REPUBLIC OF TÜRKİYE

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Electrical and Computer Engineering

AN ENHANCED ATTENTION AND DILATED CONVOLUTION BASED ENSEMBLE MODEL-BASED NETWORK INTRUSION DETECTION SYSTEM AGAINST ADVERSARIAL EVASION ATTACKS USING ENHANCED CHEETAH OPTIMIZER

Omer Fawzi Awad AWAD

Doctor of Philosophy

Supervisor

Asst. Prof. Dr. Mesut ÇEVİK

İstanbul, 2025

AN ENHANCED ATTENTION AND DILATED CONVOLUTION BASED ENSEMBLE MODEL-BASED NETWORK INTRUSION DETECTION SYSTEM AGAINST ADVERSARIAL EVASION ATTACKS USING ENHANCED CHEETAH OPTIMIZER

Omer Fawzi Awad AWAD

Electrical and Computer Engineering

Doctor of Philosophy

ALTINBAŞ UNIVERSITY

2025

The dissertation titled AN ENHANCED ATTENTION AND DILATED CONVOLUTION BASED ENSEMBLE MODEL-BASED NETWORK INTRUSION DETECTION SYSTEM AGAINST ADVERSARIAL EVASION ATTACKS USING ENHANCED CHEETAH OPTIMIZER prepared by OMER FAWZI AWAD AWAD and submitted on 05/12/2025 has been **accepted unanimously** for the degree of PhD in Electrical and Computer Engineering.

_____

Asst. Prof. Dr. Mesut ÇEVİK
Supervisor

Thesis Defense Committee Members:

Asst. Prof. Dr. Mesut ÇEVİK	Department of Electrical-Electronics Engineering, Altınbaş University	_____
Assoc. Prof. Dr. Abdullahi Abdu IBRAHIM	Department of Software Engineering, Altınbaş University	_____
Asst. Prof. Dr. Hameed Mutlag Farhan FARHAN	Department of Computer Engineering, Altınbaş University	_____
Asst. Prof. Dr. Şenay KOCAKOYUN AYDOĞAN	Department of Computer Technologies, İstanbul Gedik University	_____
Asst. Prof. Dr. Ameer Yalmaz Asaad ASAAD	Department of Computer Technologies, İstanbul Topkapı University	_____

I hereby declare that this dissertation meets all format and submission requirements for a PhD dissertation.

I hereby declare that all information/data presented in this graduation project has been obtained in full accordance with academic rules and ethical conduct. I also declare that all unoriginal materials and conclusions have been cited in the text, and all references mentioned in the Reference List have been cited in the text, and vice versa, as required by the abovementioned rules and conduct.

Omer Fawzi Awad AWAD

Signature

DEDICATION

I dedicate this work to my Advisor , who is guided me in right way to complete this work as required, as well as my family (first my brothers and sisters, my wife and my daughters, and my son) for always assisting me in my hard times.

PREFACE

I would like to express my great thanks to my great Advisor Asst. Prof. Dr. Mesut Çevik for all the kind of support that he provides with all resources that's help me to complete this dissertation. I cannot finish this work without his advising, helping and his well supervision. He was stand with me in every section of this dissertation providing me with a high-quality solution to pass all the difficulties that I was faced during the writing process.

ABSTRACT

AN ENHANCED ATTENTION AND DILATED CONVOLUTION BASED ENSEMBLE MODEL-BASED NETWORK INTRUSION DETECTION SYSTEM AGAINST ADVERSARIAL EVASION ATTACKS USING ENHANCED CHEETAH OPTIMIZER

AWAD, Omer Fawzi Awad

Ph.D., Electrical and Computer Engineering, Altınbaş University,

Supervisor: Asst. Prof. Dr. Mesut ÇEVIK

Date: 12 /2025

Pages: 94

A reliable defense system against network threats over a long time is called the Intrusion Detection System (IDS). Defenders are notified by the IDS when suspicious or malevolent activities are identified on the network. In the last ten years, machine learning has helped the IDS to become more accurate, more capable of analysis, and more adept at finding new or modified forms of known intrusions. Deep learning, an advanced version of the machine learning technique, is essential to the field of network security. Additionally, a deep learning-based Network Intrusion Detection System (NIDS) performs better than conventional IDS techniques. Recent studies, however, demonstrate that when faced with attackers in real-time, the deep learning-based IDS becomes somewhat inaccurate. There is no analysis done on how attack models will affect NIDS as well. In order to defend against adversarial evasion attacks, an enhanced deep learning-based NIDS model is designed here. The required data is first collected from commonly available websites. The best feature extraction is carried out on the collected data. Here, the Improved Cheetah Optimizer (ICO). Then, an Attention and Dilated-based Ensemble Network (ADEN) is implemented to detect the intrusions from the optimally extracted features. The Deep Temporal Convolutional Neural Network (DTCN), Long Short-term Memory (LSTM), and Gated Recurrent Unit (GRU) models are assembled together to deploy the suggested ADCEN. In the end, the ADEN detects the

intrusions and generates the respective outputs using the fuzzy ranking approach. To demonstrate well the recommended deep learning-based NIDS defends against adversarial evasion assaults, experiments are conducted against conventional models.

Keywords: NIDS, ICO, ADCEN, DTCNN, Adversarial Attack Detection, Optimal Feature Selection.

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

ABBREVIATIONS

ADCEN	:	Adaptive Dilated Cascaded Ensemble Network
ADEN	:	Attention and Dilated-based Ensemble Network
AML	:	Adversarial Machine Learning
ANN	:	Artificial Neural Network
ANNs	:	Artificial Neural Networks
APTs	:	Advanced Persistent Threats
BN	:	Bayesian Networks
CNN	:	Convolutional Neural Network
CO	:	Cheeta Optimizer
DGA	:	Domain Generation Algorithm
DNN	:	Deep Neural Network
DoS	:	Denial of Service
DRL	:	Deep Reinforcement Learning
DT	:	Decision Trees
DTCN	:	Deep Temporal Convolutional Neural Network
FDAA	:	Function-Discarding Adversarial Attack
FDR	:	False Discovery Rate
FMSA	:	Fast Model Stealing Attack
FN	:	False Negatives
FNR	:	False Negative Rate
FP	:	False Positives
FPR	:	False Positive Rate
GANs	:	Generative Adversarial Networks

GRU	:	Gated Recurrent Unit
GSO	:	Garter Snake Optimization
HIDS	:	Host-based IDS
ICO	:	Improved Cheeta Optimizer
IDS	:	Intrusion Detection System
IIoT	:	Industrial Internet of Things
JSMA	:	Jacobian-Saliency Map adversarial
LSTM	:	Long Short-Term Memory
LSTM	:	Long Short-Term Memory
MCC	:	Matthews correlation coefficient
ML	:	Machine Learning
NIDS	:	Network-based IDS
NPV	:	Negative Predictive Value
RF	:	Random Forest
RNN	:	Recurrent Neural Network
ROC	:	Receiver Operating Characteristic
SVM	:	Support Vector Machines
TCN	:	Temporal Convolutional Network
TN	:	True Negatives
TP	:	True Positives
TSO	:	Tuna Swarm Optimization
WSA	:	Water Strider Algorithm

1. INTRODUCTION

An adversarial example refers to a data input in a machine learning system designed and crafted by an attacker in such a way that the machine learning model produces incorrect results [1]. In most cases, the attacker might not even have direct access to the learning model's structure or parameters. This is called a black-box attack. On the other hand, when the attacker has complete information regarding the learning model, then the attack is referred to as the white-box attack. Moreover, attackers can also approximate the white-box attack by utilizing the concept of transferability. The principle of transferability states that an input designed to attack one model can also be used to attack another model with different parameters.

The security of a network has become an indispensable task, and types of cyber-attacks have increased at a rapid pace due to the increased development in the global network infrastructure [2]. The most advanced security system is the Network Intrusion Detection System (NIDS) and others, which function in order to protect against the threats of a network. Network Intrusion Detection Systems function as an indispensable system for maintaining network communications accurately, which plays an important part in a computer network security system [3]. Due to its advanced security features, Network Intrusion Detection Systems have drawn much more significant attention from the corresponding communities from a security point of view. A Network Intrusion Detection System is an active system that functions as a defense mechanism against attacks. They function constantly, accumulating data about traffic data from computers or from a network, creating detection patterns, and responding correspondingly against viruses, intrusions, and malicious activities that occur on a network [4]. In contemporary cyber security systems, a Network Intrusion Detection System is one of the principal defense mechanisms against attacks through observing, scanning, and monitoring traffic patterns, analyzing probable security attacks, and protecting against service discontinuance [5].

Intrusion Detection System (IDS) is designed with the intent of detecting malicious activities in the network by producing alarm notifications based on the analysis of network logs and flow patterns [6]. Depending on their application environment, IDSs can be classified in the following ways:

a. Network-based IDS (NIDS), this type of IDS observes data flowing across the network, and
b. Host-based IDS (HIDS), targets malicious activity occurring on a particular computer or server.

Moreover, IDS systems can be classified into two broad categories:

a. Signature-based IDS, in which attacks are identified by matching data with predefined malicious signatures.

b. Anomaly-based IDS, identifying possible attacks based on anomalies in normal patterns of behavior [7].

Traditionally, rule sets, or signature databases, were designed manually by cybersecurity professionals. Even though they were very effective against known attacks, they had significant shortcomings, such as their failure to protect against zero-day attacks, their need for significant cybersecurity expert input, and their high maintenance overhead in terms of time [8]. Misuse models of NIDS require large signature databases, causing them to be slow in adapting to changes in attacker strategies and highly prone to new attack patterns [9].

On the other hand, anomaly detection IDSs intend to find emerging or stealthy threats by observing deviations in normal communication patterns [10]. Anomaly detection IDSs have the benefit of detecting unknown (zero-day) attacks and updating signature sets constantly. Incorporating Machine Learning (ML) in the field of intrusion detection has achieved remarkable milestones. Machine learning models have the ability to learn complex patterns from large data sets containing normal and malicious activities, which helps in alerting threat analysts much earlier [11]. Machine learning also has the benefit of reducing threat analysts' work significantly by lowering the fp-rate and automating alert prioritization [12].

However, anomaly-based IDSs tend to suffer from high false alarm rates, in which normal communications with anomaly behaviors might result in incorrect warnings [13]. To overcome such drawbacks, hybrid approaches, integrating signature-based and anomaly-based schemes, were proposed. Machine learning has been playing an ever more important role in the development of hybrid IDSs due to their flexibility and powerful generalization ability [14].

Several ML algorithms have been effectively used in the task of intrusion detection, such as Artificial Neural Networks (ANN), Decision Trees (DT), Support Vector Machines (SVM), Fuzzy Logic Systems, and Bayesian Networks (BN). These approaches have shown remarkable accuracy in identifying malicious activities and detecting unknown attacking patterns. However, with such developments, the threat posed by Adversarial Machine Learning (AML) has also been identified, which poses a significant threat [15]. An assailant could take advantage of the flaw in the Machine Learning algorithms or models in order to bypass the intrusion detection systems thus, to undermine the security of networks [16]. Besides, the attackers could use different kinds of attacks such as attacks and evasion, depending on how familiar they are with the target environment - membership inference, model inversion, data poisoning, and model extraction. The adoption of ML in the NIDS paradigm, although a major breakthrough, has made it easier for attackers to find new ways to attack.

The adversarial attack on ML models is achieved in such a way that attack detection systems in networks can be fooled into identifying malicious activities as safe, although the attack is malicious. Thus, the evolution of attacks against networks highlights the need for more powerful, adaptable, and resilient deep learning models in the design of NIDS architectures [17].

The rapid growth in the computer networking sector and the wide adoption of internet-based services have heralded a major change in the structure of modern society. Healthcare, banking, smart power grids, industrial automation, and cloud computing are all being implemented as part of network scenarios, which will not only enhance their efficiency but also allow instantaneous data transfer. This, however, has opened up networks to various types of attacks – malware, denial of service, insider, and Advanced Persistent Threat attacks that are now more widespread due to the very high connectivity existing in networks. As it has been established that firewalls and access lists do not provide effective protection against most modern attacks, the notion of NIDS has gained the status of necessity in the layouts of today's cybersecurity solutions [1]. The Network Intrusion Detection System, imperatively designed to monitor in real-time the communication in the network and thus, detect the possible threats to confidentiality, integrity and availability of networked resources, assists in real-time response to the identified threats. The working of NIDS is different from other cyber defense solutions such as preventive measures, as it tries to spot risks only after they

have taken place. The combined effect of the sophistication of new age attacks and the vast traffic carried by the networks in today's networking environment makes intrusion detection research a very important and challenging endeavor [2]. Initially, NIDS were build up with rule-base and signatures-base methods where the patterns determined denoted the known attacks. Though such systems gave a very high accuracy level in detection of predicted threats, they faced several shortcomings while detecting the unknown or zero day attacks [3].

1.2 RESEARCH MOTIVATION

NIDS models, their robustness against adversarial evasion attacks has been found to be inadequate. Attackers take advantage of these weaknesses in models by crafting subtle deviations in input patterns, which, although invisible to even human detection, drastically change the output of models [18]. Consequently, even sophisticated DL models based on CNN, LSTM, and RNN in IDS have been found to produce high levels of false negatives in adversarially attacked networks [19].

The primary reason for doing this work is to create an IDS that is optimized, adaptable, and very accurate to quickly detect new and adversarial networks. The complexity of attacks is continuously increasing making the need for such an IDS very urgent. This is because the IDS should be able not only to learn from what has happened but also to recognize new adversarial patterns. Still, the most recent improvements in IDS are the use of both metaheuristic optimization algorithms and deep learning models which can identify feature sets with the highest accuracy [20].

The Cheetah Optimizer (CO) and the Improved Cheetah Optimizer (ICO) are two of the most recognized optimization algorithms because of their fast convergence, good balance between exploration and exploitation, and good performance in searching high-dimensional spaces [21]. By integrating the optimal feature search function with the ICO, the proposed system can achieve the following: redundancy elimination, learning efficiency increase, and NIDS models' robustness enhancement against adversarial noises.

Moreover, the application of attention mechanisms and dilated convolutional networks provides an effective means of improving the interpretability of the model and enhancing multi-scale feature learning. Attention mechanisms allow the model to concentrate on the

most significant features, which play important roles in decision-making, whereas global dependencies in the image can be captured by the application of dilated convolutional networks without increasing complexity [22]. Through ensemble learning with DTCN, LSTM, and GRU models, detection accuracy, generality, and adversarial robustness can be drastically improved.

1.3 PROBLEM STATEMENT

Intrusion Detection Systems (IDS) remain one of the most important means of ensuring the security, integrity, and confidentiality of computer networks. IDSs can be regarded as an important mechanism in defending against various malicious activities such as unlawful access, data violations, and service downtimes. However, in view of the emerging sophisticated nature of malicious activities, the normal IDS models have been facing great challenges in identifying sophisticated intrusions, especially adversarially generated ones.

An efficient IDS plays a critical role in ensuring the security of communication networks during data transfer. However, due to their high complexities, accurately detecting intrusions and evasion attacks in large-scale and diversified networks is still considered a complex task. If the IDS models are not able to accurately identify intrusions and evasion attacks, then The system protects against attacks, maintaining availability of the data being transferred over the networks become highly susceptible to exploitation by attackers. Also, one of the most significant limitations experienced in almost all common IDS models is their high False Positive Rate (FPR) (Pawlicki et al., 2020; Anthi et al., 2021).

An important problem with current IDS solutions is their inadequacy when dealing with the inspection of encrypted/obfuscated network packets. Most signature-based and simple machine learning models cannot protect against intrusions carried out over encrypted networks. Moreover, most conventional IDS models are highly prone to evasion attacks, wherein attackers design their input data in such a way that it deceitfully targets the models constructed by the detection systems. These factors emphasize the need to create more robust, adaptive, and intelligent IDS models, which can protect against adversarially generated threats in real-time (Sidi et al., 2020; Li et al., 2023).

To understand the gaps, the important features and shortcomings of some current IDS methodologies will be analyzed below:

a. Artificial Neural Network (ANN) [23]: Even though classification related to intrusion detection tasks in IDS can be accomplished with the help of ANNs, they tend to have high FPR, causing limitations regarding the efficiency of the detection mechanism in networks (Pawlicki et al., 2020).

b. EIFD AA [24]: The "Function-Discarding Adversarial Attack" detection model provides an improved level of detection in the Industrial Internet of Things (IIoT) environment, along with enhanced overall security. Still, in practical adversarial attacks, the proposed model is unrealistic, and the detection of targeted attacks does not work (Li et al., 2023).

c. Random Forest (RF) [25]: The proposed RF model is efficient in classifying the Jacobian-Saliency Map (JSMA) adversarial samples, and it is also robust against specific attacks. However, the proposed model is not generally adaptable to other adversarial attacks in machine learning (AML) (Anthi et al., 2021).

d. MaskDGA [26]: This evasion technique is highly effective in identifying weaknesses in the detection mechanisms reliant on the Domain Generation Algorithm. Although highly accurate, it is not efficient in resource-restricted settings (Sidi et al., 2020).

e. DeepMal [27]: This approach applies deep learning to adversarial instructional detection, achieving promising performance. However, the approach could suffer from overfitting, lacking sufficient generalization ability in dealing with varied network settings (Yang et al., 2021).

f. RELEVAGAN [28]: The combination of Deep Reinforcement Learning (DRL) and Generative Adversarial Networks (GANs) offers an efficient way of detecting modern adversarial evasion attacks. However, the training mechanism involves high computational complexity and is time-consuming (Hamid et al., 2023).

g. Rule-based ML IDS [29]: This technique actually conducts an in-depth analysis of DoS attack effects on supervised learning models and also makes assessments on robustness. Still, the technique is less accurate when dealing with high-dimensional data or data with disturbed feature vectors (Anthi et al., 2021).

h. FMSA [30]: Fast Model Stealing Attack. Its accuracy level in model duplication with few queries is high. However, executing this attack is expensive, and it does not stop the duplication/cloning of victim models (Fan et al., 2023).

These experiments uncover the fact that although various state-of-the-art IDS models have been developed, they do not encompass every facet of secure adversarial protective mechanisms, optimal feature determination, and adaptable deep learning. Most of the models had high computation overhead, poor generality, or were prone to attacks due to evolving adversarial evasion tactics. Consequently, the objective of this paper is to incorporate all the identified limitations with the design of an (ADCEN) optimized by means of an Improved Cheetah Optimizer (ICO) in order to suppress the false detection rates, improve accuracy, and achieve robustness in evasion attacks.

Table 1.1: Feature And Challenges.

Refrence	**Methodology**	**Features**	**Challenges**
[23]	ANN	• Intrusion detection using this method does not impact classification outcomes.	• The approach exhibits a higher False Positive Rate (FPR), degrading the overall system performance.
[24]	EIFDAA	• detecting measures achieved in identifying various IIoT-based attacks is comparatively higher.• The security provided by this model is also improved.	• Targeted adversarial attacks cannot be effectively detected using this method. • The technique lacks realistic adaptability for dynamic threat environments.
[25]	RF	• Capable of classifying JSMA-based adversarial samples with reasonable robustness.	• Requires additional defensive mechanisms to improve resilience against Adversarial Machine Learning (AML) attacks.
[26]	MaskDGA	• Accurately identifies the DGA Model Evasion Vulnerability using adversarial retraining.	• Not suitable for deployment in embedded systems due to computational constraints.

Table 1.1: Feature And Challenges "Table Continoued".

[27]	DeepMal	• Practical implementation for both static and dynamic intrusion detection environments .• Enhanced adaptability for various deep learning classifiers.	• Susceptible to overfitting and limited generalization capability. • The security performance remains unsatisfactory under complex adversarial settings.
[28]	GAN	• Provides effective identification of modern evasion attacks through a DRL-integrated GAN architecture.	• The model requires extensive training time and high computational resources.
[29]	ML	• Effectively detects the impact of DoS attacks using supervised classifiers. • Identifies key features influencing model robustness.	• Sensitive to feature perturbations, reducing detection consistency under adversarial influence.
[30]	FMSA	• Capable of performing model-stealing attacks against victim models with high accuracy and minimal query counts.	• Computationally expensive and ineffective in preventing cloning of target models.

1.4 RESEARCH QUESTIONS

Research questions are the main issues that are tackled in a study and hence are the theoretical map for scientific inquiry. For the purpose of this thesis, the questions of research are laid out with a view to finding out how well deep learning techniques that are advanced will be applied in the case of network intrusion detection systems that have been subjected to adversarial evasion attacks. The first research question considers the significance of attention mechanisms in intrusion detection. Various ways have illustrated the success of attention mechanisms in different fields which consist of the capability of the model to

linearly weigh features based on their relevance and also suppress irrelevant or noisy information. Consequently, the inquiry can be phrased as follows:

a. RQ1: In what way can attention mechanisms work miracles as far as the resilience and precision in detection of systems of deep learning–based network intrusion detection systems are concerned under adversarial conditions? The second research question has to do with the selection of features and its bearing on adversarial resilience. Selecting the right features is one of the main tasks done to reduce dimensionality, enhance model generalization, and improve the resistance of the model to adversarial perturbations. This leads directly to the following question:

b. RQ2: How does optimized feature selection through metaheuristic algorithms impact the adversarial robustness of network intrusion detection systems? The third research question is about an ensemble model's architectural design. Ensemble learning principle indicates that bringing together several learners can result in improved performance and stability. Hence, the current study focuses on:

c. RQ3: In the case of adversarial intrusion detection scenarios, can a temporal, recurrent, and attention-based ensemble framework that is integrating all the three dimensions outperform the individual detection models?

The research questions, as a whole, will take the lead in the theoretical maturity and empirical testing of the suggested framework. They give a guarantee that the research will not only tackle major issues in adversarial intrusion detection but also present new views in the area of study.

1.5 AIM OF STUDY

The major objective of the proposed work is:

a. to design an improved Network Intrusion Detection System utilizing deep learning with the primary intent of protecting network data against evasion attacks with the help of an innovative optimization approach.

b. The objective of the proposed work is to conceptualize an effective, intelligent, and adaptive intrusion detection mechanism utilizing innovative deep learning approaches,

including the application of ‘attention’ and ‘dilated convolution’ mechanisms, to accurately detect and counter the growing threat of cyber attacks.

c. The proposed method employs an Improved Cheetah Optimizer (ICO) algorithm to assist in this process. Besides reducing the complexity of computations, this also helps to increase the efficiency of learning during model training. The end product of the research is intended to be a well-rounded, secure, and efficient NIDS with the potential to provide a reliable defense mechanism.

1.6 RESEARCH CONTRIBUTIONS

The following are considered to be contributions of this work presented broadly:

a. Development of a novel NIDS system featuring robustness against adversaries: In fact, an innovative idea of Network Intrusion Detection System (NIDS) is proposed in this research work, focusing on a deep learning model that would not only identify a few cyber attacks accurately, but would also nullify evasion attacks. In actuality, the approach applies adaptive intelligence, which marks the beginning of an increase in the count of defenders in the present scenario of cyber warfare. b. Optimal Feature Selection by Using Improved Cheetah Optimizer (ICO): The cooperative approach is equipped with an effective ICO algorithm in the process of feature selection, so as to reduce data relationship and processing costs.The feature optimization approach helps in identifying the most significant and discriminative features from the raw data of network traffic, and hence, an optimal set of features promotes accuracy in intrusion detection and suppresses error levels.

b. Proposal of the Improved Cheetah Optimizer (ICO) Algorithm: This article proposes a new concept termed as the ICO Algorithm, which has been introduced from the effective Cheetah Optimizer (CO), making use of a more effective adaptive control strategy that results in optimal results. It is stated that the proposed ICO Algorithm effectively tackles both the exploitation steps as well as exploration during searching as well as accomplishes optimal searching of the parameters as well as features of the intrusion detection process.

c. Construction of an Ensemble Deep Learning Network (ADCEN): Designing a deep learning network is one of the most significant contributions of this proposed work. This proposed work presents a new concept of an adaptive dilated cascaded ensemble network (ADCEN). This proposed ensemble strategy is developed by considering the advantageous characteristics of a dilated temporal convolutional network (DTCN), gated recurrent unit (GRU), and long short-term memory (LSTM). Moreover, the proposed strategy combines the concept of the attention mechanism, as well as a dilated convolution layer, for enhanced feature extraction. Also, a fuzzy ranking strategy is employed in this proposed strategy to combine the results of various networks in this proposed strategy to generate a correct detection result.

d. Comprehensive Performance Analysis and Benchmarking: To measure the efficiency of the proposed ADCEN model, comprehensive experimental analyses were conducted by using standard intrusion detection benchmarks. The results attained by the proposed design were compared with those of the existing models in the domain of NIDS based on metrics such as Accuracy, FP Rate, Precision, Recall, and F1-Score. The experimental results affirm the superiority of the proposed ADCEN method over the competing models in the case of evasion attacks.

