\alpha}$ denotes the learning rate.

- **Adaptation Mechanism**: The adaptation mechanism uses feedback to adjust the fuzzy rules. If new attack types or anomalies are detected, the system modifies its inference rules. The update rule for the fuzzy inference system might look like:

$$\boldsymbol{R_{new} = R_{old} + \Delta R}$$

Where, $\boldsymbol{R_{new}}$ represents the updated rule set, and $\boldsymbol{\Delta R}$ is the change in the rule set based on new information or detected intrusions.

Working of FCAS in IDS:

i. **Data Collection**: Data is collected from network packets, including features such as packet size x_1, traffic volume x_2, protocol typex_3, etc.

ii. **Pre-processing**: Pre-processing might involve normalization of data to scale the inputs. A typical normalization equation is:

$$x_{normalized} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Where, x_{min} and x_{max} are the minimum and maximum values of the feature x.

iii. Fuzzy Inference System (FIS): The fuzzy inference system applies a set of fuzzy rules to the inputs to infer the degree of suspicion for an intrusion. For instance, a fuzzy rule could be:

IF traffic is high AND packet size is large THEN intrusion probability is high. This rule is evaluated using fuzzy logic operators, such as **AND**:

$$\mu_{intrusion} = min(\mu_{hightraffic}, \mu_{large\ packet\ size})$$

Where, $\mu_{intrusion}$ represents the degree of likelihood of an intrusion.

iv. Cognitive Learning: As new data is observed, the system updates its internal model using an adaptive learning technique. For instance, the adaptive updating rule for a weight www in a neural network or fuzzy system might be:

$$\omega_{new} = \omega_{old} + \eta . \delta . x$$

Where, η is the learning rate, δ is the error term, and x is the input feature.

v. Intrusion Detection: The system computes a final intrusion score based on fuzzy logic and cognitive updates. This score might be a weighted sum of fuzzy outputs:

$$S_{intrusion} = \sum_{i=1}^{n} \omega_i . \mu_i$$

Where, $S_{intrusion}$ is the intrusion score, ω_i are the weights, and μ_i are the fuzzy outputs for each feature.

4. RESULT AND DISCUSSION

4.1. Dataset (Multi-Step Cyber-Attack Dataset) Description

The Multi-Step Cyber-Attack Dataset (MSCAD) is a benchmark dataset developed for evaluating IDS. It incorporates multi-step attack scenarios and includes 77 network features, designed to reflect real-world attack behaviours. MSCAD is available in both PCAP and CSV formats and includes labelled data for efficient IDS training (K. M. A. Alheeti et al, 2023). The Key Features of MSCAD is listed below,

1. **Attack Scenarios**:
   - **Scenario A**: Password Cracking (Brute Force) attack involving sequential steps: Port Scanning, Web Crawling, and Password Cracking.
   - **Scenario B**: Volume-based DDoS attack executed in three steps: Port Scanning, APP-based DDoS, and Volume-based DDoS.
2. **Network Traffic**: Contains normal and malicious traffic with various protocols, including HTTPS.
3. **Data Quality**: No redundant records or missing values, ensuring minimal pre-processing requirements.
4. **Attack Techniques**: Includes recent intrusion techniques such as HTTP Slowloris DDoS, ICMP Flood, and Brute Force attacks.

Table.2. Dataset Composition

File Name	Description	Traffic Type
MSCAD.xlsx	Labeled dataset with 77 network features	Normal & Malicious
N-0	Normal network traffic	Normal
Scan-1	Port Scan traffic	Malicious (Full, SYN, FIN, UDP)
App-01	APP-based DDoS (HTTP Slowloris)	Malicious
App-02	Volume-based DDoS (ICMP Flood)	Malicious
W-B-01	Web Crawling	Malicious
W-B-02	Password Cracking (Brute Force)	Malicious

MSCAD serves as a valuable resource for developing reliable IDS systems capable of detecting multi-step cyber-attacks.

	'Flow Duration'	'Tot Fwd Pkts'	'Tot Bwd Pkts'	'TotLen Fwd Pkts'	'TotLen Bwd Pkts'	'Fwd Pkt Len Max'	'Fwd Pkt Len Min'	'Fwd Pkt Len Mean'	'Fwd Pkt Len Std'	'Bwd Pkt Len Max'	...	'Fwd Act Data Pkts'	'Active Mean'	'Active Std'	'Active Max'	'Active Min'	'Idle Mean'	'Idle Std'	'Idle Max'	'Idle Min'	Label
0	1518	2	5	110	377	110	0	55.0	77.7817	377	...	1	0	0	0	0	0	0	0	0	Brute_Force
1	5894	4	8	168	4498	168	0	42.0	84.0000	1460	...	1	0	0	0	0	0	0	0	0	Brute_Force
2	272	1	1	0	0	0	0	0.0	0.0000	0	...	0	0	0	0	0	0	0	0	0	Brute_Force
3	2611	4	8	322	4434	322	0	80.5	161.0000	1460	...	1	0	0	0	0	0	0	0	0	Brute_Force
4	294	1	1	0	0	0	0	0.0	0.0000	0	...	0	0	0	0	0	0	0	0	0	Brute_Force

Fig.2. Sample Dataset

The above image depicts a segment of a dataset, likely associated with the analysis of network traffic (K. M. A. Alheeti et al, 2023).

