People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

Chadli Bendjedid University - EL Tarf

Computer Science Department

THESIS

For the award of the Doctoral Degree (Third Cycle)

Privacy Protection in an E-Learning System

Field: Computer Science

Specialization: Information System and e-learning

Prepared by: Brahim Khalil SEDRAOUI

Defended on: 30/06/2026

Before the jury composed of:

N°	Full Name	Grade	Establishment	Quality
01	Ala Djeddai	MCA	Chadli Bendjedid University -El Tarf	President
02	Abdelmadjid Benmachiche	Prof	Chadli Bendjedid University -El Tarf	Supervisor
03	Amina Makhlouf	MCA	Chadli Bendjedid University -El Tarf	Co-Supervisor
04	Ibtissem Gasmi	MCA	Chadli Bendjedid University -El Tarf	Examiner
05	Sourour Maalem	MCA	ENS -Constantine	Examiner

Academic Year: 2025/2026

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Chadli Bendjedid - EL Tarf

Département d'Informatique

THÈSE

Pour l'obtention du diplôme de docteur 3ème Cycle

Protection de la vie privée dans un système d'e-learning

Filière : Informatique

Spécialité : Système d'information et e-learning

Préparé par : SEDRAOUI Brahim Khalil

Soutenue le: 30/06/2026

Devant le jury composé de :

N°	Nom et prénom	Grade	Etablissement	Qualité
01	Ala Djeddai	MCA	Universitè Chadli Bendjedid -El Tarf	Président
02	Abdelmadjid Benmachiche	Prof	Universitè Chadli Bendjedid -El Tarf	Rapporteur
03	Amina Makhlouf	MCA	Universitè Chadli Bendjedid -El Tarf	Co-rapporteur
04	Ibtissem Gasmi	MCA	Universitè Chadli Bendjedid -El Tarf	Examinateur
05	Sourour Maalem	MCA	Ecole Normale Supérieur -Constantine	Examinateur

Année Universitaire: 2025/2026

Abstract

The rapid growth of e-learning systems has considerably changed the education field introducing more access to knowledge and flexibility to learners all over the world. Nevertheless, such a change has also brought up new challenges in the area of cybersecurity, such as protecting sensitive information; safeguarding academic character; controlling cybersecurity threats. This thesis examines four novel approaches to identifying and countering various cyberattacks on e-learning systems that can improve cybersecurity in the context of e-learning environments. First is the CNN-OOA that uses Convolutional Neural Networks (CNN) in collaboration with the Orca Optimization Algorithm (OOA) to extract phishing attacks to protect the personal information and login ideas of the users. Second, the LSTM-SWAP model smartly applies the Long Short-Term Memory (LSTM) networks to examine the student behavior during online exams and detect the patterns that characterize cheating or fraudulent behavior. Third, the CNN-GWO architecture combines CNN with the Grey Wolf Optimizer (GWO) achieving the successful identification of network threats e.g. SQL injections and malware. Lastly, the CNN-CAT model integrates CNN with the Cross Attention Transformer (CAT) to enhance the detection of device anomalies in IoT environments. By using attention mechanisms, increases the sensitivity of the model to strange behaviors, making sure to keep the data of learners on IoT-enabled systems confidential and safe. These models can solve phishing, cheating, and intrusion, thereby offering a comprehensive security solution, which encourages a safer, more resilient, e-learning experience. These contributions are assessed in terms of large-scale evaluation in the actual use of applicable e-learning datasets, and they prove effective in identifying and addressing various cyber threats. The thesis points to the necessity of preprocessing and feature engineering in e-learning settings and outlines possible future directions in research, including using sophisticated machine learning methods and AI to augment the e-learning systems in regard to improved security and privacy.

Keywords: E-Learning, Cybersecurity, Intrusion detection, Phishing attacks, Machine Learning, Deep learning.

الملخص

لقد أدى النمو السريع لأنظمة التعلم الإلكتروني إلى تغيير مجال التعليم بشكل كبير مما أدى إلى زيادة فرص الوصول إلى المعرفة والمرونة للمتعلمين في جميع أنحاء العالم. ومع ذلك، فقد أثار هذا التغيير أيضًا تحديات جديدة في مجال الأمن السيبراني، مثل حماية المعلومات الحساسة، وحماية الطابع الأكاديمي، والسيطرة على تهديدات الأمن السيبراني. في هذه الأطروحة، سيتم تناول أربع أفكار جديدة لتحديد ومواجهة العديد من الهجمات الإلكترونية على أنظمة التعلم الإلكتروني من شأنها تحسين الأمن السيبراني في سياق بيئات التعلم الإلكتروني. الأول هو CNN-OOA الذي يستخدم الشبكات العصبية التلافيفية (CNN) بالتعاون مع خوارزمية تحسين أوركا (OOA) لاستخراج هجمات التصيد الاحتيالي لحماية المعلومات الشخصية وأفكار تسجيل الدخول للمستخدمين. ثانيًا، يطبق نموذج LSTM-SWAP بذكاء شبكات الذاكرة الطويلة والقصيرة المدى (LSTM) لفحص سلوك الطلاب أثناء الامتحانات عبر الإنترنت واكتشاف الأنماط التي تميز سلوك الغش/الاحتيال. ثالثا، تجمع بنية CNN-GWO بين CNN و(Gray Wolf Optimizer (GWO لتحقيق التحديد الناجح لتهديدات الشبكة على سبيل المثال حقن SQL والبرامج الضارة. وأخيرًا، يدمج نموذج CNN-CAT الشبكات العصبية التلافيفية (CNN) مع محول الانتباه المتقاطع (CAT) لتعزيز اكتشاف الحالات الشاذة في الأجهزة في بيئات إنترنت الأشياء. من خلال استخدام آليات الانتباه، يتم زيادة حساسية النموذج للسلوكيا. يمكن لهذه النماذج حل مشاكل التصيد الاحتيالي والغش والاختراق، وبالتالي تقديم حل أمني شامل، مما يشجع على تجربة التعلم الإلكتروني الأكثر أمانًا ومرونة. ويتم تقييم هذه المساهمات من حيث التقييم واسع النطاق في الاستخدام الفعلي لمجموعات بيانات التعلم الإلكتروني القابلة للتطبيق، وتثبت فعاليتها في تحديد التهديدات السيبرانية المختلفة ومعالجتها. تشير الأطروحة إلى ضرورة المعالجة المسبقة وهندسة الميزات في إعدادات التعلم الإلكتروني وتحدد الاتجاهات المستقبلية المحتملة في البحث، بما في ذلك استخدام أساليب التعلم الآلي المتطورة والذكاء الاصطناعي لتعزيز أنظمة التعلم.

الكلمات المفتاحية: التعلم الإلكتروني، الأمن السيبراني، كشف التسلل، هجمات التصيد الاحتيالي، التعلم الآلي، التعلم العميق.

Résumé

La croissance rapide des systèmes d'apprentissage en ligne a considérablement changé le domaine de l'éducation, offrant un accès accru aux connaissances et une plus grande flexibilité aux apprenants du monde entier. Néanmoins, un tel changement a également soulevé de nouveaux défis dans le domaine de la cybersécurité, tels que la protection des informations sensibles; la sauvegarde du caractère académique; le contrôle des menaces à la cybersécurité. Cette thèse examine quatre approches innovantes pour identifier et contrer diverses cyberattaques contre les systèmes d'apprentissage en ligne, susceptibles d'améliorer la cybersécurité dans ce contexte. Tout d'abord, le CNN-OOA qui utilise les réseaux de neurones convolutifs (CNN) en collaboration avec l'algorithme d'optimisation Orca (OOA) pour extraire les attaques de phishing afin de protéger les informations personnelles et les idées de connexion des utilisateurs. Deuxièmement, le modèle LSTM-SWAP applique intelligemment les réseaux de mémoire à long et à court terme (LSTM) pour examiner le comportement des étudiants lors des examens en ligne et détecter les schémas qui caractérisent la tricherie/le comportement frauduleux. Troisièmement, l'architecture CNN-GWO combine CNN avec le Grey Wolf Optimizer (GWO) pour identifier les menaces réseau telles que les injections SQL et les logiciels malveillants. Ces modèles peuvent résoudre le phishing, la triche et l'intrusion, offrant ainsi une solution de sécurité complète, qui encourage une expérience d'apprentissage en ligne plus sûre et plus résiliente. Enfin, le modèle CNN-CAT intègre CNN avec le transformateur d'attention croisée (CAT) pour améliorer la détection des anomalies des appareils dans les environnements IoT. En utilisant des mécanismes d'attention, augmente la sensibilité du modèle aux comportements étranges, en veillant à garder confidentielles et sûres les données des apprenants sur les systèmes compatibles avec l'IoT. Ces contributions sont évaluées en termes d'évaluation à grande échelle dans l'utilisation réelle des ensembles de données d'apprentissage en ligne applicables, et elles s'avèrent efficaces pour identifier et contrer diverses cybermenaces. La thèse souligne la nécessité du prétraitement et de l'ingénierie des caractéristiques dans les environnements d'apprentissage en ligne et décrit les orientations futures possibles de la recherche, y compris l'utilisation de méthodes sophistiquées d'apprentissage automatique et d'IA pour augmenter les systèmes d'apprentissage en ligne en ce qui concerne l'amélioration de la sécurité et de la confidentialité.

Mots clés : E-Learning, Cybersécurité, Détection d'intrusion, Attaques de phishing, Machine Learning, Deep learning.

To my family.

Acknowledgements

To begin with, I would like to point out a heartfelt thank you to my thesis advisor, **Pr. Abdelmadjid Benmachiche** who provided me with all the necessary support, encouragement, and advice during this study. It is their level of expertise, patience, and insightful feedback that has helped to shape this work and to further my knowledge of the field.

I give her the warmest thanks to my co-supervisor, **Dr. Amina Makhlouf**, who provided me with her help and constructive feedback. In this case knowledge and commitment have played a major role in the development and finalization of this thesis.

I would like to extend my sincere thanks to the members of the jury for accepting to evaluate this work and for the time and effort they have devoted to reviewing it.

My appreciation also goes to my institution and to all faculty members and researchers who have contributed, directly or indirectly, to my academic and research training, and who have provided a stimulating and enriching scientific environment.

Finally, I would like to express my heartfelt gratitude to my family for their unconditional love, patience, and continuous support throughout this long journey. Their encouragement has been a constant source of strength and motivation. I am also thankful to my friends and colleagues for their support and for the many enriching discussions we have shared.

Contents

Abstract I
الملخص II
Résumé III
Acknowledgements V
Contents VI
List of Figures X
List of Tables XI
Abbreviations XII
General Introduction 162
Background 2
Research Problem 16
Our Problem Statement 3
Proposed Solution 4
Structure of the thesis 5
Chapter 01: E-LEARNING 7
1. Introduction of E-learning 7
2. Historical Development and Evolution 8
3. Types of E-Learning 9
4. Platforms and Technologies 13
5. Internet of Things (IoT) in E-Learning 14
5.1 Smart Learning Environments 14
5.2 IoT for Remote and Experiential Learning 14
5.3 IoT-Enabled Personalized and Adaptive Learning 14
5.4 IoT in Mobile and Context-Aware Learning 15
5.5 Challenges of IoT Integration in E-Learning 15
6. Pedagogical Approaches in E-Learning 16
7. Benefits of E-Learning 16
8. Challenges and Limitations 17
9. E-Learning for Different Educational Levels 18
10. Global Trends and Future Directions 18
11. Conclusion 19
Chapter 02: CYBERSECURITY 22
1. Introduction of Cybersecurity 22
2. Common Types of Cyberattacks in E-Learning Environments 23

2.1 Phishing Attack 24
2.2 Malware Attack 24
2.3 Denial-of-Service (DoS) and Distributed DoS (DDoS) Attacks 24
2.4 Man-in-the-Middle (MITM) Attack 25
2.5 SQL Injection Attack 25
2.6 Zero-Day Exploit 25
2.7 Cross-Site Scripting (XSS) 25
2.8 Advanced Persistent Threats (APTs) 26
2.9 Password Attacks 26
2.10 Ransomware 26
3. Security Vulnerabilities in E-Learning Platforms 26
3.1 Weak Authentication and Authorization Mechanisms 27
3.2 Insecure Communication Channels 27
3.3 Third-Party Tool Integration 27
3.4 Software Vulnerabilities and Outdated Systems 28
3.5 Poor Session Management 28
3.6 Lack of Security Configuration Management 28
3.7 Insufficient Logging and Monitoring 28
4. Case Studies and Real-World Incidents in E-Learning Cybersecurity 29
4.1 Case Study: The University of California – Ransomware Attack (2020) 29
4.2 Case Study: University of Utah – Data Breach (2020) 29
4.3 Case Study: Moodle Platform Vulnerabilities 30
4.4 Impact of COVID-19 on E-Learning Cybersecurity 30
5. Detection and Prevention Mechanisms 30
5.1 Intrusion Detection Systems (IDS) 31
5.2 Intrusion Prevention Systems (IPS) 31
5.3 Firewalls and Endpoint Protection 32
5.4 AI-Based Detection Techniques 33
5.4.1 Machine Learning (ML) 33
5.4.2 Deep Learning (DL) 33
5.4.3 Neural Networks 33
5.4.4 Anomaly-Based Detection 34
5.4.5 Natural Language Processing (NLP) 34
5.4.6 Federated Learning (FL) 34

5.5 Security Requirements for IoT-Enabled E-Learning Systems 34
6. Conclusion 35
Chapter 03: LITERATURE REVIEW 36
1. Introduction 36
2. Related works 36
2.1 Intrusion detection 37
2.2 Phishing detection 42
2.3 Cheating detection 55
3. Comparative Study 63
4. Conclusion 71
Chapter 04: Research Methodology 73
1. Introduction 73
2. Contribution 01: LSTM-SWAP: A Hybrid Deep Learning Model for Cheating Detection [177]. 74
2.1 Description 74
2.2 Methodology 74
2.2.1 Long Short-Term Memory (LSTM) 74
2.2.2 Sliding Window Attention Processor (SWAP) 75
2.2.3 LSTM-SWAP Hybrid Model 75
2.3 Dataset 77
2.4 Results and comparison 77
2.5 Discussion 81
3. Contribution 02: CNN-OOA-Based Cyber Threat Detection: Protecting E-Learning from Phishing. ..83
3.1 Description 83
3.2 Methodology 83
3.2.1 Overview of the Proposed CNN-OOA Framework 83
3.2.2 The convolutional neural network (CNN) 84
3.2.3 Orca Optimization Algorithm (OOA) 85
3.2.3.1 Initialize Population (Candidate Solutions): 86
3.2.3.2 Fitness function: 86
3.2.3.3 Computing the Energy: 86
3.2.3.4 Update Positions: 87
3.2.3.5 Exploitation and Exploration : 87
3.2.3.6 Convergence Criteria: 88
3.2.4 Optimize Hyperparameters Using OOA 88
3.2.4.1 Hyperparameter Space : 88

3.2.4.2 Objective Function : ... 88
3.2.4.3 Categorization of Hyperparameters : ... 89
3.2.5 CNN-OOA ... 89
3.2.6 Pseudo code ... 91
3.2.7 Dataset ... 92
3.2.8 Results and comparison ... 93
3.3 Discussion ... 100
4. Contribution 03: A hybrid deep learning method employing both of Convolutional Neural Network and Grey Wolf Optimization for intrusion detection attacks ... 102
4.1 Description ... 102
4.2 Methodology ... 102
4.2.1 Preprocessing phase ... 102
4.2.2 The CNN approach ... 103
4.2.3 The GWO algorithm overview ... 104
4.2.3.1 Key Components of GWO: ... 104
4.2.4 Implementation and evaluation ... 105
4.2.5 Dataset ... 107
4.2.6 Results and comparison ... 107
4.3 Discussion ... 111
5. Contribution 04: Convolutional Neural Network and Cross-Attention Transformer for Robust Intrusion Detection in IoT Systems ... 113
5.1 Description ... 113
5.2 Methodology ... 114
5.2.1 Preprocessing phase ... 114
5.2.2 The CNN model ... 114
4.2.7 Cross-Attention Transformer (CAT) ... 114
4.2.8 Implementation and evaluation ... 115
4.2.9 Dataset ... 116
4.3 Results and comparison ... 117
4.4 Discussion ... 135
6. Conclusion ... 137
Conclusion and perspectives ... 139
Bibliography ... 143

List of Figures

Figure 1 : Different types of E-learning. 10
Figure 2 : Major Cyberattack Types Targeting E-Learning Platforms..... 23
Figure 3 : Intrusion Detection System architecture 31
Figure 4 : Intrusion Prevention System architecture 32
Figure 5 : Architecture of the LSTM-SWAP Model for Cheating Detection. 76
Figure 6 : Accuracy and loss analysis of the evaluated models in cheating detection dataset 78
Figure 7 : Training and Validation Curves of the LSTM-SWAP Model Across 5-Fold Cross-Validation over 10 epochs. 79
Figure 8 : Training and Validation Curves of the LSTM-SWAP Model Across 5-Fold Cross-Validation over 50 epochs. 79
Figure 9 : Training and Validation Curves of the LSTM-SWAP Model Across 5-Fold Cross-Validation Over 100 Epochs. 80
Figure 10 : The architecture of secure E-Learning by detecting and preventing phishing threats in real time. 84
Figure 11 : The flowchart of OOA. 85
Figure 12 : The architecture of CNN-OOA. 89
Figure 13 : The accuracy of the evaluated models in phishing email detection dataset 94
Figure 14 : The accuracy of the evaluated models in phishing websites dataset. 95
Figure 15 : Training history of CNN on phishing emails dataset. 98
Figure 16 : Training history of CNN-OOA over 10 Iterations on phishing emails dataset. 99
Figure 17 : Training history of CNN on phishing websites dataset. 99
Figure 18 : Training history of CNN-OOA over 10 Iterations on phishing websites dataset. 100
Figure 19: The accuracy of the evaluated models in NSL-KDD dataset 109
Figure 20 : Training history of CNN-GWO over 10 Iterations on NSLKDD dataset 111
Figure 21 : Comparison of Model Accuracy for Intrusion Detection on the CTU-IoT-Malware-Capture-1-1 dataset. 131
Figure 22 : Loss and Accuracy Curves of CNN-CAT for fold 1 on CTU-IoT-Malware-Capture-1-1 dataset. 132
Figure 23 : Loss and Accuracy Curves of CNN-CAT for fold 2 on CTU-IoT-Malware-Capture-1-1 dataset. 132
Figure 24 : Loss and Accuracy Curves of CNN-CAT for fold 3 on CTU-IoT-Malware-Capture-1-1 dataset. 133
Figure 25 : Loss and Accuracy Curves of CNN-CAT for fold 4 on CTU-IoT-Malware-Capture-1-1 dataset. 134
Figure 26 : Loss and Accuracy Curves of CNN-CAT for fold 5 on CTU-IoT-Malware-Capture-1-1 dataset. 134

List of Tables

Table 1 An overview of research on intrusion detection 65
Table 2 An overview of research on phishing detection 66
Table 3 An overview of research on cheating detection 69
Table 4 Description of features used for Mouse Dynamics dataset 77
Table 5 Evaluation of LSTM-SWAP Model Performance Across Different Epoch Counts. 78
Table 6 Evaluation of models using performance metrics on the Mouse Dynamics Dataset 81
Table 7 Best accuracy over 10 iterations tested on phishing emails dataset 95
Table 8 Best accuracy over 10 iterations tested on phishing websites dataset 96
Table 9 The best hyperparameters according to the highest accuracy. 97
Table 10 The performance of different models on phishing email and website datasets 100
Table 11 Best accuracy over 10 iterations on NSL-KDD dataset 109
Table 12 Comparison with other methods on the NSL-KDD dataset 110
Table 13 Final validation accuracy and loss for each fold after training 135
Table 14 Comparison with other methods on the CTU-IoT-Malware-Capture-1-1 dataset 135

Abbreviations

IA: Artificial Intelligence

ML: Machine Learning

DL: Deep Learning

IDS: Intrusion Detection Systems

IPS: Intrusion Prevention Systems

DOS: Denial-of-Service

DDoS: Distributed Denial of Service

CNN: Convolutional Neural Networks

LSTM: Long Short-Term Memory

BiLSTM: Bidirectional Long Short-Term Memory

SWAP: Sliding Window Attention Processor

RNN: Recurrent Neural Network

OOA: Orca Optimization Algorithm

GWO: Grey Wolf Optimization

CAT: Cross-Attention Transformer

IoT: Internet of Things

LMS: Learning Management Systems

MOOCs: Massive Open Online Courses

ANN: Artificial Neural Networks

DNN: Deep Neural Network

SVM: Support Vector Machine

NB: Naive Bayes

RF: Random Forest

DBN: Deep Belief Network

GRU: Gated Recurrent Unit

MLP: Multilayer Perceptron

DT: Decision Tree

LR: Linear Regression

GAN: Generative Adversary Networks

PDGAN: Phishing Detection with Generative Adversarial Network

PSO: Particle Swarm Optimization

GB: Gradient Boosting

GENERAL INTRODUCTION

General Introduction

Background

Recent years have seen a transformative explosion of e-learning that has made open, affordable, and flexible learning to all individuals and this should be a priority. In crossing geographical boundaries and time, it assists the learners on using online platforms in e-learning to acquire knowledge using electronic sources and interactive media. Learning Management Systems (LMS) are slowly becoming the norm in the education establishment worldwide, where the Moodle, Blackboard, Google Classroom, Open Classrooms [1], [2], [3], [4], and other applications are being used as learning and assessment tools, as well as a means of collaboration and course management. The growth of MOOCs [5], microlearning, and m-learning has increased the distance of learning with online programs to the extent that it has positioned individuals to learn at their own pace, as well as to follow their own path in learning. The degree of support is strong in both inclusion and scalability, and it can be applied to almost all aspects of education, including early childhood education, professional and corporate education [6].