1.7 SCOPE AND DELIMITATIONS OF THE STUDY

Within this research study, the scope sets the limits under which this research work has been accomplished. This thesis has kept its focus solely on network-based intrusion detection systems and has been concerned mainly with detecting evasion attacks by using deep learning techniques. By considering systemic defensive measures, this research has left out the defensive measures based on chemicals. This research has used only public domain datasets usually used in researching hop intrusion detection. On the one hand, these datasets offer a common ground for comparison, but on the other hand, they probably do not reflect all the features of the actual network environment. The framework also opts for a supervised learning model and does not look into training less supervised or semi-supervised detection mechanisms. The issue of encrypted traffic is another limitation. The proposed detection system is dependent on the availability of features that can be extracted from traffic data. The study does not include encrypted information analysis and privacy-preserving methods for intrusion detection. However, the research delimitations are such that the study could

still reach its aims. Besides, the meticulous marking of the research boundaries has kept the study on its theoretical track and validated its findings.

1.8 THESIS ORGANIZATION

This thesis consists of five chapters, each of which treats a different phase in the process of carrying out the proposed research. The outline of the thesis is the following:

a. Chapter 1: Introduction This chapter provides an overview of the current domain of research, specifically concerning the threat of cybersecurity attacks and the need for Network Intrusion Detection Systems in modern networks. Apart from explaining the research problem and amis of the proposed work, the chapter also identifies the limitations of other works in the domain and justifies the need for an efficient System attacks.

b. Chapter 2: Literature Review: This chapter presents an overview of the current state of the art in relation to NIDS solutions, tracing their evolution from conventional rule-based approaches to more contemporary machine learning and deep learning methodologies. There is also an in-depth discussion on approaches to detection, feature choice, and adversarial attacks. Moreover, the chapter points out the gaps in current research, which serve to introduce the proposed approach in the overall body of cybersecurity studies.

c. Chapter 3: Materials and Methodology This chapter covers the overall research methodology and presents the architectural design of the proposed Enhanced Network Intrusion Detection System. This chapter also highlights the importance of forming a new innovative model, as well as the data set used in the experimentation process. This chapter also identifies the use of the Improved Cheetah Optimizer (ICO) in formulating the best features in a minimal way that entails no redundancy as well as complexity. also illustrates the structure of the proposed model which entails a combination of Deep Temporal Convolutional Network (DTCN) model, LSTM, as well as GRU Networks. This chapter also highlights the implementation of the approaches of Attention, Dilated Convolution, as well as Fuzzy Ranking.

d. Chapter 4: Results and Discussion This chapter will discuss and interpret the experimental results obtained in implementing the proposed model. Firstly, the

experimental environment and validation criteria in model performance measurements shall be discussed. The obtained results will be interpreted in a multi-faceted manner, such as in terms of convergence validation, statistical performance, confusion matrix, and ROC curve analysis. The obtained results will also be compared with current models, establishing the higher performance, robustness, and adaptability capacity of the proposed ADCEN approach in resisting adversarial evasion attacks.

e. Chapter 5: This chapter of conclusion is utilized in summing up the important outcomes that are obtained in the course of carrying out the research activity, as well as the important results that are obtained from the experimental analysis. This chapter highlights the significance of the contribution that is brought about in the field of network security as well as intrusion detection.

2. LITERATURE REVIEW

2.1 BACKGROUND OF NETWORK INTRUSION DETECTION SYSTEMS (NIDS)

With the rapid development of the current technological age, organizations are increasingly depending on the idea of computer networks. With this increased dependence on computer networks, threats posed by attacks have also been increasing. This consists of virus attacks, denial of service attacks, phishing, ransomware, and advanced persistent threats (APTs) [31]. Threats of this nature pose a great risk to organizations in relation to the damaging of operations as well as the confidentiality of data. Network security is most imperative filed of current computer science and engineering trends.

One of the most effective techniques used in today's advanced cybersecurity solutions is the use of the Intrusion Detection System (IDS), which examines the activities that occur in the network and identifies any anomaly in the activities that hint at a possible attack [32]. There are two kinds of IDS, including the Host-based Intrusion Detection System (HIDS) that examines the data produced by a variety of computer systems in a network, such as logs, while the Network-based Intrusion Detection System (NIDS) examines activities in networks in search of a possible attack [33]. NIDS is a significant technique in identifying attacks that happen in a network.

Conventional IDS schemes rely on signature- or rule-based techniques in detecting attacks by matching incoming traffic with predefined patterns of attacks. Although these techniques possess high accuracy in detecting known attacks correctly, they fail to detect zero-day as well as unknown attacks, which keep evolving in pattern and behaviors [34]. To address these issues, Machine Learning (ML) techniques, as well as Deep Learning (DL), emerged as a solution in IDS systems that possess the ability to learn from data in terms of pattern discovery as well as generalization [35].

The transition from traditional IDS systems to intelligent IDS systems has been an important achievement in the field of cybersecurity studies. Machine learning-based IDS systems use algorithms such as Support Vector Machines (SVMs), Decision Trees (DTs), and Random Forests (RFs) to distinguish normal and malicious activities in networks [36]. These systems rely heavily on human-designed features and tend to be ineffective with high-dimensional or nonlinear data distributions.

Contrary to this, those designed with the aim of implementing deep learning algorithms, including Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), and Gated Recurrent Units (GRU), encompassing DL-NIDS designs, have been found to be more effective in acquiring hierarchical features from the data in the networks [37,38].

Nevertheless, despite such advancements, there also exist certain primary difficulties in implementing DNN-based NIDSs in a practical environment. A primary problem is that of evasion attacks by adversaries in respect to crafting subtle anomalous entries in the input data in a way that tries to induce evasion attacks on the model, leading to incorrect classification of malicious traffic as legitimate data entries [39]. Moreover, as adversaries primarily use adversarial machine learning, there arises a pressing need to develop robust systems resistant to such attacks as well. It is therefore due to this primary requirement of improvement in the field of NIDS that the academic community of researchers is focused on the development of optimized solutions in respect to combined approaches of various DNN-based techniques with intelligent feature selection strategies [40]

2.2 LITERATURE REVIEW

An important application of cybersecurity is the Intrusion Detection System (IDS) in detecting malicious activities, unauthorized accesses, and intrusions related to networks. However, in the past decade, the convergence of Machine Learning (ML) and Deep Learning (DL) approaches in IDS systems led to improvements in detection accuracy and adaptability. However, with the development of adversarial evasion attacks, which involve manipulating input data to fool detection models, the robustness of such systems has been questioned. This chapter presents a comprehensive review of the literature on recent developments in adversarial evasion detection, machine learning-based IDS, and deep learning models in network security from 2020 to 2025.

In 2020, Pawlicki et al. [23] explored the weaknesses of IDS models with adversarial attacks. They proposed and developed a number of models in ML, showing the degradation in performance and an attack detection mechanism based on the concept of an ANN to use such attacks. They concluded from their experimental work that even small deviation in network data can easily deceive the ML models used in IDS.

On the other hand, Li et al. (2023) [24] performed certain work related to adversarial attacks in the environment of the Industrial Internet of Things (IIoT). The authors have proposed a novel IDS called EIFDAA, which uses Machine Learning techniques for detecting Function-Discarding Adversarial Attacks. It has been designed to be resistant against different adversarial attacks.

Among the studies conducted in 2021, the one by Anthi et al. [25] focused on the attacks and Mapping Attack (JSMA) attack approach. The J48 and RF classified models showed a significant reduction in accuracy of 11% and 6%, respectively, with the addition of adversarial attacks. On the other hand, when these models were trained adversarially, they were more robust against different types of attack. This result indicated the potential of adversarial model training in enhancing robustness but also pointed out the computational complexity associated with adversarial training.

In the year 2020, Sidi et al. [26], have been able to implement an intrusion evasion technique designed to avoid “inline DGA classifiers”, it was found to be true, in fact, there is room for improvement in the detection of IDS with significant efficiency using the adversarial training-based evasion tactic.

The deployment of deep learning architectures in IDS has been proven to enhance the process of feature extraction and threat classification. Back in 2021, Yang et al. [27] introduced an adversarial instruction learning approach based on deep learning, called DeepMal, intended to analyze both static and dynamic network intrusions. DeepMal was designed to apply DL and ML approaches to craft adversarial attacks and identify them. The proposed approach demonstrated an increasing F1-score, along with the ability to fool current IDS solutions on one side, while also identifying sophisticated threats on the other, showcasing their hybrid adversarial learning approach's efficiency in current network security.

There has been more recent work done by Hamid et al. (2023) [28], whereby they proposed the combination of Deep Reinforcement Learning (DRL) with GANs, hence introducing the RELEVAGAN model (Relieve Adversarial GAN). The proposed model made the DRL agent work as an adversary to the discriminator in the GAN. This concept aided in speeding up the process and reducing the adversarial data needed during the training stage. The results

proved the efficiency of the proposed paradigm in reducing the training time of the generator and the detection precision.

Later, in 2021, the evasion attack capabilities of rule-based adversarial machine learning (AML) methods were further explored by Anthi et al. [29] against IDS models in an IoT-environment, specifically in the detection of Denial of Service (DoS) attacks. The outcome proved the significant effect of evasion attacks on IDS models, revealing a 47.2% reduction in detection accuracy, but adversarial learning significantly boosted the efficiency of detection.

On the other hand, the work of Fan et al. (2023) [30] specifically dealt with model stealing attacks and cloning attacks against ML-based NIDS. They proposed the Fast Model Stealing Attack (FMSA) method to demonstrate weaknesses in NIDS platforms, and they also designed a strategy to restore the distributions of private data with adversarial training. Through reducing query numbers and maximizing the efficiency of black-box cloning, their ML-based NIDS showed improved adaptability and stronger defenses against model-stealing attacks.

From the above review, it can be clearly noticed that although various models in machine learning and deep learning have been proposed to overcome adversarial attacks, most of the current solutions possess severe limitations in respect to:

a. Absence of generalization and adaptability in adversarial settings.

b. High computational cost when retraining or when constructing adversarial data.

c. Limited integration of optimization-based feature selection mechanisms.

d. Untapped potential in ensemble architectures with attention mechanisms and dilated convolution.

Thus, in an endeavor to overcome such gaps, the current study presents the ADCEN with the addition of Improved Cheetah Optimizer (ICO) to achieve the best possible feature extraction, along with the combination of LSTM, GRU, and DTCN models with Fuzzy Ranking Systems for the final decision-making process.

2.3 TAXONOMY OF NETWORK INTRUSION DETECTION SYSTEMS

2.3.1 Signature-Based Network Intrusion Detection Systems

Signature-based Network Intrusion Detection System (NIDS) identifies malicious behaviors through pattern comparison of network traffic with a pre-defined database of attack signatures. The patterns may contain byte patterns, misuse patterns, or patterns that describe specific payloads of an exploit. If signature-matched traffic is identified, an alert signal will be generated. Thus, it's effective in identifying malicious activities associated with previously defined patterns. In fact, according to recent research reports, signature-based NIDSystems are very effective in accurately identifying malicious activities without any false alarms during encounters with previously identified patterns of malicious activities.

Despite their success, signature-based NIDSs are inherently passive and unable to identify zero-day attacks or polymorphic malware. This is because an attacker can easily avoid identification either by changing bits of attack payload data or using encryption on malicious communication to prevent signature-based identification. In addition, to keep up with the latest updates on malicious activity, there is a constant need for human intervention expertise. All this narrows scalability for such systems based on environments with increasing threats [2]. As such, signature-based NIDSs are still commonly used on operational networks but have been complemented with intelligence to overcome their weaknesses.

2.3.2 Statistical-Based Network Intrusion Detection Systems

Statistical-Based Network Intrusion Detection Systems employ intrusion detection by modeling the normal network traffic using statistical measures like mean, variance, entropy, and probability distribution. Any pattern that is significantly deviation from the modeled pattern is termed as anomaly, potentially harmful. As a consequence, the statistical-based NIDS is able to identify a potential intrusion on the network, even if previously unseen, acting in a dynamic network environment [3].

Nevertheless, constructing a precise statistical model for normal behavior proves to be quite difficult because of the rather dynamic nature of today’s networks. Changes in traffic behavior, including flash crowds or changes in network topology, might be detected as an

attack, thus resulting in high false positive rates. Moreover, many statistical models incorporate assumptions about traffic that don't always turn out to be valid in actual deployments [4].

2.3.3 Machine Learning–Based Network Intrusion Detection Systems

Statistical-Based Network Intrusion Detection Systems employ intrusion detection by modeling the normal network traffic using statistical measures like mean, variance, entropy, and probability distribution. Any pattern that is significantly deviation from the modeled pattern is termed as anomaly, potentially harmful. As a consequence, the statistical-based NIDS is able to identify a potential intrusion on the network, even if previously unseen, acting in a dynamic network environment [3].

Nevertheless, constructing a precise statistical model for normal behavior proves to be quite difficult because of the rather dynamic nature of today's networks. Changes in traffic behavior, including flash crowds or changes in network topology, might be detected as an attack, thus resulting in high false positive rates. Moreover, many statistical models incorporate assumptions about traffic that don't always turn out to be valid in actual deployments [4].

2.3.4 Deep Learning–Based Network Intrusion Detection Systems

Deep Learning-Network Intrusion Detection Systems can be termed a breakthrough because they make it possible to obtain Hierarchical and Temporal Characteristics from the network data automatically. Deep Learning techniques CNN, LSTM, GRU, and TCN proved to provide better results in identifying complex-scale attacks on the network [7]. These techniques perform better when the dimensionality of data is large, and they can identify the "long-term" dependencies found in network data.

Though being excellent at detection tasks, deep learning-based NIDS systems are computationally intensive and require a considerable amount of labeled data. Another drawback is the inability to interpret results due to their black-box nature, which is a serious concern for security-critical environments. What is more alarming is that these deep models have demonstrated highly susceptibility to adversarial evasion attacks where minimal crafted

perturbations achieve severely reduced detection rates [8]. All these issues introduce requirements for developing robust intrusion detection systems.

2.4 ATTENTION MECHANISMS IN CYBERSECURITY

Attention models have become increasingly relevant for cybersecurity applications because of their efficacy to promote the discriminative power of deep learning models by selectively concentrating on relevant areas of input data. In NIDS, for network traffic data, the input features are high dimensional and explicitly and implicitly correlated in time. In standard deep learning models, all input features and temporal steps are processed uniformly, and therefore, they are more vulnerable to evasion attacks [1], [2].

Self-attention helps to establish overall dependencies in the input data, given the ability of each feature or time stamp to view all others. Its application appears to be efficacious in understanding overall dependencies among network traffic features, including packet statistics and protocol actions, and flow-level features. It has also been able to enhance the identification of overall attack patterns, which are challenging to establish through local and/or overall time stamping analysis [3].

Temporal attention works on finding the most informative time periods in sequential network traffic data. In most cases of intrusion, malicious activities happen during particular periods of a network flow, and other periods tend to look normal. The attention mechanism gives more emphasis to such crucial time periods. This helps a lot in finding low-rate attacks, slow scans, and advance persistent threats that try to evade detections through dispersing malicious activities in time periods of data. This is based on a research paper titled “Attention is Not Only for Features” published in 2017. In this paper, they also proposed an attention method for learning temporal patterns.

Channel attention models along the feature dimension to calculate how much each input channel or feature map contributes. Injections of network traffic involve a wide variety of attributes with differing levels of relevance for an ID system. Channel attention can boost distinctive attributes, such as illical usage of protocols or anomalous traffic, while filtering out redundant attributes like noise [5].

The attention mechanism also plays a major role in the defense against adversarial evasion attacks. This is because the attention mechanism narrows the focus to features which are more universal in meaning and want only in terms of time. Adversaries are thus forced to affect two or more features simultaneously to evade the model. This makes it more costly for a successful attack to occur. Thus, the attention mechanism is very important in improving the adaptability and reliability of the DL-based cybersecurity systems [6].

2.5 EFFECTIVENESS OF ATTENTION MECHANISMS AGAINST ADVERSARIAL EVASION ATTACKS

A goal of adversarial evasion attacks is to manipulate input data in order to cause misclassification while keeping semantic validity. In terms of NIDS, adversarial attacks might include minimal modifications to either packet or flow features.

Attention mechanisms are inherently adversarially robust because of the following reasons:

a. Selective Feature Emphasis. It makes attacks less dependent on weaker or more manipulable features, thus forcing the attackers to change various important characteristics at the same time, thus increasing the cost of attacks.
b. GLOBAL DEPENDENCY MODEling_: Since self-attention learning understands dependencies across dimensions and through time, it would be hard for a localized attack to mislead it.
c. Noise Removal: The channel and temporal attention masks irrelevant and noisy information to resist adversarial noise injection.
d. Adaptive Decision Making: Attention weight updates are adaptively learned so as to be capable of handling adaptating attack patterns.
e. Generalization Enhancement: Attention-based models emphasize inherent properties of attacks rather than superficial patterns; hence, their ability to generalize zero-day and polymorphic attacks improves significantly.

As a result, the implementation of attention mechanisms in deep learning-based NIDS greatly improves the robustness for adversarial evasion attacks and thus becomes an indispensable part of the cybersecurity defense system.

3. MATERIAL AND METHODOLOGY

3.1 METHODOLOGICAL FRAMEWORK OF THE PROPOSED STUDY

Intrusion Detection Systems (IDS) are longstanding trusted defense mechanisms for securing networks. Such systems alert administrators to suspected or malicious actions within a network. In the last ten years, developments in the field of artificial intelligence (AI) have greatly extended the functionality of IDS. Specifically, machine learning (ML) strategies were widely utilized to refine detection efficacy, analysis depth, and the power to discern both altered and unexpected attack patterns. Of the latter strategies, deep learning (DL) has come to play an especially significant role in securing networks. Deep learning-based Network Intrusion Detection Systems (NIDS) overall realize higher performance and accuracy than conventional IDS techniques. Nevertheless, some recent studies indicated that deep learning models—particularly those utilized for the task of image classification—see a dramatic decrease in accuracy under the threat of adversarial attack. Despite the above, there is not yet wide-ranging examination concerning how the above kinds of models might equally affect the performance of NIDS.

Thus, an optimal deep learning-driven (NIDS) will be formulated for protection from attacks that are pernicious in nature. In the envisioned framework, attack datasets will first be sourced from standard benchmark datasets, and the resulting dataset will then be utilized for the extraction of optimal features. The selection of the best attributes that are the most relevant and not redundant at all will be performed through the (ICO) [41]. The optimally chosen attributes will then be fed to the proposed (ADEN) for the detection of intrusion. The ADEN model will combine the use of three deep learning models namely, DTCN, LSTM, and GRU networks to efficaciously capture the spatial and temporal properties of the network traffic. Lastly, a fuzzy rank-based method will be utilized to produce the output for intrusion detection from the ensemble network. Detailed experiments are to be performed to prove the efficacy and robustness of the devised deep learning-driven NIDS to capture attacks pernicious in nature. The devised model for the NIDS is schematically denoted in Figure 3.1.

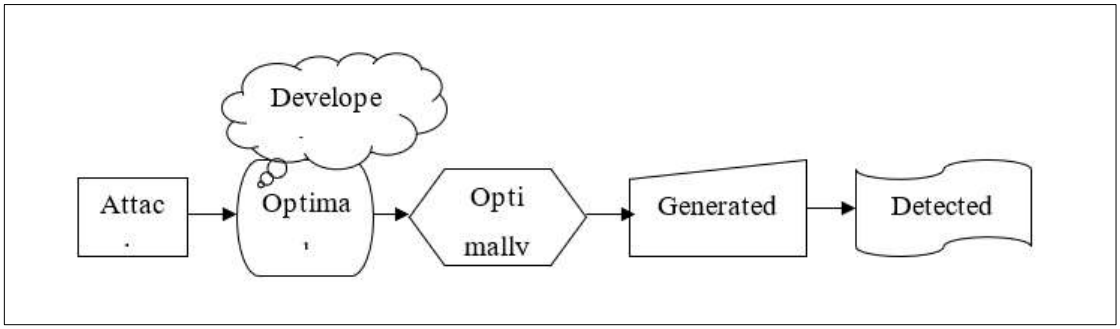


Figure 3.1: Diagrammatic Overview of the Developed DL-Based NID Model.

3.2 ROBUST NETWORK INTRUSION DETECTION FRAMEWORK FOR MITIGATING ADVERSARIAL EVASION ATTACKS

3.2.1 Rationale For Designing A Robust Network Intrusion Detection System

A Network Intrusion Detection System, or simply NIDS, is designed to find malicious action going on within a network. To achieve this task successfully, it is not uncommon for the NIDS to require real-time monitoring of all traffic, both unicast traffic flows included. Its passive method of monitoring enables a NIDS to listen in and examine traffic without interfering with normal operations. NIDS can be established at a one theme or at various peaks throughout the network to provide continuous monitoring of traffic.

NIDS matches observed traffic on the network subnets with a pre-established attack library or database consisting of known attacks. The system raises an alert whenever it finds anomalous behavior or hitherto unseen traffic patterns. Although many NIDS were constructed incorporating sophisticated detection mechanisms, the systems are prone to attack techniques that intentionally target vulnerabilities in the detection process to remain unidentified.

Special attacks that are aimed at breaking machine learning-driven NIDS are called adversarial evasion attacks. They are targeted attacks that exploit the weakness of the NIDS models, thus letting the attacker slip through the net and compromise network security. Therefore, it is very important to precisely recognize and prevent adversarial evasion attacks to protect the reliability of network defense mechanisms.

Even though detection models for the attacks evolved, the systems remain subject to some limitations, such as poor detection rates, skewed datasets, excessive false alarms, and the

requirement for longer response time. Over the past years, deep models have been popular for the detection of adversarial attacks with the intent to exploit the stronger analysis power these models possess. Nevertheless, these models are extremely prone to input-manipulating attacks, specifically adversarial attacks, that lead to misclassification and decrease the system's overall reliability.

3.2.2 Design And Development of the Proposed Network Intrusion Detection System

To protect precious transactions and information transmission, the novel Network Intrusion Detection System (NIDS) is designed with machine learning approaches. Such NIDS can recognize abnormal traffic transmissions through a network with the potential to discover malicious activities. However, it is common that the existing NIDS models are prone to various adversarial evasion attacks carried out by malicious users. To overcome this obstacle, this research applies an effective framework for detecting the adversarial evasion attack with deep learning and an intelligent feature selection process.

First, the corresponding network data are harvested and processed through the Improved Cheetah Optimizer (ICO) to find the best features that are the most informative for identifying the adversarial evasion attacks. The step for the selection of the best features decreases the computational overhead by removing the redundant or irrelevant features, thus increasing the efficiency and the model accuracy.

These optimal selected features are then passed through the Attention and (ADCEN) to achieve intrusion detection. The ADCEN model combines the usage of Long Short-Term Memory (LSTM), Deep Temporal Convolutional Network (DTCN), and Gated Recurrent Unit (GRU) models, where all the models give independent detection outputs. These independent detection outputs are then, through a fuzzy ranking technique, merged to give the final detection output for the adversarial evasion attacks. the architecture showed in figure 3. 2.