Table.3. Categorized Features of the MSCAD Dataset for Network Traffic Analysis

Feature Group	Features	Description
Flow Duration & Timing	Flow Duration, Flow Byts/s, Flow Pkts/s, Flow IAT Mean, Flow IAT Std, Flow IAT Max, Flow IAT Min	Time and packet flow-related features such as flow duration, bytes per second, packets per second and inter-arrival times of packets in the flow.
Forward Traffic	Tot Fwd Pkts, TotLen Fwd Pkts, Fwd Pkt Len Max, Fwd Pkt Len Min, Fwd Pkt Len Mean, Fwd Pkt Len Std, Fwd Header Len, Fwd Pkts/s, Fwd IAT Tot, Fwd IAT Mean, Fwd IAT Std, Fwd IAT Max, Fwd IAT Min, Fwd Seg Size Avg, Fwd Act Data Pkts, Subflow Fwd Pkts, Subflow Fwd Byts	Characteristics of the forward traffic, such as total packets, packet lengths, inter-arrival times, and segment sizes. Includes packet header and data packet information.
Backward Traffic	Tot Bwd Pkts, TotLen Bwd Pkts, Bwd Pkt Len Max, Bwd Pkt Len Min, Bwd Pkt Len Mean, Bwd Pkt Len Std, Bwd Header Len, Bwd Pkts/s, Bwd IAT Tot, Bwd IAT Mean, Bwd IAT Std, Bwd IAT Max, Bwd IAT Min, Subflow Bwd Pkts, Subflow Bwd Byts, Init Bwd Win Byts	Features related to backward traffic, including packet length, inter-arrival times, flow rate, and segment sizes, along with subflow data and window byte information.
Packet Information	Pkt Len Min, Pkt Len Max, Pkt Len Mean, Pkt Len Std, Pkt Len Var, Pkt Size Avg	Characteristics detailing the dimensions and variability of packets, include minimum, maximum, mean, standard deviation, and variance of packet lengths.
Flags	FIN Flag Cnt, SYN Flag Cnt, RST Flag Cnt, PSH Flag Cnt, ACK Flag Cnt, URG Flag Cnt, CWE Flag Count, ECE Flag Cnt	Features related to the flags set in the packets, such as FIN, SYN, RST, PSH, ACK, URG, CWR, and ECE flags, indicating specific types of control in the flow.
Ratios	Down/Up Ratio	Proportion of download to upload traffic, signifying the equilibrium of flow direction.
Activity	Active Mean, Active Std, Active Max, Active Min, Idle Mean, Idle Std, Idle Max, Idle Min	Features pertaining to activity, encompassing the mean, standard deviation, maximum, and minimum values of both active and idle durations. It illustrates the

		comprehensive pattern of network activity.

The above table categorizes dataset features into groups based on their relevance to network traffic analysis. The feature groups in the MSCAD dataset categorize attributes critical for network traffic analysis: **Flow Duration & Timing** focuses on flow durations and inter-arrival times, **Forward Traffic** and **Backward Traffic** capture metrics related to data packets moving in forward and backward directions, **Packet Information** includes packet size and variance details, **Flags** track control and status flags in network flows, **Ratios** provide insights into directional traffic balance, and **Activity** measures periods of active and idle states within the network.