The digitalization of education has made learning become a function of relying on cloud facilities, collecting user data, and trusting the internet, so that it makes the e-learning ecosystem vulnerable to a faster pace of cyber threats. In view of the growing complexity of e-learning systems, preserving the confidentiality, integrity, and availability of educational data is a challenging task.

Research Problem

As much as e-learning is accompanied by lots of benefits, the high rate of adoption has led to the emergence of profound security risks that undermine the integrity and reliability of online learning. The safety of personal and academic data is one of the most common issues everyone faces. E-learning systems gather and deposit sensitive data including the identity of users, course contents, grades, and passwords. Data breaches and identity theft through these systems will be an attractive option when there are not necessarily strong data encryption and privacy protection solutions. Furthermore, poor authentication workflows are actually prevalent, with systems in some instances still deploying just usernames and passwords without applying any sort of multi-factor authentication, making systems susceptible to brute-force attack and credential stuffing.

Phishing and Social engineering as well, the phishing and social engineering processes are widespread, mostly by using false educational portals and phony emails that aim to steal login codes. These threats help take advantage of the poor cybersecurity knowledge among students and faculty. Moreover, at the application level, vulnerabilities like APIs, less secure software, and old components open the systems to injection attacks, XSS, among others that may tamper with system integrity. Lack of real time IDS and monitoring implies that most attacks cannot be detected hence malicious users can operate in systems over a long period of time [7].

The other essential problem is the academic integrity of online evaluations. The online learning space is fraught with problems avoiding the chances of cheating, impersonation, and unprecedented usage of resources by the examinees. Also, due to the variety of access devices and networks including personal laptops to public Wi-Fi, there are variations in the security posture of the user environments. The security situation is further confounded by the presence of insider threat of the authorized users, lack of resources and skills that educational organizations can use to ensconce the comprehensive cyber security frameworks. The two sets of challenges that are overlapping highlight the necessity of smart, flexible, and situational aware security solutions, which are also specific to e-learning platforms.

Our Problem Statement

The rapid evolution of cyber threats poses a significant challenge to the security of e-learning systems, which must safeguard users' personal information, learning activities, and digital resources. Conventional security measures—such as firewalls, rule-based access controls, and signature-based detection—are increasingly inadequate against intelligent attacks that can adapt and evade detection.

E-learning platforms face unique challenges due to the hybrid and complex nature of user activity. Learners interact through both synchronous (real-time) and asynchronous channels, access the system from multiple devices, and often operate concurrently in large numbers. This complexity makes it difficult to distinguish between legitimate and malicious activity, whether the threat is phishing, cheating during online exams, network intrusions, or anomalies in IoT-enabled devices.

Additionally, there is a limited availability of cybersecurity datasets specific to e-learning, which constrains the development, training, and evaluation of effective detection models. The diverse types of threats and the dynamic nature of user behavior demand adaptive, intelligent, and context-aware security solutions that go beyond traditional approaches.

Proposed Solution

To address the diverse and evolving security challenges in modern e-learning environments, this dissertation proposes multiple hybrid deep learning approaches, each targeting a specific category of cyber threats. Rather than relying on a single detection mechanism, the research introduces four distinct hybrid architectures, with each model independently designed to handle a particular security problem within e-learning systems.

First, the CNN–OOA model combines CNN with OOA to enhance phishing attack detection. CNN automatically extracts hierarchical and discriminative features from structured and semi-structured data, while OOA optimizes critical model parameters to improve detection accuracy. This model focuses on safeguarding users' personal information and login credentials against sophisticated phishing attempts.

Second, the LSTM–SWAP model is developed to detect cheating and fraudulent behaviors during online examinations. Leveraging the temporal modeling capability of LSTM networks, this model analyzes sequential student behavior logs and interaction patterns. The SWAP mechanism enhances sensitivity to abnormal behavioral trends, enabling accurate identification of cheating activities while minimizing false alarms.

Third, the CNN–GWO architecture integrates CNN with GWO to identify network-level threats, including SQL injection attacks and malware. CNN extracts deep feature representations from network traffic and system logs, while GWO optimally tunes hyperparameters to strengthen classification performance and robustness against complex intrusion patterns.

Finally, the CNN–CAT model incorporates CNN with CAT to detect device-level anomalies in IoT-enabled e-learning environments. The attention mechanism enables the model to focus on subtle and context-aware deviations in device behavior, significantly improving sensitivity to anomalous activities and ensuring the confidentiality and integrity of learner data.

Collectively, these four hybrid models constitute the core contributions of this dissertation, each addressing a distinct category of cyber threats—including phishing attacks, cheating behavior during online examinations, SQL injection and malware attacks, XSS threats, and anomalies in IoT-enabled devices. The proposed models are developed and evaluated using preprocessed and carefully curated e-learning datasets, with particular emphasis on domain-

specific preprocessing and feature engineering to mitigate data imbalance and heterogeneous usage patterns.

By achieving high detection accuracy, low false-positive rates, and adaptability to emerging attack vectors, the proposed framework provides a secure, reliable, and continuous digital learning experience, demonstrating strong potential for real-world deployment in modern e-learning platforms.

Structure of the thesis

This thesis begins with a general introduction that outlines the scope, objectives, and significance of the research, situating e-learning within the context of emerging technologies and cybersecurity challenges. It comprises four main chapters. Chapter 1 introducing e-learning by outlining its concept, evolution, types, platforms, and pedagogical foundations, while emphasizing emerging technologies such as IoT, artificial intelligence, and cloud computing, along with the associated privacy and security challenges in modern e-learning systems. Chapter 2 examines cybersecurity in e-learning, covering major threats, platform vulnerabilities, real-world incidents, and preventive measures such as IDS, firewalls, and AI-based detection, emphasizing a layered approach to ensure secure, private, and resilient digital learning environments. Chapter 3 provides a literature review on ML and DL techniques for intrusion detection, phishing detection, and cheating detection, including a comparative study that highlights performance, strengths, and persistent challenges such as computational cost and dataset limitations. Chapter 4 details the proposed methods, introducing four novel models: CNN-OOA for phishing detection, LSTM-SWAP for cheating detection, CNN-GWO for network threat identification, and CNN-CAT for IoT anomaly detection. These approaches collectively provide a comprehensive cybersecurity solution, validated through large-scale evaluations on real e-learning datasets, demonstrating effectiveness in detecting and mitigating diverse cyber threats. The thesis concludes with a conclusion and perspectives section, summarizing the key findings, discussing practical implications, and outlining future research directions to advance secure, intelligent, and efficient e-learning systems.

CHAPTER 01
E-LEARNING

Chapter 01: E-LEARNING

1. Introduction of E-learning

E-learning, or electronic learning, is the use of digital tools to help and improve teaching and learning. It includes a wide range of activities, including fully online courses, blended learning environments, virtual classrooms, and mobile learning apps [8]. E-learning has changed the way people learn around the world in the last few decades, making it easier and more flexible for students everywhere [9]. E-learning has made education more accessible to a wider range of people, including working adults, students in rural or underprivileged areas, and people with disabilities, by giving them an option other than traditional face-to-face instruction [10].

The rising use of digital devices and the internet has made it easier for e-learning to be implemented in many different types of schools [11]. The COVID-19 epidemic and other events pushed up this transformation furthermore, compelling schools to employ online teaching methods on a scale never seen before. Because of this, e-learning is now a key element of how schools, colleges, and professional development programs teach [12]. The e-learning trend is also a component of greater changes in society, such as globalization, new technology, and the necessity to continuously learn in a knowledge economy that is always changing [13].

Moreover, e-learning has supported a paradigm shift in the role of both educators and learners. Instructors are no longer only producers of content but facilitators of learning, curators of digital resources, and designers of interactive experiences [14]. Likewise, learners are expected to take greater control of their education, displaying autonomy, time management, and digital literacy [15]. This shift has led to the development of learner-centered pedagogies that leverage multimedia content, real-time collaboration tools, and adaptive learning technologies to enhance engagement and retention [16].

E-learning systems allow the capture and analysis of large amounts of data on learners that can be used to customize training and to identify areas of improvement [17]. These findings encourage a shift in our approach to education towards one that is more data-driven and dynamic [18]. In addition, e-learning also facilitates inclusion through multiple means of delivery of content (such as videos, audio lectures, transcripts, and interactive simulations), offering different mediums of presentation for varied learning styles and needs [19].

On a policy level, governments and education groups are also gaining awareness for the strategic value of digital education. Global investments into broadband infrastructure, open educational resources (OER) [20], and teacher training programs are expanding [21]. Scalability of e-learning makes it also an ideal tool for bridging educational divides and ensuring lifelong learning opportunities for all [22].

2. Historical Development and Evolution

Computer-assisted instruction (CAI) systems were first used in the 1960s, which is when e-learning began. These early systems were aimed at enhancing traditional teaching by giving programmed instruction using mainframe computers [23]. Although basic by today's standards, CAI laid the framework for future educational innovations by demonstrating that computers could play a meaningful role in learning. They delivered linear, text-based lessons that allowed learners to advance at their own pace, presenting the idea of self-directed study with automatic feedback—a principle that remains basic in modern digital education [24].

The University of Illinois's work on the PLATO (Programmed Logic for Automated Teaching Operations) system in the 1970s was a big step forward for e-learning. PLATO was a game changer for its time because it had features like discussion forums, real-time messaging, graphical games, and assessment tools [25]. These are all things that are now standard in modern LMS. Its interactive features encouraged group learning long before the internet was widely available, and its impact on the design of later e-learning platforms is hard to overstate [26].

The internet and web browsers changed the possibilities of e-learning in the 1990s. Web-based delivery made it possible to share multimedia files like pictures, sounds, and videos over networks all over the world [27]. The first web-based learning platforms came out during this time. They offered interactive experiences that went far beyond what CAI and PLATO could do [28]. Hyperlinks made it possible to learn in ways that weren't straight lines, and early multimedia tools helped make content more interesting. These things set the stage for constructivist approaches in online education [29].

In the 2000s, LMS such as Blackboard, Moodle, and subsequently Canvas gained popularity. These tools enabled the management of the courses, content delivery, communication, and performance tracking without the need to switch between them [30]. This was the reason why adoption within universities, colleges, and enterprises was a turning point in that online learning became not only an option but a normal delivery method [31].

LMS systems provided faculty with an opportunity to arrange syllabuses, tasks, tests, and grading with the help of digital tools and facilitated academic functioning and hybrid and distance learning counseling [32].

The e-learning world was shaken in the 2010s when MOOCs became a reality [33]. Platforms like Coursera, edX, and Udacity gave people access to courses offered by the most prestigious universities and institutions at minimum prices or even free of charge and gave information to millions of people all over the world [34]. MOOCs popularized the notion of lifelong learning and furthered the notion of open education but also questioned up to what degree these kinds of online courses could be completed, the degree to which learners were engaged, and how valuable mass online courses are as a pedagogical tool despite their popularity within the education system [35], [36].

Nowadays, rapid developments in e-learning have arisen, as new technologies–such as mobile learning (m-learning), cloud computing, or AI–are becoming increasingly popular, alongside their combination with immersive media [37]. With the advent of m-learning, time and location have seemingly been transcended on smartphones and tablet devices [38]. People who are engaged in learning should be able to access content and their learning record wherever and whenever it is relevant to supporting their learning [39]. Cloud computing provides scalable, secure storage for educational content and learner information. By transitioning to cloud-based platforms, remote collaboration and real-time updating are enabled [40]. Now, AI-driven tools are helping personalize learning paths, automate grading, and deliver intelligent insights and recommendations [41].

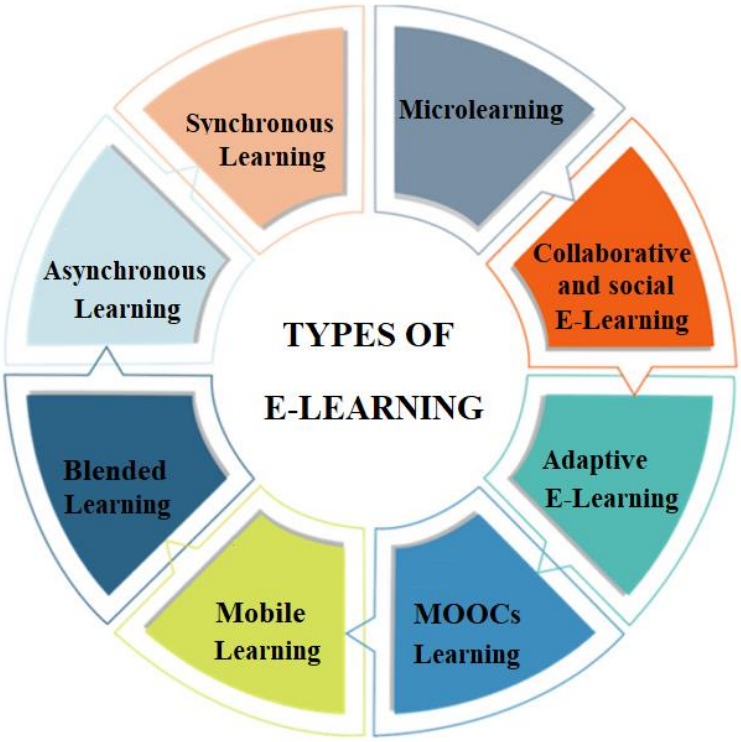
This change in technology has been associated with a larger transformation in the educational paradigm, which involves moving from teacher-based learning to learner-based learning and from a standardized approach to content delivery to an adaptive data-guided approach. The history of e-learning is not merely one of the evolution of technology; it is also one of the evolution of teaching, of the society, and of education that became more accessible to us all [42].

3. Types of E-Learning

There are different types of e-learning based on how it is delivered, how much interaction there is, and the technology that is used. These types are designed to meet the needs of different types of students, different types of schools, and different types of teaching goals, as shown in Figure 1.

To create effective educational interventions that meet the needs of learners and fit within the limits of the situation, you need to know the different types of e-learning [43].

Figure 1: Different types of E-learning.



- **Synchronous E-Learning:** This is real-time, instructor-based online learning that takes place between learners and educators as they both communicate in real-time using digital technology. In this format, novel tools like virtual classrooms (Zoom, Microsoft Teams), live webinars, and video conferencing become central. Synchronous learning itself reproduces the immediacy of face-to-face teaching, and dynamic discussions can be made, live question and answer sessions and group activities [44]. It will enable lively conversation, live polls and questionnaires, as well as group work. This will facilitate the social presence and cooperative learning that are relevant processes in engaging the students and motivating them. This model is particularly appropriate where feedback must be given immediately, such as in the learning of a language, the performing arts, or any technical demonstration. Nonetheless, the synchronous form of e-learning needs a good internet connection and its planning of various time zones that might not be universally accessible to all learners [45].

- **Asynchronous E-Learning:** this allows the learners to go through the educational material on his or her own time frame. It consists of lectures recorded in advance, interaction modules with learning activities, discussion texts, email messages (correspondence), and online assignments delivered in LMS. The model is very flexible and very independent, and the learner has the opportunity to go back, to think and read the concept, and to learn at his or her own pace. Asynchronous e-learning can be of great use with adult students, working people, and students whose learning styles differ or who cannot attend classes due to the location barrier. It also supports reflective, considered learning practices, though it can become on occasion prohibitively slow and decontextualized unless augmented with collaborative software and course instructor presence [44], [46].
- **Blended Learning (Hybrid Learning):** also referred to as hybrid learning, is a hybrid of traditional face-to-face education and online learning. It is rising in popularity in tertiary education and primary-to-secondary education and has the advantages of physical environments and digital ones. Most blended models use lectures online and face-to-face time in discussions, skills training, or exams. It is an instructional strategy that helps to provide differentiated instruction and, at the same time, direct teacher- student interactions. Blended learning accommodates the flipped classroom, in which the introduction of new material happens online and the development of a problem-solving or application activity happens in the classroom. When properly utilized, blended learning has the potential to make learning more effective as well as more engaging and is highly flexible and student-led [47].
- **Mobile Learning (m-Learning):** is an educational method of learning that is transacted using mobile devices like smartphones, tablets, and wearable gadgets. As more people gain access to mobile internet and mobile applications, m-learning is becoming an effective method of increasing access to education. It enables microlearning opportunities, which consist of brief, targeted content, which are particularly useful in just-in-time learning, language acquisition, and professional development. The gamification, interactive apps, and push notifications are added features that improve engagement and retention. It is specifically beneficial to m-learning, particularly in areas where access to conventional learning infrastructure and facilities is lacking and, therefore, learners can learn when and where they want. Nevertheless, issues like the inability to zoom in/out or screen size and digital side distractions and diverse device compatibility need to be taken into account when designing m-learning [48].

- **Massive Open Online Courses (MOOCs):** MOOCs are another interesting form of e-learning and entail scalable, mostly no-cost, access to educational material for a worldwide audience. The common elements are pre-recorded video lectures and discussion forums with peers and completion passes and certificates[49]. They are presented by Herzog in part by means similar to other platforms such as Coursera, edX, FutureLearn, and Khan Academy [33] [50]. MOOCs make education democratic and expand geographic and economic access to instruction, with all or a significant amount of content created on behalf of prestigious institutions being available to anyone with access to the internet. Nonetheless, phenomena like poor completion rates, low levels of interaction between the instructor and the learner, and absence of individualized feedback have been reported, and researchers as well as developers have started investigating more interactive and adaptive forms of MOOCs [51] [52].
- **Adaptive E-Learning:** This is a system that adapts the content being presented, taking into consideration the learner profile, i.e., prior knowledge, learning rate, and performance metrics. Adaptive platforms create individualized learning pathways that are made possible by using AI and ML algorithms to dynamically vary difficulty level, sequencing, and presentation of learning materials. This would be employed to maximize the effectiveness and interest in learning, and it provides a personalized learning plan. Adaptive e-learning is becoming increasingly popular in educational as well as corporate training communities, where learning analytics comes in to guide the input and help learners to advance in their studies [53].
- **Collaborative and social E-learning:** The social learning platforms and collaborative tools have opened a new aspect in e-learning whereby researchers have encouraged learners to co-create knowledge through the use of a social peer. This involves internet forums, wikis, blogs, group work, and integration of social media. Group e-learning will enable critical thinking and communication ability and enhance in-depth knowledge, as collective intelligence is harnessed. It is also in line with constructivist theories of learning with a focus on active learning and learner engagement with learning issues that have direct applicability to the real world [54].
- **Microlearning:** This method involves the delivery of didactic information in short, digestible bits, which are usually narrowed down to several minutes in length. Its main aim is to suit the reduced focus abilities that define modern learners and at the same time focus on reflexive, predetermined achievements. This mode is particularly beneficial in a

business onboarding environment, second-language learning, and developing specialized skills, as trainees have the benefit of efficient and on-demand learning that can be integrated into daily activities smoothly. The teaching content can include videos, infographics, quizzes, or flashcards. The format is ideal for just-in-time learning and offers high flexibility, although its impact can be limited without adequate follow-up or reinforcement [55].

4. Platforms and Technologies

Technological infrastructure is a key component of e-learning. The following platforms and tools form the backbone of modern digital education:

- **Learning Management Systems (LMS):** Provide a centralized platform for delivering, tracking, and managing educational content. LMSs support features such as grading, attendance, discussion forums, and analytics. Examples include Moodle, Blackboard, and Canvas [56].
- **Virtual Learning Environments (VLEs):** Interactive platforms that integrate communication, assessment, and content delivery. They often include tools for quizzes, peer review, and discussion boards. VLEs are designed to simulate classroom-like interactions in a virtual space [57].
- **Video Conferencing Tools:** Such as Zoom, Microsoft Teams, and Google Meet, which enable synchronous learning and collaboration. These tools have become essential for live lectures, group discussions, and office hours [58].
- **Mobile Applications:** Enhance accessibility and offer features like push notifications, offline learning, and gamified elements. Apps such as Duolingo and Kahoot have gained popularity for their engaging formats [59].
- **AI-Powered Tools:** Used for adaptive learning, automated grading, and personalized feedback. AI can identify learning gaps, recommend resources, and streamline administrative tasks [60].
- **Cloud-Based Solutions:** Facilitate scalable and flexible delivery of content. Cloud storage and SaaS platforms ensure learners and educators can access resources from any location [61].

5. Internet of Things (IoT) in E-Learning

The Internet of Things is a revolutionary technological trend that can take e-learning beyond the traditional digital education and education environment that offers the possibility to make smart devices connected to each other and retrieve, communicate, and process data in real-time. The IoT is made out of a network of physical things, which include sensors, wearable computers, smart boards, cameras, and environmental monitoring devices, and have software and connectivity built into them, thus allowing them to communicate via the Internet. In the case of e-learning, the IoT is used to improve the learning conditions, effectively connecting the physical and the virtual world, thus making educational experiences more interactive, contextual, and intelligent [62].