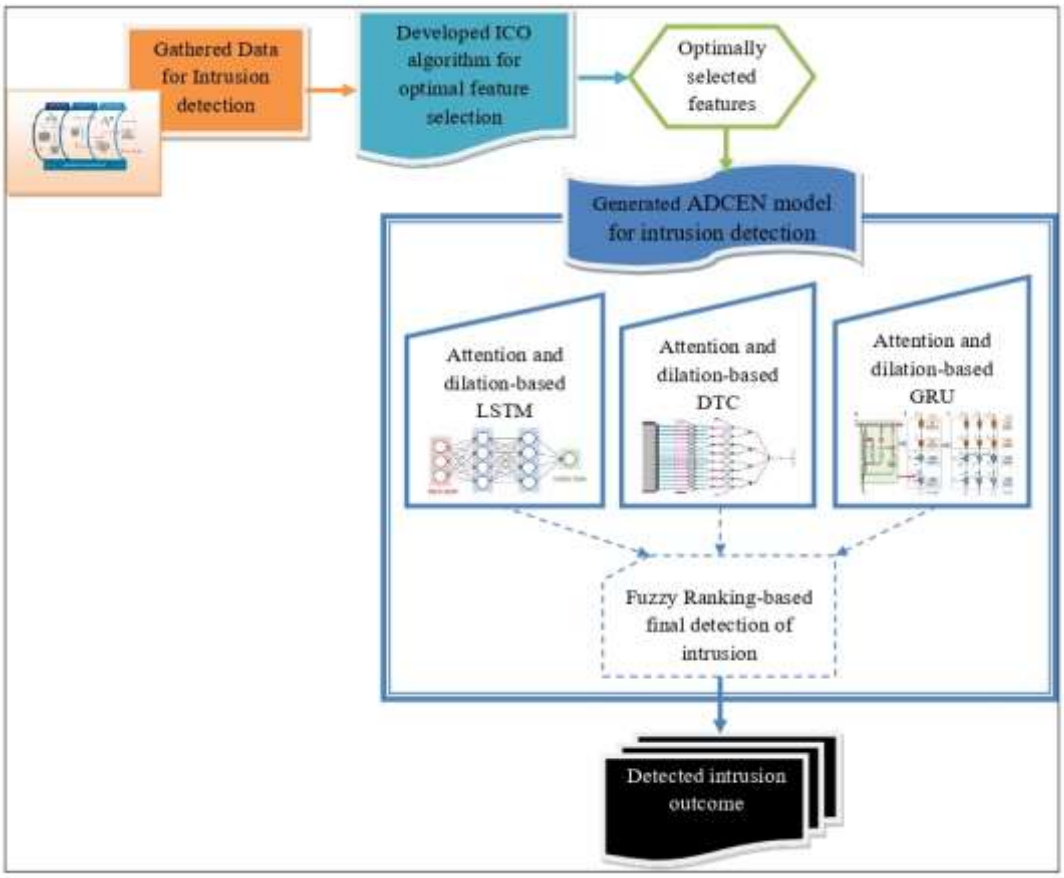


Figure 3.2: Architecture of the DL-Based ADCEN Model for Adversarial Evasion Attack Detection.

3.3 THE DATASET EMPLOYED FOR INTRUSION DETECTION

The attack data employed in this work were obtained from a publicly available standard data benchmark, which is used both in training and in the testing of the detection model. The data involves a broad array of network traffic records, both legitimate and malicious. These network traffic records contain several features that allow the model to be trained on them, learning patterns in the data pertaining to the attacks. describes the characteristics of the Chosen dataset The Parameters and Features of the Data, along with the Distribution of Attack Categories, have been included in Table 3.1.

Table 3.1: Characteristics of the Chosen Dataset.

Dataset Name	Dataset Link	Description
"CICIDS2017"	"https://www.kaggle.com/datasets/ cicdataset/cicids2017/data access date: 2023-10-30"	• This dataset includes the recent and prominent attacks that match the actual attack scenario. Together with the "time stamp based annotated flows, port information regarding of source and destination, the IP address of source and destination, attacks, and protocols", this dataset also contains the findings of the network traffic is obtained with the aid of "CICFlowMeter". • This dataset was generated with the goal of resembling the actual background traffic. This dataset uses 25 users' abstract behavior on the basis of "email, FTP, HTTP, SSH, and HTTPS" protocols. For 5 days, starting from July 3 to July 7 2017, the data were collected. "Denial of Service (DoS), Brute Force FTP, Heartbleed, Brute Force SSH, DDoS, Botnet, Web Attack, and Infiltration" are some of the attacks that have been obtained in this dataset. Among all these, only data regarding Friday morning and afternoon are considered for our research work.

3.4 ENHANCED CHEETAH OPTIMIZER FOR OPTIMAL FEATURE SELECTION IN NETWORK INTRUSION DETECTION SYSTEMS

3.4.1 Developed Improved Cheetah Optimizer

We here choose the Cheetah Optimizer (CO) to be the optimization method for this task, given that it is very efficient for searching for the best solution, it has strategies for evading the local optima, and it has mechanisms to avoid early convergence. Nevertheless, if the number of the CO random variables is large, it is not easy to find the actual best solution and it will be computationally expensive. Therefore, to compensate for this weakness, the CO random variable is changed, with the strategy being formulated in the equation (1), so the Improved Cheetah Optimizer (ICO) is derived.

$$h = \frac{fit_{bst}}{25 * fit_{mean}} \tag{3.1}$$

In Equation (1), the terms are the best and average fitness value, respectively. The random variable in the standard Cheetah Optimizer (CO) is distributed within a predefined range. The predefined range is replaced with the range provided in Equation (1) to refine the selection process, and the resulting optimization is the ICO Improved . The ICO is then utilized to optimally choose the features from the aggregated dataset, making the ensuing intrusion detection process more efficient and accurate.

CO [26] is also inspired by the hunting behavior of cheetahs, wild cats that are native to the African and Iranian forests. The cheetah is famous for being very agile and moving at incredible speeds, with the natural consequence that it is an efficient hunter. When hunting, the cheetah mixes stealth with short bursts of speed, but it cannot maintain the high-speed chase for very long. To achieve maximum hunting success, the cheetah's hunting strategy is to adopt a "hide and catch" strategy: it hides close to the potential prey until the optimal time, then charges quickly to catch the prey. The natural hunting strategy is mathematically modeled to design the CO. Several assumptions are implemented in the CO to model the cheetah's searching and capture strategy, the foundation for optimization for feature selection and other computational tasks.

Assumptions and operations of the primary mechanisms for the Cheetah Optimizer (CO) are given below:

a. Population Matrix Presentation: The locations of the cheetahs around the decision variables of the search space are the columns of the CO population matrix, and individual cheetahs at various iterations are represented through the rows. The fitness is computed for each individual cheetah, and the cheetah with the overall highest fitness is taken to be the cheetah with the greatest probability to catch the prey successfully.

b. Hunting Energy: The cheetah's stored hunting energy is not depended on the prey itself. The energy is calculated with some parameters $\hat{h}_{a,c}^{-1}$and$\breve{h}_{a,c}$.to consider the hunting potential precisely.

c. Interactions among Cheetahs: Interactions among other cheetahs from the populace are feasible with the optimization process.

d. Random movements: The random movements for searching and hunting stages are those of cheetahs. The random movements are controlled through parameters $\hat{h}_{a,c}^{-1}$,$\breve{h}_{a,c}$,$\delta_{a,c}^{d}$, and$\beta_{a,c}^{d}$to obtain precise modeling for the CO behavior.

e. Stages Without Movement: The cheetah is not moving at the foraging and the waiting stages; it is moving only at the hunting stage.

f. Phase Selection Strategy: The hunting phase and the waiting phase are chosen through two randomly selected variables $h_3 \leq h_2$, t. When some parameter meets some criterion, the waiting phase is implemented. The hunting phase is initiated with reference to some other randomly selected variable. The randomly selected parameters $E = exp^{2\left(1-\frac{d}{D}\right)}(2h_1 - 1)$ are also bounded within some specified range, while lies within some other specified range$[0,1]$.

g. Leader Cheetah Evaluation: When the leader cheetah that is chosen is not able to catch the prey within the time for two or three continuous iterations, it is said to be the worst solution and is shifted to the lowest position in the population matrix. Exploration is facilitated through the maintenance of a fixed position for the prey at a very restricted region in the search space.

h. Running Out of Energy and Position Update: When the energy of the cheetah runs out, it goes back to the initial position if the global solution is not reached. However, if the position of the leader cheetah is held fixed, the position of the cheetah is then updated to prevent getting stuck at local optima.

i. Evolution of the Population: Some of the population is given evolutionary enhancements at the end of each iteration to speed up the whole searching process.

CO runs the three major phases, i.e., foraging, waiting, and hunting. The phases are also simulated for the CO to bring the cheetahs to the optimal solution. The reason behind each phase is described below.

Foraging: The cheetah, to begin with, is supposed to sense the perfect prey inside or outside the search (lookup) space. The above can be achieved with the help of an "active patrolling" or an "ideal sitting" strategy, contingent on the activity of the prey. The decision among these strategies is contingent on a number of aspects, some of which are the coverage space, the status of the cheetah, and the state of the prey.

For mathematical formulation of the searching ability of the cheetah, the first step for the algorithm is the initialization of the population. Set the size of the population to be the number of cheetahs $a = 1,2,..,b$ in the searching space, and the searching space to include problem variables, the dimension of the problem space to be provided by each variable $c = 1,2,..,A$. The position of the cheetah is described with, and the position where the prey is with the optimal solution with the decision variables.

Based on their current positions, the cheetah's location $B_{a,c}^{d}$is updated during the search process with respect to the decision variables. This update mechanism is mathematically formulated in Equation (2), allowing the cheetah to progressively approach the optimal solution.

$$B_{a,c}^{d+1} = B_{a,c}^{d} + \hat{h}_{a,c}^{-1} \cdot \beta_{a,c}^{d} \tag{3.2}$$

In equation (2), the terms $d$are described accordingly: is the time spent searching for hunting$\beta_{a,c}^{d}$, a^{th} is the step size for the cheetah moving through the decision variables d^{th}, is randomly drawn with a normal distribution $\hat{h}_{a,c}^{-1}$, is the limit for hunting time, is the current

location for the cheetah, and is the location for the cheetah at the subsequent hunting time. The step size is always given a positive value, and for the foraging period, it is mostly initialized to a lower value due to the perception that the cheetah is a slow-searching entity.

Step size can also be changed through the position of other cheetahs, and the algorithm can choose a tighter or further away searching strategy. The random parameter is given to each cheetah uniquely at various hunting time instances. The location of the cheetah for each decision variable is calculated through the multiplication by the maximum allowable step size, which is restricted with the upper and lower limits of the searching space. The above formulation is utilized to find the position of the leader cheetah for the decision variables.

For all cheetahs except the leader, the locations are accordingly updated to the distance between the cheetah and any other randomly selected cheetah. The leader cheetah is at a measurable distance from the best prey, and the position is accordingly updated to the location of the prey. The leader cheetah approaches the prey stepwise with the elapse of the hunting time, with the position iteratively updated until the hunting process is completed. Waiting: The prey is believed to be within an easily accessible distance to the cheetah during the waiting period. Not to disturb the prey earlier than necessary, the cheetah stays concealed and immobile, restricting unwanted movements that might give the prey a chance to escape. Mathematically, this action is represented in Equation (3), mimicking the strategy the cheetah applies to wait for the prey to be at a hunting distance before it starts the hunting process.

$$B_{a,c}^{d+1} = B_{a,c}^{d} \tag{3.3}$$

These position upgrades for the Cheetah Optimizer (CO) are not carried out at the same time for all cheetahs. Such updating is staggered to assist the algorithm to gradually converge to the optimal solution for the hunting phase and prevent early convergence.

Hunting: During the hunting process, the cheetah applies its swiftness and agility to seize the target. Upon identifying the target, the cheetah embarks on a high velocity attack with all force aimed at achieving the target. When the target tries to run away, the movement and direction of the cheetah are continually changed to match the direction the target is running. Besides, the position of the leader cheetah is also dictated by the position of other

surrounding cheetahs to achieve a synchronized and efficient hunting process. The hunting process of the cheetah is mathematically represented in Equation (4) below:

$$B_{a,c}^{d+1} = B_{C,c}^{d} + \check{h}_{a,c} \cdot \delta_{a,c}^{d} \tag{3.4}$$

Here, h denotes the current position of the cheetah, with the other notations standing for the current position, step changes, and direction changes that move the cheetah toward the prey. The hunting process with this expression truly balances exploration and exploitation of the searching space.

In Equation (4), $\hat{h}_{a,c}$ is the turn parameter for the a^{th} cheetah under the d^{th} decision variable, and $B_{C,c}^{d}$ is the optimum position for the prey under the corresponding decision variable. The formula $\delta_{a,c}^{d}$ is the interaction constant for the cheetah, defining interactions.

Its movements and transformations are regulated through the turn factor. It is calculated through the formula: $| h_{a,c}^{-1} |^{\exp(h_{a,c}/2)} \cdot \sin(2\pi h_{a,c})$ where $h_{a,c}$ is a normal distribution variable. The variable is then fine-tuned with the concept of fitness introduced in Equation (1) to further enhance the ability of the algorithm to exploit and explore the space.

Algorithm 1: Executed ICO

The population count of cheetahs at the beginning of the algorithms b is preset

The A dimension and the problem data are initialized

The population of the cheetah $B_a(a=1,2,..,b)$ is initialized

Fitness values *fit* are computed for all the cheetahs

The leader cheetah, prey, and the home of the cheetah are generated

$d=0$

$g=1$

Fix the maximum iteration count as G

$$D=60\times\left(\frac{A}{10}\right)$$

While $(g\le G)$

- Arbitrarily choose a member $i(2\le i\le b)$ from the cheetah population
- For $a\in i$
  - The neighbour of a is determined
  - For $c\in\{1,2,...,A\}$
    - The value of $\hat{h}$, $\tilde{h}$, β, δ, and E are determined
    - **The value of the arbitrary parameter h is amended using the fitness concept as given in Eq. (1)**
    - The value of h_2 and h_3 is initialized as $[0,1]$
    - If $h_3\le h_2$
      - Generation of h_4 in the range $[0,3]$
      - If $h_4\le E$
        - The hunting step is performed to update the location of the cheetah
      - Else
        - The foraging step is used to update the position of cheetah
    - Else
      - The waiting step is executed to update the position of the cheetah
    - End
  - End
  - The leader, as well as the a^{th} member's positions, are updated
- End
- $d=d+1$
- If $(d>rnd\times D)$ along with no change in the leader's location
  - The cheetahs return to their home
  - The location of the leader is amended
  - The position of the prey is given to the cheetah
- $d=0$
- End
- $g=g+1$
- The best solution is amended

End

Figure 3.3: Algorthim1 Executed ICO.

The flowchart illustrating the improved cheetah optimizer (ICO) is presented in figure 3.3.

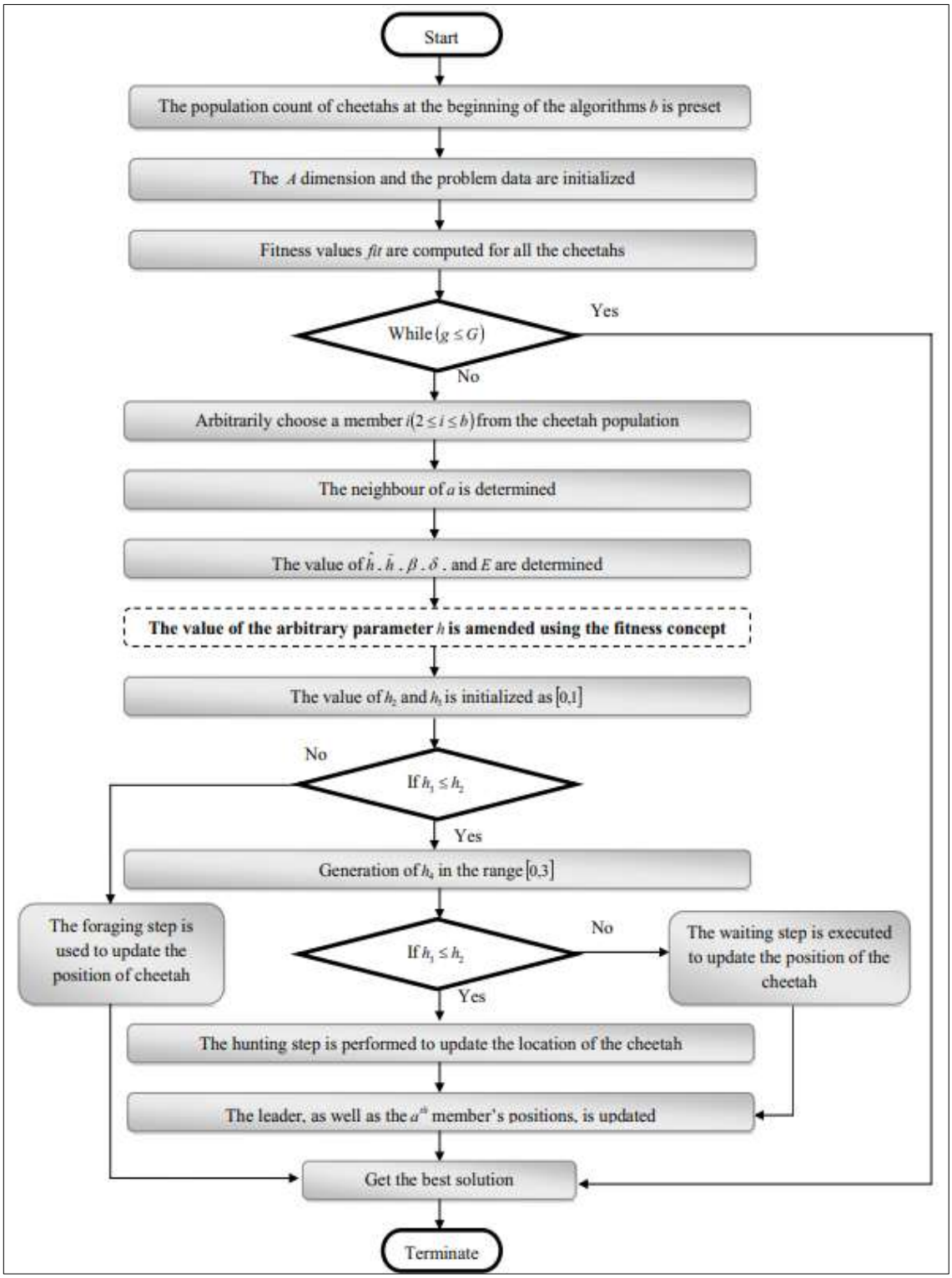


Figure 3.4: Flowchart Illustrating the Operational Process of the Improved Cheetah Optimizer (ICO)**.**

3.5 OPTIMAL FEATURE DETERMINATION USING THE IMPROVED CHEETAH OPTIMIZER (ICO)

The collected dataset (AD_{ad}^{data}) is utilized to perform an optimal feature selection procedure prior to identifying adversarial evasion attacks using the developed deep learning–based detection framework. This dataset (AD_{ad}^{data}) contains multiple features, some of which are redundant or lack meaningful contribution to the adversarial attack detection process. Including such irrelevant attributes can increase computational complexity, cause unnecessary processing overhead, and raise the overall computational cost. Hence, it is essential to select only the most informative and relevant features before feeding the data into the deep learning–based adversarial evasion detection system. In this study, the optimal feature selection is achieved through the use of the Improved Coyote Optimization (ICO) algorithm. The ICO efficiently identifies and retains only the significant features, which are then provided to subsequent analysis. The best chosen feature subset generated by the ICO is represented as (OF_{of}^{ICO}) in Figure 3.4.

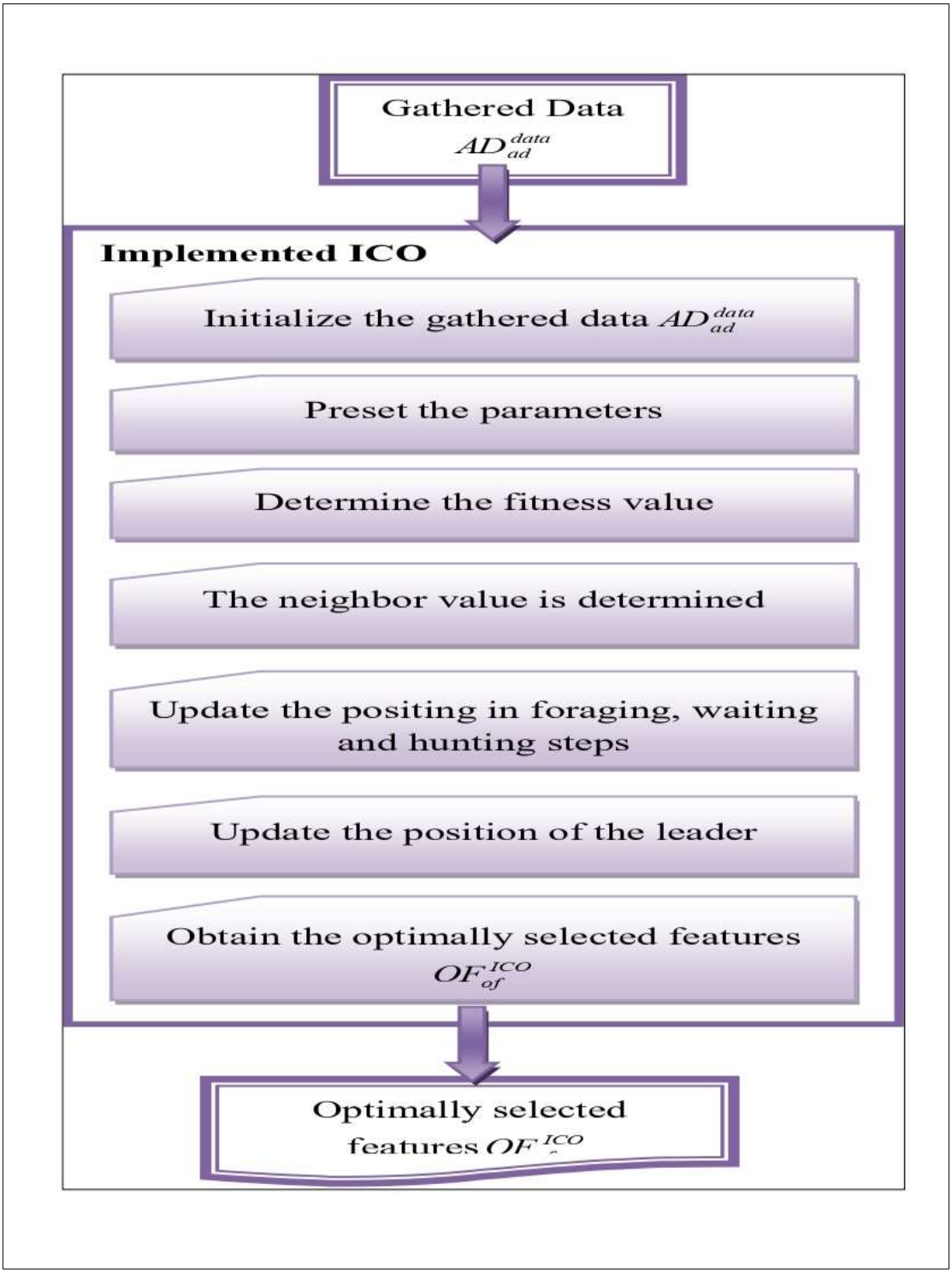


Figure 3.5: Schematic Representation of the Optimal Feature Selection Process Implemented through (ICO)**.**

3.6 OBJECTIVES OF THE IMPROVED CHEETAH OPTIMIZER–DRIVEN FEATURE SELECTION FRAMEWORK

The Improved Coyote Optimization (ICO) algorithm performs the optimal feature selection process with the objective of maximizing intra-class correlation while minimizing inter-class correlation. This dual objective ensures that the features within the same class remain closely related, whereas the distinction between different classes is emphasized, ultimately improving the accuracy of adversarial evasion attack detection. The objective function employed in the proposed detection framework is expressed in Eq. (5).

$$ob = \underset{\{OF_{of}^{ICO}\}}{\arg\min}\left(\frac{1}{IaC} + IeC\right) \tag{3.5}$$

The feature selection process operates within the range $[1, of]$, where of corresponds to the number of features chosen from the dataset. The correlation coefficient for different classes is calculated using Eq. (6), while that for the same classes is determined using Eq. (7).

$$I_aC = \sum \frac{(OF_{of}^{ICO_d} - mn(OF_{of}^{ICO_d})) \times (OF_{of}^{act_d} - mn(OF_{of}^{act_d}))}{sm \times sd(OF_{of}^{ICO_d}) \times sd(OF_{of}^{act_d})} \tag{3.6}$$

In Eq. (6), $OF_{of}^{ICO_d}$ represents the features selected from different classes, ρ_d denotes the correlation coefficient between features of different classes, $OF_{of}^{act_d}$ corresponds to the actual features, sm indicates the sample size, mn represents the mean value, and sd denotes the standard deviation.

$$IeC = \sum \begin{pmatrix} OF_{of}^{ICOs} \\ -mn\left(OF_{of}^{ICOs}\right) * \frac{OF_{of}^{acts} - mn\left(OF_{of}^{acts}\right)}{sm * sd\left(OF_{of}^{ICOs}\right) * sd\left(OF_{of}^{acts}\right)} \end{pmatrix} \tag{3.7}$$

In Eq. (7), $OF_{of}^{ICO_s}$ signifies the features extracted from the same class, ρ_s denotes the correlation coefficient among features belonging to the same class, and $OF_{of}^{act_d}$ indicates the actual feature set used for comparison.