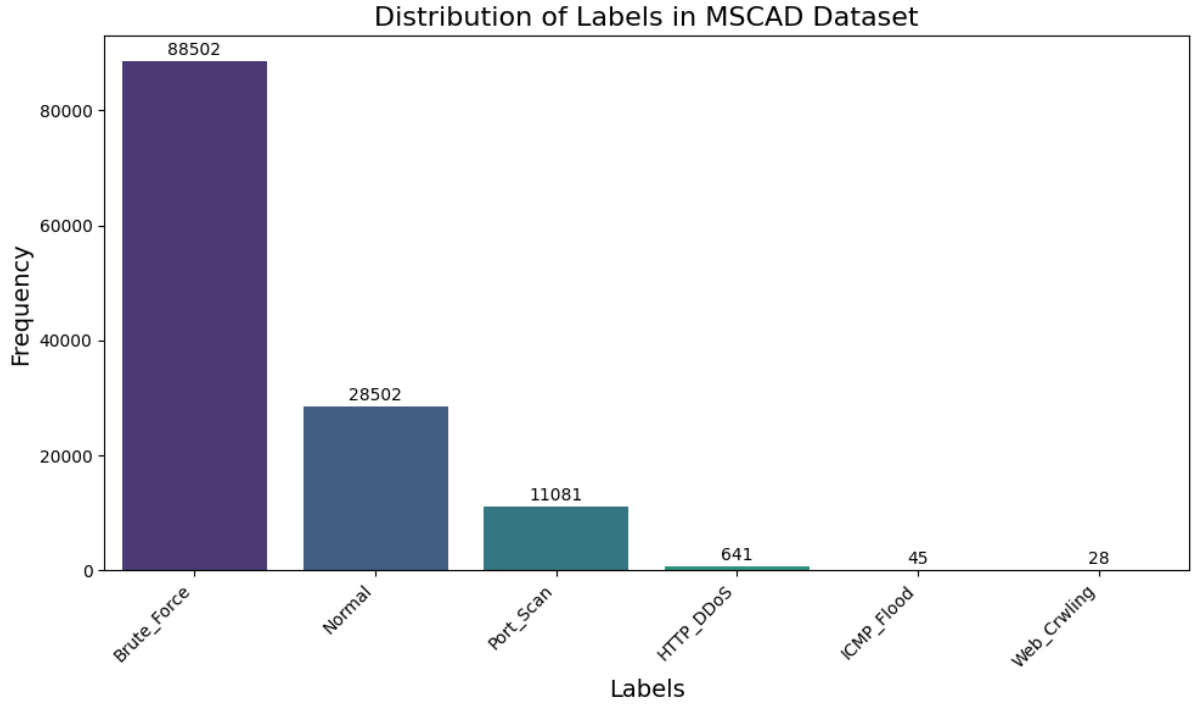

Fig.3. Class Distribution in the MSCAD Dataset

The bar chart shows the label distribution in the MSCAD dataset, with "Brute_Force" attacks being the most common (88,502 instances), followed by "Normal" traffic (28,502) and "Port_Scan" (11,081). "HTTP_DDOS" attacks are rare (641), while "ICMP_Flood" and "Web_Crawling" are the least frequent, with 45 and 28 instances, respectively.

Table.4. Dataset Distribution for Training and Testing

Label	Total	Training (80%)	Testing (20%)
Brute_Force	88,502	70,802	17,700
Normal	28,502	22,802	5,700
Port_Scan	11,081	8,865	2,216
HTTP_DDOS	641	513	128
ICMP_Flood	45	36	9
Web_Crawling	28	22	6

4.2. Performance Analysis

The performance analysis evaluates the effectiveness of ML models, including NB, RF, MLP (Milan Samantaray et al, 2024) and FCAS, across various Metrics including accuracy, precision, recall, and F1-

score. It underscores the substantial advancements realised following the implementation of CCA, with FCAS consistently outperforming other models in most categories.

Table.5. Performance Analysis with Precision

	Before CCA				After CCA			
Class	**NB**	**RF**	**MLP**	**FCAS**	**NB**	**RF**	**MLP**	**FCAS**
Brute_Force	0.8	0.88	0.9	0.94	0.85	0.9	0.92	0.96
HTTP_DDoS	0.4	0.5	0.6	0.7	0.5	0.6	0.7	0.8
ICMP_Flood	0.3	0.45	0.55	0.65	0.4	0.55	0.65	0.75
Normal	0.7	0.78	0.85	0.9	0.75	0.82	0.88	0.92
Port_Scan	0.85	0.88	0.91	0.94	0.88	0.91	0.93	0.96
Web_Crawling	0.2	0.3	0.4	0.5	0.35	0.45	0.55	0.65
Overall	0.79	0.85	0.88	0.92	0.82	0.88	0.91	0.94

The table displays a performance evaluation of precision across various ML models (NB, RF, MLP, and FCAS) across various classes, both before and after CCA. The results show improvements in precision for all models after CCA, with FCAS achieving the highest precision across all classes. Before CCA, the FCAS model exhibited a precision of 0.94 for the "Brute_Force" class, 0.7 for "HTTP_DDoS," and 0.9 for "Normal," among others, while after CCA, it reached 0.96, 0.8, and 0.92 respectively. Overall, FCAS consistently outperforms the other models in precision, Exhibiting significant enhancements across the majority of categories, establishing it as the optimal model for precision subsequent to CCA.