5.1Smart Learning Environments

So-called smart learning environments are IoT-based environments that allow institutions to track the engagement of learners, their attendance, and their interaction patterns automatically. As an illustration, the use of smart classrooms that have IoT sensors can monitor the presence of students, environmental conditions (lighting, temperature, noise), and equipment utilization, thus providing the best learning environment and enhancing energy savings. The LMS can be used to incorporate these data-driven insights to assist real-time analytics and informed decision-making by educators and administrators [63].

5.2IoT for Remote and Experiential Learning

Wearable sensors and smart devices are examples of IoT technologies that can be integrated into remote and online learning to facilitate experiential and practical learning, which is otherwise hard to implement in a wholly virtual context [64]. IoT-based laboratories can also be used in other fields like engineering, healthcare and medical education [65], [66], [67], robotics [68], [69], [70], [71], and science instruction where students can remotely interact with real-world equipment, perform experiments, and receive real-time feedback. The functionality adds a major boost to hands-on learning and minimizes the need to be in proximity, thus making practical education more affordable and extensible [72].

5.3IoT-Enabled Personalized and Adaptive Learning

Furthermore, in personalized and adaptive e-learning, IoT has a significant role. The IoT-based platforms will support more effective learner models through the automatic process of

collecting data on the behavior, the activity level, and the patterns of interaction of a learner. Along with AI and learning analytics, this information contributes to building adaptive learning paths that adapt in real time to the needs, preferences and performance edge of specific learners. This kind of customization drives engagement, retention and learning [73], [74].

5.4IoT in Mobile and Context-Aware Learning

IoT enhances mobile learning as well; IoT allows the smooth communication between learners and their immediate environments. Smart sensors and devices that are installed on ordinary objects enable learning to happen within a real-life setting, hence facilitating situated and context-aware learning. As an example, learning applications can also be location-based; that is, with the physical location of the learner, educational proceedings can be delivered, and smart notifications can offer a learner timely prompts and feedback to improve just-in-time learning [75]. Furthermore, the integration of IoT with artificial intelligence technologies, including autonomous systems and speech recognition, can enhance human–machine interaction in intelligent learning environments. Speech recognition systems provide natural and accessible interfaces through which learners can interact with educational platforms and smart devices [76], while previous research on the evolutionary optimization of Hidden Markov Models (HMMs) [77], [78], [79], Gaussian mixture models, and voice-based biometric authentication has demonstrated the potential of intelligent speech-processing technologies for developing adaptive and interactive digital environments. Similarly, autonomous mobile robots equipped with intelligent navigation and IoT communication capabilities can support educational experimentation [80], remote laboratories, and practical training activities [81], [82]. Therefore, the convergence of IoT, speech recognition, and autonomous robotic systems can contribute to the development of more interactive, intelligent, adaptive, and context-aware e-learning environments [83], [84].

5.5Challenges of IoT Integration in E-Learning

Although it has its benefits, the integration of IoT into e-learning presents several challenges. Data privacy and security are major concerns since IoT devices continuously collect sensitive learner information. Secure communication, data encryption, and compliance with privacy regulations are therefore essential[85]. Moreover, IoT infrastructure is costly to implement and maintain, while resource-constrained embedded systems introduce additional challenges related to processing, energy consumption, and real-time performance [86], [87].

6. Pedagogical Approaches in E-Learning

Effective e-learning depends not only on technology but also on sound pedagogical design. Several learning theories underpin e-learning practices:

- Constructivism: Emphasizes active learning through exploration, problem-solving, and real-world applications. Learners build knowledge through interaction with content, peers, and instructors [88].
- Connectivism: Highlights the role of networks, collaboration, and information flow in learning. This theory reflects the dynamic nature of digital environments where learners engage with diverse sources of knowledge [89].
- Flipped Classrooms: Students study content independently at home and engage in interactive activities during online sessions. This approach maximizes synchronous time for discussion and problem-solving [90].
- Gamification: Involves the use of game mechanics such as points, badges, and leaderboards to increase learner motivation. Gamified learning environments can make education more enjoyable and competitive [91].
- Project-Based Learning: Encourages learners to work on complex tasks and produce tangible outcomes. This method supports critical thinking, creativity, and real-world problem-solving [92].

These approaches enhance learner engagement, promote DL, and support diverse cognitive styles. They also underscore the importance of learner autonomy, collaboration, and contextualized knowledge [93].

7. Benefits of E-Learning

Online learning is a revolution to pupils and the schools that offer services to them:

- Flexibility: You can start and stop material at any time day or night, and at the speed you can learn and when your busy life gets in the way.
- Accessibility: Since everything is on the net, geographical distance can no longer inhibit your ability to have access to a particular course in a university or to achieve a desired degree. The same can be said of anyone who has mobility or other physical concerns.
- Scalability: A single lecture can be viewed simultaneously by thousands of users, which allows schools to save money and time to provide the material to a large group of people.

- Reduced Expenditure; When compared to such industries that use a great deal of expenditure on traveling as well as physical infrastructure, digital learning cuts down the costs and allows the university to make better use of available resources. On top of this, the digital materials can be adjusted and reused in an easier manner.
- Personalization: Rather than delivering all topics at the same level, adaptive platforms tailor learning content to individual learners' needs and abilities. AI-based recommendation systems further enhance personalization by analyzing learner preferences, behaviors, and interactions to recommend relevant courses and educational resources. Advanced recommendation approaches can also improve resource discovery and support more adaptive and learner-centered e-learning environments [94], [95].
- Data Analytics: In the background are computing programs that gather and process data on how you and your fellow users are performing. The information can help faculty to modify lessons, indicate areas of difficulties, and highlight reminders in advance to prevent trouble [96].

Taken as a whole, these advantages are boosting the potential of e-learning globally. It does not matter whether you live in a Technopolis or in a small village, digital learning can support lifelong learning and allow you to achieve an equal education.

8. Challenges and Limitations

E-learning is the much-hyped flexibility and accessibility, but there are a number of challenges that need to be marginalized to make it work:

- Digital divide: Laptops and smartphones are not available to all people and even fewer of them have a good high speed internet connection. Such injustice deters access particularly, in poorer neighborhoods or rural area [97].
- A passion and drive: They are able to fall asleep in case a student does not get the sufficient guidance or motivation as in the course that was set self-paced [98].
- Technical breakdowns: Learning can be interrupted by system down time, defective software and compatibility issues that increase levels of frustration [99].
- Pedagogical limits: Skills that require practical practice such as in laboratory experiment skills in science classes cannot be perfected online [100].
- Assessment integrity: It is difficult to keep exams clean in remote environments without excellent proctoring tools [101].

These issues need substantial institutional commitment, investment in facilities, ongoing instructor training, and pensive instructional design to be solved. Most importantly, policies and practices should be ruled by equity to ensure that all people get the chance of becoming included in digital learning.

9. E-Learning for Different Educational Levels

E-learning serves various educational levels, each with distinct needs:

- **Primary and Secondary Education**: Focuses on interactive content, digital literacy, and parental involvement. Platforms must be age-appropriate and support developmental goals [102].
- **Higher Education**: Offers flexible course structures and supports research, collaboration, and peer-to-peer learning. Universities leverage LMS platforms for hybrid and online degree programs [103].
- **Professional Development**: Tailored for working adults seeking continuous learning. Content is modular, self-paced, and often linked to career advancement [104].

Understanding the context of the learner enables the design of more effective and relevant e-learning experiences. Each level benefits from specialized tools and pedagogies aligned with its objectives. In this context, Intelligent Tutoring Systems (ITSs), supported by AI techniques, can provide personalized instruction, adaptive feedback, and individualized learning support across different educational levels, contributing to the development of more intelligent and learner-centered educational environments [105].

10.Global Trends and Future Directions

The future of e-learning is shaped by ongoing technological advancements and changing societal needs [106], [107]:

- **Artificial Intelligence (AI)**: Will further personalize learning pathways and automate administrative tasks [108].
- **Virtual and Augmented Reality**: Will enhance experiential learning through immersive simulations and interactive environments [109].
- **Learning Analytics**: Will provide deeper insights into learner behavior and enable predictive modeling to improve retention and performance [110].
- **Microlearning**: Short, focused learning modules will gain popularity, especially in mobile and corporate training [111].

- **Blockchain Technology**: May be used for secure credentialing and verification of learning outcomes [112] [113].

As these trends unfold, educators and institutions must remain agile and forward-thinking to leverage new opportunities while addressing emerging challenges.

11.Conclusion

Lifelong learning E-learning is a radical change in the form of education and in the process of learning for a modern, fast developing world, where learning, work and personal life is no longer separate. It has evolved from a supplemental form of education into one of the mainstays of academic and professional learning systems in the last decades. Digital transformation of the economy, globalization and the growing need for flexible and lifelong learning opportunities make e-learning one of the essential answers to many problems to which traditional education has to respond to. Although a number of developments have contributed to these advancements, the successful roll out of e-learning is still a complex picture with many components. Obstacles such as digital divide, infrastructure problem, pedagogy mismatch, and effectiveness of learner's engagement have still been concerns.

These limitations highlight the importance of approaching e-learning not merely as a technological substitution but as a comprehensive educational reform that requires thoughtful planning, inclusive strategies, and continuous evaluation. Nevertheless, the benefits of e-learning are both wide-ranging and profound. It offers unprecedented accessibility to education, enabling learners from remote regions, underprivileged communities, and non-traditional backgrounds to participate in quality learning experiences. E-learning also supports personalization, allowing content to be tailored to individual learning needs, preferences, and paces—thereby fostering deeper understanding and improved outcomes. Moreover, with the integration of technologies such as AI, VR, and Learning Analytics, the potential for innovation in instructional design and learner support is virtually limitless. Success in e-learning hinges on the synergy between technology, pedagogy, and policy. Robust instructional design, user-friendly platforms, teacher preparedness, and institutional commitment are all vital to ensuring that e-learning environments are effective, inclusive, and equitable. Educators must be empowered with training and resources to create engaging digital content, while students should be supported with digital literacy skills and access to learning technologies. Furthermore, collaborative efforts among stakeholders—governments, academic institutions, technology providers, and learners—are essential for establishing frameworks that promote quality assurance, data privacy, and universal access. As the demand for hybrid and remote learning

models continues to grow, educational systems must evolve to ensure that digital learning contributes meaningfully to human development, economic growth, and social equity. In conclusion, e-learning is not merely a response to technological change; it is a reflection of broader shifts in how societies value, pursue, and deliver education. As we look toward the future, a sustainable and inclusive vision for e-learning must be anchored in principles of equity, innovation, and learner-centeredness. By embracing these principles, we can harness the full potential of e-learning to create more resilient, adaptable, and accessible educational ecosystems for generations to come.

CHAPTER 02
CYBERSECURITY

Chapter 02: CYBERSECURITY

1. Introduction of Cybersecurity

E-learning has become one of the main pillars of modern-day learning, which offers learners unlimited accessibility and time flexibility, in the prevailing online environment. This transition, however, implies a high level of cybersecurity threats: poorly maintained online learning settings can lose their confidentiality, integrity, or even availability. Integrating cloud services, real-time communication technologies, and LMS into the infrastructures of institutions has also created a pluralistic attack surface prone to both external and internal risks. Since teaching, evaluation, administrative roles are all becoming digital solutions that rely on these digital networks, Information Technology (IT) security is no longer a technical friendly option but an ethical necessity [114].

In e-learning, cybersecurity involves the protection of digital structures, valuable user information, knowledge assets, and the integrity of systems. Unlike most existing IT systems, educational systems deal with a diverse group of stakeholders, including students, instructors, administrators, and a variety of third-party service providers, and these stakeholders have different access controls and different levels of technical expertise [115]. This heterogeneity increases the likelihood of accidental security breaches, which is why extensive user education, in addition to technological mechanisms and policy frameworks, should also become comprehensive.

The need to practice intense cybersecurity has been heightened by key demands of the world like the COVID-19 pandemic, which has created a high and massive migration to online learning. In this period, the attacks on educational institutions have greatly increased [116]. A report by Microsoft in 2021 has shown that education sector suffered over 60% of the reported malware attacks in the institutions of learning across the entire world during the pandemic. These trends represent a need to strengthen Defensive measures that are specific to particular vulnerabilities of digital education spaces [117], [118].

Regulatory forces also emphasize the need to prioritize cybersecurity in e-learning. Laws like the General Data Protection Regulation (GDPR) in the European Union and other similar ones in other regions are making educational institutions develop strict data-protection policies. The lack of adherence is subject to legal penalties and loss of confidence, and therefore cybersecurity is important in the long-term viability and credibility of online learning [119].

This chapter will thus provide a wholesome overview of cybersecurity in e-learning systems. It summarizes the major cyber threats experienced in digital learning spaces, the vulnerability that they have, and preventive and remedial security countermeasures. It also looks at the role of the emergent technologies, especially the AI in cyber resilience. As a result of adopting a systematic approach to studying the various complexities of security in e-learning, institutions will be able to develop safer and more reliable educational experiences among all users.

2. Common Types of Cyberattacks in E-Learning Environments

In the modern education field, the e-learning platforms have been specifically enticing candidates to cybercriminals considering the large amounts of sensitive data they process, including academic, personal, financial [120], and research holdings, and how business continuation is a priority [119]. The related cyber threats are of diverse nature in terms of motive and complexity, as shown in Figure 2, beginning at opportunistic single-incident exploits and extending to the well-organized long-term incursions. This complex threat environment requires a sophisticated insight into the process of designing effective cybersecurity architectures within digital-education environments [121].

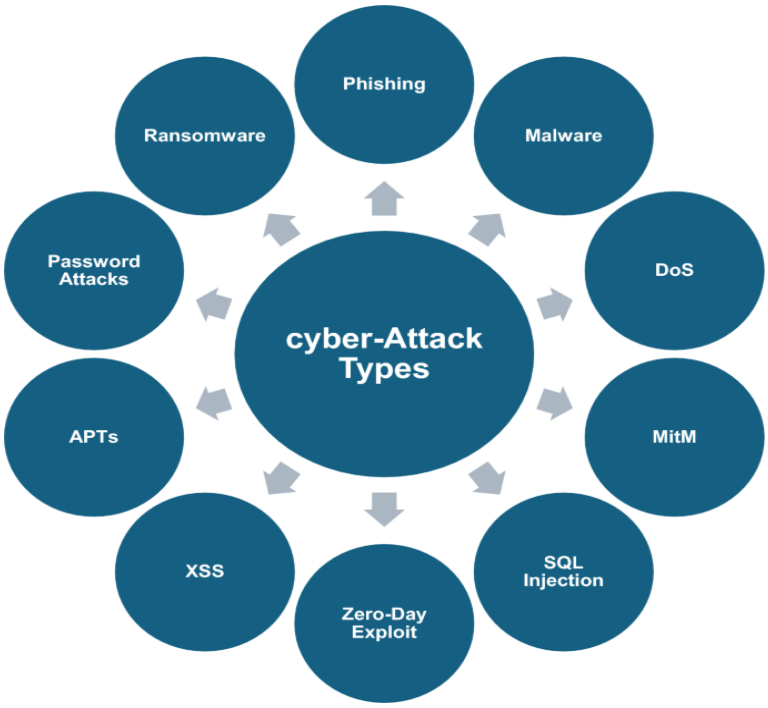


Figure 2: Major Cyberattack Types Targeting E-Learning Platforms

2.1Phishing Attack

Pedagogically, phishing is a type of electronic fraud where the attacker will create a message, usually a message like an email or instant message that will appear to be sent by an entity the target considers legitimate and will as a consequence lead to a target giving up sensitive materials like the login and password or financial information [122]. In academia, these identity thieves often pretend to be administrators in their messages or even recreate the Learning Management System sign-on pages. Such attacks are often successful culminating in account hijacking, academic information theft and mass malware infection. Since email is still an important part of digital learning environment, phishing is still one of the most tenacious and potentially harmful threats higher learning has to face [123].

2.2Malware Attack

Malware the actual term is the broadening of the word virus to include worms, trojan horses, ransomware, spyware, and the rootkit. These structures are designed with ill intentions of invading and harming computing systems without seeking any authority to do so on the part of the system owner. In an e-learning environment, malware may take the form of emails, courseware that can be downloaded, or a compromised browser add-on [124]. The consequences such as unauthorized surveillance and loss of control over the computing environment or, in the case of ransomware, the lock-up of the entire system that can lead to significant disruption of the learning process and the exposure of confidential information of the organization may follow [125].

2.3Denial-of-Service (DoS) and Distributed DoS (DDoS) Attacks

In the sphere of educational technology, the occurrence of DoS and DDoS attacks is a current risk aimed to bring online learning systems out of commission by inflating a huge amount of traffic aimed at critical servers. The DDoS variant differs by the attack being launched by a set of hostile computers that have been compromised into a set, usually organized into botnets, thus increasing the size of the assault, as well as the level of operations associated with the attack. In the past, students and faculty were not able to access the critical platforms due to such attacks in critical periods and times such as during examination and when students had to submit their assignments. It is also notable that DoS request can be used as a feint in that they are used sometimes to serve as a distraction to defenders in that decoy actions are pursued as more targeted intrudes in other areas are on-going [126].

2.4Man-in-the-Middle (MITM) Attack

An unlucky situation that is regularly noticed in modern kinds of e-learning is MITM assault. In such scenarios, traffic will be intercepted by some malicious third party and it might even be altered; a student and a LMS server as an example and the parties will not show any kind of awareness. They usually launch such attacks using unsecured Wi-Fi connections or by using DNS spoofing methods. The results can be dire to students: stealing of credentials, tampered correspondence, and the rottening of academic information, especially due to the fact that a large proportion of the students use publicly-owned or otherwise managed networks pay little security to protect them when using LMS websites [127].

2.5SQL Injection Attack

The issue of SQL injection attacks should be regarded in the context of the fleedom of digital pedagogy. These exploits can be used to execute arbitrary SQL queries in the back-end databases because they target flaws with poor security that adversely affect the input fields. Lax inputs and authorization checks may lead to information retrieval, corruption, and deletion: grades, student profiles, assessment data, and other important data. In this respect, SQL injection is a rather dangerous vulnerability in e-learning systems that often occurs due to legacy codes, poor development habits, or suboptimal database setup [128].

2.6Zero-Day Exploit

Zero-Day Exploit attacks on unknown weaknesses of software or systems, the ones which are not being addressed, patched or even known by developers yet. The zero-day exploits are very risky because it does not offer any defenses when the attack occurs. They can be utilized to flood e-learning systems with malware, unauthorized access, or denial of critical services without the institutional IT teams having any idea [129].

2.7Cross-Site Scripting (XSS)

XSS attack is about injecting malignant scripts into credible web applications. It is also possible in e-learning environment considering that a discussion post, assignment comment pages, and the profile pages all contain vulnerable input fields. Such scripts run in the browsers of other Internet users, causing session hijacking, theft of credentials, or redirecting to bad websites. The risk is especially severe in the systems that do not robustly implement input validation or output encoding [130].

2.8Advanced Persistent Threats (APTs)

APTs are targeted attacks by advanced threat actors that are long-term. That is, these attacks are specific and executed by highly skilled individuals or groups, usually state-sponsored or part of organized cybercrime groups. Such attacks consist of several processes such as reconnaissance, infiltration, lateral movement, and exfiltration. In e-learning systems, APTs can be aimed at research data, administrative systems or intellectual property. APTs are slowly and quietly evolving and thus they are known to be very difficult to identify and address [131].

2.9Password Attacks

Password attacks are used to break into a system using weak or otherwise compromised credentials, re-used credentials, or weak username/ password pairs. The methods used are brute-force attack, dictionary attack, and credential stuffing. Online educational systems, hacked user profiles can cause unacceptable grade manipulation, loss of information, and fraudulent representation of staff and students. Although this trend is on the rise, weak or single-factor authentication is still present on many platforms subjecting them to these types of attacks [132].

2.10 Ransomware

A ransomware is a type of malware that encrypts data and systems and after encrypting it, the malware asks money, usually in cryptocurrency to decrypt them. In online education, a ransomware outbreak may prevent the educators and learners to enter the LMS, encrypt academic data, and neutralize institutional processes. Some of the prevalent infection vectors involve phishing emails, malicious software updates, and unsecure remote access. With the increased adoption of e-learning, this trend is increasing its attractiveness as a victim to ransomware attacks, particularly on impoverished educational institutions with low cybersecurity resources [133].

3. Security Vulnerabilities in E-Learning Platforms

The design of the e-learning platform is usually composed of a range of elements like LMS, cloud, 3rd party tools, and modules of communication. Each of these components may create a security issue when they are not designed properly, configured, or maintained accordingly [134]. It is imperative to find such weak points to be able to strengthen the resilience of e-learning ecosystems in general.

3.1 Weak Authentication and Authorization Mechanisms

Most e-learning systems are still using usernames and passwords and are still lightly enforcing strong password security. Brute-force attacks and credential stuffing are very efficient against weak or reused passwords. Also, there are poorly implemented role-based access control (RBAC), which results in unauthorized access to sensitive data like student records, test banks, or graders [135].