3.7 DESIGN OF AN ATTENTION-ENHANCED DILATED CONVOLUTION ENSEMBLE FRAMEWORK FOR NETWORK INTRUSION DETECTION

3.7.1 Deep Temporal Convolutional Neural Network Architecture (DTCNN)

An efficient form of the conventional CNN, optimized for time series data and providing enhanced accuracy, speed, and compact modeling, is termed the Temporal Convolutional Network (TCN) [42]. The TCN represents a Deep Learning algorithm optimized for sequence data via the utilization of causal convolution and dilation. Contrary to the conventional CNNs, the TCNs are designed for the attainment of a vast receptive field while keeping the memory requirements small. The network proceeds sequentially for the input information, thereby being the most suitable for the modeling of time series information. In terms of architecture, the TCN comprises input/output layers of equal size, featuring both causal 1D convolution layers and dilated convolution layers.

Causal Convolutional Layers: A key property of the TCN network is the fact that it uses causal convolutions, which allow it to generate an output array of the same size as the input array; this proves to be very useful for the TCN, much like an RNN architecture. In the process, each output at a certain point in time only depends on the input values at the current point in time and previous points in time, meaning the input at a latter point in time cannot affect the current output. Unlike the traditional CNN architecture, the TCN uses kernels aligned through the time axis. Dilated Convolutional Layers However, the process of using conventional causal convolution operations for large input data might require the consideration of deeper architectures and larger filter sizes, thus increasing computation costs and the risk of training instability associated with the complexity of such networks. To address the above problems and limitations, dilated causal convolution techniques are used in TCNs, where the convolutional filters are extended via a dilation rate. Through this process, the receptive field grows within an exponential rate without the need for additional layers and/or parameters to handle long-range dependencies in sequence data effectively. The receptive filed of the given TCN can be computed via equation (8).

$$Q = 2^{M}(J - 1) \tag{3.8}$$

In equation (8), the represents the receptive field, whereas K represents the size of the kernel, which gets enlarged based on the predefined dilation rate. The dilation operation causes the network's output to be affected by inputs from the K·(J−1) preceding time steps, where J represents the number of layers in the architecture. With the addition of every new layer based on dilated convolution, the receptive field gets considerably enlarged, thus permitting the modeling of long-range dependencies within the time series without a substantial increment in the parameter size, keeping the architecture lightweight and efficient.

Residual Units: In the TCN architecture, residual connections are used to ease the information transfer process between layers. The input is passed to the subsequent layers using skip connections, which are the combination of the input w and the transformed representation

$$N = w + S(w) \tag{3.9}$$

In the residual block, the output from the convolution layer, say w, and the output from the residual block, denoted as S(w), are used to produce the output from the residual unit. The formula with this residual representation, the TCN is able to utilize both the identity mapping and learned residual mapping to improve the training process of the network. A residual block within the TCN architecture contains two dilated causal convolution layers, two Rectified Linear Unit activation functions, and a dropout layer. The dropout layer prevents the development of overfitting within the network, whereas the Rectified Linear Unit introduces nonlinearity into the network for faster convergence during training. The architecture of the Dilated Temporal Convolutional Network (DTCN) is depicted in Fig. 3.5.

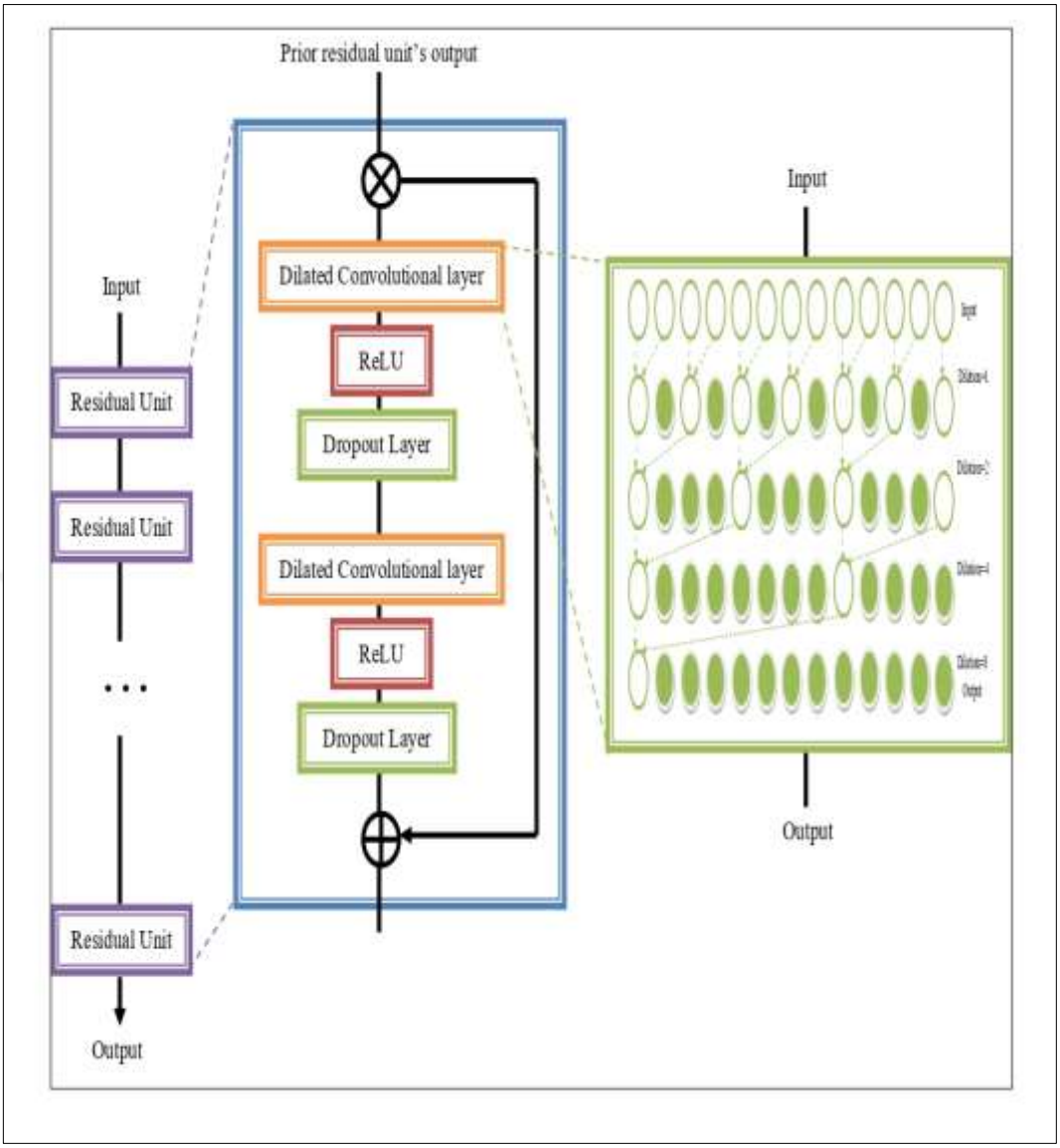


Figure 3.6: Architecture of the DTCN.

3.7.2 Long Short-Term Memory–Based Neural Network Model

The Long Short-Term Memory (LSTM) network is a new and powerful variant of recurrent neural architecture that was designed to improve the learning of long sequential data and to eliminate some of the main issues that traditional Recurrent Neural Networks (RNNs) have. One of the main problems that LSTM has tackled is the vanishing gradient that comes with long training sequences. To solve this problem, the LSTM model has special memory cells and gate mechanisms that control the storing, altering and passing on of information. This

way, the network can maintain stability and learning efficiency while capturing long-range temporal dependencies. [43].

The functional process of the LSTM is controlled by four gates, namely the input gate, the forget gate, the output gate, and the update gate, which manage the storing, updating, and flushing of information associated with the sequence. In an LSTM, the storing process of information is achieved by the combination of the sigmoid and tanh functions, and the network may have one or two memory cells based on the architecture.

The input gate controls the degree to which the new information goes into the memory cell; it is given by equation (10):

$$\alpha_p = \sigma\left(H_\alpha^r \times r_p + H_\alpha^q \times q_{p-1} + s_\alpha\right) \tag{3.10}$$

In equation (10), α_p , p denotes the input gate at the pth time step H_α^q and indicate the weights used for the input and hidden layers, respectively; q (p–1) represents the former state of the hidden layer denotes the bias and σ represents the sigmoid function, calculated via equation (11):

$$\sigma = \frac{1}{1+exp^{-r}} \tag{3.11}$$

The update gate, responsible for updating the information retained in memory, is expressed in Eq. (12):

$$\chi_p = \sigma(H_\chi^r \times r_p + H_\chi^q \times q_{(p-1)} + s_\chi) \tag{3.12}$$

Here, $\chi_p$denotes the update gate, $H_\chi^r$and $H_\chi^q$are the input and recurrent weight matrices, and $s_\chi$represents the bias.

The forget gate determines which information should be removed or retained in the memory cell. It is formulated as Eq. (13):

$$\varepsilon_p = \sigma\left(H_\varepsilon^r \times r_p + H_\varepsilon^q \times q_{(p-1)} + s_\varepsilon\right) \tag{3.13}$$

In Eq. (13), $\varepsilon_p$represents the forget gate, $H_\varepsilon^r$and $H_\varepsilon^q$are the weights corresponding to the input

and hidden layers, and $s_\varepsilon$is the bias. Based on the previous state $p-1$, the memory cell's contents are updated through the interaction between the forget and input gates.

The output gate controls the portion of the memory cell's content that should be output to the next layer, as shown in Eq. (15):

$$\phi_p = \sigma\left(H_\phi^r \times r_p + H_\phi^q \times q_{(p-1)} + s_\phi\right) \tag{3.15}$$

Here, $\phi_p$represents the output gate, $H_\phi^r$and $H_\phi^q$are the weight matrices, and $s_\phi$is the bias term.

The hidden state of the LSTM, which represents the processed information at each time step, is computed using Eq. (16):

$$q_p = \tanh\left(\varepsilon_p\right) \times \phi_p \tag{3.16}$$

In Eq. (16), tanhdenotes the hyperbolic tangent activation function, calculated as follows:

$$\tanh \quad = \frac{e^r - e^{-r}}{e^r + e^{-r}} \tag{3.17}$$

Finally, the output of the LSTM at each time step is expressed in Eq. (18):

$$t_p = H_q^t \times h_p + s_t \tag{3.18}$$

In equation (18), H_q^t refers to the weight matrix for the final hidden state, s_t indicates the bias term, and h_p refers to the hidden state at time t_p . The weights and biases for the LSTM network are then optimized via the backpropagation through time algorithm.

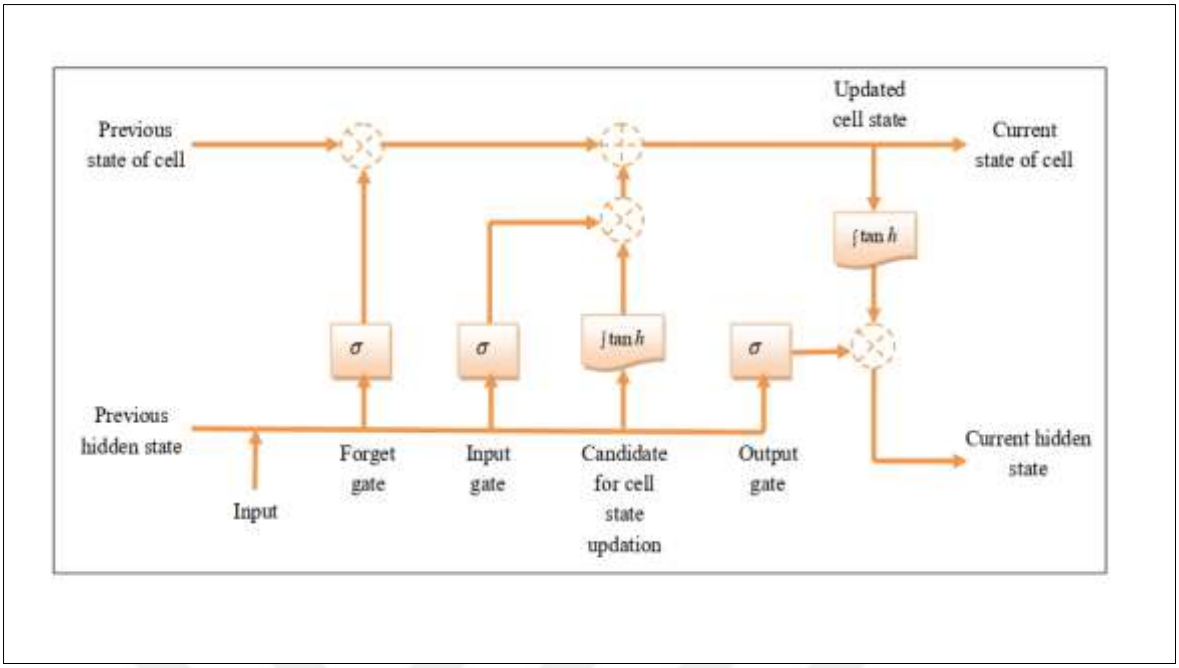


Figure 3.7: Diagrammatic Representation of the Conventional LSTM Model.

3.7.3 Gated Recurrent Unit

The Gated Recurrent Unit (GRU), a Recurrent Neural Network (RNN), is designed for the efficient execution of the algorithm over the input sequences for modeling their dependencies [44]. In some cases, the performance of the GRU may be superior to the performance of the LSTM because the former consumes less memory during training and takes less training time; however, for sequences that contain very long sequences, the performance of the former may be inferior since the complexity of the memory management process of the latter is high.

Unlike the LSTM, the GRU doesn't contain an output gate, thus fewer parameters are used during training. In the GRU architecture, the process of information passage through the cell states is governed by two gates, which are the update gate & reset gates. The statutes gates ensure the model's capability to either remember and forget the information at an appropriate time by balancing between the short-term and long-term dependencies.

The architecture of the GRU is generally patterned after the architecture of the LSTM, the only difference being the combination of the input and the forget gate into the update gate in the former architecture. The update gate decides the portion of the previous hidden state

to be carried over to the next time step as the new hidden state; therefore, the GRU training converges faster compared to the training of the LSTM.

In certain cases, the results obtained by the GRU have proved to be superior to the results obtained by the LSTM. Like the previously mentioned LSTM, the GRUs are also able to handle the 'vanishing gradient' issue, which generally takes place during the training process for the sequence data.

The proposed GRU network uses only the update and reset gates, which are both essential for the information management process within the network. The reset gate defines the level of information from the previous layer to be ignored during the computation of the new candidate activation. The formula used in the reset gate process is described in equation (19).

$$m_k = sgm([j_{k-1}, n_k] + o_m) \tag{3.19}$$

In equation (19), the term (sgm)refers to the sigmoid activation function, $(m_{k)}$)
represents the reset gate, j (k−1) signifies the previous hidden state,
(j_{k-1}) represents the bias term related to the reset gate, and
(n_k), on the other hand, refers to the input at the current time step. The reset gate decides the level of information to be forgotten for the calculation of the new candidate activation.
The functioning of the update gate, which decides the extent to which the old information from the hidden layer should be kept and to which extent the new information should replace it, is given by the equation in (20).

$$l_k = sgm([j_{k-1}, n_k] + o_l) \tag{3.20}$$

In Eq. (20), the term $l_k$represents the update gate, while $o_{(l)}$denotes the bias associated with the update gate. The update gate regulates how much of the previous information should be preserved and how much should be replaced with the new candidate activation at the current time step.
The candidate state of the GRU cell, which holds the potential new information to be added to the hidden state, is computed using Eq. (21):

$$\tilde{j}_k = \tanh\left(F_j n_k + m_k \otimes \left[F_j j_{(k-1)}\right] + o_j\right) \tag{3.21}$$

In Eq. (21), the term tanhdenotes the hyperbolic tangent activation function, $F_j$represents the weight matrix, $o_j$indicates the bias for the candidate state, and $\tilde{j}_k$corresponds to the candidate activation state. The operator $\otimes$signifies element-wise (scalar) multiplication.
The hidden state of the GRU at the current time step is determined using Eq. (22).

$$j_k = (1 - l_k) \otimes j_{(k-1)} + l_k \otimes \quad (3.22)$$

In Eq. (22), $j_k$denotes the hidden state of the GRU. This equation combines the previous hidden state $j_{(k-1)}$and the newly computed candidate state $\tilde{j}_k$, weighted by the update gate l_k. This mechanism allows the GRU to retain essential past information while incorporating relevant new input, ensuring an efficient balance between memory and adaptability. The overall architecture of the GRU is illustrated in Fig. 3.7.

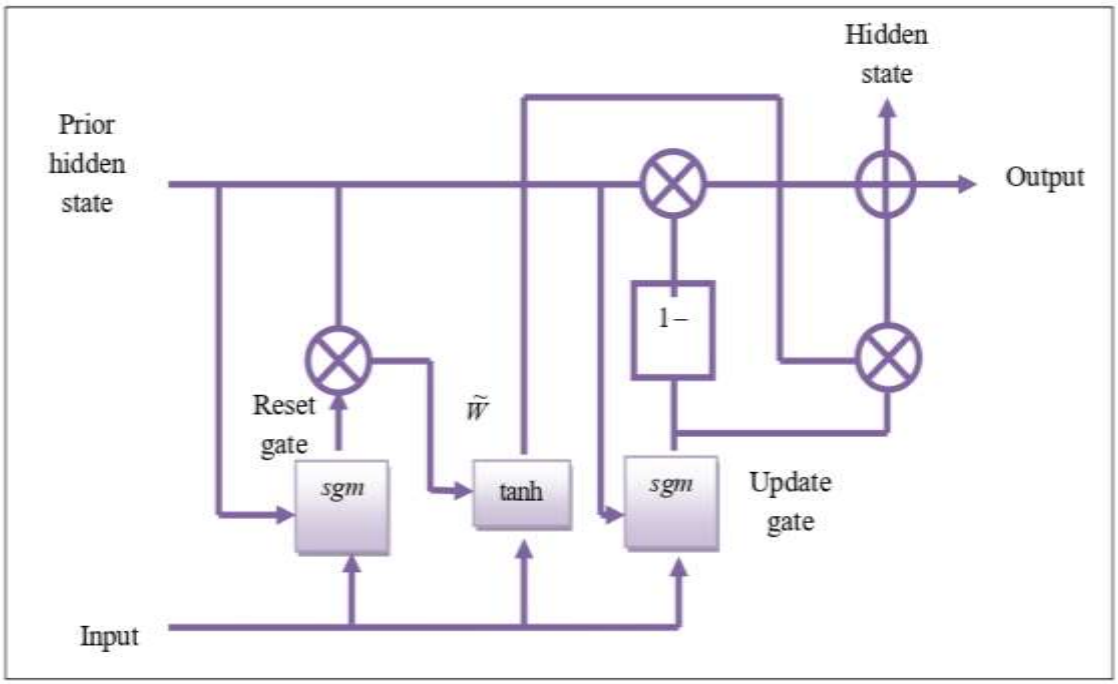


Figure 3.8: Diagrammatic Representation of the Gated Recurrent Unit (GRU) Model**.**

3.8 PROPOSED ADCEN-BASED NETWORK INTRUSION DETECTION FRAMEWORK

The Improved Cheetah Optimizer (ICO) has delivered the optimal feature subset that is now entered into the ADCEN framework proposed. This framework merges Deep Temporal Convolutional Networks (DTCN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU) models creating a unified ensemble structure. Attention mechanisms and dilated convolution layers are being used in the process to improve learning and increase detection accuracy. The attention layer highlights the important features while reducing the irrelevant information thus causing a decline in the computational complexity. On top of that, the use of dilated convolution has provided the model with a wider receptive field, which allows the model to recognize the larger contextual patterns in the network traffic data.

The optimally selected features (OF_{of}^{ICO}) from the ICO are passed to the ADCEN model designed. The ADCEN is designed by ensembling the DTCN, the LSTM, and the GRU networks together. The attention and the dilated convolution layers are appended to the networks for enhanced performance results. The attention layer is appended to the network to reduce the complexity involved in the computation process for unnecessary information passed to the network. The receptive field is therefore enhanced by the addition of the dilation layer.

The score for the individual detections comes from the attention mechanism using the Dilated Convolution-Based DTCN, LSTM, and GRU models. Finally, the fuzzy ranking technique is employed to achieve the final prediction score from the above-detection scores. The fuzzy ranking technique is an effective technique in decision-making processes. Fuzzy quantities are used in fuzzy decision-making to measure the performance level of alternatives during the modeling of real-world situations. A vast majority of the existing ranking techniques are unreasonable and do not allow the ranking of fuzzy quantities. Ranking fuzzy quantities is not possible since they are overlapping among themselves since they are obtained using the probability distribution function. Unlike the real quantities, which are ordered in a straight line, the fuzzy quantities are often partial and cannot be compared among themselves.

The attention-based outputs of the DTCN, LSTM, and GRU modules provide the detection confidence scores. Then, the confidence scores are fused together with the help of a fuzzy ranking method to make the ultimate classification decision. By considering different prediction results as one, the fuzzy ranking method simplifies the decision-making process and results in a single, strong inference.

Every fuzzy value is converted into a real value and then compared among each other based on the development of the ranking function using the set of real and fuzzy values for assigning every fuzzy value to a real value whenever there exists a natural order among them, thus facilitating the ranking of the fuzzy values. After the completion of the fuzzy ranking technique, the result for the final detection would be obtained from the ADCEN model. The pictorial representation of the implemented ADCEN based adversarial evasion attack detection model is presented in the Figure 3.8 below.