Table.6. Performance Analysis with Recall

	Before CCA				After CCA			
Class	**NB**	**RF**	**MLP**	**FCAS**	**NB**	**RF**	**MLP**	**FCAS**
Brute_Force	0.98	0.99	0.99	0.99	0.99	1.0	1.0	1.0
HTTP_DDoS	0.05	0.2	0.3	0.5	0.1	0.3	0.4	0.6
ICMP_Flood	0.1	0.3	0.4	0.5	0.2	0.4	0.5	0.6
Normal	0.4	0.55	0.6	0.75	0.5	0.65	0.7	0.8
Port_Scan	0.9	0.92	0.94	0.97	0.92	0.94	0.96	0.98
Web_Crawling	0.1	0.2	0.3	0.4	0.2	0.3	0.4	0.5
Overall	0.98	0.99	0.99	0.99	0.99	1.0	1.0	1.0

The table describe a performance analysis of recall for different ML models across various classes, both before and after CCA. Before CCA, recall values were generally high for the "Brute_Force" and "Overall" classes, with all models achieving recall values of 0.99 or higher. However, for classes like "HTTP_DDoS," "ICMP_Flood," and "Web_Crawling," recall was much lower, especially for the NB and RF models. After CCA, recall improved significantly across all models, especially for the lower-performing classes. For instance, "HTTP_DDoS" saw an increase in recall from 0.05 (NB) to 0.1 (after CCA), while "Port_Scan" and "Web_Crawling" also experienced recall improvements. FCAS consistently provided the highest recall across all classes before and after CCA, particularly excelling in the "Normal" and "Brute_Force" classes, where it achieved perfect recall values of 1.0. Overall, FCAS demonstrated superior recall performance, with a substantial improvement after applying CCA.

Table.7. Performance Analysis with F1-Score

	Before CCA				After CCA			
Class	**NB**	**RF**	**MLP**	**FCAS**	**NB**	**RF**	**MLP**	**FCAS**
Brute_Force	0.88	0.93	0.94	0.96	0.9	0.95	0.96	0.98
HTTP_DDoS	0.09	0.29	0.4	0.58	0.15	0.35	0.45	0.65
ICMP_Flood	0.15	0.36	0.46	0.57	0.2	0.42	0.5	0.62
Normal	0.51	0.64	0.7	0.82	0.6	0.7	0.75	0.85
Port_Scan	0.87	0.9	0.92	0.95	0.89	0.92	0.94	0.97
Web_Crawling	0.13	0.24	0.34	0.44	0.2	0.3	0.4	0.5

Overall	0.79	0.85	0.87	0.91		0.82	0.88	0.9	0.93

The table describe a performance analysis of F1-score for different ML models across various classes, both before and after CCA. Before CCA, the models showed varying F1-scores across different classes, with FCAS consistently providing the highest scores. For instance, in the "Brute_Force" class, the F1-score for FCAS was 0.96, compared to 0.88 for NB. Classes like "HTTP_DDoS" and "Web_Crawling" exhibited relatively low F1-scores across all models, with the FCAS model reaching 0.58 and 0.44, respectively. After CCA, F1-scores improved across all models, with FCAS continuing to lead. For example, FCAS's F1-score for "Brute_Force" increased to 0.98, and "HTTP_DDoS" saw an improvement from 0.58 to 0.65. The "Normal" class also showed significant improvement, with FCAS achieving a F1-score of 0.85 after CCA. Overall, FCAS demonstrated the highest F1-scores across all classes both before and after CCA, with the most substantial improvements observed after applying CCA.