Organizations that fail to install multi-factor authentication (MFA) or biometric verification are the ones that are highly exploitable in terms of account takeovers and privilege escalation attacks [136].

3.2 Insecure Communication Channels

The communication between users and servers including the login information, personal details, or test scores should be also encrypted with a protocol like HTTPS and SSL/TLS [137]. Still, certain platforms or integrated tools can transfer data in plain text or make use of weak encryption algorithms, which can be eavesdropped or subjected to the MitM attack or even session hijacking [138].

E-learning is becoming more in-depth with video conferencing and live streaming, which does pose the risk of no longer being audio-only when these events are not properly secured in a manner that utilizes meeting passwords, waiting rooms, and participant management [139].

3.3 Third-Party Tool Integration

E-learning platforms in the modern environment tend to be the systems that have the third-party simulators, extensions, or APIs to cover some or all of quizzes, analytics, discussion boards, and plagiarism tracking. Although these tools may make the implementation be more functional, they may also come with security risks unless they are frequently updated, deliberated, or kept securely linked [140].

Insufficiently sandboxed or higher than necessary permissioned plugins become a source of attack in that they could compromise the whole platform. Intruders can also take advantage of their flaws in such tools in order to inject malicious codes or do unauthorized accesses.

3.4 Software Vulnerabilities and Outdated Systems

All software that is available is subject to bugs and weaknesses. Old Moodle, Blackboard or home-made systems may lack critical security code changes, thus can be vulnerable to technologies such as SQL injections or remote code execution (RCE) and XSS [141].

Evaluation and monitoring of dependencies are not performed regularly in open-source tools which makes the risk even more significant. Institutions with weak IT personnel might postpone the process of patching because of compatibility issues or insufficient human resources.

3.5 Poor Session Management

Poor management of sessions can lead to the phenomenon of session fixation, session hijacking, or unauthorized persistence [142]. As an example, when a user logs out or a timeout happens, invalidation of enterprises is not made a reality, thus giving the malicious actor a chance to pose as a legitimate user. Besides, systems that do not use secure cookies, IP binding, or user-agent validation expose session tokens to interceptions [143].

3.6 Lack of Security Configuration Management

Inadequate security configuration of servers, databases, or cloud storage (e.g., AWS S3 buckets) resulted in many data leakages in business and academic settings. Among the most prevalent security mistakes that lead to reduced security are using default credentials, unlimited security access, and lacking encrypted backup among others [144].

With e-learning, exposed databases can give away information that is sensitive such as student progress, internal communication or even unpublished research content.

3.7 Insufficient Logging and Monitoring

Most of the e-learning systems do not provide real-time tracking and means of logging access, all failed logins, changes of privileges, or file downloads. Institutions cannot see what is going on in their networks with these controls in place, lateral movements, ongoing intrusions, or data exfiltration [145].

Because the monitoring tools may prevent an incident, a good monitoring tool must have alerts, audit trails and anomaly detection features to facilitate post incident forensics and be notified of suspicious activity.

4. Case Studies and Real-World Incidents in E-Learning Cybersecurity

There are real-life examples that give us an idea about the vulnerabilities and attack methods prevailing in e-learning systems. With the spread of digital learning across the world, there has been an upsurge in cyberattacks against institutions in education. Case studies show the principle of exploitation of weak security measures, poorly prepared infrastructures, and humans offered by cybercriminals to steal information, provide disruption of the service, or perform spying activities [146]. In this section, the reported cases of cybersecurity breaches in learning settings will be provided, alongside the effect that the COVID-19 pandemic has had on the e-learning threat landscape.

4.1 Case Study: The University of California – Ransomware Attack (2020)

In June 2020, the University of California, San Francisco (UCSF) School of Medicine was hit with a major ransomware attack that targeted them. The attackers locked important research and academic information and threatened the university with a ransom of more than 1 million dollars, which the university eventually paid so as to have access to its systems again [147].

This incident highlights several important lessons:

- High-quality research data may be a valuable asset, and utilizing it could become worthwhile when crises prevail all over the world.
- The attack surface is increased because there is no segmentation of networks and protection of sensitive systems.
- Institutions require integrated backup and incident response and cyber insurance.

4.2 Case Study: University of Utah – Data Breach (2020)

The University of Utah was attacked through the computer world, where it suffered unauthorized access to its IT systems [147]. The hack has made the confidential information visible, and the university had to pay 457 thousand dollars not to publish the data. During the investigation, it was found that there were malfunctions in terms of storing and access to internal data [148]. This incident highlights several key implications:

- Colleges and universities can be attacked as a target of extortion.
- Another trend is a common attack of backup systems and data repositories by cybercriminals.
- Active surveillance and encryption should be used.

4.3 Case Study: Moodle Platform Vulnerabilities

Moodle, one of the most popular open-source LMS, has become the target of different vulnerabilities that may affect the security of the user-related information and course-related materials. Such areas of vulnerabilities include authentication mechanisms, handling files and maintaining sessions that have put at risk unauthorized access, data leakage and remote code execution. Hackers using such flaws may have unauthorized access to the accounts, the stealing of sensitive information, or interruption of normal workflow of educational processes. As a reply, the Moodle community constantly tries to remedy the security issues and improve resilience of the platform. All these threats can be curbed by regular updates, secure coding and adherence to security monitoring and keeping a safe learning environment to users. This case reemphasizes the necessity of protecting online learning environments and the insistence of active care when keeping their integrity against the emerging online attacks. [149].

4.4 Impact of COVID-19 on E-Learning Cybersecurity

The COVID-19 pandemic immensely increased the rate of online learning globally, and a lot of learning institutions did not do much to match the rate and size of the shift, and this contributed to heavy cybersecurity weaknesses[150]. This shift witnessed a meteoric increase in phishing attacks typically posing as university administrators or health officials and a jump in ransomware attacks on school districts and universities. Also, insecure video conferencing tools, like Zoom, have been faced with the so-called Zoom-bombing [151]. The outcomes were manifold in the form of security implications: the institutions required infrastructure that was cloud-secure, scalable, and user-friendly; cyber hygiene and digital literacy became as important as an internet connection; and incident response protocols had to be adjusted to the conditions of hybrid and remote learning.

5. Detection and Prevention Mechanisms

Since cyber threats against online learning systems are currently becoming increasingly complicated, institutions need to implement a series of tools and measures in order to identify, preclude, and react to malicious action. As noted in this section, four critical defense mechanisms are mentioned, including IDS, IPS, firewalls, and endpoint protection, and AI-based detection mechanisms [152].

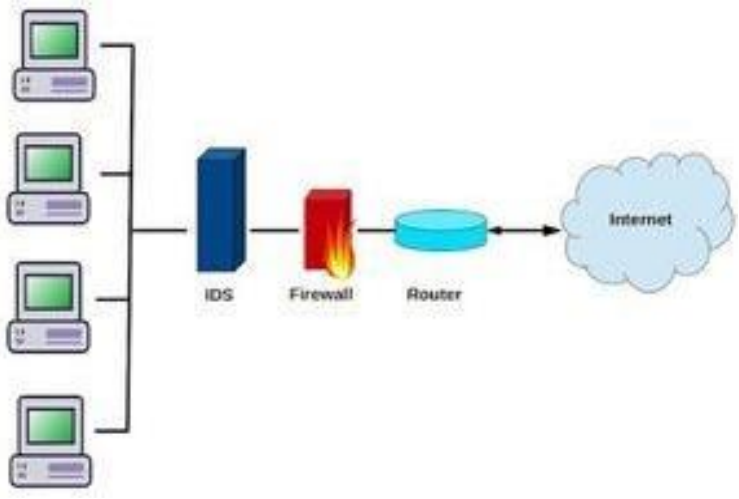
5.1 Intrusion Detection Systems (IDS)

IDSs are systems which are specified to monitor and to analyze activities occurring in a network or a system in case there are signs of malicious behavior [153]. These systems typically function using two main approaches:

- Signature-based and by detecting known threats based on prepackaged signatures of attacks.
- Anomaly-based, i.e. detecting something normal, which is not normal and which can reflect a new or zero-day attack [154].

In e-learning environment, IDS has been very instrumental in security of online education where information in terms of unusual sign in, bad files download amongst other things is constantly monitored [155]. With these suspicious actions realized in a timely fashion by IDS, the security teams can discover that an intrusion has occurred and can act accordingly. Security is one of the major strengths of implementing an IDS in e-learning systems because it enhances protection to sensitive educational data and resources against a wide range of cyber-attacks, which undoubtedly makes the learning environment safer and more reliable to educators and learners [156]. Figure 3 shows that the IDS is used, together with a firewall and a router as layers of defense to provide proper security and monitoring of the traffic within a network.

Figure 3: Intrusion Detection System architecture

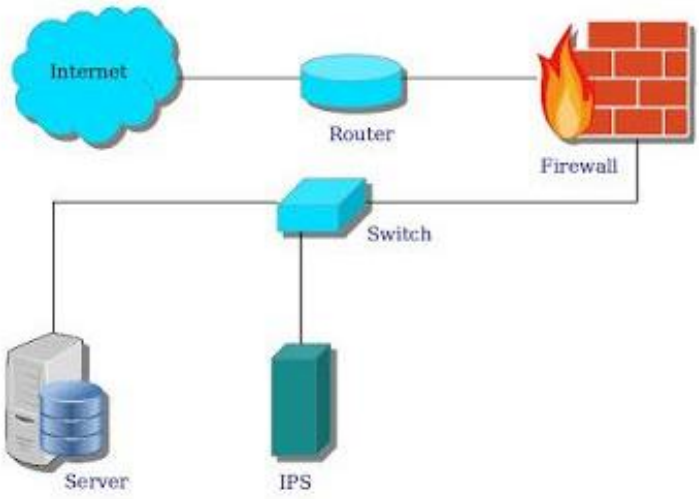


5.2 Intrusion Prevention Systems (IPS)

IPSs complement the existing IDS as it is proactive and prevents recognized threats in real-time. Inserted directly in the network traffic stream, IPS has the ability to take decisive measures like dropping malicious packets, terminating malicious connections and blocking traffic

originating at known malicious IP addresses. IPS is particularly essential in the protection against the following threats, DDoS attacks of LMS servers, buffer overflow attacks, SQL injection attacks, and the infiltration of malware. With a good IPS, institutions are able to secure and safeguard continuity and integrity of online learning services against disruption and ensure that there is a secure stable learning environment [157]. Figure 4 below illustrates the integration of an IPS within a network security architecture.

Figure 4: Intrusion Prevention System architecture



5.3 Firewalls and Endpoint Protection

Firewalls can be used as protection against untrusted external traffic like the internet with trusted internal traffic. These observe and filter the network traffic either inbound or outbound according to some well-defined security regulations. Firewalls help in preventing malicious data packets to reach the sensitive systems by filtering the traffic and blocking unauthorized connectivity, thereby neutralizing most threats of external attacks, including malware, ransomware, and DoS attacks. Firewalls may be either software-based, hardware-based, or a combination of both, and may be used at any layer of the network stack, anything up to deep packet inspection and carrying out application-level filtering [158].

Endpoint Protection denotes security backend that defends individual systems or point of end i.e., computers, mobile devices, and other communicative gadgets/devices against hazard. Endpoint protection normally addresses antivirus and antimalware software, and other mechanisms which attempt to identify, prevent and eliminate potentially malicious programs or breaches at the individual computer endpoint. Behavioral analysis, encryption and data protection against data loss are common features of the advanced endpoint protection products

to that they can be comprehensive in addressing a wide range of threats such as zero-day exploits, ransomware and phishing [159].

5.4 AI-Based Detection Techniques

Cybersecurity systems using AI-based detection techniques are transforming the manner in which they respond to cyber threats and to detect them. AI can be used to improve techniques because they extend the possibility to detect malicious activities, automate the process of detecting threats, and optimize reaction time. The main AI-based detection techniques are ML, DL, and NN, which analyze the used extensive data, detect the patterns, and arrive at real-time decisions by identifying the potential threats. These methods are more than signature-based detecting as they can identify new or developing threats including zero-day attacks and advanced persistent threats (APTs) [160].

5.4.1 Machine Learning (ML)

ML models are also trained on network traffic or system logs on high volumes of data to recognize patterns of normal and abnormal operating patterns. After being trained in these patterns, the system will automatically identify when it deviates and this may be a sign of a cyberattack, including a DDoS attack or data exfiltration. Such methods can include supervised learning, unsupervised learning and reinforcement learning [161].

5.4.2 Deep Learning (DL)

DL is a subcategory of ML and refers to using deep networks (neural networks with very many layers) to learn representations of extremely complicated data sets like images, sounds, and sequential vision. The DL that serves cybersecurity is capable of finding more complex patterns and correlations in data and, thus, is especially successful at identifying complex attacks that are otherwise challenging to detect using classical techniques [162].

5.4.3 Neural Networks

Artificial Neural Networks (ANNs) are the computerized networks based on the organization and processes of the human brain. Those networks are networks of layers of interconnected nodes (neurons) processing and learning input data. ANNs are applied in IDS and malware identification to classify malicious traffic and predict the occurrence of malicious activity on the basis of training data. They affect their effectiveness in identifying threats that emerge as they are capable of learning using large quantities of data [163].

5.4.4 Anomaly-Based Detection

Anomaly-based detection systems usually employ AI as part of their system, where normal behavior is gradually learnt and anything out of that behavior is marked as a suspicious incident. The anomaly-based detection is able to detect new, unknown threats, whereas signature-based detection can detect only the known ones. This is especially helpful in dynamic environments where tactics of the attackers keep on evolving [164].

5.4.5 Natural Language Processing (NLP)

Cybersecurity is another area where NLP techniques are being implemented, especially in identifying phishing attacks, and social engineering. NLP has the ability to study and comprehend patterns of texts in emails, messages, or social media posts and detect suspicious text or messages, which can be run through phishing-scams [165].

5.4.6 Federated Learning (FL)

Federated Learning (FL) is an emerging AI approach that enables multiple devices or institutions to collaboratively train ML models without sharing their raw data. In e-learning and IoT environments, FL can enhance cybersecurity and privacy by allowing distributed devices and platforms to contribute to threat detection while keeping sensitive learner data locally stored. This decentralized approach reduces data exposure and supports the development of privacy-preserving intrusion and anomaly detection systems. When integrated with AI and IoT technologies, FL can therefore contribute to building more secure, scalable, and privacy-aware e-learning ecosystems [166].

5.5 Security Requirements for IoT-Enabled E-Learning Systems

The process of ensuring the security of IoT-enabled e-learning systems requires a multi-layered response to the threats across devices, networks, and applications. Authentication and access-control measures cannot be done without to make sure that all users and devices are correctly identified and authorized, but sensitive learner data should be protected by a strong encryption algorithm and privacy-aware policies [167]. The instrumentation of IoT should include secure boot, integrity checking of firmware, and verified protocols of updating, and communication channels should be strengthened to resist interception and modification. Constant monitoring and intrusion-detection solutions are used to identify and ensure that abnormal behavior is detected in good time. Additionally, security solutions are expected to maintain system availability and performance at scale with a large group of users and devices

and ensure safe interoperability with third-party tools and cloud-based services. As a result, the security of IoT-based e-learning platforms requires a layered and adaptive approach to be implemented and keep usability and educational effectiveness [168].

6. Conclusion

The integration of e learning technologies has completely transformed education as it has enhanced accessibility, scalability, and flexibility. Yet, the associated risks of cybersecurity are increasing and, without proper measures, threatens to undermine institutional trust, violate user privacy, and discontinue academic work. The chapter has covered an in-depth look into the key cyber threats impacting e-learning settings that include phishing, ransomware, denial-of service attacks, insider threats, and use of zero-days vulnerability. Moreover, it has established that architectural and operational vulnerabilities exist in LMS in the form of insufficient authentication mechanisms, third-party integrations, and misconfiguration of software. To deal with such threats, there should be a layered approach to security. Advanced AI-driven detection technologies in conjunction with such technologies as IDS, IPS, firewalls, and protection on endpoint devices should be implemented. The use of AI and ML was highlighted in specific detail since it will be able to facilitate the application of predictive analytics, real time anomaly detection and automated response systems. To step down, not only should institutions put more money in technological defense, but also change the culture of cybersecurity into one of the students, faculty members, and administrators. This should entail regular risk assessment, adherence to international data protection laws and regulations (such as GDPR, FERPA), and constant surveillance, as well as the utilization of the privacy-by-design concept. Deploying e-learning ecosystems will hence be an extremely collaborative process with educational, technological and policy-making organizations as key players in ensuring high level of security, inclusiveness and resilience of an e-learning ecosystem.

CHAPTER 03
LITERATURE REVIEW

Chapter 03: LITERATURE REVIEW

1. Introduction

The continuous advancement of digital technologies has brought about significant benefits to modern society; however, it has also given rise to a range of cyber threats that challenge the security and integrity of information systems. Among these threats, phishing, network intrusions, and academic cheating in online environments have emerged as critical issues that demand sophisticated detection and prevention mechanisms. Consequently, the field of cybersecurity has increasingly turned toward ML and DL techniques as powerful tools capable of automating detection processes, identifying complex attack patterns, and reducing human dependency in decision-making.

This chapter provides a comprehensive overview of related works that apply ML and DL approaches to the domains of phishing detection [169], IDS, and cheating detection. The review aims to synthesize existing research efforts, emphasizing their methodologies, algorithms, datasets, performance metrics, and comparative strengths and limitations. By analyzing these studies, the chapter highlights how the integration of intelligent models contributes to improving detection accuracy, minimizing false positives, and enabling adaptive learning in dynamic cyber environments.

The chapter is organized into three main sections. The first focuses on intrusion detection, summarizing major ML and DL-based techniques designed to recognize and classify network anomalies in real time. The second section discusses phishing detection, examining recent approaches that leverage URL analysis, email content, and hybrid DL frameworks to enhance detection performance. The third section explores cheating detection, a growing research area within educational cybersecurity, where ML and DL models are used to ensure integrity in online learning and assessment platforms. A comparative discussion at the end of the chapter consolidates the findings and underscores the persistent challenges—such as computational costs, dataset imbalance, and real-time scalability—that continue to shape the evolution of intelligent cybersecurity solutions.

2. Related works

The given chapter examines the current literature on phishing and IDS with application of ML and DL. The aim is to discuss the methods, algorithms and outcomes existing in literature and to draw attention to the strengths and limitation of various methods in solving cybersecurity

challenges, especially phishing and cheating detection. In the reviewed studies, datasets are diverse and include the features based on URLs that recognize phishing to more complex models to recognize academic cheating. The chapter seeks to give a detailed idea of the existing approaches and precondition the suggested work to assist in this developing sphere. The ensuing sections have been divided into two major categories as Phishing Detection and IDS. In each section, one elaborates on the relevant studies, algorithms applied, data sets, outcomes, strengths, and limitations, and therefore concludes on future work performed in the said areas.

2.1 Intrusion detection

Jing et Chen, 2019 [170] proposed a ML-based intrusion detection method using a Support Vector Machine (SVM) model enhanced with a nonlinear logarithmic scaling approach for improved data preprocessing. The study focused on the UNSW-NB15 dataset, which consists of 257,673 records (175,341 for training and 82,332 for testing) divided into one normal class and nine attack classes, including DoS, Exploits, Fuzzers, Backdoor, and others. The authors utilized the RBF kernel in the SVM model and implemented both binary and multi-class classification tasks. Their results demonstrated that the proposed method achieved a testing accuracy of 85.99% for binary classification and 75.77% for multi-class classification, outperforming other conventional algorithms such as Naive Bayes (NB), Random Forest (RF), and Expectation-Maximization (EM) clustering. Moreover, the detection rate reached 97%, while the FPR was as low as 3%, indicating the model's strong ability to distinguish between normal and attack traffic. The main advantages of the proposed approach lie in its high accuracy, reduced FPR, and adaptability to modern and complex attack types in network traffic. However, the authors also acknowledged limitations such as sensitivity to class imbalance within the dataset and the computational burden of SVM training on large-scale data, which could restrict real-time deployment.

Tian et al., 2020 [171] introduced an intrusion detection approach based on an improved Deep Belief Network (DBN) to address challenges such as overfitting, low classification accuracy, and high FPR in traditional IDS. The authors incorporated a combined sparsity penalty term based on Kullback–Leibler (KL) divergence and non-mean Gaussian distribution into the likelihood function of the unsupervised training phase of DBN, forming the KG-DBN model. The model structure includes stacked KG-CRBM (Continuous Restricted Boltzmann Machine) and KG-RBM layers, followed by a backpropagation (BP) neural network for fine-tuning. For data preprocessing, the study applied probabilistic mass function (PMF) encoding

to convert nominal features into numeric values and Min–Max normalization to scale continuous data. Experiments were conducted on two benchmark datasets — NSL-KDD and UNSW-NB15, with three hidden layers trained via the contrastive divergence (CD) algorithm. The proposed model achieved 96.17% accuracy with a 1.46% FPR on NSL-KDD and 86.49% accuracy with a 5.56% FPR on UNSW-NB15, surpassing several state-of-the-art methods such as basic DBN, CDBN, RF, and kNN-based models. The advantages of this method include its strong generalization ability, reduced feature redundancy, and improved detection of both known and new attacks. However, the authors noted limitations, including increased computational cost due to multiple hidden layers and the model's dependence on hyperparameter tuning, which may affect its scalability for real-time intrusion detection environments.