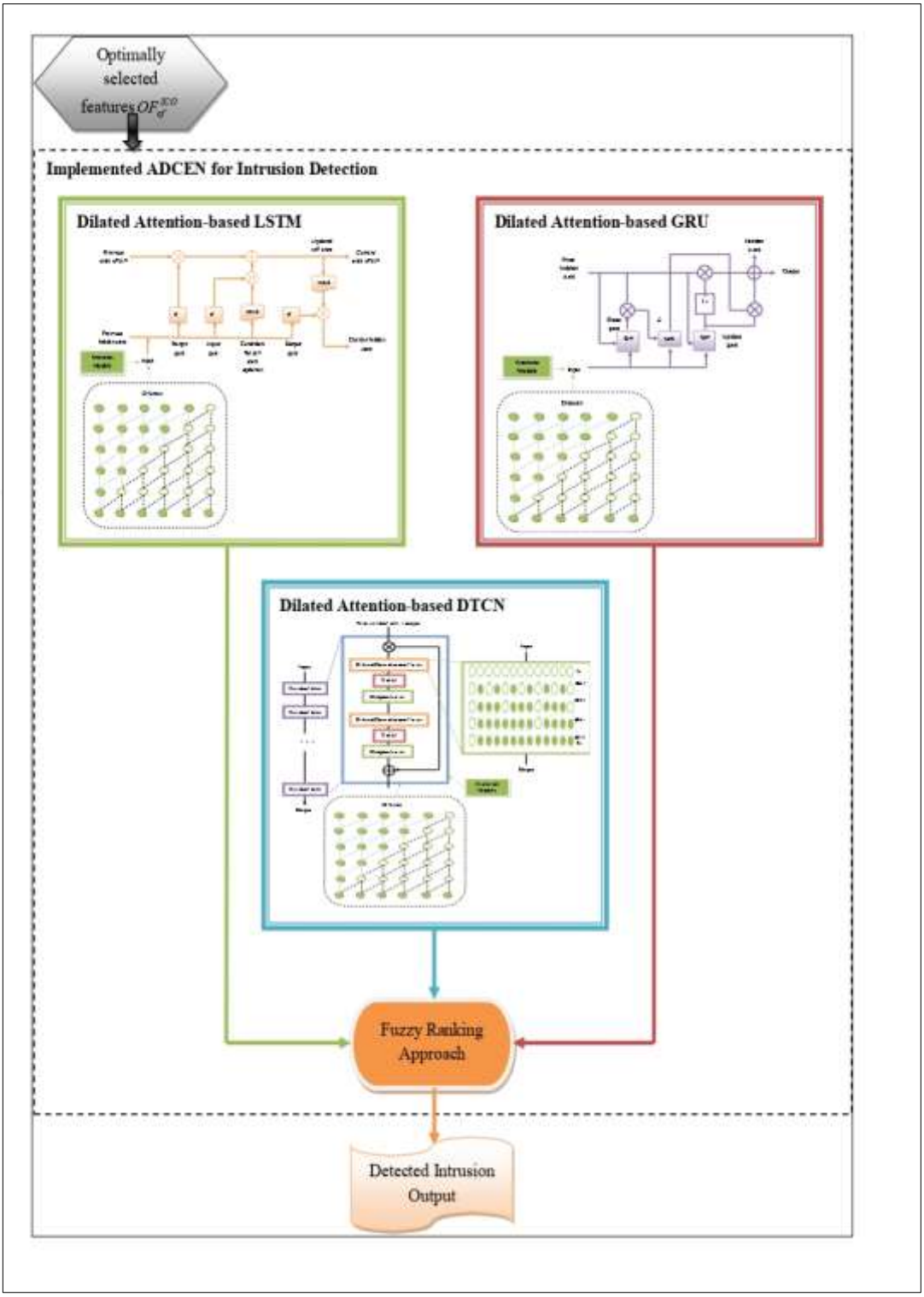


Figure 3.9: Structural Implementations of the Proposed Method.

4. RESULTS AND DISCUSSION

4.1 EXPERIMENTAL SETUP

The designed Adversarial Dilated Convolutional Ensemble Network (ADCEN)-based adversarial evasion attack detection model was written and implemented utilizing the programming paradigm of Python. The reason for the choice of the programming paradigm of Python for the development of the proposed work was the support for machine learning, deep learning, and data analysis, as well as the support for the computation of complex numerical problems within the programming paradigm of Python. Comprehensive simulations and analysis of the experiment have been carried out in order to demonstrate the effectiveness of the proposed approach within the experimental analysis process.

In the implemented ADCEN-based detection model, the structure configuration was designed based on the careful consideration of the settings of the following key parameters: the maximum number of iterations to be carried out was fixed at 50, the size of the chromosome was fixed at 25, and the size of the population was set at 10.

Additionally, to ensure the efficiency and effectiveness of the proposed ADCEN model, it was tested for comparison purposes with a number of traditional and up-to-date algorithms. These algorithms include the Tuna Swarm Optimization algorithm [45], which takes inspiration from the foraging behavior of tuna; the Water Strider Algorithm [46], which takes inspiration from the movement of water striders on the water's surface; the Garter Snake Optimization algorithm [47], which takes inspiration from the hunting behavior of garter snakes; and the Coyote Optimization algorithm [41], which takes inspiration from the cooperative behaviour of coyotes.

In addition, the experiment compared the results with other Deep Learning architectures like MobileNet [48], a lightweight CNN architecture designed for efficient computation; the Long Short-Term Memory (LSTM) networks [43], an architecture used for learning long-term dependencies within data; the Dilated Temporal Convolutional Network (DTCN) [42], which increases the receptive field via dilated convolution layers; and the Recurrent Neural Network (RNN) [49], an architecture for modeling the dependencies within the data in a sequence. Via the above thorough experiments and evaluations, the proposed ADCEN-based adversarial evasion attack detection model showed the superior performance for adversarial

evasion attack detection compared to other algorithms regarding the metrics of accuracy, stability, and convergence speed. The experiment results proved the effectiveness of the combination of attention networks, dilated convolution, and ensemble strategies to improve the capability of the model for modeling complex dependencies in traffic data for adversarial perturbations detection.

4.2 VALIDATION METRICS

In order to analyze and verify the performance and reliability of the implemented adversarial evasion attack detection model, certain statistical metrics have been used. These metrics allow the analysis of the capability of the machine learning model in classification, the rate of error, and the robustness of the results obtained during the experiment. The used metrics in the research are explained below:

a. False Discovery Rate (FDR): The False Discovery Rate measures the proportion of the incorrect positive predictions among all predictions made by the classifier that came up positive. This measure could be useful for determining the probability of the detected attack being a false alarm among the detected ones. The False Discovery Rate formula uses equation (4.1):

$$Eg = \frac{\text{RS}}{PQ+RS} \tag{4.1}$$

In equation (23), the PQ refers to the True Positives (TP), which are the instances correctly identified as attacks, while the RS refers to the False Positives (FP), which are the samples classified as attacks even though they are actually normal.

b. Negative Predictive Value (NPV): measures the proportion of correctly classified samples as ‘normal’ among samples classified as ‘normal’. This measure shows the performance of the model for the negative examples (non-attack examples). The equation for this measure is given in Eqn. (4.2):

$$Rv = \frac{\text{OP}}{OP+QR} \tag{4.2}$$

In this case, OP stands for the True Negatives (TN), denoting correctly identified normal cases, and QR stands for the False Negatives (FN), indicating attacks mistakenly classified as normal cases.

c. Specificity: it measures defines the capability of the machine learning algorithm to identify the correct normal samples from the total actual normal samples. The equation for SPECIFICITY is given as follows:

$$Tj = \frac{\text{OP}}{OP+RS} \tag{4.3}$$

The greater the specificity value, the fewer false alarms the model makes (false positives), thus distinguishing the normal traffic effectively.

d. False Positive Rate (FPR): This measure defines the rate at which regular instances are wrongly identified as assault situations by the learning algorithm. Formula:

$$Wk = \frac{\text{RS}}{RS+OP} \tag{4.4}$$

Where Tp denotes True Positive instances and Fn represents False Negative instances.

e. Sensitivity: Sensitivity (also named Recall, True Positive Rate)

The Sensitivity evaluates the ability of the pattern recognition device to identify real occurrences among the possible attacks correctly. Equation (4.5) defines the Sensitivity measure:

$$Qi = \frac{\text{PQ}}{PQ+QR} \tag{4.5}$$

A higher sensitivity value means the model succeeds in picking up most of the actual attack examples.

f. Precision: The Precision measure defines the percentage of samples correctly identified as attacks among the samples identified as such, from the total samples identified as attacks. Precision is calculated using the formula given by Eq. (4.6):

$$Ol = \frac{PQ}{PQ+RS} \tag{4.6}$$

Precision is crucial for determining the correctness of the predictions made by the model for the attacks.

g. Accuracy: The Accuracy reflects the ratio of the total correctly classified samples (both attack and normal samples) to the total samples in the group, as given by the equation below:

$$\text{Wc}=\frac{OP+PQ}{OP+RS+PQ+QR} \tag{4.7}$$

"This measure provides a single score for the performance on all the classes."

h. Matthews correlation coefficient (MCC): The MCC measures the correlation between the observed and predicted classifications. The formula takes into consideration the four aspects of the confusion matrix (TP, TN, FP, and FN), especially used for imbalanced problems. The calculation formula for MCC shows in equation (30):

$$\text{Ek}=\frac{\text{OP*PQ-RS*QR}}{\sqrt{(PQ+RS)(PQ+QR)(OP+RS)(OP+QR)}} \tag{4.8}$$

Values for MCC lie between -1 and +1, where +1 represents perfect prediction, the value of 0 reflects random prediction, and -1 represents complete disagreement between prediction and observation.

i. F1 score F1 – Score: The F1 - Score is the harmonic mean measure between precision and sensitivity (or recall), where a single measure simultaneously considers the errors of both the false positives and the false negatives. Its calculation uses equation (4.9):

$$\text{To}=\frac{\text{2*PQ}}{2*(PQ+RS+QR)} \tag{4.9}$$

F_1 - Score A high F1-score value represents effective modeling for the prediction of adversarial attacks based on the precision and recall metrics.

j. False Negative Rate (FNR): The False Negative Rate indicates the percentage of actual attacks misclassified into normal traffic. It is represented by the formula in equation (4.10):

$$Xc=\frac{\text{QR}}{PQ+QR} \tag{4.10}$$

A smaller value of FNR indicates a superior attack detection model with fewer missed detections.

these validation metrics jointly assess the performance of the constructed ADCEN-based adversarial evasion attack detection model from diverse aspects such as accuracy, reliability, robustness, and error control to ensure the high precision and good generalization capability of the proposed model in diverse evasion attacks.

4.3 CONVERGENCE BEHAVIOR ANALYSIS OF THE PROPOSED FEATURE SELECTION METHOD

Figures (4.1, 4.2.4.3) shows the convergence validation for the implemented algorithms of the proposed feature selection technique on the given problem, which results in effective visualization of the convergence process within a fixed number of iterations for the algorithms used in this work.

The convergence curve effectively reflects the relationship between the cost function and the number of iterations. The cost function measures the difference between the actual results and the results obtained during the process of Optimization. The cost function value decreases as the quality of the feature subsets increases, thereby increasing the quality of the features' classification process and minimizing redundancy among the selected features.

During the convergence phase, the cost function was computed through changes in the parameter settings for the selection of features in both algorithms. With an increasing iteration number, the algorithms generally tried to reduce the error, thus optimizing the selected features to allow for the most accurate detection with the lowest possible complexity.

The convergence rate of the ICO-based feature selection algorithm was found to be very rapid and stable, thus establishing the algorithm's superiority in determining the key features for the task. The performance of the implemented ICO-based algorithm was compared with four other algorithms based on optimization process:

a. Water Strider Algorithm (WSA)
b. Tuna Swarm Optimization (TSO)
c. Crow Optimization (CO)

d. Garter Snake Optimization (GSO

In the 20th iteration, the ICO-based approach made a great leap forward among other approaches. In particular, the performance improvements for the ICO-based feature selection algorithm are as follows:

a. Improvement of 0.97% compared
b. Improvement of 2.39% compared
c. 5.12% improvement over CO
d. Improvement of 13.56% over

Such percentage differences indicate the effectiveness of the ICO-based approach over the traditional approach by showcasing the faster convergence rate and higher precision achieved during the process of feature selection.

The reason for the superior convergence property of the ICO algorithm lies in its intelligent balance between exploration and exploitation, thus avoiding convergence to local optima and effectively exploring the search space. Through the effective minimization of the cost function, the ICO algorithm-based feature selection approach always picks the optimal combination of features, which are notably beneficial in improving the classification capability of the model against adversarial examples. Therefore, from the results of the convergence validation, it can be concluded that the implemented ICO-based feature selection approach ensures faster convergence, superior performance, and superior error minimization capability compared to the existing techniques based on optimization algorithms for feature selection.

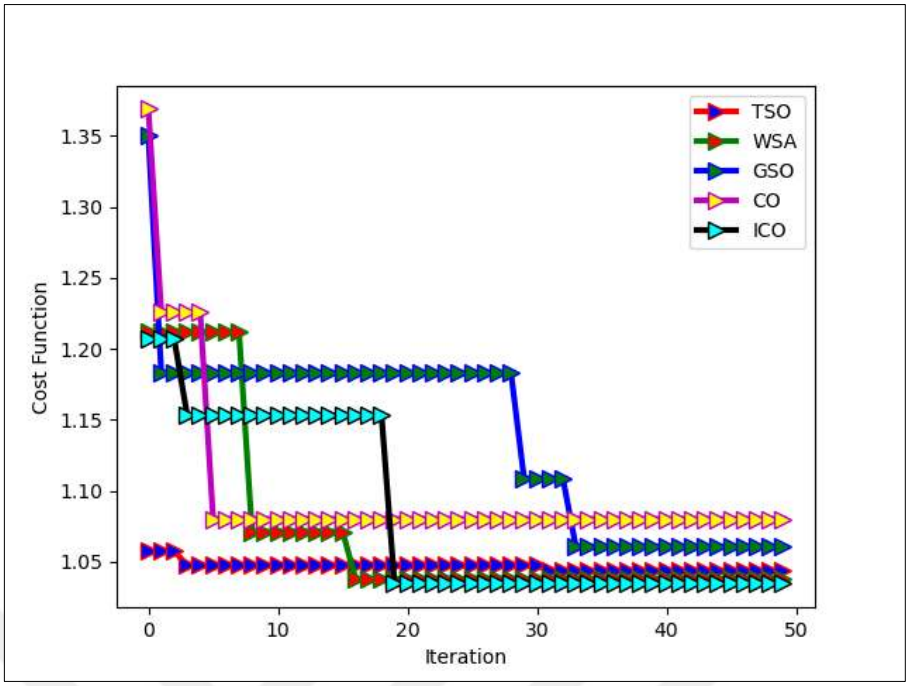


Figure 4.1: Validation of Convergence for The Implemented Feature Selection Technique Across Multiple Algorithms, Evaluated On (A) Dataset 1

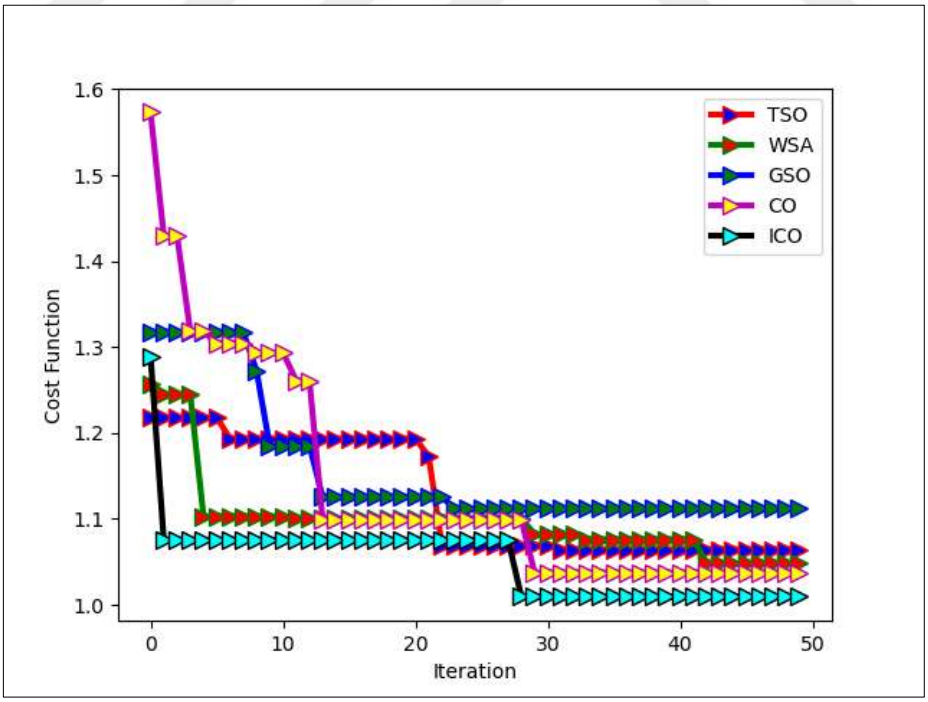


Figure 4.2: Validation of Convergence for The Implemented Feature Selection Technique Across Multiple Algorithms, Evaluated on (b) Dataset 2**.**

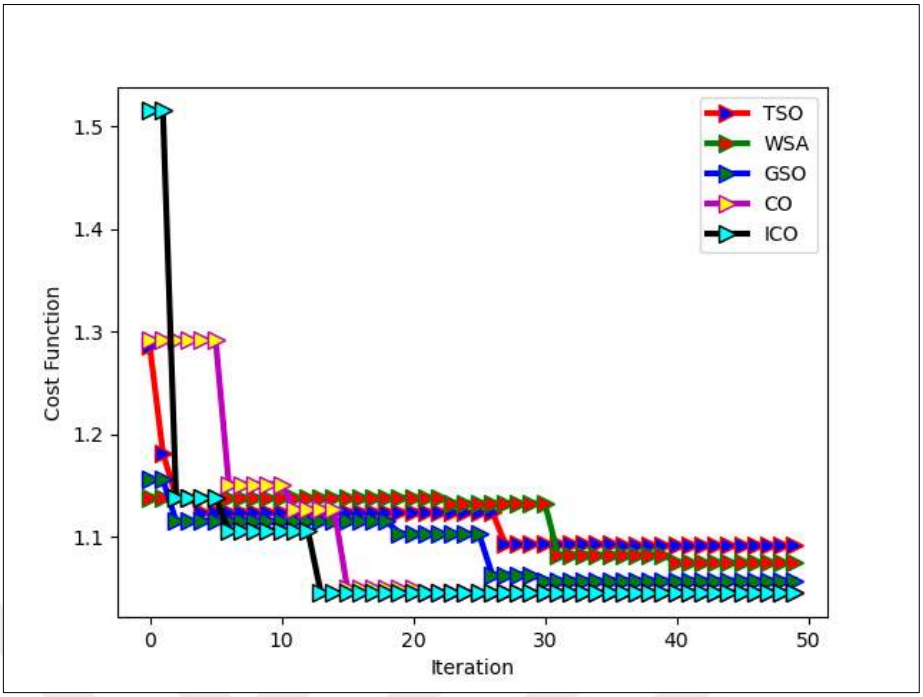


Figure 4.3: Validation of Convergence for the Implemented Feature Selection Technique Across Multiple Algorithms, Evaluated on Dataset 3.

4.4 STATISTICAL EVALUATION OF THE IMPLEMENTED FEATURE SELECTION APPROACH

A quantitative analysis of the proposed feature selection method based on the ICO approach, compared to other existing techniques for optimization, has been carried out statistically to measure the performance of the proposed method quantitatively based on statistics such as mean, standard deviation, and variance for optimized results obtained for varied datasets. The results are summarized in the table below as Table III.

In particular, the assessment centers on Dataset 3, as the comparative analysis in the latter underscores the mean performance enhancement of the ICO approach over four existing algorithms, namely the Tuna Swarm Optimization algorithm, the Water Strider Algorithm, the Garter Snake Optimization algorithm, and the Crow Optimization algorithm.

As indicated by the results presented, the performance improvements realized by the proposed ICO-based feature selection technique over conventional algorithms are as follows:

a. 3.05% higher than the TSO algorithm,
b. 3.14% higher than the WSA algorithm,
c. 0.74% higher than the GSO algorithm, and
d. 1.19% higher than the CO algorithm.

These results indicate the enhanced performance of the ICO-based approach over other existing methods for the proposed task, which signifies the superiority of the approach over other techniques based on the given features' characteristics and the task's requirements. The enhanced mean value shows the ICO algorithm's increased capability to achieve an optimal trade-off between exploration and exploitation during the search process, thus avoiding local optima traps and reaching convergence towards the global optima reliably.

Moreover, the small variation and standard deviation found in the statistical results indicate the consistency of the feature selection process offered by the ICO-based approach over various iterations and executions. Such consistency assumes utmost importance in an adversarial setting where a stable feature extraction process directly influences the robustness of the underlying machine learning technique against noise, perturbations, and adversarial attacks.

In conclusion, based on the results obtained from the statistics highlighted within Table 4.1, it is clear that the implemented ICO based approach for feature selection performs better in terms of mean performance, consistency, and robustness compared to the other algorithms of TSO, WSA, GSO, and CO. Therefore, the proposed ICO based feature selection approach proves to be an efficient technique for the selection of the most informative features required for the evasion attack detection process.

The results presented in Table 4.1 illustrate the statistical performance of the proposed feature selection approach in comparison with several benchmark optimization algorithms across three experimental datasets. Multiple performance indicators, including optimal, average, median, and least optimal values, as well as performance variability, are reported to provide a comprehensive evaluation. The findings demonstrate the robustness and competitiveness of the proposed approach under different experimental conditions.

Table 4.1: Quantitative Performance Analysis of The Proposed Feature Selection Approach Compared With Benchmark Optimization Algorithms.

Experimental Dataset I					
Performance Metric	TSO [30]	WSA [31]	GSO [32]	CO [26]	ICO
Optimal Value (BEST)	1.044	1.038	1.061	1.08	1.035
Median Performance	1.048	1.038	1.183	1.08	1.035
Average Performance	1.047	1.071	1.139	1.098	1.083
Performance Variability	0.003	0.063	0.064	0.055	0.063
Least Optimal Value	1.058	1.212	1.35	1.369	1.207
Experimental Dataset II					
Performance Metric	TSO [30]	WSA [31]	GSO [32]	CO [26]	ICO
Optimal Value (BEST)	1.065	1.048	1.112	1.038	1.01
Median Performance	1.069	1.1	1.112	1.098	1.076
Average Performance	1.125	1.098	1.157	1.135	1.051
Performance Variability	0.067	0.048	0.076	0.13	0.047
Least Optimal Value (Worst)	1.218	1.257	1.318	1.573	1.289
Experimental Dataset III					
Optimal Value (BEST)	1.092	1.075	1.057	1.046	1.045
Median Performance	1.124	1.133	1.103	1.046	1.045
Average Performance	1.114	1.115	1.088	1.093	1.08
Performance Variability	0.031	0.028	0.03	0.082	0.094
Least Optimal Value	1.286	1.138	1.156	1.292	1.515

4.5 DETECTION PERFORMANCE ANALYSIS OF THE PROPOSED ADVERSARIAL EVASION FRAMEWORK

The performance analysis of the proposed framework for ADCEN-based adversarial evasion attacks. The performance analysis of the proposed framework for ADCEN-based adversarial evasion attacks is given by the graph in Figures (4.4:4:18), which indicates the performance and efficiency of the proposed framework for three different datasets.

Careful analysis of the performance graphs reveals that the False Positive Rate (FPR) of the proposed ADCEN-based framework is considerably lower than other models. In other words, based on the analysis of the performance of the proposed framework in the context of the second dataset with 2-fold cross-validation, the results indicate noticeable improvements of 4.44%, 5.62%, 7.43%, and 8.67% in the FPR performance over the RNN, DTCN, LSTM, and MobileNet models, respectively.

Moreover, the proposed framework has also been validated for the reliability, consistency, and generalization capability of the proposed model by implementing the five-fold cross-validation technique. In the five-fold cross-validation technique, the entire data set is divided into five equal subsets, and each subset is used alternately for testing purposes, while the other four subsets are used for training purposes at a time.

The results obtained from the k-fold validation experiment demonstrate the superior performance of the proposed ADCEN-based adversarial evasion attack detection framework on every single dataset and fold used for the experiment. The results obtained from the proposed framework demonstrate the ability of the designed model to achieve high precision in the task of distinguishing benign and adversarial samples, thus the prevention of false alarms associated with a high FPR.

The reduction of the FP rate directly contributes to enhanced robustness and reliability of the system, which is very essential for the real-world intrusion prevention and cyber security applications. The distinct values for the FPR indicate the improved precision of the framework, which results from the successful incorporation of attention techniques, dilated convolution layers, and ensemble-mode Deep Learning Architectures like DTCN, LSTM, and GRU in the ADCEN architecture. In summary, the experimental results confirm the validity of the designed ADCEN-based adversarial evasion attack detection framework in

exceeding the performance of traditional models such as RNN, DTCN, LSTM, and MobileNet and establishing itself as an effective and optimized technique for the task of AEAD. The characteristics of the ensemble architecture, features, and attention-driven process associated with the proposed framework allow for effective results to be achieved in an accurate, stable, and optimized manner.