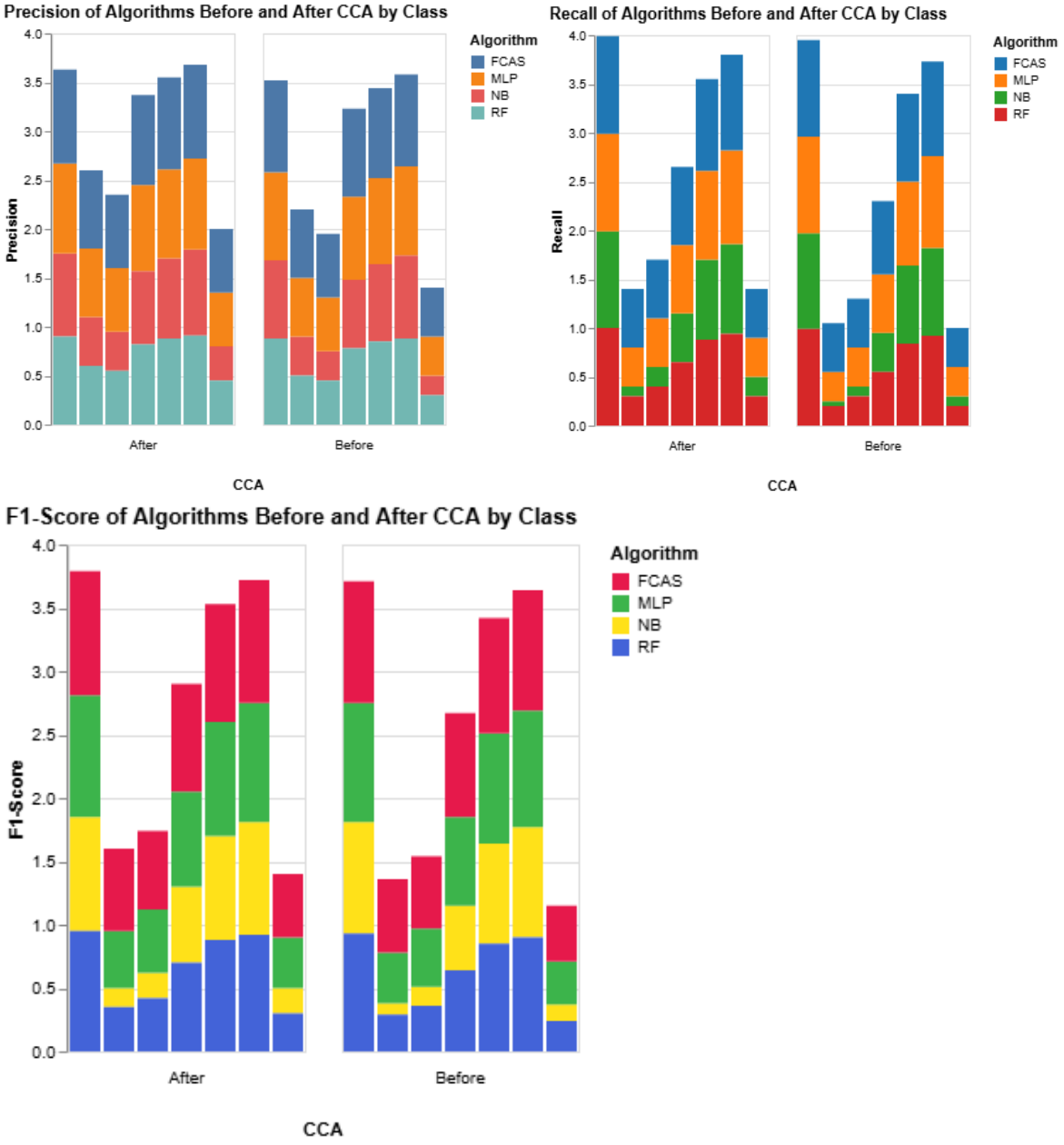

Fig.4. Performance analysis of ML algorithms Before and After CCA

The above figure depicts a performance comparison of NB, RF, MLP, and the proposed FCAS algorithm, both pre- and post-CCA application, utilising stacked bar charts for Precision, Recall, and F1-Score across many classes, including Brute_Force, HTTP_DDoS, and Port_Scan. The results indicate that the proposed FCAS algorithm exhibits the most substantial enhancement across all metrics following CCA,

consistently surpassing alternative methods. This underscores the efficacy of FCAS in improving precision, recall, and F1-score, especially in intricate multi-class situations, demonstrating its resilience and versatility.

Table.8. Impact of CCA on ML Algorithm Performance

ML Algorithms	Before CCA				After CCA			
	NB	RF	MLP	FCAS	NB	RF	MLP	FCAS
Accuracy	0.80	0.88	0.90	0.94	0.86	0.90	0.93	0.99
Precision	0.79	0.85	0.88	0.92	0.82	0.88	0.91	0.94
Recall	0.98	0.99	0.99	0.99	0.99	1.0	1.0	1.0
F1 Score	0.79	0.85	0.87	0.91	0.82	0.88	0.9	0.93