Almiani et al, 2020 [172] proposed a Deep Recurrent Neural Network (DRNN) model for an IoT IDS designed to enhance Fog computing security. The model features a multi-layered recursive neural network architecture that performs intelligent classification of IoT traffic at the network edge, using two cascaded detection layers—each fine-tuned to identify specific attack types such as DoS, Probe, R2L, and U2R. The authors employed the NSL-KDD dataset, which contains over 125,000 records, and applied comprehensive preprocessing steps including symbolic-to-numeric transformation, feature reduction (reducing from 41 to 26 attributes), min–max normalization, and data oversampling to address class imbalance. The system was trained using a modified backpropagation algorithm with recursive feedback and Kalman filtering to stabilize weight updates and prevent gradient explosion. Experimental results demonstrated that the proposed DRNN achieved 94.27% detection rate, 92.18% accuracy, 9.8% false positive rate, and high Mathew Correlation Coefficient (84.44%) and Cohen's Kappa (84.36%), indicating strong stability and robustness across classes. The model particularly excelled at detecting DoS (98.27%) and Probe (97.35%) attacks, while maintaining acceptable performance for R2L (64.93%) and U2R (77.25%). Advantages of the approach include its real-time capability and its resilience to data imbalance. However, the authors noted limitations such as the computational overhead associated with deep network structures and the sensitivity to hyperparameter tuning, which may challenge deployment on resource-constrained IoT devices.

Ashiku et Dagli, 2021 [173] proposed a DL-based network IDS that leverages a CNN architecture integrated with a regularized Multilayer Perceptron (MLP) to identify and classify network attacks in modern digital environments. The study utilized the UNSW-NB15 dataset,

a comprehensive benchmark reflecting realistic network traffic containing nine categories of cyberattacks. The proposed CNN architecture incorporated double-stacked convolutional layers, max pooling, and dropout regularization to mitigate overfitting and enhance feature learning. Experimental results demonstrated 94.4% accuracy on the pre-partitioned UNSW-NB15 dataset and 95.6% accuracy on a user-defined 70–30 split dataset. The model achieved competitive performance compared to other DL methods, significantly improving detection rates while maintaining stability across multiple attack classes. The advantages of this system include its adaptive hyperparameter optimization, robust multiclass classification capability, and high detection accuracy even under class imbalance. However, limitations were observed in handling underrepresented attack categories, where detection rates declined due to dataset imbalance, and in the computational complexity inherent in deep CNN architectures. The authors suggested future work focusing on transfer learning, feature reduction, and bootstrapping techniques to further improve real-time detection of zero-day attacks.

Al-Emadi et al., 2020 [174] proposed an intelligent NIDS using DL techniques—specifically CNN and RNN—to effectively detect and classify various types of cyberattacks. The study employed the NSL-KDD dataset. Approximately 85% of the data was used for training and 15% for testing and validation. Both CNN and RNN (using LSTM and Gated Recurrent Unit (GRU) variants) models. The CNN architecture consisted of one convolutional layer, max-pooling, a dense layer, and dropout to prevent overfitting, while the RNN architectures used two LSTM or GRU layers with dropout and a dense output layer. Experimental results demonstrated that CNN outperformed the other models, achieving 97.01% accuracy, 100% precision, 97.01% recall, and an F1-score of 98.48%, whereas RNN-LSTM achieved 81.6% accuracy and RNN-GRU performed lowest with 50.25% accuracy. The advantages of the proposed approach include high detection accuracy, robustness, and reduced overfitting through dropout regularization, making CNN a suitable choice for real-time NIDS with limited computational resources. However, limitations were noted, particularly high computational costs during training and sensitivity to hyperparameter tuning, which hinder scalability for large-scale or real-time deployment. The authors suggested future work focusing on enhanced computing resources and hybrid DL architectures to improve detection speed and accuracy.

Mebawondu et al, 2020 [175] developed a NIDS based on a supervised learning paradigm utilizing an ANN with a MLP architecture. The study aimed to create a lightweight and efficient IDS by applying feature selection via Gain Ratio and binarization techniques to the UNSW-NB15 dataset. During preprocessing, continuous features were binarized while nominal attributes were discretized into numeric form to simplify computation. The Gain Ratio method was then employed to rank and select the top 30 most relevant features, reducing redundancy and computational overhead. The model was implemented and trained with the dataset divided into 70% training and 30% testing. Experimental results showed that the ANN-MLP achieved an accuracy of 76.96%, precision of 79.8%, and a FPR of 23.95%, while the Matthews Correlation Coefficient (MCC) reached 0.57, indicating a moderate correlation between predicted and actual classifications. The proposed IDS was capable of effectively distinguishing between normal and attack traffic, validating the robustness of the feature selection method and ANN classifier. The advantages of the model include its computational efficiency, good generalization capability, and adaptability for real-time intrusion detection due to its lightweight design. However, its limitations involve relatively low accuracy compared to DL approaches, a high false alarm rate, and restricted testing across varying feature subsets. The authors recommended future research focusing on optimizing feature selection thresholds and integrating faster classification algorithms to enhance detection accuracy and support real-time network security applications.

Dutta et al, 2021 [176] proposed a hybrid DL model for network intrusion detection that combines a Classical Autoencoder (CAE) for feature engineering with a Deep Neural Network (DNN) for classification, aiming to enhance detection accuracy and reduce false positives. The model was trained and tested on the UNSW-NB15 dataset, which includes 2,540,044 labeled instances (175,341 for training and 82,332 for testing) encompassing nine attack types and one normal class. The CAE employed one-hot encoding and Min–Max normalization to preprocess 196 features, extracting latent representations subsequently fed into a four-layer DNN trained using stochastic gradient descent with the Adam optimizer and binary cross-entropy loss. The proposed hybrid approach achieved 91.29% accuracy, 92.08% precision, 90.64% recall, 91.35% F1-score, and an 8.05% FPR, outperforming baseline models such as standalone DNN (88.15%) and RF (85.14%). The advantages of the model include superior classification performance, automatic nonlinear feature extraction, and enhanced robustness, while its limitations involve higher computational complexity and reduced accuracy for minority attack classes due to dataset imbalance. The authors recommended future work focusing on improving

multi-class attack detection, balancing minority samples, and optimizing scalability for real-time network environments.

Said et al, 2023 [177] introduced a hybrid DL model, termed CNN-BiLSTM, for enhancing NIDS within Software-Defined Networking (SDN) environments. The proposed model combines the CNN for spatial feature extraction with Bidirectional Long Short-Term Memory (BiLSTM) for temporal feature learning, leveraging the strengths of both architectures for superior intrusion detection accuracy. The researchers utilized three benchmark datasets—InSDN, NSL-KDD, and UNSW-NB15—each preprocessed through label encoding, random oversampling to balance data, and a hybrid feature selection strategy based on RF importance ranking and Recursive Feature Elimination (RFE), which reduced the features to ten highly significant attributes. The experimental results demonstrated outstanding detection performance, achieving 97.77% accuracy for binary classification on the InSDN dataset, 98.42% on NSL-KDD, and 84.23% on UNSW-NB15, outperforming conventional DL models such as CNN, AlexNet, LeNet-5, and CNN-LSTM. The CNN-BiLSTM also maintained high recall and precision rates across multiple attack categories, proving its robustness for both binary and multi-class detection. The advantages of the model include enhanced accuracy, efficient feature utilization, and reduced overfitting. However, limitations include higher training time and computational complexity compared to simpler architectures, potentially limiting deployment in real-time SDN systems. The authors suggested exploring transfer learning and real-world SDN deployment in future research to improve scalability and operational performance.

Alsharaiah et al, 2024 [178] proposed an innovative hierarchical DL-based NIDS that integrates an LSTM network with an attention mechanism to enhance the detection of malicious network activities using the UNSW-NB15 dataset. The dataset, which includes 49 features and multiple attack categories, was used without manual feature engineering—allowing the model to automatically extract essential features. Preprocessing involved data normalization, conversion of symbolic data to numeric form, and binary labeling, with clustering applied to reduce redundancy and enhance training efficiency. The proposed Attention-based LSTM (AT-LSTM) model was trained with 82,332 samples and tested on 175,341 samples. The model achieved an accuracy of 92.2%, outperforming other well-known models such as CNN-BiLSTM (77.16%), RF (75.56%), Least Square SVM (78.86%), and rule-based classifiers (57.01%). The advantages of the AT-LSTM model include its ability to capture long-term temporal dependencies, dynamically focus on important features through attention, and

eliminate the need for manual feature selection, resulting in improved precision and interpretability. However, the limitations identified were the computational cost and training time associated with deep architectures, as well as potential vulnerability to adversarial attacks in real-time environments. The authors suggested future improvements through synthetic oversampling to address class imbalance and XGBoost-based feature selection to further refine detection performance in practical network defense systems.

2.2 Phishing detection

Do et al., 2021 [179] involved classifying phishing webpage based on DNN, CNN, LSTM, and GRU algorithms and were based on an empirical investigation. The study sought to establish how well such algorithms perform in identifying phishing webpages using 30 website characteristics retrieved on the basis of URLs and the data set considered entailed 11,055 URLs (4,898 phishing and 6,157 genuine). There was much in the way of parameter tuning as the method also employed experiments varying the learning rates, the dropout rates, the architecture of the neural networks, the batch size, and the epochs to obtain the best performance of the experiments. The findings have shown that no particular DL algorithm has surpassed all the others in relation to all of the performance measures. Nonetheless, DNN and LSTM attained the greatest accuracies of 97.29 and 97.20 respectively. The shortest training and testing time but with the least accuracy of 96.56% belonged to CNN and GRU was almost the same as CNN. DNN model achieved most superior overall performance in terms of accuracy, precision, recall and F1-score. Among the limitations of the study, it is possible to note the manual and labor-dependent manipulation process of hyperparameters, which can be enhanced with the application of optimization methods. Future research outlines in the study involve using the different algorithms together to form ensemble models in order to enhance the performance and address some challenges such as computational complexity.

Mughaid et al., 2022 [180] proposed a DL model that implements a phishing detection system which applies the ML approach to determine whether an email is phishing or not. The authors trained and tested the performance of the detection models on three data, which contain 22 (3000 training samples), 50 (5000 training samples), and purely text features (2500 training samples). Classification was performed based on seven ML classification algorithms, such as Locally-Deep SVM, Boosted DT, and NN among others. The accuracies indicated that the Boosted DT and NN recorded the best accuracies at 1.00 of the second dataset with 50 features. The lowest percentage accuracy was recorded with the Decision Forest algorithm which

was 0.953. The model was more effective than its predecessors, particularly in balancing the data set among other factors, which made the model give accurate predictions. Nonetheless, the measure of predefined dataset and the necessity to constantly advance the approach to feature selection was also process limitations that hindered the conduct of the study. The suggested system was effective in phishing identification, and authors propose feature extraction and model further improvement to increase precision.

Alshingiti et al., 2023 [181] introduced a phishing detection system on the basis of DL that uses LSTM, CNN, and the hybrid of LSTM-CNN to identify phishing URLs. The objective of the study is to enhance the current phishing detection performance using the comparisons of these DL methods of classifying either phishy or legit URLs. The dataset that we will study has 80 features and 20 000 URLs. Data preprocessing was performed with SelectKBest to choose the features and then models were trained and tested on 80 and 20% data points respectively as a training and test set. The finding was that CNN performed better than LSTM and LSTM-CNN models and CNN was more accurate (99.2%) in comparison to LSTM-CNN (97.6%) and LSTM (96.8) models. CNN also performed best in terms of accuracy, recall, and F1-score and had the most accurate and efficient identification of phishing URLs. CNN model would be very useful when addressing the problem of text classification, and LSTM would be effective when learning the relationship between the sequence data. The LSTM-CNN composite model is quite good and was not able to out-compete CNN because of its improved performance speed and the number of parameters that ought to be trained, making the model simpler. The robust findings were complemented with the fact that, as observed by the authors, during the model, the state of the site, including whether it is currently active or not, is not verified which might affect precision. In future work, there will be an emphasis in enhancing feature engineering, the training process will be accelerated and the state of the site will be verified to increase the accuracy of the detection.

Adebowale et al., 2023 [182] suggested a framework of Intelligent Phishing Detection System (IPDS) integrating both LSTM and CNN to check phishing sites. The method combines the use of image and text characteristics and content features of the Web page frame to detect accurately. The algorithm learned with the help of a collection containing 1 million links and more than 10,000 images (phishing and non-phishing sites). Considering all the standards of the model, accuracy was 93.28% indicating good performance improvement in time with the average of 25 seconds compared to the previous techniques that have been used which consume 30 seconds. It is important to note that the CNN had the best time efficiency, whereas the LSTM

performed better on the tasks associated with texts. The LSTM-CNN hybrid model was the best one that took into consideration the advantages of the two architectures. The effect of cross-validation was that the Holdout method recorded the best performance and the accuracy was 93.28%. As compared to the existing previous studies such as the one by Yang et al. (2019) that demonstrated an accuracy rate of 98.99 and another by Bahnsen et al. (2017) with another accuracy rate of 98.7, the IPDS presented a lower accuracy rate but outperformed in real time performance and robustness as the multiple features (the image, the text and the frame) were used simultaneously to detect. The system also allowed identifying phishing websites in shorter periods as it targeted particular elements that were phishing URL specific.

Aldakheel et al., 2023 [183] offered a DL-based method of phishing sites identification applying a CNN model. The authors concentrate on the analysis of the Uniform Resource Locators (URLs) as one of the characteristics to differentiate between the phishing sites and their authentic versions. The approach uses a CNN based model that has been trained on the data provided by PhishTank, that includes 10,000 phishing sites and 10,000 legitimate websites. The data is performed to preprocessing such as conversion of URLs into character vectors via character embedding. Results of the experiment describe that the proposed CNN model demonstrated a high level of accuracy (98.77%) that was higher than other models, such as KNN (87.98%), SVM (94.13%), RF (94.26%), and RNN (97.4). The very low FPR and the high accuracy of the model indicate the model would be helpful in classifying phishing sites. The model of the study follows the idea of integrating DL, where not only the URL features are taken into account but also the upcoming threats that will contribute to an overall phishing detection system. The model provided promising performance even though the authors do not deny a few weaknesses related to being dependent on the PhishTank dataset and the inability to provide necessary detection capabilities in real-time. Further studies will be dedicated to better identifying more advanced phishing attacks, testing the system in security platforms, and decreasing false recognized data. Practical uses of the suggested system are browser extensions or deployment on the server side of the phishing detection in real time.

Karim et al., 2023 [184] discussed a hybrid ML phishing URL detection methodology was spoken of. They have trained different ML algorithms using a dataset containing more than 11,000 URLs comprising approximately 33 attributes scraped both phishing and genuine websites. Methodology involves preprocessing, feature selection, using canopy clustering and

also using cross-fold validation and grid search optimization. The research uses various models (Decision Tree (DT), Linear Regression (LR), NB, RF, SVM, Gradient Boosting Machine, and Hybrid model (LR+SVC+DT) incorporate both soft voting and hard voting). It was found that RF algorithm had the highest performance, 96.77% accuracy, 96.73% precision, and an F1-score of 97.12. The hybrid model (LR+SVC+DT) that involves soft and hard voting attained the accuracy of 95.23%. The strengths of such method include high rates of accuracy and an effective processing of phishing detection. Nevertheless, the authors indicate that potential further research should entail integration of list-based and ML-based techniques in order to optimize phishing URL detection systems further. The paper has also illustrated the significance of feature selection and model tailoring as well as the perfect result.

In their 2024 study, Bezerra et al, [185] presented a ML model that identifies phishing email with the collaboration of E-goi, a multi-channel marketing automation enterprise. The data pertained to more than 500,000 emails, among which the phishing emails amounted to a small percentage (less than 1%). Several ML models were tested DT, LR, support vector classification (SVC), RF, as well as neural networks and great attention was paid to the solution of the problem of the class deficit by means of the method of under-sampling. The last model is a neural network that performed over 80% accuracy in detecting the phishing emails with minimal chance of blocking the genuine emails (4.62% of blocked accounts). The findings indicated that there are certain characteristics whose nature was to a greater extent relevant in identification of phishing including email content, email address length and hour of send. The RF model performed equally as well recording 97.42 to 97.45% accuracy and precision of 99.07 to 99.47% but with a rather low recall of 49.26 and 49.90% meaning that there is a bias towards non-phishing email messages. The lowest sampling techniques were to use K-means clustering with K=4clusters with a better performance measure especially in precision and recall aspect with better improvements as the phishing ratio increased. In spite of the success, the study reported limitations of the work (on quality of data and feature extraction) that were restricted by errors in the email formatting. The possible future direction can be working on features that will enhance feature extraction, incorporating both SMOTE and GAN to address class imbalances, and training the model toward the goal of identifying more recent phishing attacks.

Shafin, 2024 [186] proposed an innovative feature selection (FS) mechanism of detecting web phishing, which uses SHapley Additive exPlanations (SHAP) and Local Interpretable

Model-agnostic Explanations (LIME) to weigh global and local feature significance. This dataset consists of 11,430 samples with 87 features which are identified in the form of URL-based features, content-based features and external features. The methodology is the application of SHAP to the global feature importance and LIME to local calibration, and the result is agglomerated to pick the most relevant features. Through the evaluation, it has been revealed that the RF and XGBoost (XGB) achieved 97.41% and 97.21% accuracy respectively after using the SLA-FS framework which outperformed the performance of the traditional FS methods of Recursive Feature Elimination (RFE) and Information Gain. RF model improved by 0.65 on its accuracy and XGB improved by 0.41. Among the primarybenefits of this framework, one can note its capabilities of increasing the level of phishing detection considerably in the meantime decreasing the period of processing by 0.38 ms/sample due to RF processing. The study however says that the KNN performs relatively lower in terms of accuracy 84.51%, despite feature selection. The limitation is that the framework is based on the reduction of features that do not necessarily account all the details on phishing attacks. It is possible that the future work of the project involves increasing the flexibility of the framework and looking into some other models and ways of developing it further.

Fares et al. 2024 [187] provided a ML-based approach to detect email phishing successfully, with the implementation of such models as SVM, RF, and XGBoost. The data used in this paper was found on Kaggle, in terms of email text and email type (Phishing or Safe) features. The data are preprocessed to clean data, process the text (tokenizing, removing stopwords, stemming), vectorize them (TF-IDF and BoW formats), and encode the labels. As determined by the experimental results, the SVM model recorded the best accuracy of 97.61%, giving more accuracy that the RF (95.04%) and XGBoost (96.61%). The paper has shown that SVM performed best regarding the values of classification measures (precision, recall, and F1-score) in addition to execution time, and it provides a stable and efficient way of extracting phishing emails. Nevertheless, the research also states that better accuracy in detection could be achieved by enlarging the models with the help of ensemble approaches. The shortcomings involve the use of a single dataset that might not reflect the changing phishing tactics and the necessity to apply more feature selection techniques in order to make the accuracy rate even higher.

Kavya and Sumathi 2024 [188] discussed the recent innovations in phishing detection with the microscope put under the developments and new methods, including ML-based, DL,

Generative Adversary Networks (GANs), and ensemble. List-based techniques, similarity-based approaches, ML techniques, and DL models are some of the methods examined by the authors. One of the most significant results is that CNNs and RNNs are effective in phishing detection since they demonstrate high accuracy (up to 98.74%) and low FPR in comparison with other traditional ML models. In addition, GAN-based models such as PDGAN have promising potential as well, with greater robustness to adversarial attacks and eliminating data imbalance due to generation of synthetic phishing samples. PDGAN model in particular showed an accuracy of 97.58%, being excellent in the case of zero-day phishing attacks. Nevertheless, GANs and DL models have a high computational price and difficulties in applying them in real-time. The paper identifies the need to further research concerning contextual analysis, adversarial training, and model interpretability to enhance phishing detection in the future. Also, integrating explainable AI (XAI) and federated learning is proposed in the article to improve privacy preserving detection systems. The study points out that there are shortcomings in robustness, diversity of the data, and wide-ranging applicability, which indicates that it is necessary to do more research to bring such techniques to real-life settings.

Gupta et al. 2024 [189] introduced the model of phishing detection that involved combining RNN and Whale Optimization Algorithm (WOA). The used data is a set of more than 11,000 website URLs that can be marked as phishing or legitimate, and 30 different features of such websites. The WOA tuning of the hyperparameters is applied to the RNN and leads to better results of model performance. It has produced a report of an overall accuracy as 92 with a value of precision as 0.92 on phishing and non-phishing websites, 0.94 on a phishing site and 0.90 on legitimate websites. F1-doc on phishing sites was 0.93 which means the detection ability of phishing websites is truly balanced. The model obtained high true positive rate 94% and 90.2% on phishing and legit-websites although some misclassifications were obtained. The amount of the RNN parameters and the iterative search population algorithm of WOA affect the computational complexity of the model. The proposed model performs better than other conventional models such as the LR, SVM and DT, particularly the ability to work with non-linear and complex patterns in the phishing URL. By evaluating the ability of obtaining an optimal solution using RNN, and WOA combination in phishing attack detection, the study establishes a record of future research in cybersecurity.