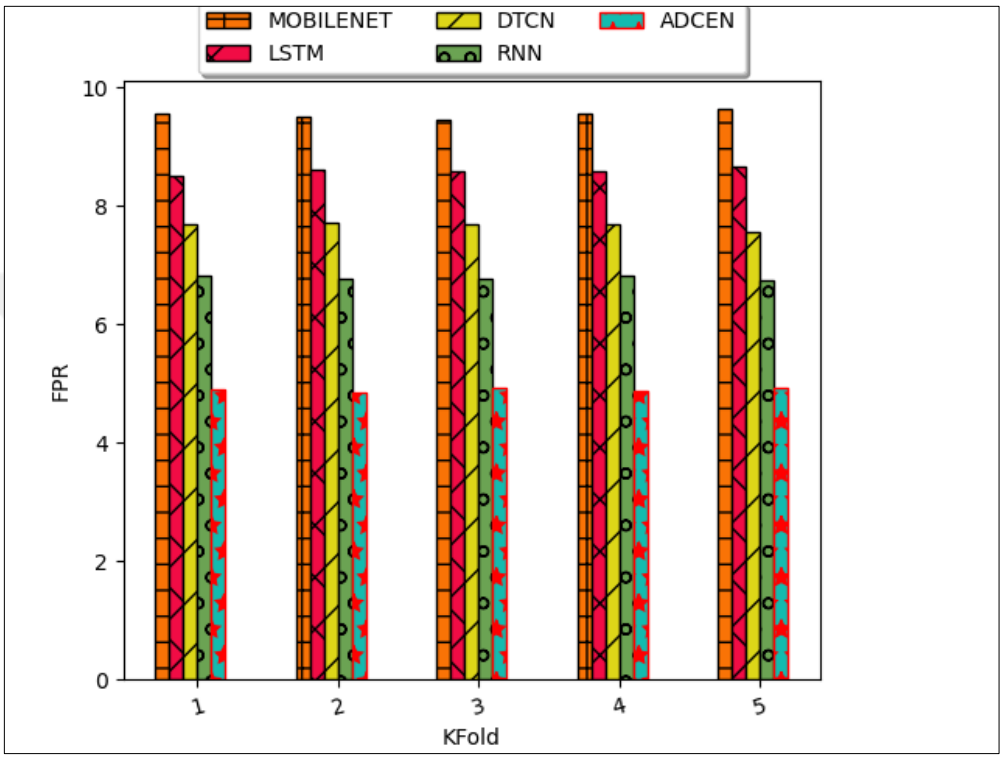


Figure 4.4: Comparative Performance of the Proposed Approach in Comparison with Existing Techniques on Dataset 1 for (a) Specificity.

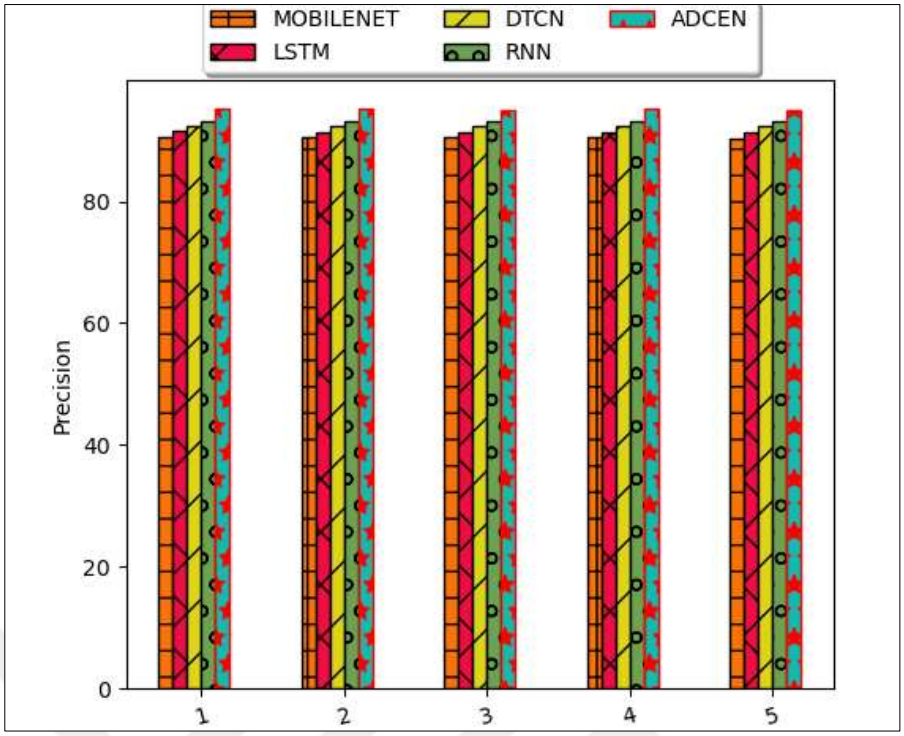


Figure 4.5: Performance Comparison of The Proposed Approach in Comparison with Existing Techniques on Dataset 1 In Terms of Sensitivity.

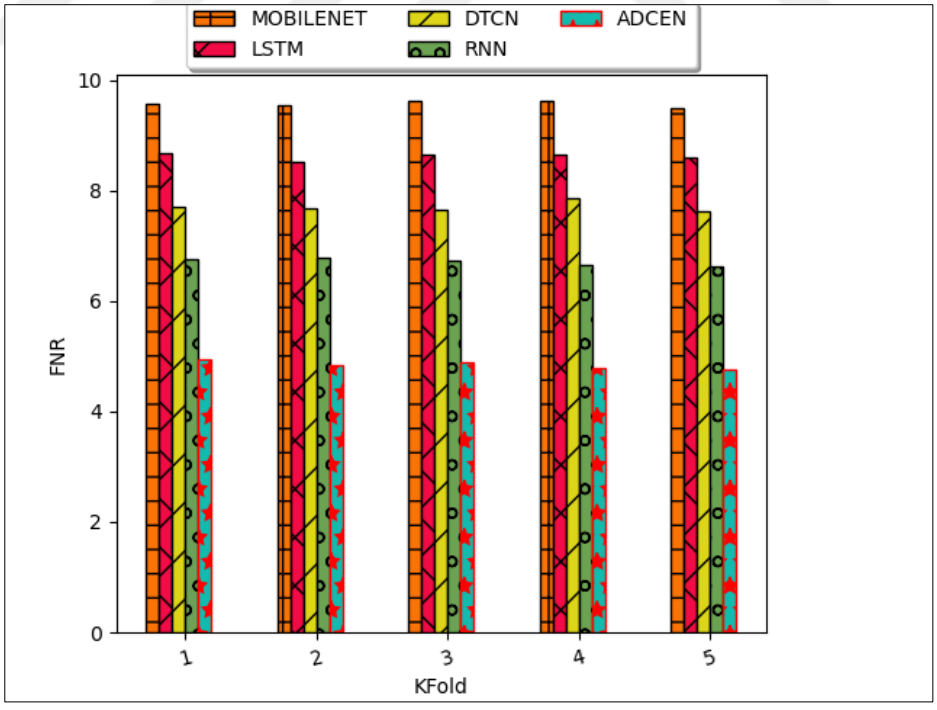


Figure 4.6: Precision-based Evaluation of the Proposed Approach in Comparison with Existing Techniques on Dataset 1.

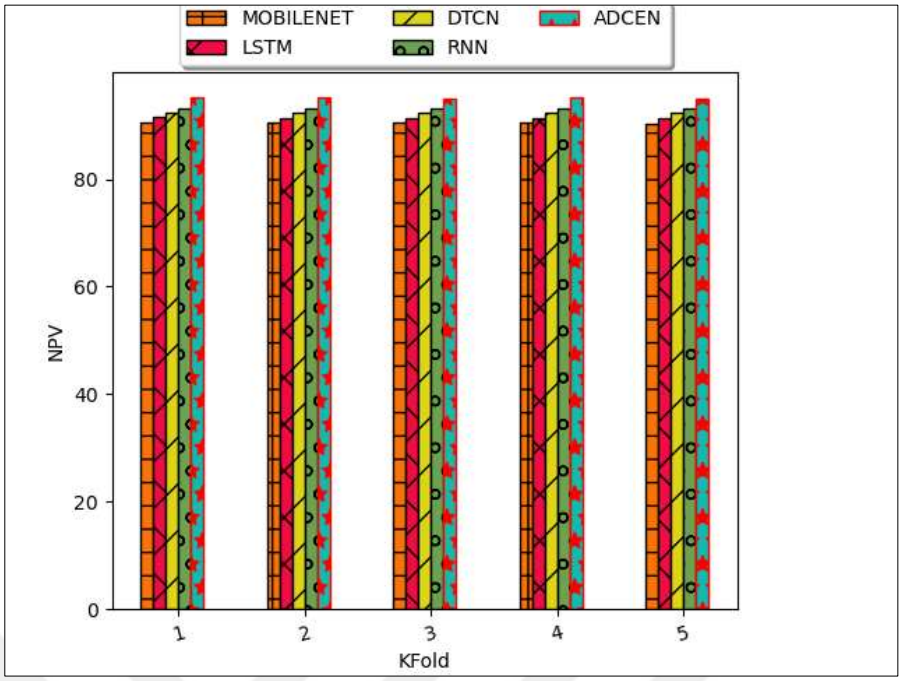


Figure 4.7: False Positive Rate (FPR) Comparison Between The Proposed Detection Framework and Conventional Methods on Dataset 1.

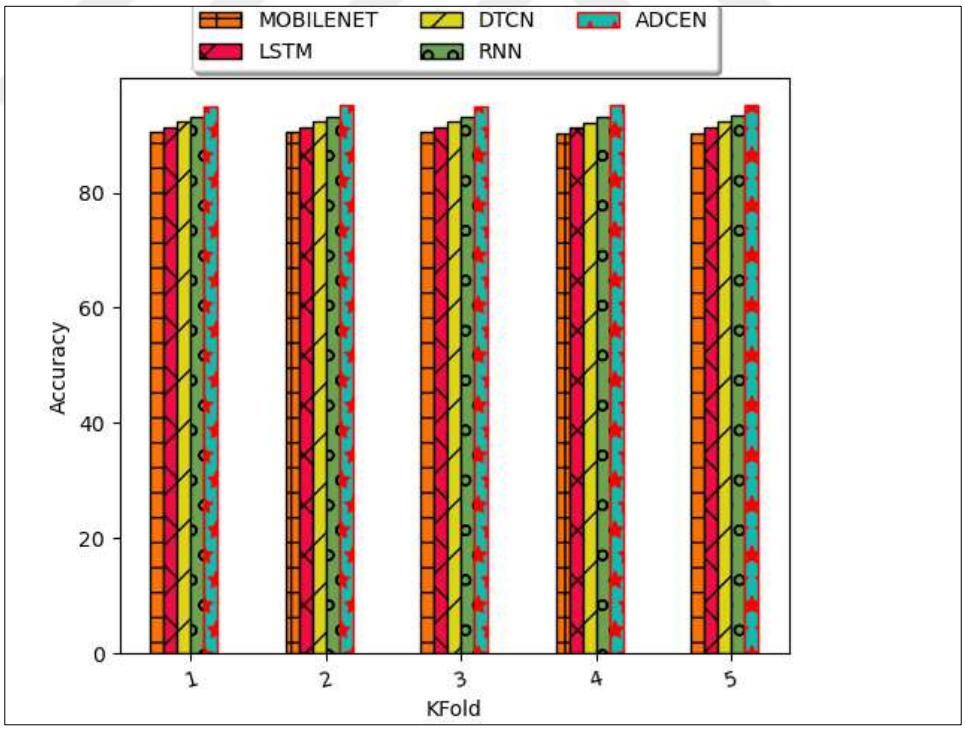


Figure 4.8: Accuracy Analysis of the Proposed Approach in Comparison with Existing Techniques on Dataset 1.

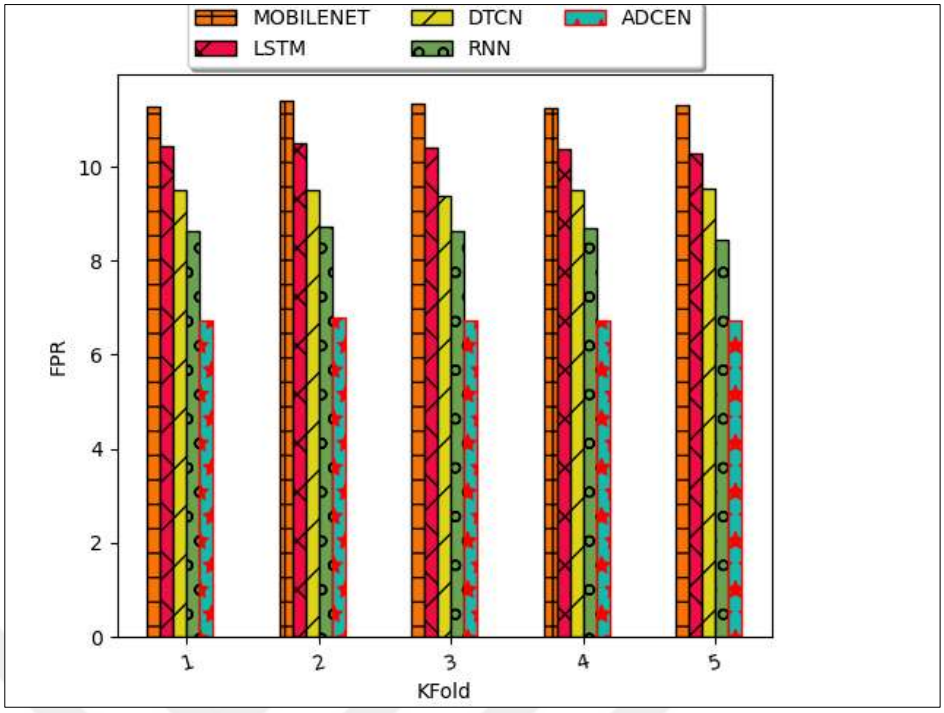


Figure 4.9: Specificity Comparison of the Proposed Model and Baseline Methods on Dataset 2.

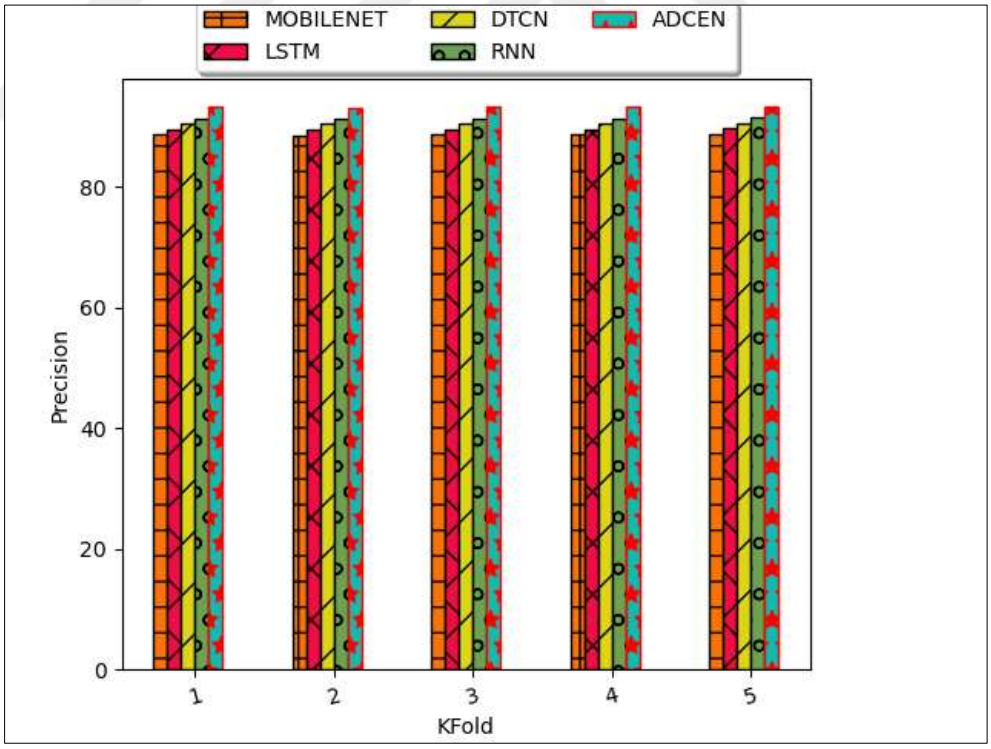


Figure 4.10: Sensitivity Performance Evaluation of the Proposed Detection Model Versus Existing Approaches on Dataset 2.

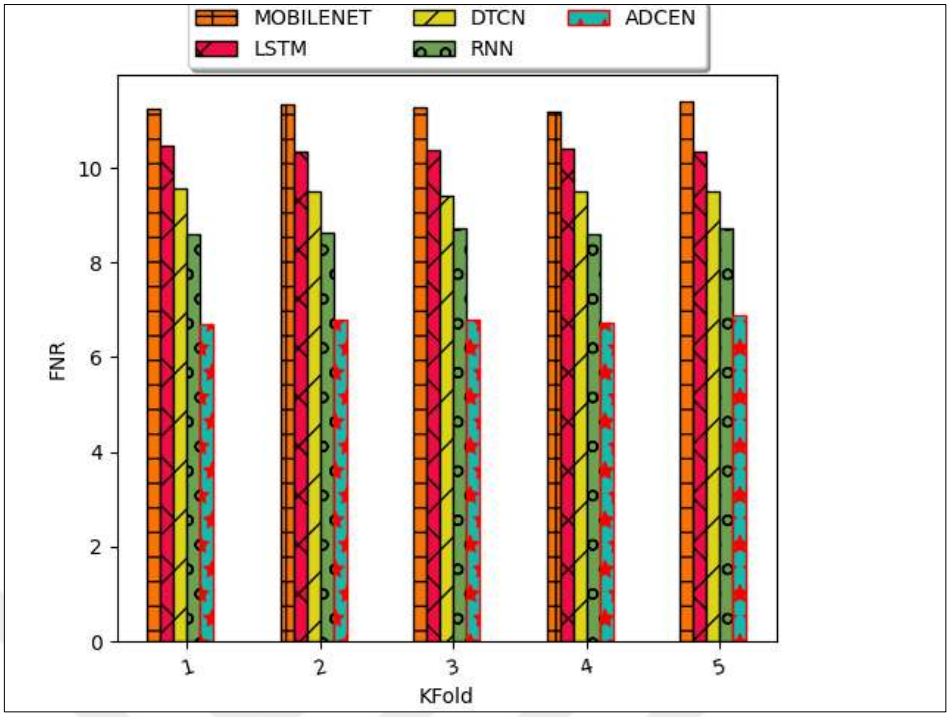


Figure 4.11: Precision Comparison of the Proposed Approach and Existing Methods Using Dataset 2.

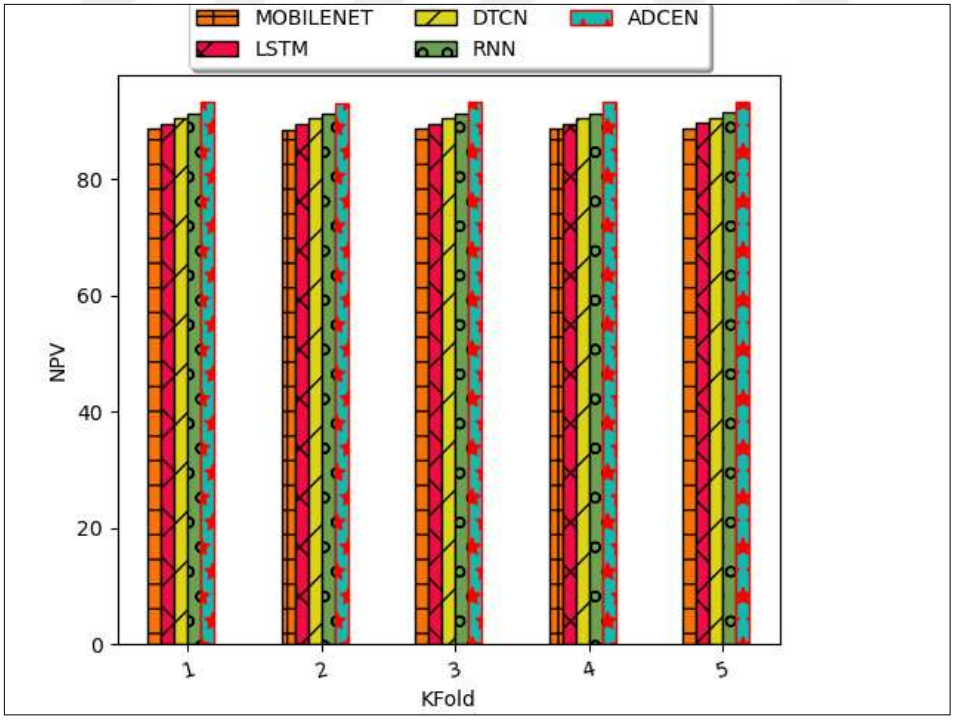


Figure 4.12: False Positive Rate (FPR) Comparison of the Proposed Approach in Comparison with Existing Techniques on Dataset 2..

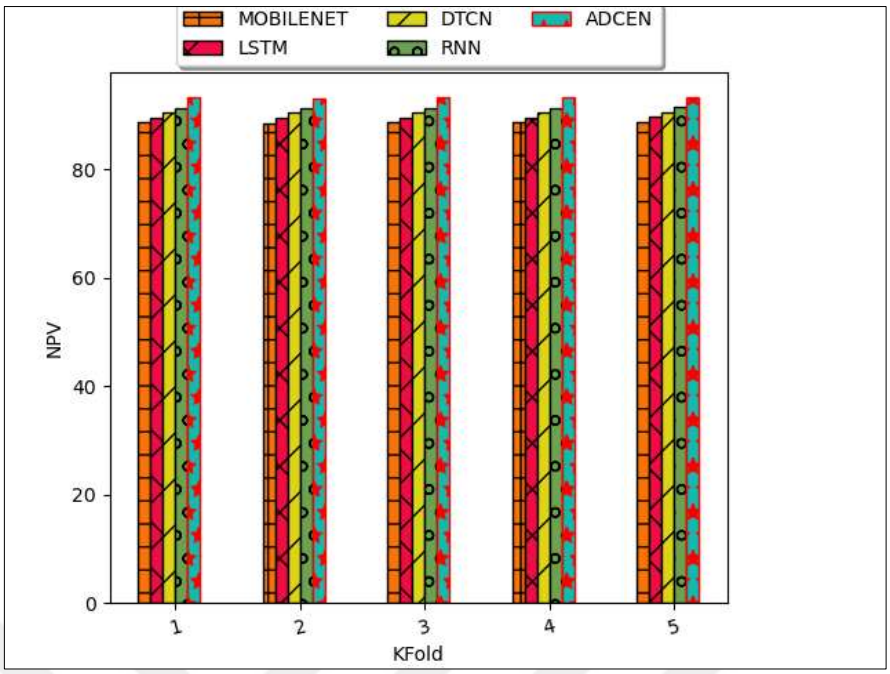


Figure 4.13: Accuracy Evaluation of the Proposed Approach In Comparison with Existing Techniques on Dataset 2.

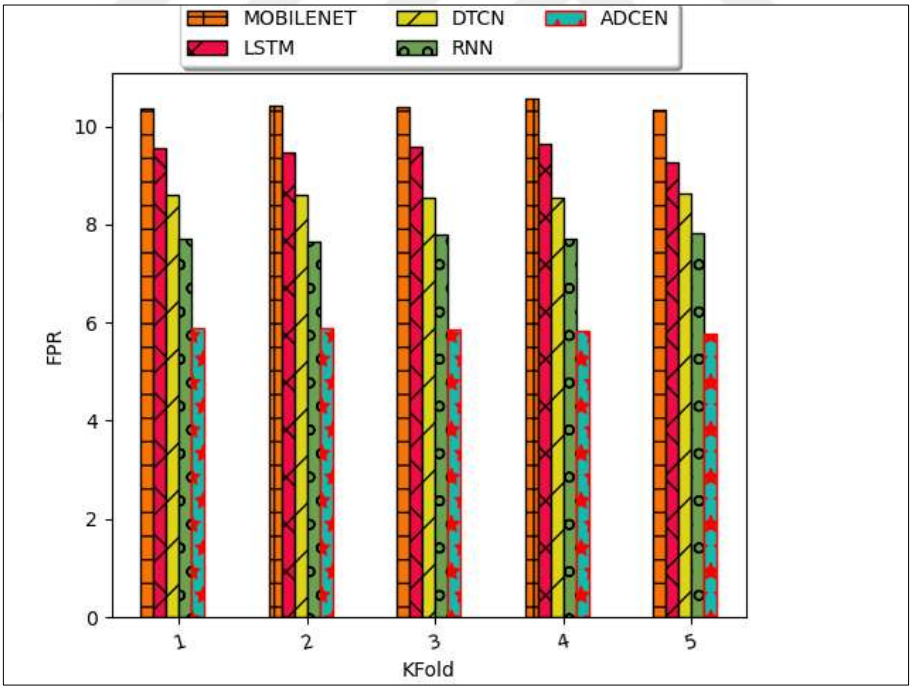


Figure 4.14: Specificity-Based Performance Comparison of the Proposed Methods Detection and Benchmark Methods on Dataset 3.