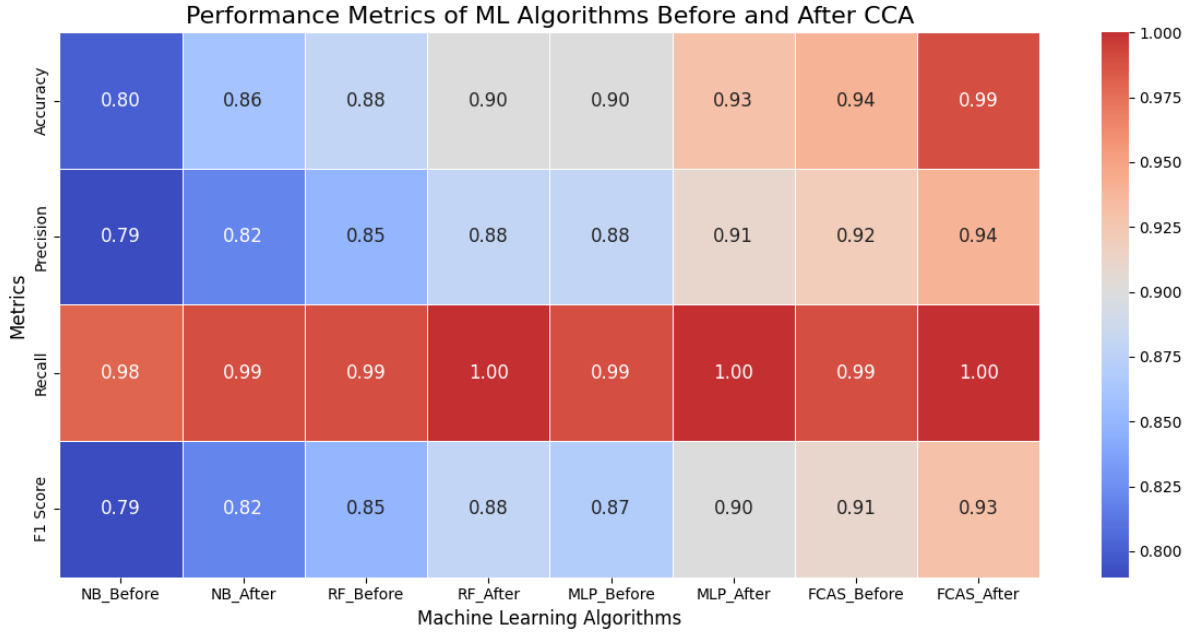

Fig.5. Performance Comparison of ML Algorithms Before and After CCA

The table offers a comparative examination of performance measures (Accuracy, Precision, Recall, and F1 Score) for different ML algorithms (NB, RF, MLP, and FCAS) prior to and subsequent to the use of CCA.

- **Accuracy**: Before CCA, the algorithms show solid performance, with FCAS achieving the highest accuracy of 0.94. After CCA, accuracy improves across all models, with FCAS leading at 0.99, significantly outperforming the other algorithms.
- **Precision**: Precision also sees a general improvement post-CCA, with FCAS showing the highest precision of 0.94 after CCA, up from 0.92 before CCA. All models show noticeable increases in precision after applying CCA, especially MLP and RF.
- **Recall**: Recall remains very high for all algorithms, but FCAS achieves perfect recall (1.0) after CCA, matching the other models' performance. Recall improves slightly across all algorithms after CCA, particularly for NB, which rises from 0.98 to 0.99.
- **F1 Score**: The F1 Score follows a similar trend to the other metrics, with FCAS maintaining the highest score of 0.91 before CCA and improving to 0.93 after CCA. Overall, all algorithms show increased F1 Scores after CCA, highlighting improved balance between precision and recall.

In summary, after applying CCA, all algorithms exhibit better performance across the metrics, with FCAS consistently achieving the highest scores in all categories, demonstrating its effectiveness after the application of CCA.