Barik et al., 2025 [190] offered a phishing URL detection model as the optimization of DL in a web-based structure. The cited methodology combines CNN and Enhanced Grid Search

Optimization (EGSO) methodology to enhance model performance. This data comprises phishing and legitimate URLs which were retrieved using the following sources, Kaggle, Alexa and PhishStorm, and contains 27,534 samples (13,767 phishing URLs and 13,767 legitimate URLs). The model attained a remarkable overall accuracy of 99.44%, precision of 99.98%, a recall score of 99.21 and an F1-score of 99.32. Its validity was tested with a secondary dataset and the results were quite comparable (99.40% accuracy), which indicates the reliability of the model applied. Detection of this technique has its strengths as its accuracy is high, false-positive is low, and it can be used in real-life situations like integrating web browsers to detect phishing content. Nevertheless, there were also shortcomings found in the study as well, such as the failure to find sites that utilize embedded objects, such as Flash or JavaScript, and the need to consistently update the data sets used in the model without which it would not be possible to predict phishing sites due to the ever-changing nature of that methodology. In further studies, the area may look into integrated learnings through transfer and extending the model with more options in phishing techniques.

Rao et al. 2025 [191] introduced a super learner ensemble fine-tuned to phishing URLs in mobile devices. The approaching consists in extracting the mixture of handcrafted features, DL features, and transformer-based embeddings to categorize the URLs. The data set employed in experiments consists of 190,062 legitimate URLs and 141,490 phishing URLs and it was gathered with the use of PhishTank and OpenPhish to collect the phishing sites and the Google search engine to collect the legitimate sites. The model was able to demonstrate an excellent performance rate with the 98.93 accuracy, 99.15 precision, 99.07 F1-score, and 97.81 MCC with the super learner ensemble, which are considerably higher than that of conventional ML and DL systems such as XGBoost and SVM. Within the scope of the study, the following benefits of the model are noted, their ability to be operated on resource-limited mobility devices, and the possibility to ensure real-time detection, the average time to response of which was 480ms. The major advantage of the model is that it is able to integrate other ML algorithms and can be made more effective by taking advantage of the distinct behavioral characteristic of individual classifiers. The limitations however are; the difficulty in identifying shortened URLs and obfuscated domains and that the model at the moment is English-language URLs based. In future, development will be conducted to broaden the ability of the system to work in a multilingual and more intricate URL structures.

Alsaidi et al. 2025 [192] introduced the HawkPhish-DNN cybersecurity model that hybridized Harris Hawks Optimization (HHO) and DNN into managing phishing URLs to

increase phishing URL detection. Its model incorporates complex methods of feature extraction (entropy extraction and TF-IDF vectorization) that extract important features of URLs, and the problem of class imbalance is solved by means of SMOTE (Synthetic Minority Over-sampling Technique). So, the used dataset contains phishing data collected by PhishTank and OpenPhish and benign URLs from Alexa. Multi-objective HHO was applied to optimize the model in order to balance the accuracy of detection and eliminate false positives. In the results, it can be observed that HawkPhish-DNN was able to generate a false positive rate that is as low as 0.2, whilst also returning an accuracy of 99.6%. Also, it scores higher than the classic models of DNN, improving precision (99.8%), recall (99.4%), and F1-score (99.6%). This model can withstand such advanced schemes of phishing and has low-cost computing overhead, rendering it applicable in real-time applications. The study recognizes, however, limitations, including the difficulty of catching such phishing websites with the use of more sophisticated techniques of obfuscation, such as shortened URLs or domain generation algorithms. Further research will concentrate on making the model more versatile to novel phishing processes and increasing its implementability to other forms of cyber threats.

Garki et al., 2024 [193] discussed recent challenges of phishing attacks in the environment of Open and Distance Learning (ODL) by offering predictive ML-based framework. In the paper, it is pointed out that technical countermeasures (e.g. encryption, firewalls) are not enough to counter phishing threats based on social engineering. In order to obtain such humanized vulnerability, the authors provide the simulation of real phishing situations in ODL systems to obtain user behavioral information. The paper decides to use the given dataset to compare the work of three classifiers, including RF, DT, and RNN. The models will be evaluated by precision, recall, F1-score, and accuracy. The findings are that RF has a good degree of high point recall and F1-point (very much at 0.97), and RNN has a good precision (1.0) and accuracy (0.98), hence making it very successful in the detection of phishing behaviors with a very low false positive detection. The uniqueness of the work is the practical behavior-based method of detection, as well as the emphasis on the awareness of users and the prediction of their behavior as an important aspect of preventing phishing in ODL. It can serve as a basis to incorporate intelligent phishing detection models into the educational approaches to cyber-security, with a particular focus on proactive and user-centered defense tools and methods appropriate in the distance-learning environment.

A solution by Yoo et Cho 2022 [194] presented an Intelligent Chatbot Security Assistant (ICSA) to help in SNS phishing attacks with the Text-CNN used to detect the stages of the attack. DNACST includes the development of a chatbot analyzing phishing stages in chatrooms of SNSs in real-time to provide corresponding protection measures to victims. It has been built with actual phishing examples and unattacked sentences, and experiments revealed that Text-CNN performed better than LSTM in accuracy during training and testing. Findings showed the real-time capability of SNS Defense of ICSA. The merits of this method are that it involves proactive protection, and the weak points consist of the necessity to keep the dataset constantly updated and the more sophisticated implementation of the multi-sentence translator algorithm. In the future, the datasets will be improved, models will be made more accurate, and they should be retrained in real time. The implementation of the study through Google Dialogflow and the Telegram Chatbot on AWS is proved to be efficient due to its large experimentation.

Alazaidah et al., 2024 [195] tested how to identify phishing sites using ML models. The paper seeks to determine the most effective classifier and feature selection technique to use in enhancing performance in phishing. The authors consider the entire set of 24 different classifiers provided by 6 strategies of learning on 2 phishing datasets: Dataset1 (binary ML, 11055 instances, 30 predictors) and Dataset2 (multiclass, 1353 instances, 9 predictors). The classification accuracy is measured by a number of evaluation measures such as accuracy, precision, recall, F1-score, and ROC area in the case of the classifiers. The outcome indicated that RF had performed the best with regard to Dataset2 being the most accurate (97.26%) and having the best true positive rate (0.973). The best accuracy and precision were obtained by Filtered Classifier and J-48 on Dataset1. In terms of feature selection, the accuracy (96.58%) provided by InfoGainAttributeEval was better compared to any other feature selection method, and the precision and recall were the best with 0.966 standard. The study emphasizes that RF is a good choice of dataset, especially when the features are more complex and multiple classes are to be integrated, whereas FilteredClassifier and J-48 are more suitable when datasets are less complex and binary. The primary benefit of the research is that it critically examines the classifier and feature selection techniques of detecting phishing, which can be utilized to improve security. The limitations of the study are, however, the availability of a single dataset in some of the classifiers and the need to test other phishing data in order to scale up the range of generalizability.

Sahingoz et al., 2024 [196] proposed the architecture of a system on the basis of DL to identify phishing websites. The logic behind is training the system with a large dataset with the help of five DL algorithm, including ANN, CNN, RNN, Bidirectional RNNs (BRNN) and Attention Networks (ATT). They gathered a data set consisting of about 5.1 million URLs (2.3 million phishing URLs 2.8 million legitimate URLs). It is also based on a character level embedding of vectorization and can efficiently classify the URLs. The experimental outcomes reveal that CNN model performed best with 98.74% detection accuracy, thus it is the best model. It has a speed of classifying more than 130,000 URLs within a second, thus having the provision of a quick real-time phishing detection.

The research study of Trad et al., 2024 [197] explored the ability of large language models (LLMs) used to identify phishing URLs. They contrasted two procedures prompt engineering and fine-tuning. The research has a phishing URLs dataset consisting of 1000 URLs (500 phishing and 500 legitimate), and these two methods are applied to the models such as GPT-3.5-turbo and Claude 2. In the results, prompt engineering using Claude 2 demonstrated an F1-score of nearly 93%, compared to GPT-3.5-turbo (F1-score of 78%) in the various prompts (zero-shot, role-playing and chain-of-thought). Nevertheless, when the authors fine-tuned base models such as GPT-2 and DistilGPT-2, the results got considerably better. An F1-score of 97.29 % is the highest performance with GPT-2 (medium) and is even better than the current phishing detection practices on the market. There was higher AUC (99.56%) in the fine-tuned models than in prompt-engineered models as well. The paper ends by concluding that prompt engineering is less resource demanding and rapid to implement but fine-tuning gives a better output of phishing detection, especially in a real-life setting phishing URLs are scarcer compared to real ones. The study points out that even though the fine-tuned models require more resources than the general-purpose LLMs, it is much more practically effective as a task-specific solution to detecting phishing. Data privacy and model maintenance is another pursuit in the research, with it mentioned that fine-tuned models provide greater security due to the ability to process locally as compared to prompt-engineered models using third party servers. Comprehensively, fine-tuning was indicated as the better way to detect phishing, as opposed to the state-of-the-art methods.

Borra et al., 2024 [198] introduced a strong classification system of the emails that aspires to detect a spam, phishing and ham email. The given OEC-Net methodology would use a combination of datasets (UCI, CSDMC and SpamAssassin) with a hybrid strategy of Principal Component Analysis (PCA) feature extraction, Particle Swarm Optimization (PSO) feature

selection, and Deep Learning Convolutional Neural Network (DLCNN) to classify. That non-traditional system demonstrates high performance in relations to classical methods such as LR, SVM, NB and reaches such results as 98.43 accuracy, 97.78 precision, 96.41 recall, and the F1-score of 97.07. Using PCA and PSO, feature extraction and selection become optimized and, therefore, result in improved classification efficiency and classification accuracy. Although the model has a success, its demand on considerable computational resources and being a black-box result in identifying such limitations. The OEC-Net is the complete solution to advance the security of emails, as it proves to be beneficial in the categorization of spam and phishing emails.

Jayaraj et al., 2024 [199] suggested the ML method of detecting phishing and authorized interventions in computer systems. The approach proposes a Hybrid Ensemble Feature Selection (HEFS) approach that involved Cumulative Distribution Function gradient (CDF-g) algorithm in generating primary feature subsets, and data perturbation ensemble in generation of secondary feature subsets. Such features are then categorized based on the ML techniques like, DT, RF, k-NN, and Neural Networks. Based on the findings of the study, the proposed model yields the maximum accurateness of 97.8, sensitivity of 98.2 and specificity of 98.17. This model has higher performance in comparison to other typical approaches, such as DT and RF. The stated phishing detection infrastructure proves to be computationally efficient and it can easily be implemented on resource-constrained devices such as routers to provide real-time detection of phishing websites. Nevertheless, the study also states the drawbacks that exist with the classic list-based phishing detection systems illustrating that these systems should evolve to select new features so that to be able to address the dynamic phishing strategies. The authors propose CDF-g algorithm as desirable and flexible approach to address these shortcomings in modifying existing approaches in feature selection.

The study by AL-Ahmadi et al., 2022 [200] offered a PDGAN (Phishing Detection with Generative Adversarial Networks) model of phishing detection. This model makes use of GANs (in this case LSTM networks to generate synthetic phishing URLs and CNN as a discriminator to tell whether the URL is phishing or legitimate. As compared to more typical methodologies that utilize content served on a webpage or even third-party services, PDGAN only requires a URL of a given website achieving 97.58% accuracy and 98.02% precision on a dataset of almost 2 million phishing and legitimate addresses sourced upstream resources such as PhishTank and DomCop. The model shows better performance than the state-of-the-art models. The structure of PDGAN architecture is applied to the advantageous identification of phishing

links that are generated by LSTM and classified by CNN as a part of the adversarial process. This method demonstrates that there is a substantial decrease in the false positive rate in comparison to the initial value reaching an amazing 2.1% FPR. The paper also reflects the ability of PDGAN to enhance its detection by examining new phishing URL pattern which was not included in the training set. These findings imply that the proposed PDGAN is effective and efficient in detecting phishing websites and can outperform the traditional and the recently developed DL models by using the URLs only. Nevertheless, not all the URLs generated are perfect in terms of semantically correct information and this is a weakness that can be improved in the subsequent revisions.

Orunsolu et al., 2022 [201] introduced a powerful solution in the form of an anti-phishing ML system. The task of feature selection module is applied in this system by extracting appropriate features of URLs, web properties and web behavior to create an effective feature vector. The method employs the SVM as well as NB to classify the presence of phishing websites which are trained in a 15-dimension feature space. The results indicate experimental dataset, comprising of 2541 phishing URLs and 2500 benign URLs that achieved 99.96 accuracy and <0.04 % FP occur reported with 10-fold cross-validation. The major benefit of the system is that it identifies the discriminative features in the available phishing datasets, an aspect that significantly enhances the efficiency of finding the phishing protocols since one is not required to conduct an endless production of new features. The model provides robustness and the high computational cost effectiveness with incremental build-up of sets of features. The weak points are the complex phishing attacks such as websites clones and the reliance on some of the URL characteristics. The omitted response to these limitations considers future work where higher-end methods like DNS query validation will be used and language independence in phishing detection will be considered. The system shall be implemented as middleware in order to overcome the problem of deploying the system as a client-server deployment and enhance scalability.

Recently, a paper by Boussougou et al., 2025 [202] introduced the vishing-detection problem and proposed the use of ML and DL models to detect vishing. The selected task of the problem to consider is to improve performance of phishing detection models in terms of their ability to identify imbalanced dataset, based on the two strategies applied to the data: back-translation (BT) and SMOTE. The dataset that is used by the authors is KorCCVi v2 a Korean voice phishing dataset and they suggest augmenting of the same with the use of back-translation and use of English, Chinese, and Japanese as intermediate languages. The research concludes that

models trained with BT-augmented dataset yielded higher results than those trained and tested on the original imbalanced dataset, resulting in F1-scores of 98.91 when comparing BT against SMOTE of 97.23, representing an improved detection of phishing attempts. The BT approach maintains contextual and linguistic precision, and it brings variety into the dataset, whereas SMOTE is more likely to increase the recall and cause noise. The research, however, accepts limitations, e.g., computing costs, and possible overfitting when examining some of the models, which means that it is recommended that intermediate languages should be selected carefully in the BT process. Based on the research findings, the multilingual BT strategy can be considered as a better method that would improve voice phishing schemes.

Chinta et al., 2025 [203] implemented a complex ML system on phishing email detection. This paper capitalizes on several ML models, such as CNN, XGBoost, RNN, and SVM and designs an effective solution to detect phishing emails. The authors relied on large size phishing email with both phishing and legitimate emails and began to use rich features engineering strategy like tokenization, HTML tags, elimination of points, and exclusion of stop words to bring forth top-notch performances of models. In the findings of the proposed study, it has been evident that the BERT-LSTM combination model has produced excellent outcomes with the F1-score of 99.24, accuracy of 99.55, precision of 99.61, and recall of 99.55 that surpass the conventional models such as NB, RNN, and SVM. BERT-LSTM model had excellent generalization capabilities and little overfitting, and it is a potentially useful method in real-time email phishing detection. Nevertheless, the study also observes some difficulties provoked by a reliance on computational resources and the necessity of big labeled sets of data in terms of learning and the possibility of regular updates based on changing phishing techniques. These results demonstrate the methodological suitability of the BERT-LSTM model in phishing-email detection and its suitability to practical application.

Nayak et al., 2025 [204] examined the topic of detection of phishing with the help of ML and DL models. The dataset used in the research consists of 58,645 URLs that comprise 111 features that are used to differentiate between phishing websites and legitimate websites. The research focuses on feature selection, hyperparameter tuning and DL (Feedforward Neural Network (FNN), DNN, Wide and Deep, and TabNet) enhancement of phishing identification. It has been identified that the FNN model displayed the best accuracy with 94.46 accuracy on 14 chosen features. A grid search of the hyperparameters was applied to train the model so that it could perform optimally. The researchers also introduce a new measure that takes into

account factors other than accuracy which includes the anti-phishing score that measures time consumption, false positives, and false negatives. Based on the results, it can be assumed that feature selection with the help of the Permutation Importance technique contributes greatly to the efficiency and robustness of the model. The incorporation of these models in real-life phishing detection environment also forms part of the discussion in this study and the opportunity that the Feedforward Neural Networks may present the application into practical use especially in real-time program which must trade precision against minimal load is mentioned. The study proposes the further enhancement and its interaction with higher-level cybersecurity platforms, with the purpose of countering the changeability of phishing attacks.

Taofeek, 2024 [205] introduced a new method to detect phishing sites that is real-time based on ML. The article brings out the shortcomings of the traditional methods of phishing detection which usually requires the user to report an emerging threat delaying a system against a new phishing attack. The proposed method obviates interaction with a user because it involves an automated process of detection that ensures quicker and better outcomes. This method utilizes the idea of ML models along with binary visualization methods to look at the URL and find phishing sites. This is done by scraping the HTML code of the websites, representing the same as 2D images using a script called BinVis and then inputting the same into a ML model constructed on TensorFlow to be classified. The study proved the use of the proposed model was found to be of high accuracy in the detection and the final evaluation results indicated an overall accurate detection of 94.16%, precision rate of 95.83%, and the recall rate of 87.50%. These findings indicate that the model has the potential of identifying legitimate and phishing websites. By using binary visualization (BinVis) transformation, conversion of the data on websites into the visual representation has been achieved, which demonstrates easier classification of websites according to their structural characteristics by the ML algorithms. It was the XGBoost classifier that was confirmed as the most effective as compared to other models such as SVM, RF, and MLP regarding attitudes and accuracies. Future research involves the addition of more training examples to the system, the use of GPUs to speed up calculations and improve the model against a wider variety of phishing attacks including those that are not English-based.

2.3 Cheating detection

Hassan Hosny et al., 2022 [206] introduced an intelligent system that measures fair assessment in online laboratory based on the behavior of how students interact with their mouse. The data to be analyzed is the data obtained on the students taking laboratory exercises and

simultaneously passing online exams with the aim of identifying the signs of cheating. The approach is to examine the mouse activity, including the mouse movement, the button state and coordinates, to develop a cheat pattern. The mouse behavior of students was classified with the good results using different AI algorithms, such as KNN, SVC, RF, LR, XGBoost, and LightGBM which achieved the greatest accuracy since it reported an accuracy of 90%, precision of 88% and separation of 95% between the cheaters and non-cheaters. The principal merit of this system lies in its affordable cost since they only need the mouse dynamics data but not any other resource such as the cameras. Nevertheless, the study also features the limitations, including the possible influences of the emotional conditions of the students or the various screen resolutions that might result in the behavior change observed in the mouse dynamics. These limitations involve the proposed future research to improve the performance of the system.

Siddiqui et al., 2022 [207] investigated the use of mouse dynamics as a behavioral biometric to authenticate users continuously. The authors measured their sample of 40 participants who executed a task of high intensity during a 20-minute exercise. Three binary and multi-class classification problems were tried with three ML algorithms and three DL models. The single best model with the highest performance was a 1-dimensional Convolutional Neural Network (1D-CNN), which had an average 85.73% accuracy in the classification procedure. With multi-class classification, an ANN attained peak accuracy of 92.48%, which is the highest that was observed with this data input. The experiment shows that DL architecture, specifically ANN, can demonstrate a better performance in relation to more classical ML models such as SVM and KNN, with ANN leaving other methods far behind in terms of accuracy and F1- score. Nevertheless, there are also some limitations that the study recognizes, such as the employment of relatively simpler DL designs and the necessity of additional bigger batches of data to create stronger models. It is possible to conduct further research by perfecting these models, implementing more intricate DL architectures, and testing the impact of data processing protocols. The findings propose that mouse dynamics might be an unaffirmative and non-impeding approach to persistent authentication.

The research study by Valizadeh, 2022 [208] investigated how students of Turkish higher education perceive online course cheating in terms of methods and motifs and the possible countermeasures to it. The research relied on the mixed-methods design, i.e., the combination of quantitative (based on the Likert-scale questions) and qualitative (based on the open-ended

questions) data collected among 163 online learners. It was found that over half of the respondents felt that it is more common and more effortless to cheat in an online course than it is in a traditional classroom. Most widespread forms of cheating involved Googling or using some other online source to find the answers, asking other people, and referring to personal notes or the coursebooks. Reasons given as to why people cheat were due to lack of knowledge, technical difficulties, stress during exams, and no proctoring. Some of the solutions given by the respondents included deactivating copy-pasting ability, a mandatory use of webcams, and development of various quizzes per student to mitigate the use of cheating. Not all students were convinced though; some students doubted the ability of the system due to technical restrictions as well as the inability of this system to proctor appropriately. With these challenges the study entails recommendations on how to approach them, which should include better proctoring systems, reduction of the course syllabus, and other forms of assessment like homework or essay-based exams.

Zhou et Jiao, 2022 [209] analyzed data augmentation in ML to detect cheating of large-scale assessment. The research indicates that dilemma of ensemble learning style and extension of it with anomaly detecting and feature selecting technique may enhance a detection performance. The data that the study will be based on is the one gathered in a large-scale licensure test, the data in which include 1636 test-takers, 46 of whom were flagged as potential cheaters. The methods of outlier detections such as Isolation Forest, Elliptic Envelope, or DBSCAN were used to produce scores of a set of outliers which were concatenated with the input data of the meta-model. These outliers scores were included in the meta-model development along with the most useful features of the original dataset which have been determined through application of such methods as Chi-square, Pearson Correlation, and Recursive Feature Elimination (RFE). The findings revealed that the meta-model with the performance that was the most significant (blending with two forms of data augmentation) had its accuracy, precision, and recall increased significantly, and the value of the F1-score equaled 0.7857. This was better in comparison with other base models such as the GB and RF. The researchers find that data augmentation through anomaly detection and feature selection is a promising method to significantly increase the efficiency of the models used to detect cheating on large-scale tests, providing a strong solution to data imbalance and serves to improve the results of predictor ship.