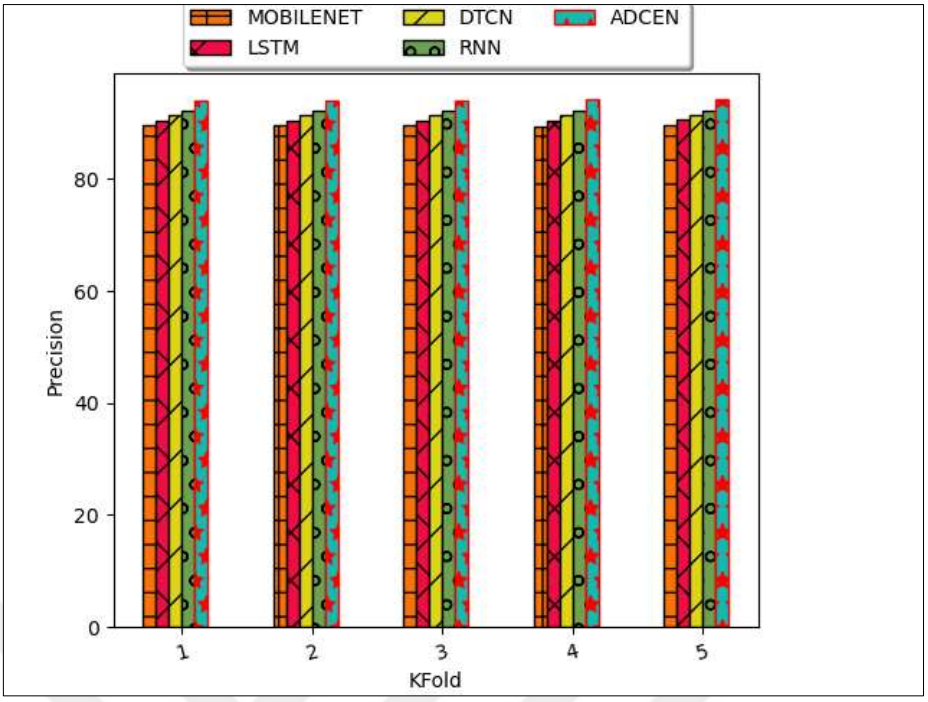


Figure 4.15: Sensitivity Analysis of the Proposed Detection Framework Versus Existing Intrusion Detection Methods on Dataset 3..

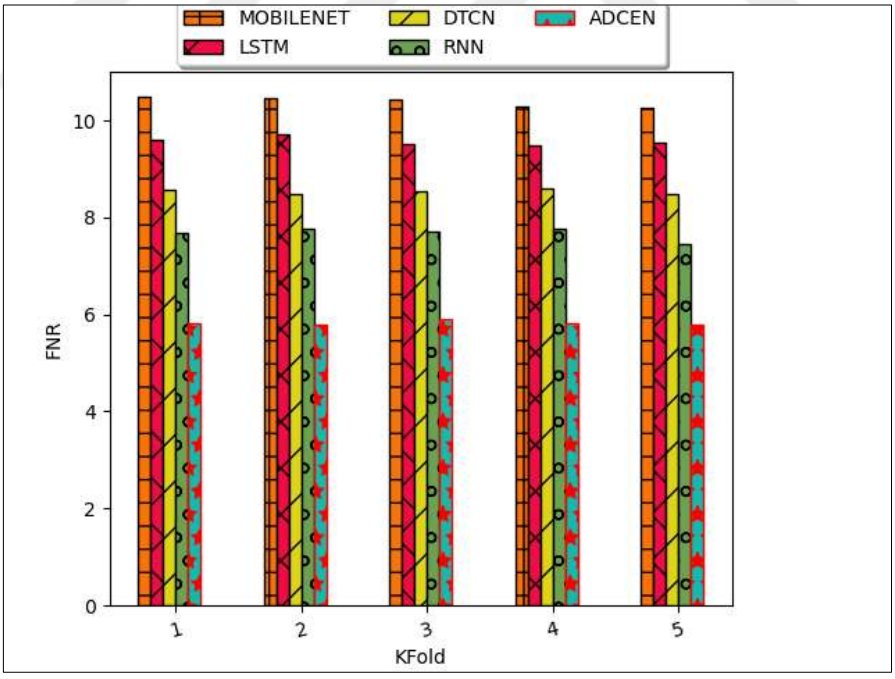


Figure 4.16: Precision Performance Comparison Between The Proposed Adversarial Evasion Detection Approach and Baseline Models on Dataset 3.

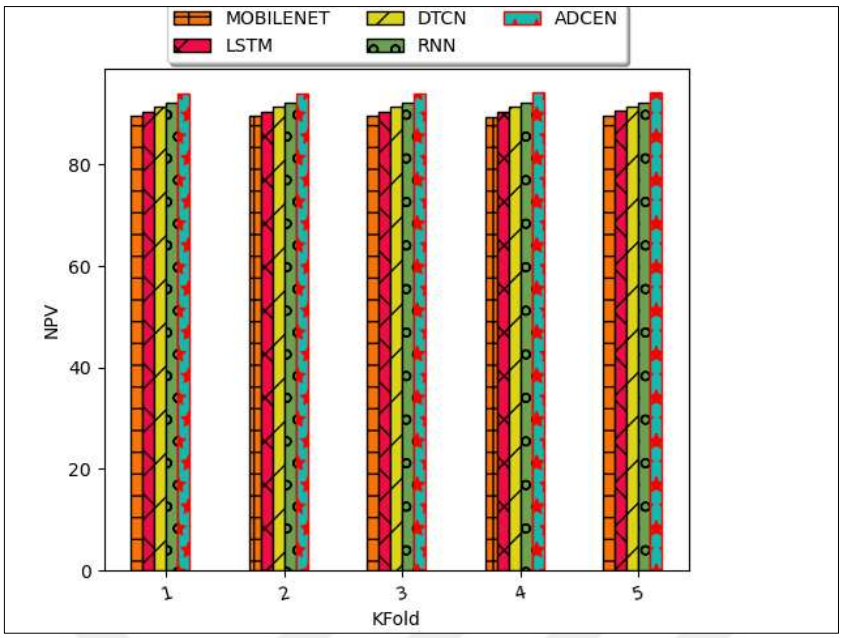


Figure 4.17: False Positive Rate (FPR) Analysis of the Proposed Model Compared with Conventional Techniques on Dataset 3.

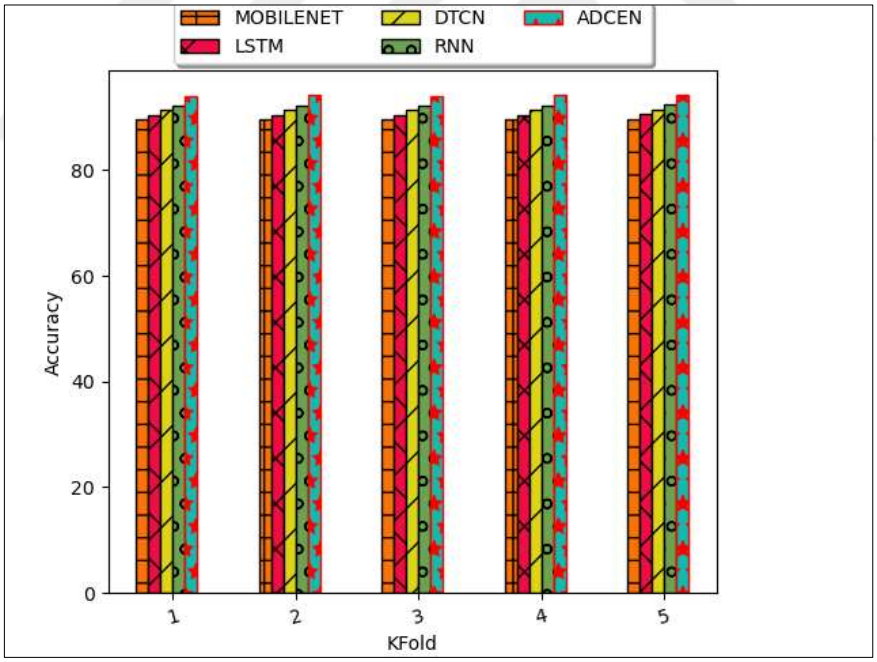


Figure 4.18: Accuracy Comparison of the Proposed Framework And Existing Approaches On Dataset 3.

4.6 OPERATIONAL CHARACTERISTICS AND COMPUTATIONAL ANALYSIS OF THE PROPOSED DETECTION MODEL

Table 4.2 presents a detailed operational analysis of the proposed model. For Dataset 1, the ADCEN model achieves higher classification accuracy compared to several benchmark models: it outperforms MobileNet by 5.21%, LSTM by 4.13%, DTCN by 2.97%, and RNN by 1.97%. These improvements indicate that the ADCEN design—which integrates attention-enhanced dilated convolutions, recurrent components (LSTM and GRU), and an ensemble fusion scheme driven by fuzzy ranking over optimized features from ICO—consistently produces more accurate detection results.

The accuracy gains reported in Table 4.2 translate into tangible benefits for real-time deployment. Higher accuracy reduces both false alarms and missed detections, which in turn lowers the operational overhead for security analysts and increases the reliability of automated mitigation actions. The ADCEN's superior performance can be attributed to three principal factors: (1) optimal feature selection by ICO that supplies the ensemble with highly discriminative inputs, (2) attention mechanisms that direct computational focus to salient temporal patterns while ignoring irrelevant data, and (3) ensemble fusion that combines complementary strengths of DTCN, LSTM, and GRU to capture both long-range temporal dependencies and fine-grained sequential variations.

Taken together, the results in 4.2 demonstrate that the generated ADCEN-based model not only improves nominal accuracy over established architectures but also offers practical advantages for real-time adversarial-evasion detection systems, where both promptness and correctness of decisions are critical.

Table 4.2: Quantitative Evaluation of Deep Learning–Based Intrusion Detection Models Datasets1.

Evaluation Metrics	MobileNet [33]	LSTM [28]	DTCN [29]	RNN [34]	ADCEN
FNR	%9.5	%8.5	%7.6	%6.6	%4.7
Specificity	%90.3	%91.3	%92.4	%93.2	%95.1
Accuracy (%)	%90.4	%91.3	%92.3	%93.3	%95.15
FPR	%9.6	%8.6	%7.5	%6.7	%4.9
Sensitivity (%)	%90.4	%91.4	%92.3	%93.3	%95.2
NPV (%)	%90.3	%91.3	%92.4	%93.2	%95.1
MCC	%0.810	%0.83	%0.85	0.87	%0.91
F1-score (%)	%90.5	%91.4	%92.4	%93.3	%95.2
Precision (%)	%90.4	%91.4	%92.5	%93.3	%95.1
FDR	%9.7	%8.7	%7.6	%6.8	%4.931

Table 4.3: Quantitative Evaluation of Deep Learning–Based Intrusion Detection Models Dataset2.

Evaluation Metrics	MobileNet [33]	LSTM [28]	DTCN [29]	RNN [34]	ADCEN
Sensitivity (%)	%88.	%89.6	%90.5	%91.2	%93.2
FNR	%11.4	%10.4	%9.5	%8.8	%6.9
Precision (%)	%88.7	%89.8	%90.5	%91.6	%93.3
Accuracy (%)	%88.7	%89.7	%90.5	%91.5	%93.2
NPV (%)	%88.7	%89.8	%90.5	%91.6	%93.3
FPR	%11.3	%10.3	%9.6	%8.5	%6.8
F1-score (%)	%88.7	%89.7	%90.5	%91.5	%93.2
MCC	0.78	0.80	0.82	0.83	0.87
Specificity (%)	%88.8	%89.8	%90.4	%91.6	%93.3
FDR	11.4	10.3	9.6	8.5	%6.89

Table 4.4: Quantitative Evaluation of Deep Learning–Based Intrusion Detection Models Dataset3.

Evaluation Metrics	MobileNet [33]	LSTM [28]	DTCN [29]	RNN [34]	ADCEN
FNR	%10.3	%9.6	%8.5	%7.5	%5.8
Specificity (%)	%89.7	%90.8	%91.4	%92.2	%94.2
FDR	%10.4	%9.3	%8.7	%7.8	%5.8
FPR	%10.3	%9.3	%8.7	%7.9	%5.8
NPV (%)	%89.7	%90.8	%91.4	%92.2	%94.2
Accuracy (%)	%89.7	%90.6	%91.5	%92.4	%94.2
Precision (%)	%89.7	90.7	91.4	%92.2	%94.2
MCC	%0.80	%0.81	%0.83	%0.85	%0.88
Sensitivity (%)	%89.8	%90.5	%91.6	%92.6	%94.2
F1-score (%)	%89.7	%90.6	%91.5	%92.4	%94.3

Tables (4.2,4.3 and 4.4) provides a detailed quantitative assessment of the proposed ADCEN model against several deep learning-based intrusion detection methods over three benchmark datasets. The reported numbers offer a clear picture of the advantages and consistency of the proposed framework during the application of different evaluation metrics. For Dataset 1, ADCEN gains the top accuracy (95.147%), sensitivity (95.232%), and F1-score (95.251%), while at the same time, it shows the lowest false negative count (4.768) and false positive count (4.938). The development of performance in this way signifies that the model is more capable of correctly spotting malicious traffic and at the same time, preventing the misclassification of the benign network activities. A trend in the performance, which is alike to that of Dataset 1 is seen also for Dataset 2, where ADCEN has the winning position over all rival models by all main metrics. To be specific, the accuracy of the proposed framework is 93.287% and the sensitivity is 93.305%, together with the 6.895 and 6.730 FND and FPD respectively. The continuous decline in error-related metrics across datasets signifies the strength and versatility of the proposed detection framework. Moreover, the improved Matthews Correlation Coefficient (MCC) values suggest the presence of class imbalance, which is a common challenge in intrusion detection

datasets, but still, the classification performance is well-balanced. In Dataset 3, ADCEN keeps on demonstrating its excellent detection ability by being the best in accuracy (94.205%) and sensitivity (94.196%) and at the same time, having the lowest FNR (5.804) and FPR (5.785) among all the models that were assessed. The constant upward trend in precision and NPV further corroborates the trustworthiness of the proposed model in segmenting malicious traffic from the legitimate network behavior. All these results, when taken together, confirm that the ADCEN model is the superior model.

4.7 CLASS-WISE DETECTION ANALYSIS USING CONFUSION MATRIX EVALUATION

Figure (4.19) shows the confusion matrix analysis of the proposed ADCEN-based adversarial evasion attack detection system. The analysis ensures that the proposed model effectively categorizes the input samples into their respective classes. A 2x2 confusion matrix is used for the analysis purposes such that the two classes are the benign (or normal) and the adversarial (or malicious). A comparison between the actual results and the prediction results of the proposed ADCEN-based adversarial evasion attack detection model is used for the construction of the confusion matrix analysis.

In the assessment at hand, the four key elements for statistical assessment of a classification system's accuracy, precision, and robustness are used, which are True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). Together, these elements testify to the ability of the algorithm to distinguish between valid and malicious inputs.

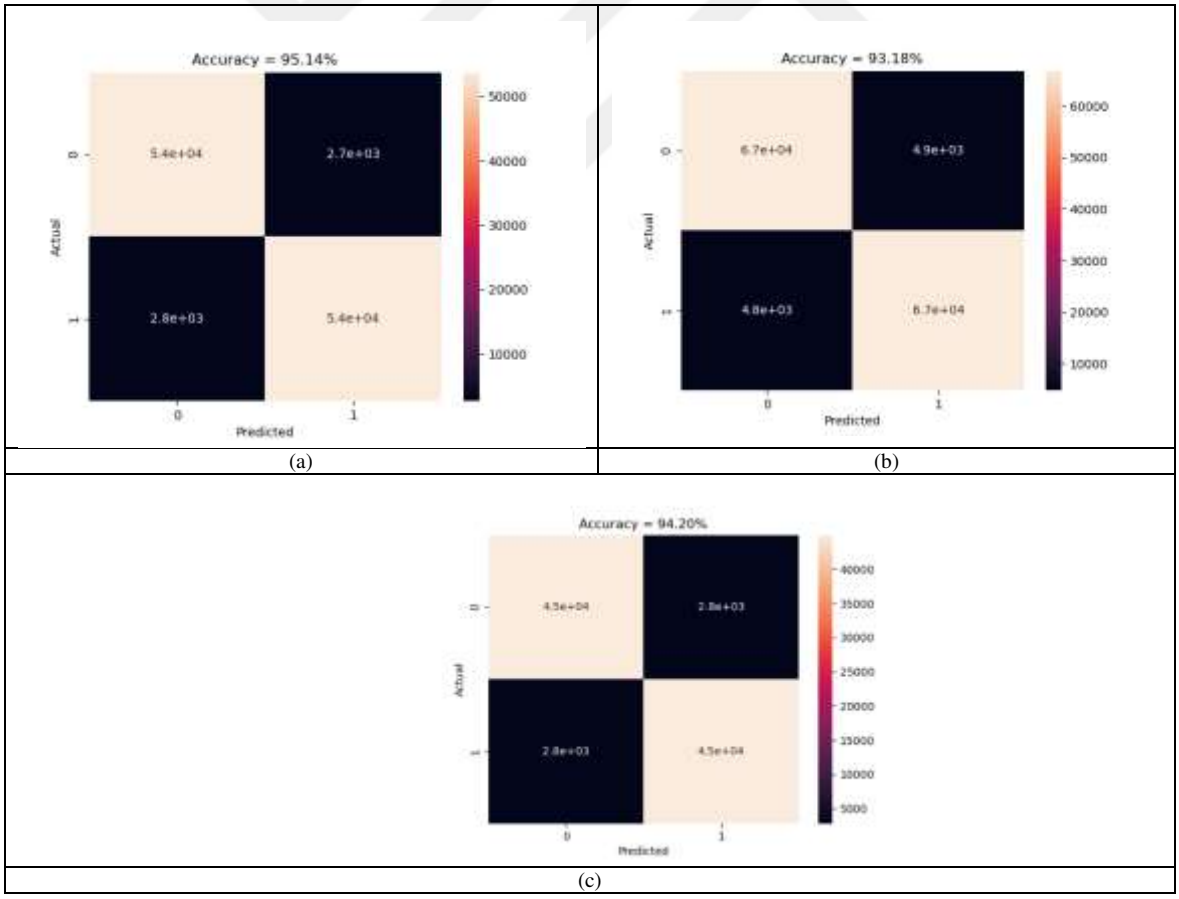
Experimental results indicate the proposed ADCEN-based adversarial evasion attack detection system produced high accuracy levels for all the used datasets:

a. 95.14% for Dataset 1,
b. 93.18% for Dataset 2, and
c. 94.20% for Dataset 3.

Such results support the fact that the precision of the proposed system remains high irrespective of the environment it operates in. The high values for the measures of accuracy indicate the ADCEN model's ability to avoid misclassifications to produce valid results.

In the cases where the data was composed of 50,000, 60,000, and 40,000 samples, the model was able to correctly detect about 5.4×10^4, 6.7×10^4, and 4.5×10^4 samples, respectively. These results indicate the ability of the proposed model to handle large samples of data without affecting

The superior performance obtained from the analysis of the confusion matrix again proves the effectiveness and efficacy of the designed ADCEN-based detection framework. By utilizing the most appropriately selected features (from the ICO-based feature selector), the attention mechanism, and the combination of both DTCN, LSTM, and GRU, the proposed framework effectively succeeds in improving the discrimination power between the adversarial and non-adversarial examples. In conclusion, the results from the confusion matrix analysis are clear and demonstrate the high efficacy of the ADCEN-based adversarial evasion attack detection in providing high levels of detection accuracy, very small error levels, and superior generalization performance on a variety of datasets.



(a) (b)

(c)

Figure 4.19: Confusion Matrix–based Evaluation of the Proposed Framework Across Three Datasets

4.8 ROC-BASED DISCRIMINATIVE CAPABILITY ASSESSMENT OF THE PROPOSED DETECTION MODEL

Figures (4.20: 4.22) shows the Receiver Operating Characteristic (ROC) analysis for the proposed ADCEN-based adversarial evasion attack detection protocol. Receiver Operating Characteristic (ROC), an important analysis technique for performance measurement, plots the relationship between the True Positive Rate (TPR) and the False Positive Rate (FPR). Practically, the graph offers insight into the capability of the designed algorithm in classifying the adversarial examples correctly while keeping the rate of the false-positive cases to a very low level.

In the proposed experiment, the ROC curves for the designed ADCEN-based detection scheme are compared to other benchmark models such as RNN, LSTM, DTCN, and MobileNet by plotting the graph based on three different data sources to ensure a holistic analysis for the assessment of the capability of the detection process.

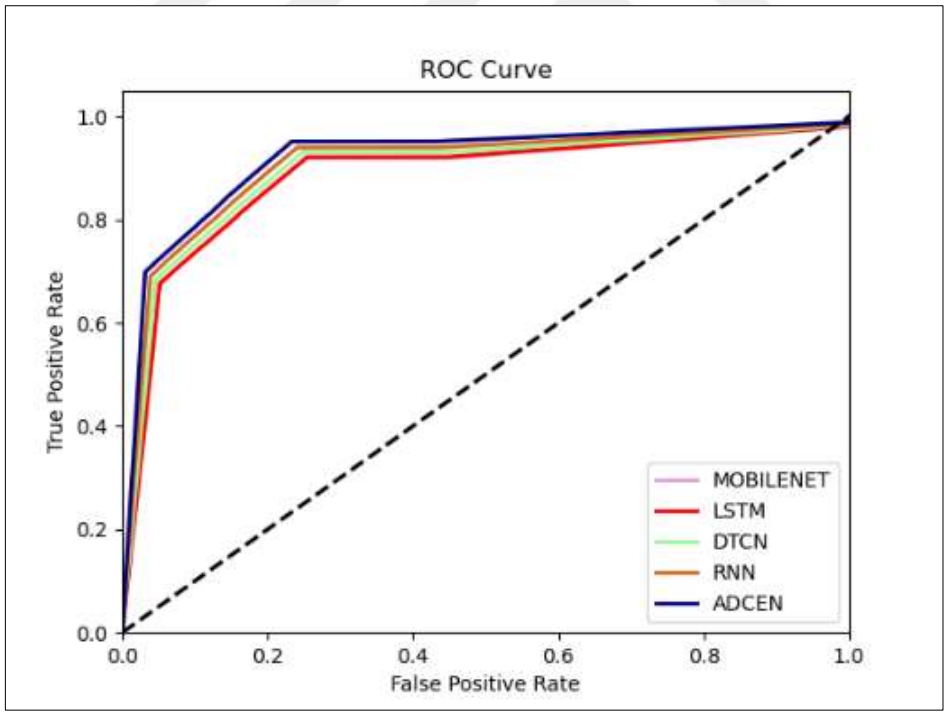


Figure 4.20: Comparison of ROC Curves for the Proposed Feature Selection Method and Other Algorithms Across (a) Dataset 1.