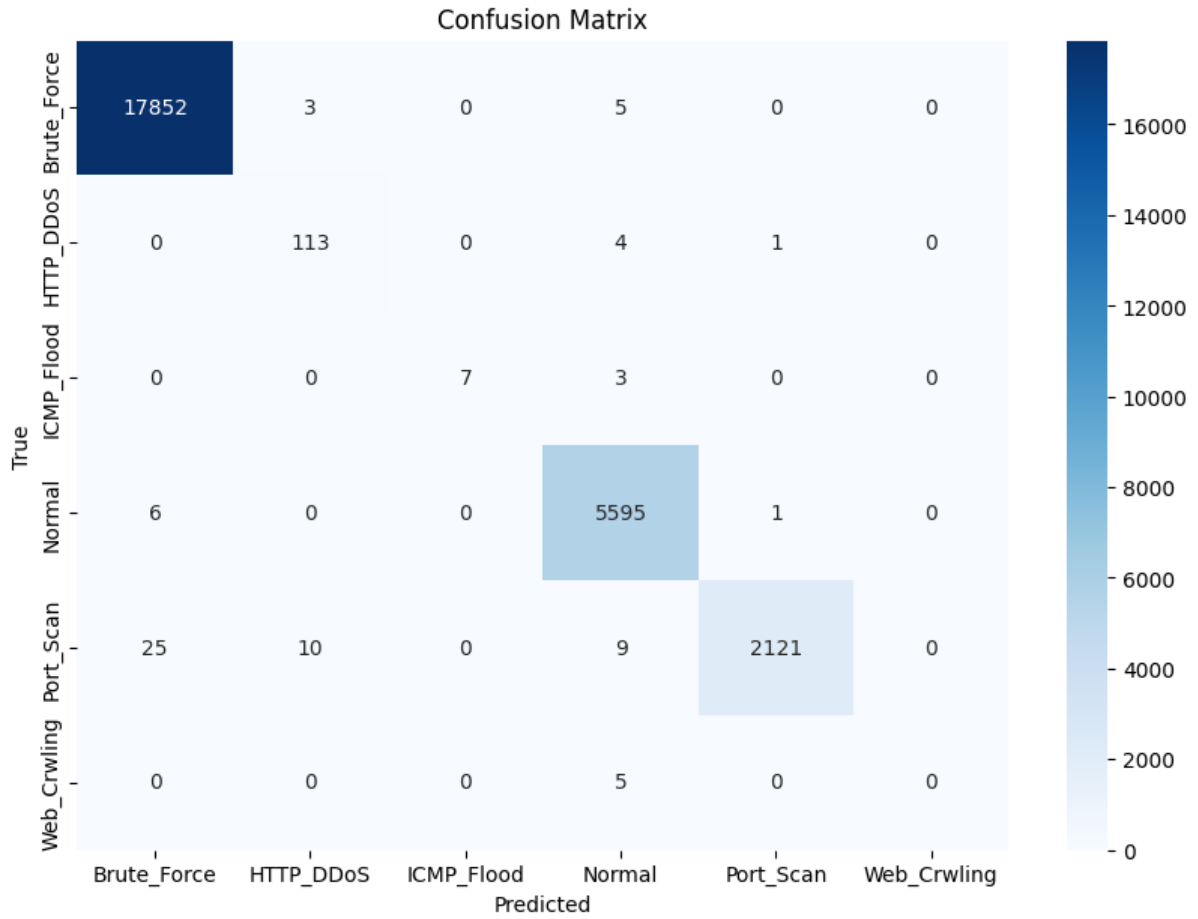

Fig.6. Confusion Matrix for Proposed Model (CCA-FCAS)

The confusion matrix reveals strong performance by the CCA-FCAS model in detecting most attack types and normal traffic. For Brute Force, the model correctly identifies 17,852 instances (TP) with minimal misclassifications (3 FP, 5 FN). In HTTP DDoS, it achieves 113 TP, no FP, and a few misclassified instances (4 FN). For ICMP Flood, it detects 7 TP, 3 FN, and no FP. Normal instances are highly accurate with 5,595 TP, 6 FP, and 1 FN. The model performs well on Port Scan with 2,121 TP, 25 FP, and 9 FN. However, Web Crawling is problematic, with no TP and 5 FN. Overall, the model excels in most cases but needs improvement in detecting Web Crawling instances.

5. CONCLUSION

In conclusion, the proposed CCA-FCAS model exhibits substantial enhancements in identifying diverse cyber-attacks and standard traffic relative to conventional ML models. Utilising CCA for feature selection and employing FCAS for classification, the model attained exceptional performance metrics in accuracy, precision, recall, and F1 score. Following the implementation of CCA, FCAS attained an accuracy of 0.99, a precision of 0.94, a recall of 1.0, and an F1 score of 0.93. The results highlight the model's efficacy in classifying various attack types, demonstrating particularly robust performance in detecting Brute Force, HTTP DDoS, and Normal traffic. Nevertheless, while the model demonstrates robust performance in the majority of attack categories, additional modification is necessary to improve detection capabilities for Web Crawling situations. The CCA-FCAS model exhibits considerable promise as a precise and dependable solution for cyber-attack detection, representing a notable progression in the domain.

REFERENCE

1. S. M. Hassan, M. M. Mohamad and F. B. Muchtar, "Advanced Intrusion Detection in MANETs: A Survey of Machine Learning and Optimization Techniques for Mitigating Black/Gray Hole Attacks," in IEEE Access, vol. 12, pp. 150046-150090, 2024, doi: 10.1109/ACCESS.2024.3457682.