Duhaim et al., 2022 [210] investigated the problem of verifying cheating in online tests to combat the Covid-19 pandemic on the basis of the data mining methodology. The system is

implemented on the Moodle platform with three layers; the first one is based on statistical analysis of the data available which will be using the IP address, time taken and time late; the second layer utilizes the similarity measures to compare the answers of the subjects given by the students; the third layer uses clustering algorithms (K-Means, EM and Hierarchical clustering) to group the students in accordance with the similarity of the answers given. As the results indicate, the best performance of K-Means clustering has been found, and the overlap similarity algorithm comparably worked best when dealing with the essay responses. Duplicate IP address, odd time signature, and similarity in responses were some of the IP parameters that the system was used to identify cheating. An example of this is, 16 students were found to be highly similar in regard to their answers on their essays, and they were clusterized by their similarity in their answers on their essay. The research emphasizes the usefulness of such modes of improving fairness and integrity in online tests. Nevertheless, the system will not be accurate enough as student behavior can be inconsistent, and more enhancement will be required to monitor it in real time.

Alsabhan, 2023 [211] introduced a method of ML that would count cheating cases in higher education examination based on LSTM networks. The methodology takes into consideration the 7WiseUp behavior dataset that incorporates the data concerning the attendance of the students, academic performance and data regarding the behavior. The data is used to establish the behavior of cheating depending on the performance of students in their exams and grade patterns. The model attained a higher accuracy of 90% along with LSTM with dropout layers, dense layers, and Adam optimizer compared to previous methods. The results obtained in the study point out to the fact that the optimized architecture and hyperparameters, as well as the clean dataset, led to the improved accuracy level. On its design, the model adopts factors like the grades given to students during various examination periods and defines that as normal or cheating. To produce synthetic datasets to simulate real world test fraud scenarios, the 7WiseUp dataset was employed and four synthetic datasets and a single real-world dataset were involved in the experimentation. The synthetic datasets include various degrees of challenge when it comes to identifying the existence of cheating through discrepancies in the test scores. Nonetheless, along with having the success associated with the model, the research does present a few weaknesses, such as involving only one dataset and the difficulty in identifying more advanced rules of cheating behavior. Additional models and more importantly considering a few datasets should be tackled in the future.

The problem of cheating in online exams as presented by El Rhezzali et al., 2023 [212] gained importance especially in the online educational situation such as e-learning, which has increased exponentially during the COVID-19 pandemic. The authors suggest a three-layered model of cheating identification such as continuous authentication, and online proctoring. The first is a facial recognition and authentication layer and the second is a behavioral one (e.g., action on abnormal mouse motions, eye gaze and suspicious activities via a webcam). This system uses ML to evaluate the behavior of the students during examinations, seeking patterns of abnormal behavior that code as cheating and include IP addresses anomalies, non-standard timing patterns, etc. Different methods of cheating such as collaboration, the use of prohibited devices, or Internet search are also outlined in the study. The authors suggest that pairing real-time proctoring with analysis performed after the exam (detecting answer similarity) creates an opportunity to produce precises the detection of cheating. Their proposed architecture is a merge of remote proctoring, video streaming, and answer comparison which increases the accuracy of detection. Although the system offers a strong solution, the authors have identified the weakness during the technical issues, like the failure in internet connectivity that could affect the accuracy of detecting the cheating. The future study may seek to increase the reliability of the system and to fit it to suit different online exam platforms.

Moreno-Marcos et al., 2023 [213] developed Statoodle, a learning analytics tool to be used to analyze the actions of Moodle students and assist to identify a possible way of cheating during distant examinations. Statoodle is an engine that converts Moodle data such as quiz reports, log reports and grade reports into detailed PDF and Excel reports on activity within the software by students. Among its most important properties, the tool evaluates the Item Response Theory (IRT) to determine the difficulty of questions on the tests, the efficiency of the students to rapidly find solutions to the tasks, and detect malpractices such as possible attempts at cheating. It is a tool that validates logs in order to identify the unauthorized access of materials during exams like changing the IP address or actions involving resources not related to the exam quiz being taken at that time. This system has been experimented in two courses where it has been able to detect cases of fraud in exam as well as report these cases to the instructors. The tool does have limitations though, which are sometimes false positives (e.g. IP address change that can be likely attributed to changing adjacent networks), and cannot identify out of platform activities, such as offline cheating. Nevertheless, despite the above

limitations, Statoodle can be useful to the instructors as it offers a glimpse into the behaviors of the students and contributes to the early identification of the cheater.

Volarić et al., 2023 [214] aimed at identifying academic cheating during online tests, especially during the COVID-19 pandemic by using the ML-based approach. The participants in the experiment were 54 students who took two tests online separated by time (first and final tests) and all were enrolled in computer science courses during which some gained advice to act as cheaters with different means (mobile devices, same browser, and different browsers). In carrying out the research, 500,000 uncollated logs of student activities were obtained via a Student Activity Tracker Chrome extension, which tracked mouse movements, keyboard clicks and inactivity periods among other activities that may suggest the possibility of cheating. These data were then trained on four ML models that included LR, DT, RF, and k-NN. The outcomes indicated that the models did well in the accuracy with the logistic regression being the most accurate at 96.67%. In the study, it has been established that suspicious activities such as unusual waste of time and toggling of browser are associated with cheating behaviors. The authors, however, state that the sample required to perform this experiment is small, and the surrounding environment was uncontrolled in some way, which makes the research hard to generalize, and they propose to increase the number of members of the experiment group and diversify it. The next line of research will be devoted to incorporating DNNs and GAN in the detection model.

Ong et al., 2023 [215] proposed a method of detecting cheating based on clustering when taking online exams. The system used a web camera in tracking and monitoring student behaviors concerning their facial expressions, their body gestures, and their eyeballs movements in order to detect the cheating. The data in this study was synthesized by creating 200 video recordings of 50 subjects performing different acts of cheating, including use of smartphone, head and eye movements, and blocking off the eyes with a hand. These videos were further analyzed by a clustering mechanism that was able to group the cheating behaviors. Three clustering algorithms such as Self-Organizing Maps (SOM), Fuzzy C-Means and Time Series K-Means were tested where the best accuracy of 83.6% was attained due to Time Series K-Means. Similar patterns of behavior can be clustered to achieve effectiveness in detection of cheating behavior, i.e., looking at a phone, or eye blocking. The finding of the detection results in the system reveals its ability to detect and identify a behavior such as eye motion, face position, and phone usage. According to the findings of the study, clustering-based approach can be the most effective in identifying cheating behaviors without assuming labels, thus being

flexible when it comes to new methods of cheating. The research cites limitations, though, such as possible classification errors of unobtrusive behavior and failure to detect activities beyond the web browser. In the future, one should expect an increase in the dataset, enhancements in clustering, and addition of more features.

With the help of skeleton recognition and ML models, Tran et al., 2023 [216] proposed one of the most recent methods of cheating detection during classroom examinations. In the methodology it is used that the YOLO-Pose detects the skeletal key points in the video recording and then it is classified with the help of ML algorithm such as SVM, DT, RF, XGBoost, and Ac-LSTM (a specialized Long Short-Term Memory network). The simulated dataset was made up of paper-based tests, in which 6 videos comprised of different angles of cameras in recordings of 12 participants. The behaviors recorded where the act of cheating whereby students were caught sharing and exchanging papers during their exam, copying of answers, and peeping at other papers. The Ac-LSTM model performed second best after the XGBoost model with accuracy, recall, and F1-score 95, 90 and 92, respectively, beating other models like SVM and DT. The earlier skeleton key points (18 key points) were used to detect fine actions and computations are simplified. The results of the study indicate that the system proposed is very viable to be used as a real-time cheating detection system in exam scenarios, and this stands as a cost effective and efficient scalable solution. Nevertheless, it has shortcomings such as false positive in situations of a student pondering or a small rotational head movement. The current challenges and future efforts towards accuracy shall be improved.

Zhao et al., 2023 [217] studied academic cheating in elementary school children by examining self-reported cheating behaviors, utilizing ML to evaluate attitudes toward cheating. The sample consisted of 2,094 students ranging from 2nd to 6th grade in China. The study aimed to identify various predictors of cheating, such as children's attitudes toward cheating, the observation of peers cheating, and their socioeconomic status. The study employed several ML algorithms, including RF, LR, XGBoost, and MLP. Among these, RF proved to be the most accurate, achieving an accuracy of 81.43% and a high AUC of 80.29% when predicting cheating behaviors. Key predictors included children's beliefs about whether cheating is acceptable (7.07% contribution), the frequency of peer cheating, and the observed prevalence of cheating in the school system. Additionally, factors like school type (with lower socioeconomic status correlating with higher cheating rates) and academic achievement also played a role. The study further confirms that the factors influencing cheating in older students

apply to younger children and suggests strategies to address academic integrity at early educational stages. The results also demonstrate the effectiveness of ML in analyzing developmental data and provide insights into cheating behavior among young students.

Gopane et al., 2024 [218] introduced an AI system to check cheating during online testing, where they minimize human proctoring. It is a system, which is the hybrid of FaceNet to recognize the face and the Lucas-Kanade algorithm to track the head-related movement, and the Active Appearance Model (AAM) will be used to sense eye gaze and keep track of the motion. The approach entails recording a video feed of a student on an exam through his or her webcam, which would then be processed using detection of cheating behaviors including abnormal head turns, eye tracks, and no motion (indicating that a fixed image may be used to trick the system). Video data of more than 1000 online exams was used to train and test the system, which indicated an F-score of 0.94 corresponding to a great accuracy in the identification of cheating behaviors. The advantages of the proposed system over the baseline methods included its effectiveness in the detection of the cheating regardless of such factors as occlusions, different angles, and different lighting features. Nonetheless, the system is dependent on the input of webcam and it is possible that in future they can include auditory monitoring and data mining in order to make it even more effective in detecting cheater. Results indicate that online proctoring of exams is a possible idea that can be implemented through low cost, efficient and autonomous systems.

Deliberating on the paper exams paper in classroom, Abeba et al., 2024 [219] discussed the DL approach to detection of cheating in such exams. This study offers a CNN design to predict a two-class label, which involves cheating or non-cheating on the images depicting the exam sheet. The data set would be composed of 1236 pictures of cheating trials and 1565 non-cheating pictures of the Woldia University and other sources on the internet. This model was trained with augmented data, giving a total 5286 images which was processed under strategies such as image resizing, conversion to grayscale, and reduction of noise to enhance the accuracy. The CNN model was able to achieve an impressive accuracy of 92% on the larger images, (resized to 300x300 pixels) which are better with baseline methods. The paper ends with the following focus on the necessity of dataset enlargement and additional investigations into incorporating additional perspectives on the types of cheating into increasing the performance of the model.

3. Comparative Study

This part of the paper shows the comparative research analysis of network intrusion detection research, phishing research, and cheating research to emphasize the advancement and application of ML and DL methods in various aspects of cybersecurity.

Authors	Algorithm	Descrption	Dataset	Advantages	Limitations
Jing et Chen, 2019 [170]	SVM	Focuses on improving intrusion detection using SVM with a nonlinear scaling method. Evaluates the performance of binary and multi-classification.	UNSW-NB15 dataset	- Improved Accuracy: 85.99% for binary classification - Robustness: Better than traditional methods in handling complex attack types - Low FPR: Particularly for multi-class classification	- Computational Cost: SVM with nonlinear scaling may require more time for training - Class Imbalance: Some attack classes may have skewed results
Tian et al., 2020 [171]	DBN	Introduces an improved DBN with combined sparsity penalty terms to enhance feature extraction and reduce overfitting in network IDS.	NSL-KDD, UNSW-NB15 datasets	- Improved Classification Accuracy: 96.17% (NSL-KDD) and 86.49% (UNSW-NB15) - Low FPR: Effective for handling imbalanced datasets - Robustness: Suitable for modern and sophisticated network attacks	- Complexity: Requires careful parameter tuning and could be computationally intensive - Imbalanced Class Distribution: Minor attack classes may perform poorly in F-measure
Almiani et al, 2020 [172]	RNN	The paper presents a deep RNN model designed to detect IoT-related intrusions, leveraging recursive neural networks in a multi-layered approach.	NSL-KDD dataset	- High Detection Accuracy: 94.27% for DoS, 97.36% for Probe attacks - Low FPR: Capable of detecting DoS attacks with 0% FPR - Real-Time Detection: Optimized for IoT environments	- High Computational Overhead: Requires extensive computational resources for training - Imbalanced Dataset: Issues with rare attack classes like R2L and U2R, addressed by oversampling
Ashiku et Dagli, 2021 [173]	CNN	Proposes a DL-based NIDS using CNN and a semi-dynamic hyperparameter optimization approach for network attack detection	UNSW-NB15 dataset	- High Accuracy: Achieved up to 95.6% for the testing dataset - DL Architecture: Utilizes CNN and MLP for improved detection - Optimized for Real-World Network Traffic	- Class Imbalance: Significant challenge with underrepresented attack classes - Computational Overhead: Requires GPU for optimization and processing
Al-Emadi et al., 2020 [174]	CNN, RNN (LSTM, GRU)	Proposes a DL-based NIDS using CNN for feature extraction and RNN (LSTM/GRU) for sequence-based attack detection.	NSL-KDD dataset	- High Precision: CNN achieved 100% precision - Superior Recall: RNN-LSTM and RNN-GRU performed well on recall - Effective with Large Data	- Computational Cost: High training time and resource demand - Limited Dataset Use: Focused on NSL-KDD, other datasets could be explored
Mebawondu et al, 2020 [175]	ANN and MLP	Proposes a lightweight NIDS using feature selection (Gain Ratio) and MLP ANN to detect network attacks, including novel attacks.	UNSW-NB15 dataset	- High Accuracy: Achieved 76.96% accuracy - Lightweight: Suitable for real-time IDS - Feature Selection: Improved detection by selecting relevant features	- Class Imbalance: Detection of underrepresented classes such as Worms and Backdoors is low - False Alarm Rate: 23.95%
Dutta et al, 2021 [176]	CAE + DNN	Proposes a hybrid anomaly detection system that first applies CAE for feature engineering and then uses a DNN for classification.	UNSW-NB15 dataset	- Improved Accuracy: 91.9% in comparison with baseline methods - Low FPR: Achieved 8.05% - Feature Engineering: Efficient use of CAE for feature reduction	- Class Imbalance: Still challenges with detecting minority classes - Complexity: May require more computational resources for training
Said et al, 2023 [177]	CNN-BiLSTM	Hybrid approach combining CNN for spatial feature extraction and BiLSTM for temporal feature learning.	InSDN, NSL-KDD, UNSW-NB15	- High Accuracy: Achieved up to 97.77% accuracy in binary classification - Improved Detection: Outperformed traditional models - Efficient Feature Selection RF and RFE used for feature selection	Training Time: CNN-BiLSTM takes longer to train compared to other models like LeNet5 or CNN - Complex Architecture: Hybrid models may increase computational overhead
Alsharaiah et al, 2024 [178]	AT-LSTM	Proposes a binary classification model that uses LSTM for capturing temporal dependencies and an Attention Layer for highlighting important features.	UNSW-NB15 dataset	- High Accuracy: Achieved 92.2% - No Feature Engineering Needed: Model learns features automatically from the data - Efficient Handling of Sequential Data	- Computational Demands: Requires substantial computational resources

					- Potential Vulnerability to Adversarial Attacks

Table 1 allows considering the issue of intrusion detection in more detail because it applies various algorithms, including SVM, DBN, CNN, and DL technologies based on the development of LSTM with attention models. The articles of such authors as Jing and Chen (2019), Tian et al. (2020), and Ashiku and Dagli (2021) present a model that is very efficient in recognizing advanced attacks as it has high accuracy (up to 97.77%) and low false positive rates. The drawbacks to these systems though include their high computational expense, the requisite large-scale data, and class imbalances.

Table 2 changes gears and focuses on phishing identification, showing a selection of ML algorithms, such as DNN, CNN and LSTM, which managed to perform with accuracy of over 99% of cases. The effectiveness of practicing notable works by Do et al. (2021) and Aldakheel et al. (2023) in identifying phishing websites and emails is especially discussed since they are burdened by the shortcoming of requiring a huge amount of labeled data and the inability of identifying new phishing methods, which has to carry out continuous updates and effective resource usage.

Table 3 focuses on the detection of cheats during online exams and applies such algorithms as KNN, SVC, RF, and LSTM on data on dynamic mouse moves and facial expression. Other authors, e.g., Siddiqui et al. (2022) and Hassan Hosny et al. (2022) indicate the high accuracy (up to 96.67%), along with flexibility to fit various exam formats, but problems with data quality, complexity of the feature extraction, and processing limitations in real-time remain open. In the three avenues, although the detection accuracy and efficiency continue to rise with usage of advanced ML techniques, problems of computational power, limitation of dataset and constant improvement of models still provide major challenges of implementing such systems at scale.

Table 1 An overview of research on intrusion detection

Authors	Algorithm	Descrption	Dataset	Advantages	Limitations
Jing et Chen, 2019 [170]	SVM	Focuses on improving intrusion detection using SVM with a nonlinear scaling method. Evaluates the performance of binary and multi-classification.	UNSW-NB15 dataset	- Improved Accuracy: 85.99% for binary classification - Robustness: Better than traditional methods in handling complex attack types - Low FPR: Particularly for multi-class classification	- Computational Cost: SVM with nonlinear scaling may require more time for training - Class Imbalance: Some attack classes may have skewed results
Tian et al., 2020 [171]	DBN	Introduces an improved DBN with combined sparsity penalty terms to enhance feature extraction and reduce overfitting in network IDS.	NSL-KDD, UNSW-NB15 datasets	- Improved Classification Accuracy: 96.17% (NSL-KDD) and 86.49% (UNSW-NB15) - Low FPR: Effective for handling imbalanced datasets - Robustness: Suitable for modern and sophisticated network attacks	- Complexity: Requires careful parameter tuning and could be computationally intensive - Imbalanced Class Distribution: Minor attack classes may perform poorly in F-measure
Almiani et al, 2020 [172]	RNN	The paper presents a deep RNN model designed to detect IoT-related intrusions, leveraging recursive neural networks in a multi-layered approach.	NSL-KDD dataset	- High Detection Accuracy: 94.27% for DoS, 97.36% for Probe attacks - Low FPR: Capable of detecting DoS attacks with 0% FPR - Real-Time Detection: Optimized for IoT environments	- High Computational Overhead: Requires extensive computational resources for training - Imbalanced Dataset: Issues with rare attack classes like R2L and U2R, addressed by oversampling
Ashiku et Dagli, 2021 [173]	CNN	Proposes a DL-based NIDS using CNN and a semi-dynamic hyperparameter optimization approach for network attack detection	UNSW-NB15 dataset	- High Accuracy: Achieved up to 95.6% for the testing dataset - DL Architecture: Utilizes CNN and MLP for improved detection - Optimized for Real-World Network Traffic	- Class Imbalance: Significant challenge with underrepresented attack classes - Computational Overhead: Requires GPU for optimization and processing
Al-Emadi et al., 2020 [174]	CNN, RNN (LSTM, GRU)	Proposes a DL-based NIDS using CNN for feature extraction and RNN (LSTM/GRU) for sequence-based attack detection.	NSL-KDD dataset	- High Precision: CNN achieved 100% precision - Superior Recall: RNN-LSTM and RNN-GRU performed well on recall - Effective with Large Data	- Computational Cost: High training time and resource demand - Limited Dataset Use: Focused on NSL-KDD, other datasets could be explored
Mebawondu et al, 2020 [175]	ANN and MLP	Proposes a lightweight NIDS using feature selection (Gain Ratio) and MLP ANN to detect network attacks, including novel attacks.	UNSW-NB15 dataset	- High Accuracy: Achieved 76.96% accuracy - Lightweight: Suitable for real-time IDS - Feature Selection: Improved detection by selecting relevant features	- Class Imbalance: Detection of underrepresented classes such as Worms and Backdoors is low - False Alarm Rate: 23.95%
Dutta et al, 2021 [176]	CAE + DNN	Proposes a hybrid anomaly detection system that first applies CAE for feature engineering and then uses a DNN for classification.	UNSW-NB15 dataset	- Improved Accuracy: 91.9% in comparison with baseline methods - Low FPR: Achieved 8.05% - Feature Engineering: Efficient use of CAE for feature reduction	- Class Imbalance: Still challenges with detecting minority classes - Complexity: May require more computational resources for training
Said et al, 2023 [177]	CNN-BiLSTM	Hybrid approach combining CNN for spatial feature extraction and BiLSTM for temporal feature learning.	InSDN, NSL-KDD, UNSW-NB15	- High Accuracy: Achieved up to 97.77% accuracy in binary classification - Improved Detection: Outperformed traditional models - Efficient Feature Selection RF and RFE used for feature selection	Training Time: CNN-BiLSTM takes longer to train compared to other models like LeNet5 or CNN - Complex Architecture: Hybrid models may increase computational overhead
Alsharaiah et al, 2024 [178]	AT-LSTM	Proposes a binary classification model that uses LSTM for capturing temporal dependencies and an Attention Layer for highlighting important features.	UNSW-NB15 dataset	- High Accuracy: Achieved 92.2% - No Feature Engineering Needed: Model learns features automatically from the data - Efficient Handling of Sequential Data	- Computational Demands: Requires substantial computational resources - Potential Vulnerability to Adversarial Attacks