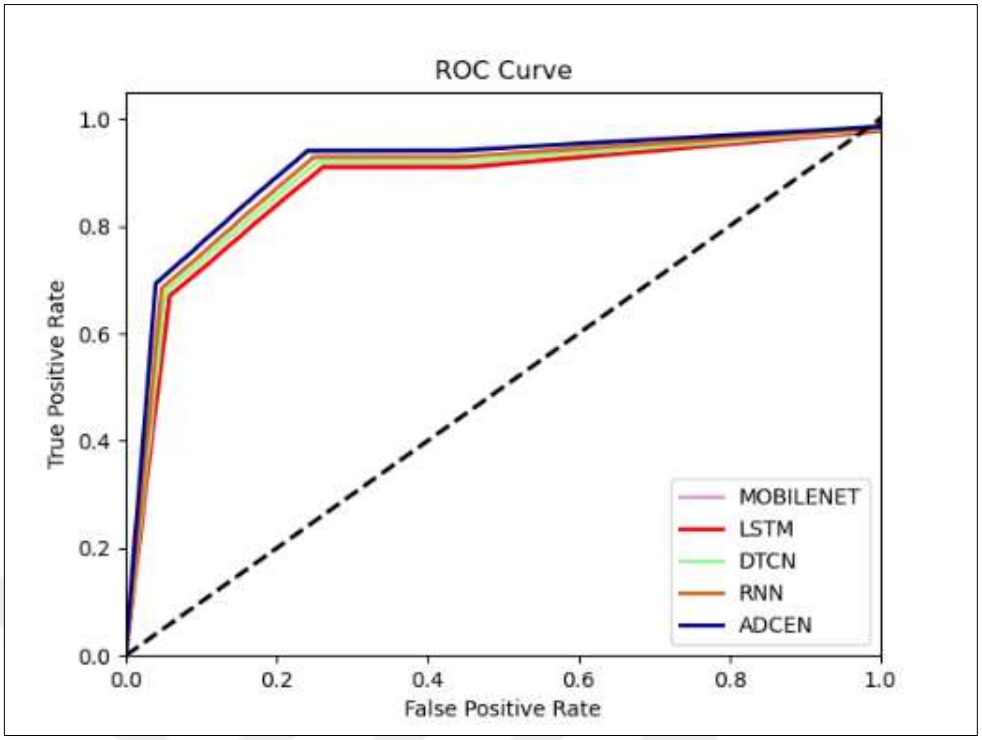


Figure 4.21: Comparison of ROC Curves for the Proposed Feature Selection Method and Other Algorithms Across (b) Dataset 2.

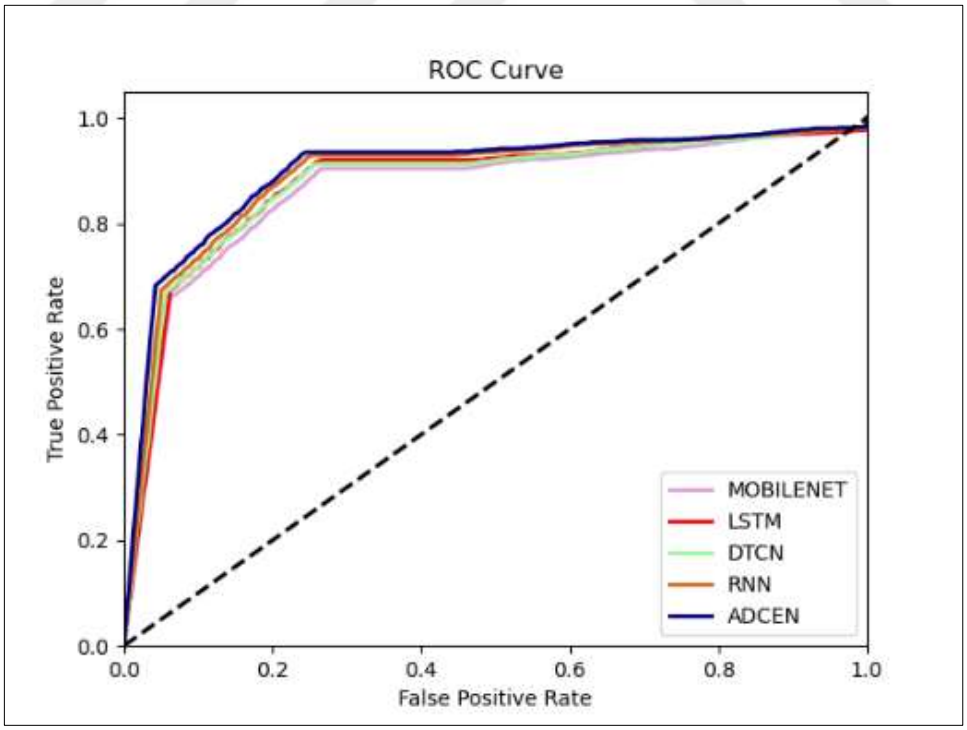


Figure 4.22: Comparison of ROC Curves for the Proposed Feature Selection Method and Other Algorithms Across (c) Dataset 3.

5. CONCLUSION

In the proposed work, an innovative deep learning-based Network Intrusion Detection System (NIDS) was successfully designed and developed to effectively defend against sophisticated evasion attacks, which remain one of the significant threats in current cybersecurity settings. The primary objective of the proposed work was to design a robust, intelligent, and adaptive detection mechanism to effectively identify and defend against sophisticated adversarial attacks.

The proposed approach commences with gathering attack data sourced from multiple standard benchmarks, ensuring diversity in the collected data. After collecting the data, there was a preprocessing and feature selection step, which plays an important part in ensuring efficiency in the attack detection approach.

To achieve the same, the Improved Coyote Optimization (ICO) algorithm was used. The ICO algorithm was able to identify the important features from the data while removing unnecessary features. The detection process became simpler with the reduced dimensionality, achieving lower computational complexity. The efficiency of the detection process was also improved due to the optimized features.

After identifying the optimal features, they were fed into the proposed Adversarial Deep Cascaded Ensemble Network (ADCEN) model, which is the heart of the detection mechanism. The proposed ADCEN model was expressly designed to comprise the combination of three deep learning models, namely the Deep Temporal Convolutional Network (DTCN) model, the LSTM network, and the Gated Recurrent Unit (GRU) model. The combination of the three models makes use of the temporal learning properties of LSTM and GRU models, along with the hierarchical feature extraction property of DTCN, such that the model can effectively exploit both the spatial and sequence dependencies in the network traffic.

To further improve the decision-making process, the fuzzy ranking mechanism was incorporated in the later stage of the proposed approach. The decision-making approach devised on the basis of fuzzy logic helps to further improve the results obtained from the ADCEN model, yielding more accurate and understandable outcomes in terms of classifying the intrusion.

Extensive experimentation and analysis of simulations were conducted with various benchmark datasets to evaluate the efficiency of the proposed approach. The adversarial evasion attack detection system based on ADCEN outperformed the conventional models based on deep learning, including MobileNet, LSTM, DTCN, and RNN. With respect to Dataset 1, the accuracy improvement of the proposed ADCEN approach was found to be 5.21%, 4.13%, 2.97%, and 1.97%, respectively, over the models MobileNet, LSTM, DTCN, and RNN.

The superior performance validates the fact that the proposed ADCEN framework is highly effective in recognizing even subtle adversarial attacks designed to circumvent conventional intrusion detection solutions. The combination of the ICO approach with the feature selection technique and fuzzy decision combination proved highly useful in making the system more robust and flexible.

In conclusion, the proposed ADCEN-based NIDS exhibits highly reliable, scalable, and intelligence-enabled approaches in defending adversarial evasion attacks in the network. The deep learning approach with ensemble mechanisms, along with optimal feature identification and fuzzy decision combination, ensures accurate real-time detection with significantly reduced false positives. The results of this work not only enhance the current status of adversarial defense mechanisms in NIDS, but they also provide prospects in the future combination of explainable AI, FL, and BLS security layers in order to further improve the resilience of CPS, including SCADA, IoT, and industrial automation networks, against emerging and evolving adversarial attacks.

The obtained results indicate that the TPR values achieved by the proposed ADCEN-based approach for the adversarial evasion attack detection are substantially higher than the other models used for comparison. In the context of Dataset 3, for instance, the TPR achieved by the proposed ADCEN approach is greater than the TPR achieved by the other models based on RNN, LSTM, DTCN, and MobileNet by a margin of 5.88%, 8.43%, 8.46%, and 16.88%, respectively.

The reason for the improved performance might be contributed by the nature of the ADCEN model's architecture, which follows the serial cascaded ensemble pattern, combined with the

attention-driven approach for optimizing the process embedded within the architecture of the ADCEN model.

Furthermore, the Area Under the Curve (AUC), an important measure reflecting the performance quality of the detection algorithm, shows a remarkable enhancement in the proposed ADCEN-based system over the existing detection algorithms. The higher the AUC measure, the greater the discriminating power between adversarial and non-adversarial examples. In other words, the proposed ADCEN algorithm ensures a high detection rate regardless of the adversarial environment, thus validating the reliability, robustness, and correctness of the proposed ADCEN algorithm. Conclusion

In conclusion, the ROC analysis proves the validity of the proposed model protocol's superior performance compared to traditional deep learning approaches like RNN, LSTM, DTCN, and other models like MobileNet. In fact, the proposed approach balancing TPR and FPR ensures high accuracy, consistency, and stability for the detection task. Additionally, the broader and upper graph of the ROC for the proposed ADCEN approach indicates the high generative capability and validates the proposed approach's suitability for real-time adversarial attack detection for SCADA and other cyber-physical infrastructure networks.

5.1 FUTURE WORK

Even though the proposed adversarial evasion attack detection mechanism based on ADCEN has shown remarkable improvements in detecting complex adversarial attacks, there also exist several possible directions to further enhance this work in the future.

a. Integration of Explainable Artificial Intelligence (XAI):Future work should be dedicated to incorporating explainable AI methodologies in explaining the decision-making process of the ADCEN model. This will play an important role in enhancing the transparency and trustability of the model, helping network administrators to understand the explanations behind the classification of certain traffic patterns as adversarial or benign.

b. Federated and Distributed Learning Frameworks: To mitigate privacy and scalability issues, the proposed ADCEN model can be extended in a federated learning paradigm, which helps multiple distributed nodes or organizations work together in training

detection models without exchanging raw data. This will improve data security, adaptability, and robustness in various network environments.

c. Deployment in Resource-Constrained, Real-Time Scenarios: Since deep learning models consume high computational resources, future work can be oriented towards crafting a lightweight or edge-optimized variant of the proposed ADCEN model. This would promote fast detection with reduced latency in practical, resource-constrained settings.
d. Improving Robustness: Although the proposed ADCEN system works effectively against current adversarial evasion attacks, in future work, it is important to incorporate adversarial training, distillation, and gradient masks simultaneously to improve robustness against future adaptive attacks.
e. Multi-Modal Intrusion Detection: The current model is almost exclusively concerned with data obtained from network traffic. However, multi-modal data fusion, involving other data sources such as system logs, usage patterns, and hardware sensors, could be researched in further works in order to enhance the accuracy of detection of the system.
f. Adaptive and Self-Learning Mechanisms: To facilitate continuous improvement and adaptability, reinforcement learning or online learning modules can be added to the ADCEN approach to enable learning updates based on changing patterns of attacks and behaviors in networks. Large-Scale Validations in Future work must include comprehensive practical implementations of the proposed approach in industrial SCADA, IoT, and cloud infrastructures. This would serve to ascertain the scalability, resilience, and integrity of such proposed schemes.
g. Hybrid Security Integration: The ADCEN model could be incorporated into a hybrid cybersecurity framework integrating blockchain solutions for safe data exchange and threat intelligence platforms to improve traceability, accountability, and trust in distributed networks for intrusion detection.
h. Cross-Dataset Generalization and Transfer Learning: However, more research in the realm of transfer learning can allow the ADCEN framework to be more effective in learning between varied networks and in adapting to new attack patterns. Energy-Efficient Model Optimization Given the growing need for sustainable computing, future work can be done in the optimization of the ADCEN approach in terms of energy efficiency via pruning, quantization, and hardware acceleration.

REFERENCES

[1] João Vitorino, Isabel Praça, and Eva Maia, "SoK: Realistic adversarial attacks and defenses for intelligent network intrusion detection," Computers & Security, vol. 134, pp. 103433, November 2023.

[2] Wang, Qiuhua, Hui Yang, Guohua Wu, Kim-Kwang Raymond Choo, Zheng Zhang, Gongxun Miao, and Yizhi Ren, "Black-box adversarial attacks on XSS attack detection model," Computers & Security, vol. 113, pp. 102554, 2022.

[3] Alshahrani E., Alghazzawi D., Alotaibi R., and Rabie O., "Adversarial attacks against supervised machine learning based network intrusion detection systems," Plos one, vol. 17, no. 10, pp. e0275971, 2022.

[4] Marc Chalé, Bruce Cox, Jeffery Weir and Nathaniel D. Bastian, "Constrained optimization based adversarial example generation for transfer attacks in network intrusion detection systems," Optimization Letters, 2023.

[5] Guangrui Liu, Weizhe Zhang, Xinjie Li, Kaisheng Fan and Shui Yu, "VulnerGAN: a backdoor attack through vulnerability amplification against machine learning-based network intrusion detection systems," Science China Information Sciences, vol. 65, no. 170303, 2022.

[6] Rui Shu, Tianpei Xia, Laurie Williams and Tim Menzies, "Omni: automated ensemble with unexpected models against adversarial evasion attack," Empirical Software Engineering, vol. 27, no. 26, 2022.

[7] Kishor Datta Gupta, Dipankar Dasgupta and Zahid Akhtar, "Determining Sequence of Image Processing Technique (IPT) to Detect Adversarial Attacks," SN Computer Science, vol. 2, no. 383, 2021.

[8] Jianhua Wang, Xiaolin Chang, Yixiang Wang, Ricardo J. Rodríguez and Jianan Zhang, "LSGAN-AT: enhancing malware detector robustness against adversarial examples," Cybersecurity, vol. 4, no. 38, 2021.

[9] Hyun Kwon, Yongchul Kim, Hyunsoo Yoon and Daeseon Choi, "Classification score approach for detecting adversarial example in deep neural network," Multimedia Tools and Applications, vol. 80, pp. 10339–10360, 2021.

[10] Alper Sarıkaya, Banu Günel Kılıç, and Mehmet Demirci, "RAIDS: Robust Autoencoder-Based Intrusion Detection System Model Against Adversarial Attacks," Computers & Security, pp. 103483, 15 September 2023.

[11] Phan The Duy, Nghi Hoang Khoa, Do Thi Thu Hien, Hien Do Hoang, and Van-Hau Pham, "Investigating on the robustness of flow-based intrusion detection system against adversarial samples using Generative Adversarial Networks," Journal of Information Security and Applications, vol. 74, pp. 103472, May 2023.

[12] Ying-Dar Lin, Jehoshua-Hanky Pratama, Didik Sudyana, Yuan-Cheng Lai, Ren-Hung Hwang, Po-Ching Lin, Hsuan-Yu Lin, Wei-Bin Lee, and Chen-Kuo Chiang, "ELAT: Ensemble Learning with Adversarial Training in defending against evaded intrusions," Journal of Information Security and Applications, vol. 71, 103348, December 2022.

[13] Jonathas A. de Oliveira, Vinícius P. Gonçalves, Rodolfo I. Meneguette, Rafael T. de Sousa Jr, Daniel L. Guidoni, José C.M. Oliveira, and Geraldo P. Rocha Filho, "F-NIDS — A Network Intrusion Detection System based on federated learning," Computer Networks, vol. 236, pp. 110010, November 2023.

[14] Phan The Duy, Le Khac Tien, Nghi Hoang Khoa, Do Thi Thu Hien, Anh Gia-Tuan Nguyen, and Van-Hau Pham, "DIGFuPAS: Deceive IDS with GAN and function-preserving on adversarial samples in SDN-enabled networks," Computers & Security, vol. 109, pp. 102367, October 2021.

[15] Bhukya Madhu, M. Venu Gopala Chari, Ramdas Vankdothu, Arun Kumar Silivery, and Veerender Aerranagula, "Intrusion detection models for IOT networks via deep learning approaches," Measurement: Sensors, vol. 25, pp. 100641, February 2023.

[16] Na-Eun Park, Yu-Rim Lee, Soyoung Joo, So-Yeon Kim, So-Hui Kim, Ju-Young Park, Seo-Yi Kim, and Il-Gu Lee, "Performance evaluation of a fast and efficient intrusion detection framework for advanced persistent threat-based cyberattacks," Computers and Electrical Engineering, vol. 105, pp. 108548, January 2023.

[17] Haoyu Liu, and Paul Patras, "NetSentry: A deep learning approach to detecting incipient large-scale network attacks," Computer Communications, vol. 191, pp. 119-132, 1 July 2022.

[18] M. M. Hasan, R. Islam, Q. Mamun, M. Z. Islam and J. Gao, “Adversarial Attacks on Deep Learning-Based Network Intrusion Detection Systems: A Taxonomy and Review,” Heliyon, preprint, Feb. 2025. doi:10.2139/ssrn.5096420.

[19] M. A. Akbari, M. Zare, R. Azizipanah-Abarghooee, S. Mirjalili and M. Deriche, “The Cheetah Optimizer: a nature-inspired metaheuristic algorithm for large-scale optimization problems,” Scientific Reports, vol. 12, art. 10953, June 2022. doi:10.1038/s41598-022-14338-z.

[20] A. P. AbdelHalim, A. Abo-Alian and N. Badr, “Deep Learning Techniques for Network Intrusion Detection: A Comparative Survey,” International Journal of Intelligent Computing and Information Sciences, vol. 25, no. 2, pp. 74–87, June 2025. doi:10.21608/ijicis.2025.395907.1404.

[21] Z. A. Memon, M. A. Akbari and M. Zare, “An Improved Cheetah Optimizer for Accurate and Reliable Estimation of Unknown Parameters in Photovoltaic Cell and Module Models,” Applied Sciences, vol. 13, no. 18, art. 9997, Sept. 2023. doi:10.3390/app13189997.

[22] E. C. P. Neto et al., “Deep Learning for Intrusion Detection in Emerging Technologies,” Artificial Intelligence Review, 2025 (in press). doi:10.1007/s10462-025-11346-z

[23] Marek Pawlicki, Michał Choraś, and Rafał Kozik, "Defending network intrusion detection systems against adversarial evasion attacks," Future Generation Computer Systems, vol. 110, pp. 148-154, September 2020.

[24] Shiming Li, Jingxuan Wang, Yuhe Wang, Guohui Zhou, and Yan Zhao, "EIFDAA: Evaluation of an IDS with function-discarding adversarial attacks in the IIoT," Heliyon, vol. 9, no. 2, pp. e13520, February 2023.

[25] Eirini Anthi, Lowri Williams, Matilda Rhode, Pete Burnap, and Adam Wedgbury, "Adversarial attacks on machine learning cybersecurity defences in Industrial Control

Systems," Journal of Information Security and Applications, vol. 58, pp. 102717, May 2021.

[26] L. Sidi, A. Nadler and A. Shabtai, "MaskDGA: An Evasion Attack Against DGA Classifiers and Adversarial Defenses," IEEE Access, vol. 8, pp. 161580-161592, 2020.

[27] Chun Yang, Jinghui Xu, Shuangshuang Liang, Yanna Wu, Yu Wen, Boyang Zhang and Dan Meng , "DeepMal: maliciousness-Preserving adversarial instruction learning against static malware detection," vol. 4, no. 16, 2021.

[28] Rizwan Hamid Randhawa, Nauman Aslam, Mohammad Alauthman, Muhammad Khalid, and Husnain Rafiq, "Deep reinforcement learning based Evasion Generative Adversarial Network for botnet detection," Future Generation Computer Systems, 7 September 2023.

[29] Eirini Anthi, Lowri Williams, Amir Javed, and Pete Burnap, "Hardening machine learning denial of service (DoS) defences against adversarial attacks in IoT smart home networks," Computers & Security, vol. 108, pp. 102352, September 2021.

[30] Kaisheng Fan, Weizhe Zhang, Guangrui Liu and Hui He, "FMSA: a meta-learning framework-based fast model stealing attack technique against intelligent network intrusion detection systems," Cybersecurity, vol. 6, no. 35, 2023.

[31] H. Hindy, D. Brosset, E. Bayne, A. Seeam, C. Tachtatzis, and R. Atkinson, “A Taxonomy of Network Threats and the Effect of Current Datasets on Intrusion Detection Systems,” *arXiv preprint arXiv:1806.03517*, June 2018.

[32] O. I. Adelaiye, A. Ajibola, and S. Faki, “Evaluating Advanced Persistent Threats Mitigation Effects: A Review,” *International Journal of Information Security Science*, vol. 7, no. 4, pp. 159-171, Dec. 2018.

[33] L. Diana, “Overview on Intrusion Detection Systems for Computers and Networks,” *Computers*, vol. 14, no. 3, art. 87, Mar. 2025.

[34] L. Mohammadpour, T. C. Ling, C. S. Liew, and A. Aryanfar, “A Survey of CNN-Based Network Intrusion Detection,” *Applied Sciences*, vol. 12, no. 16, art. 8162, Aug. 2022.

[35] B. I. Farhan, “Survey of Intrusion Detection Using Deep Learning in the Internet of Things,” *International Journal of Cyber-Safety & Mobility*, vol. 11, no. 1, pp. 45-63, 2022. ijcsm.researchcommons.org

[36] A. Kaissar et al., “A Survey on Network Intrusion Detection Using Convolutional Neural Networks,” *ITM Web of Conferences*, vol. 43, art. 01003, 2022.

[37] E. C. P. Neto et al., “Deep Learning for Intrusion Detection in Emerging Technologies,” *Artificial Intelligence Review*, 2025 (in press).

[38] Z. Li, “Toward Deep Learning Based Intrusion Detection System,” *Proceedings of the ACM Conference*, 2024. dl.acm.org

[39] D. T. Salim, “A Systematic Literature Review for APT Detection and Mitigation,” *PMC Frontiers*, vol. 13, art. 10336420, 2023.

[40] R. Buchta, “Advanced Persistent Threat Attack Detection Systems,” *ACM Transactions on Cyber-Security*, vol. 10, no. 2, 2024. dl.acm.org

[41] Mohammad Amin Akbari1, Mohsen Zare2, RasoulAzizipanah-abarghooee, SeyedaliMirjalili, & Mohamed Deriche, "The cheetah optimizer: a nature-inspired metaheuristic algorithm for large-scale optimization problems Springer Nature," Scientific Reports, vol.12, pp.10953, 2022.

[42] Bockrath, Steffen, Vincent Lorentz, and Marco Pruckner, "State of health estimation of lithium-ion batteries with a temporal convolutional neural network using partial load profiles," Applied Energy, vol. 329, pp. 120307, 2023.

[43] Hasan, Mustafa Wassef, "Building an IoT temperature and humidity forecasting model based on long short-term memory (LSTM) with improved whale optimization algorithm," Memories-Materials, Devices, Circuits and Systems, pp. 100086, 2023.

[44] Thanh, Phuong Nguyen, Ming-Yuan Cho, Chun-Lung Chang, and Meng-Jie Chen, "Short-Term Three-Phase Load Prediction With Advanced Metering Infrastructure Data in Smart Solar Microgrid Based Convolution Neural Network Bidirectional Gated Recurrent Unit," IEEE Access, vol. 10, pp. 68686-68699, 2022.

[45] Xie, Lei, Tong Han, Huan Zhou, Zhuo-Ran Zhang, Bo Han, and Andi Tang, "Tuna swarm optimization: a novel swarm-based metaheuristic algorithm for global optimization," Computational intelligence and Neuroscience, vol. 2021, pp. 1-22, 2021.

[46] Kaveh, A., and A. Dadras Eslamlou, "Water strider algorithm: A new metaheuristic and applications," Structures, vol. 25, pp. 520-541, 2020.

[47] Naghdiani, Maryam, and Mohsen Jahanshahi, "GSO: A New Solution for Solving Unconstrained Optimization Tasks Using Garter Snake's Behavior," In 2017 international conference on computational science and computational intelligence (csci), pp. 328-333, 2017.

[48] Wang, Wei, Yutao Li, Ting Zou, Xin Wang, Jieyu You, and Yanhong Luo, "A novel image classification approach via dense-MobileNet models," Mobile Information Systems, vol. 2020, 2020.

[49] Medsker, Larry R., and L. C. Jain, "Recurrent neural networks," Design and Applications, vol. 5, no. 2, pp. 64-67, 2001.