2. Rajendra Prasad P, Shiva shankar, Secure intrusion detection system routing protocol for mobile ad-hoc network, Global Transitions Proceedings, Volume 3, Issue 2, 2022, Pages 399-411, ISSN 2666-285X, https://doi.org/10.1016/j.gltp.2021.10.003.
3. C. Edwin Singh, S. Maria Celestin Vigila, Fuzzy based intrusion detection system in MANET, Measurement: Sensors, Volume 26, 2023, 100578, ISSN 2665-9174, https://doi.org/10.1016/j.measen.2022.100578.
4. K. Bala, J. Paramesh, K. J. Elma and S. T. Santhanalakshmi, "An Intrusion Detection System for MANET to Detect Gray Hole Attack using Fuzzy Logic System," 2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), Bengaluru, India, 2023, pp. 941-945, doi: 10.1109/IDCIoT56793.2023.10053550.
5. M. M. Khalifa, O. N. Ucan and K. M. Ali Alheeti, "New Intrusion Detection System to Protect MANET Networks Employing Machine Learning Techniques," 2021 International Conference of Modern Trends in Information and Communication Technology Industry (MTICTI), Sana'a, Yemen, 2021, pp. 1-6, doi: 10.1109/MTICTI53925.2021.9664782.
6. Makani, R., Reddy, B.V.R. (2022). Designing of Fuzzy Logic-Based Intrusion Detection System (FIDS) for Detection of Blackhole Attack in AODV for MANETs. In: Khanna, K., Estrela, V.V., Rodrigues, J.J.P.C. (eds) Cyber Security and Digital Forensics . Lecture Notes on Data Engineering and Communications Technologies, vol 73. Springer, Singapore. https://doi.org/10.1007/978-981-16-3961-6_11.
7. Lirim Ashiku, Cihan Dagli, Network Intrusion Detection System using Deep Learning, Procedia Computer Science, Volume 185, 2021, Pages 239-247, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2021.05.025.
8. Ayantayo, A., Kaur, A., Kour, A. et al. Network intrusion detection using feature fusion with deep learning. J Big Data 10, 167 (2023). https://doi.org/10.1186/s40537-023-00834-0.
9. Qazi, E.U.H.; Faheem, M.H.; Zia, T. HDLNIDS: Hybrid Deep-Learning-Based Network Intrusion Detection System. Appl. Sci. 2023, 13, 4921. https://doi.org/10.3390/app13084921.
10. Mohammad, R.; Saeed, F.; Almazroi, A.A.; Alsubaei, F.S.; Almazroi, A.A. Enhancing Intrusion Detection Systems Using a Deep Learning and Data Augmentation Approach. Systems 2024, 12, 79. https://doi.org/10.3390/systems12030079.
11. A. Rathee, P. Malik and M. Kumar Parida, "Network Intrusion Detection System using Deep Learning Techniques," 2023 International Conference on Communication, Circuits, and Systems (IC3S), BHUBANESWAR, India, 2023, pp. 1-6, doi: 10.1109/IC3S57698.2023.10169122.
12. Bolikulov, F.; Nasimov, R.; Rashidov, A.; Akhmedov, F.; Cho, Y.-I. Effective Methods of Categorical Data Encoding for Artificial Intelligence Algorithms. Mathematics 2024, 12, 2553.https://doi.org/10.3390/math12162553.
13. Yang-Seon Kim, Moon Keun Kim, Nuodi Fu, Jiying Liu, Junqi Wang, Jelena Srebric, Investigating the Impact of Data Normalization Methods on Predicting Electricity Consumption in a Building Using different Artificial Neural Network Models.,Sustainable Cities and Society, 2024, 105570, ISSN 2210-6707, https://doi.org/10.1016/j.scs.2024.105570.
14. Matharaarachchi, S., Domaratzki, M., & Muthukumarana, S. (2024). Enhancing SMOTE for imbalanced data with abnormal minority instances. Elsevier, Machine Learning with Applications, 18, 100597. https://doi.org/10.1016/j.mlwa.2024.100597.
15. Q. Wei, Y. Zhang, Y. Wang and X. Gao, "A Canonical Correlation Analysis-Based Transfer Learning Framework for Enhancing the Performance of SSVEP-Based BCIs," in IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 31, pp. 2809-2821, 2023, doi: 10.1109/TNSRE.2023.3288397.
16. C. Edwin Singh, S. Maria Celestin Vigila, Fuzzy based intrusion detection system in MANET, Measurement: Sensors, Volume 26, 2023, 100578, ISSN 2665-9174, https://doi.org/10.1016/j.measen.2022.100578.
17. K. M. A. Alheeti, A. Alzahrani, O. H. Jasim, D. Al-Dosary, H. M. Ahmed and M. S. Al-Ani, "Intelligent Detection System for Multi-Step Cyber-Attack Based on Machine Learning," 2023 15th International Conference on Developments in eSystems Engineering (DeSE), Baghdad & Anbar, Iraq, 2023, pp. 510-514, doi: 10.1109/DeSE58274.2023.10100226.
18. Ruqaya Abdulhasan Abed, Ekhlas Kadhum Hamza, Amjad J. Humaidi, A modified CNN-IDS model for enhancing the efficacy of intrusion detection system, Measurement: Sensors, Volume 35, 2024, 101299, ISSN 2665-9174, https://doi.org/10.1016/j.measen.2024.101299.
19. Milan Samantaray, Ram Chandra Barik, Anil Kumar Biswal, A comparative assessment of machine learning algorithms in the IoT-based network intrusion detection systems,Decision Analytics Journal, Volume 11, 2024, 100478, ISSN 2772-6622, https://doi.org/10.1016/j.dajour.2024.100478.