Table 2 An overview of research on phishing detection

Authors	Algorithm	Description	Dataset	Advantages	Limitations
Do et al., 2021 [179]	DNN, CNN, LSTM and GRU	DL algorithms for phishing webpage detection using 30 website features from URLs.	UCI dataset (11,055 URLs: 4,898 phishing, 6,157 legitimate)	DNN and LSTM achieved high accuracy (97.29% and 97.20%), CNN and GRU had faster training/testing times.	No algorithm outperformed all others, manual hyperparameter tuning is time-consuming;
Mughaid et al., 2022 [180]	Boosted DT, NN, SVM, LR, Averaged Perceptron, etc.	A ML-based system to detect phishing emails by analyzing email features and comparing different algorithms' performances	Phishing Email Collection, Phishing Legitimate Full, Spam or Not Spam Dataset	-High accuracy rates -Robust results across multiple datasets. -Comprehensive comparison of several algorithms.	-Limited by dataset features. -Some phishing emails may evade detection due to similarity with legitimate emails.
Alshingiti et al., 2023 [181]	CNN, LSTM, LSTM-CNN	A DL-based approach using CNN, LSTM, and LSTM-CNN for phishing detection based on URL classification.	Yahoo Directory, PhishTank, 20,000 URLs with 80 features	-High accuracy in phishing detection. -CNN-based model outperforms others with 99.2% accuracy. - LSTM and LSTM-CNN also provide strong detection results.	- Does not check if the website is active, impacting the results. -Requires significant computing power. - Need for better parameter selection and feature engineering.
Adebowale et al., 2023 [182]	CNN, LSTM	Hybrid DL model for phishing detection using URL features	Dataset with phishing and legitimate URLs	High detection accuracy, robust feature learning	Computationally intensive, requires large datasets
Aldakheel et al., 2023 [183]	CNN	A CNN-based DL model for phishing detection using URL features. The approach includes character embedding, feature	PhishTank (10,000 phishing URLs, 10,000 legitimate URLs)	-High accuracy (98.77%). -Outperforms other models like RNN, SVM, and KNN. -Capable of processing large datasets and handling	- Limited to PhishTank dataset, which may not cover all types of phishing attacks. - Requires crawling and URL analysis, which may not be suitable for real-time detection.
Karim et al., 2023 [184]	DT, LR, RF, SVC, KNN, NB, GBM, Hybrid LSD Model (LR+SVC+DT)	A hybrid ML-based approach for phishing detection, utilizing several algorithms combined with a feature selection technique and hyperparameter tuning.	Kaggle (11,054 records, 33 attributes, phishing and legitimate URLs)	-High accuracy and precision (up to 98.12%). -Hybrid LSD model outperforms individual models with a precision of 97.31% and recall of 96.33%. - Cross-fold validation and grid search optimization	-Computationally intensive due to ensemble methods. -Some algorithms may show lower performance compared to ensemble models. -May not detect unknown phishing sites without prior training data.
Bezerra et al, [185]	NN, RF	A hybrid approach combining a neural network and RF for detecting phishing emails based on various features, including email content, sender's email address, and message format.	514,027 emails from E-goi	-High accuracy for phishing detection (NN: 99.16%). -Reduced false positives (4.62% blocked accounts). -Comprehensive feature extraction methods improve detection performance	-Challenges with feature extraction due to varied email formats. -Unbalanced dataset affects performance. -Needs continuous retraining to account for new phishing tactics.
Shafin, 2024 [186]	SHAP + LIME (SLA-FS)	Combines SHAP (global) and LIME (local) to identify features for phishing detection. It enhances feature selection (FS) for web phishing by identifying class-specific, significant features from both global and local perspectives.	Web Page Phishing Dataset (Hannousse & Yahiouche, 2021)	High accuracy (97.41% for RF), increased detection speed, interpretability, adaptable to specific attack types, improved real-time detection	Computational cost may increase for large datasets, relies on a good balance between global and local feature importance.
Fares et al. 2024 [187]	SVM, RF, XGBoost	A ML approach to detect phishing emails using three different algorithms (SVM, RF, XGBoost) and a feature selection technique.	Kaggle phishing email dataset	- High accuracy rates (SVM: 97.6%) - Robust detection of phishing emails - Minimal execution time	- May require tuning to handle new, sophisticated phishing methods - Dependence on the dataset quality and diversity
Kavya and Sumathi 2024 [188]	LR, RF, SVM, CNNs, RNNs, LSTMs), GANs (PDGAN)	Review of state-of-the-art phishing detection methods, covering classical, ML, and advanced DL techniques like GANs	PhiKitA, PhishTank, UCI phishing dataset, Ethereum-specific blockchain dataset	-ML: Adaptive, scalable; good for structured data -DL: High accuracy, automated learning of complex patterns -GANs: Robustness through synthetic data generation	- Traditional Methods: Prone to high false-negative rates - ML: Needs manual feature extraction, vulnerable to adversarial attacks - DL: Resource-heavy - GANs: Computationally expensive

Gupta et al. 2024 [189]	RNN-WOA	A framework combining RNN's ability to process sequential data with WOA's optimization capabilities for phishing detection, resulting	Kaggle dataset with over 11,000 URLs	- RNN: Effective in learning complex patterns from sequential data -WOA: Efficient hyperparameter optimization	- RNN: Computationally intensive, particularly in large datasets - Requires significant computational resources for training - WOA optimization can increase complexity and time
Barik et al., 2025 [190]	EGSO-CNN	A DL-based model to detect phishing URLs, combining CNN with an EGSO technique. The model uses feature extraction and data preprocessing techniques like VAE and StandardScaler	Kaggle, Alexa, PhishStorm	-EGSO-CNN:High accuracy 99.44% - Optimization reduces error rates and false positives - Flexible and adaptable for real-world scenarios	Dataset Limitations: The model may not detect phishing sites using embedded Flash/JavaScript content - Relies on updated datasets for performance - High computational cost
Rao et al. 2025 [191]	Phish-Jam (Hybrid Super Learner Ensemble, LSTM, Transformer models)	Phish-Jam is a mobile phishing detection system using a super learner ensemble that integrates multiple ML algorithms. It uses handcrafted, transformer-based, and DL features for phishing URL classification	PhishDump dataset: 190,062 legitimate URLs and 141,490 phishing URLs	-Fast Detection: Low response time for real-time operation -Hybrid Ensemble: Combines multiple models for improved accuracy (98.93%) and robustness -Mobile-Friendly: Lightweight model suited for mobile devices	-Limited to English URLs: The model may struggle with non-English URLs -Shortened URLs: Difficulties detecting phishing sites using shortened URLs - Dependence on URL Structure: Can be circumvented by sophisticated URL modifications
Alsaidi et al. 2025 [192]	HHO and DNN	A novel phishing URL detection system that integrates HHO to optimize the DNN's weights and biases. It uses advanced feature extraction and multi-objective optimization to balance accuracy and minimize false positives.	PhishTank and OpenPhish datasets	-High Accuracy: 99.6% -Low False Positives: 0.2% -Optimized for Real-Time: Efficient and quick detection - Dynamic Optimization: HHO helps avoid local minima	- Computationally Intensive: Some real-time constraints on hardware - Limited to URL-based features: May miss phishing tactics based on visual similarities or embedded objects
Garki et al., 2024 [193]	RF, DT, RNN	The study investigates phishing attack prediction in e-learning platforms using ML techniques. It highlights vulnerabilities in ODL systems, particularly in relation to social engineering, and tests several models to detect these attacks.	Simulated phishing data from real-world phishing tactics,	-RNN: High precision (1.0) and accuracy (0.98) -RF: Best overall balance across metrics (recall 0.97, F1-score 0.97) -DT: Strong recall (0.95)	-RNN: Lower recall (0.89), missing some true positives -RF: Slightly lower precision compared to RNN -DT: Lower accuracy (0.94)
Yoo et Cho 2022 [194]	Text-CNN	Real-time defense against SNS phishing attacks using a chatbot	SNS phishing messages	Multi-phase detection, real-time action response	May struggle with advanced social engineering attacks
Alazaidah et al., 2024 [195]	RF, J-48, FilteredClassifier, GainRatio, InfoGain, SMO, MLP	The study evaluates various ML techniques to detect phishing websites. It compares 24 classifiers, investigates feature selection, and identifies the best performing methods for phishing detection	Dataset1: 11055 instances, 30 binary features. Dataset2: 1353 instances, 9 features with 3 classes.	-RF: High accuracy in Dataset2 (97.26%) -J-48 and FilteredClassifier: High performance for binary classification -InfoGainAttributeEval: Best feature selection method for accuracy, precision, and recall	- Rules learning strategy: Lowest performance across all datasets - RF: May not perform well on datasets with very high feature counts and class labels
Sahingoz et al., 2024 [196]	ANN, CNN, RNN, BRNN, ATT	DEPHIDES, a phishing detection system based on DL algorithms to analyze URLs. It aims to identify phishing URLs with high accuracy using five DL models	PhishTank for phishing URLs (2.3 million) and Common Crawl for legitimate URLs (2.8 million)	-High Accuracy: CNN achieved 98.74% accuracy -Real-Time Detection: Processes over 130,000 URLs per second -Zero-Day Attack Detection: -Language-Independent: Can detect phishing across	-Hardware Intensive: Requires substantial GPU resources for processing large datasets - URL Length Limitation: Model uses a fixed 200-character limit for URLs, which may impact detection of extremely short or long URLs
Trad et al., 2024 [197]	Prompt Engineering, Fine-Tuning GPT-2	Compares prompt engineering and fine-tuning for phishing URL detection using LLMs. Results show fine-tuned models significantly outperform prompt-engineered ones	Hannousse & Yahiouche Dataset (11,430 URLs, balanced with 500 phishing and 500 legitimate URLs for test)	-Fine-Tuned GPT-2: Achieved F1-score of 97.29%, surpassing state-of-the-art techniques -Prompt Engineering: Quick, no training needed -Real-World Application: Tested on varied phishing-to-legitimate ratios	-Prompt Engineering**:** Lower performance **-**Fine-Tuning**:** Requires significant resources (computational power, domain-specific training) **-**Data Privacy**:** Concerns with prompt engineering via APIs
Borra et al., 2024 [198]	PCA, PSO, DLCNN	OEC-Net, a hybrid method that utilizes PCA for feature extraction, PSO for feature selection, and DLCNN for email classification (spam, ham, phishing).	Hybrid dataset combining UCI Spambase, CSDMC, and SpamAssassin datasets.	-High Accuracy: 98.43% -Optimal feature selection -DL integration for superior classification -Hybrid dataset for more robust training	-Computational Overhead: High resource consumption for DL models -Limited interpretability: DL models often lack transparency in decision-making - Sensitive to dataset quality: Relies on diverse and comprehensive datasets for optimal performance

Jayaraj et al., 2024 [199]	HEFS, CDF-g, DT, RF, k-NN	Focuses on detecting phishing URLs to enhance IDS. The paper introduces a hybrid ensemble feature selection method that utilizes CDF-g for identifying optimal features.	Custom dataset with phishing URLs collected from various sources	-High Accuracy: 97.8% -Fast Processing: Low train time and test time - Efficient Feature Selection	- Scalability: The system may face challenges in handling very large datasets in real-time - Limited Generalization: Focuses on phishing URLs, limiting broader intrusion detection use
AL-Ahmadi et al., 2022 [200]	LSTM (Generator), CNN (Discriminator)	Phishing detection model using GAN for URL-based classification	PhishTank, DomCop URLs	High accuracy (97.58%), no reliance on third-party services	Dependent on synthetic data generation
Orunsolu et al., 2022 [201]	SVM and NB	Proposes an enhanced ML-based model for phishing detection, integrating feature selection to improve accuracy and reduce false positives. The model uses features from URL, webpage properties, and behavior.	2541 phishing instances and 2500 benign instances from standard phishing datasets	- 99.96% Accuracy: Achieves near-perfect detection rates - Low False Positive (0.04%): High precision with minimal false alarms - Efficient Runtime: SVM and NB both perform well with low computational cost	- Limited to Selected Features: Relies on predefined features and doesn't account for some visual or newer phishing tactics - May Struggle with Sophisticated Phishing Techniques
Boussougou et al., 2025 [202]	BT, SMOTE, LSTM, BiLSTM, CNN	Augments dataset for better detection of voice phishing (vishing) using AI models.	Korean Call Content Vishing (KorCCVi) dataset	High F1 scores (98.91%), enhanced detection accuracy	Limited to Korean dataset, challenges in real-time application
Chinta et al., 2025 [203]	CNN, XGBoost, RNN, SVM, BERT-LSTM	Detection of phishing emails using ML models.	Phishing email dataset	High accuracy (99.55% F1 score), real-time detection	Dependency on large labeled datasets
Nayak et al., 2025 [204]	FNN, DNN, Wide and Deep, TabNet	This paper explores ML techniques for phishing detection, leveraging feature selection and DL models to enhance detection efficiency and reduce feature extraction costs	PhishTank, Mendeley 2020 dataset with 58,645 URLs, including both phishing and legitimate URLs	- High accuracy (94.46%) with Feedforward Neural Network - multi-model approach (FNN, DNN, Wide & Deep, TabNet) improves robustness - Feature selection boosts computational efficiency - Introduces the Anti-Phishing Score metric for better evaluation	- Complexity in training: High computation time for deep models - May not handle new phishing techniques - Dependence on feature relevance: Effectiveness varies with feature quality
Taofeek, 2024 [205]	DT, RF, MLP, XGBoost, Autoencoder, SVM	The paper proposes a novel ML approach to detect phishing websites using various algorithms. It employs feature extraction and a binary visualization method to identify phishing URLs and protect sensitive data	Phishing URLs dataset from PhishTank (5,000 URLs) including phishing samples from companies like PayPal, Microsoft, etc	- High accuracy (94.16%) in phishing detection - Binary Visualization for better classification - Fast detection with TensorFlow - MobileNet optimizes testing speed	- Model Interpretability: DL models may lack transparency - Data Quality: Relies on the availability and quality of datasets - Evolving Phishing Techniques: New tactics could affect model effectiveness - False Positives and Negatives: Need for improvement to avoid misclassifications

Table 3 An overview of research on cheating detection

Authors	Algorithm	Description	Dataset	Advantages	Limitations
Hassan Hosny et al., 2022 [206]	KNN, SVC, RF, LR, XGBoost, LightGBM	The system detects cheating in online laboratory exams using AI algorithms that analyze mouse dynamics. Mouse movements are used as biometric data to distinguish cheating behavior	Data from 50 students completing both VRL-based and static exams.	High accuracy (up to 90% for LightGBM); no additional hardware required (uses only mouse data); effective for both VRL and static exams	System performance may be affected by the emotional state of students; redundant data collection; different screen resolutions can affect results.
Siddiqui et al., 2022 [207]	SVM, KNN, RF, LSTM-RNN, 1D-CNN, ANN	This study evaluates ML (SVM, KNN, RF) and DL (LSTM-RNN, 1D-CNN, ANN) models for continuous user authentication based on mouse dynamics	Dataset of 40 users performing high-intensity tasks (Minecraft)	High performance of DL models (1D-CNN, LSTM-RNN, ANN); ANN achieved 92.48% accuracy in multi-class classification	SVM and KNN struggled with high-dimensional data; feature scaling required; DL models are computationally expensive and hard to interpret
Valizadeh, 2022 [208]	Quantitative/ Qualitative Analysis	The study explores Turkish students' perceptions of cheating in online education and provides suggestions to minimize it, including more stringent exam protocols and alternative assessment methods.	Data from 163 Turkish students	Mixed-methods approach provides both quantitative and qualitative insights	Potential cultural bias in results; limited to Turkish higher education context.
Zhou et Jiao, 2022 [209]	Blending Ensemble Learning, Anomaly Detection	Explores data augmentation in ML for cheating detection in large-scale assessments, focusing on anomaly detection techniques.	Large-scale assessment data	Data augmentation improves detection accuracy; robust for large datasets.	Requires large datasets; focuses on large-scale assessments which may not generalize to smaller settings.
Duhaim et al., 2022 [210]	Data Mining Techniques	Uses data mining techniques to detect cheating in online exams during the COVID-19 pandemic, proposing methods for analyzing students' answers to identify suspicious patterns.	Data from online exams during COVID-19	Uses clustering and similarity measures for effective cheating detection.	Focused on data mining techniques; may not work in all exam formats.
Alsabhan, 2023 [211]	LSTM	Proposes a ML-based method to detect cheating in online exams using the 7WiseUp behavior dataset. The model employs LSTM to identify suspicious patterns in student exam data.	7WiseUp behavior dataset	- 90% Training Accuracy, 92% Validation Accuracy - High Efficiency: 90% accuracy in training data, improving over traditional methods - Data Preparation: Cleaned data contributed to improved accuracy	- Single Dataset: The model uses one dataset (7WiseUp) which may not be generalizable to all settings - Imbalanced Dataset: Low recall and F1 score could affect real-world applicability - Limited Factors: The model does not fully analyze why cheating occurs beyond grade sequences.
El Rhezzali et al., 2023 [212]	Vision-based Detection, Online Proctoring	Discusses various techniques for detecting cheating in online exams, including behavior monitoring through webcam and software tools.	Data from online exam observations	Easy to implement with existing technologies; offers real-time monitoring.	Relies on webcam monitoring, which may not be accessible to all students.
Moreno-Marcos et al., 2023 [213]	Moodle Data Analytics	Statoodle analyzes Moodle data to detect cheating by inspecting students' actions during online exams. It provides detailed performance and cheating detection reports for instructors.	Moodle data from exported logs	Simple tool for instructors; works with existing LMS data exports.	Requires Moodle-based system; may not be suitable for all learning platforms.
Volarić et al., 2023 [214]	LR, DT, RF, KNN	The study develops a ML-based model to detect academic fraud during online tests by analyzing student behavior through a web browser extension	Data collected from 52 students, each performing two tests with various cheating methods	High accuracy (up to 96.67% with LR; non-intrusive method via browser extension; good for detecting cheating via inactivity time, browser switching, and text manipulation.	Small sample size (52 students), leading to potential overfitting; reliance on browser-based data, which cannot track activities outside the browser; data from uncontrolled testing environments
Ong et al., 2023 [215]	Clustering (Time Series K-Means, SOM, Fuzzy C-Means)	The study uses clustering-based methods to detect cheating in online examinations by analyzing students' facial expressions, body gestures, and eye movements, utilizing a simple webcam	Custom dataset with 200 samples from 50 students,	Effective with 83.6% accuracy in detecting various cheating behaviors; non-intrusive (uses only a webcam); adaptable to various online examination settings.	The clustering approach may not distinguish subtle motion patterns; relies on camera positioning; misclassification due to noisy data; challenges with dynamic motion detection.

Tran et al., 2023 [216]	YOLO-Pose, SVM, DT, RF, LSTM	This model integrates skeleton tracking (YOLO-Pose) and ML models to detect cheating behaviors in classrooms, focusing on analyzing motion and gestures.	Classroom cheating behavior dataset	High accuracy in detecting cheating motions; real-world applicability.	Dataset limitations; focused on physical classroom environments rather than online setups.
Zhao et al., 2023 [217]	RF, MLP, LR, XGBoost,	The study used ML to predict self-reported cheating in elementary school children. Predictors include children's perceptions of cheating, peer behaviors, and socio-demographic factors	Data from 2,094 children from grades 2 to 6 across three schools in Hangzhou, China.	High predictive accuracy (81.43% on holdout set with RF; effective use of ML for understanding developmental psychology; identified key predictors of cheating.	The model relies on self-reported cheating, which may underestimate actual behavior; results based on one sample from China, limiting generalizability to other regions or educational systems
Gopane et al., 2024 [218]	FaceNet Model, Lucas Kanade Algorithm, AAM	The system combines AI, computer vision, and image processing techniques to detect cheating in online exams by analyzing students' facial movements, gaze, and activity patterns using webcams	Video data collected from over 1000 online examinations at the K.J. Somaiya Institute of Technology	High accuracy in detecting cheating (F-score of 0.94); uses cost-effective and non-intrusive webcam technology	Limited to webcam-based video input; the accuracy may vary based on lighting conditions, occlusions, or face angles.
Abeba et al., 2024 [219]	CNN	Focuses on classroom-based exam cheating detection using DL models (CNN). It applies image classification to identify cheating behaviors based on exam images.	Image dataset from classroom exam cases	-High accuracy for image classification (92%). -Innovative approach for classroom cheating detection.	-Limited to paper-based exams. -Dataset may not be fully comprehensive.

4. Conclusion

This chapter has reviewed a broad spectrum of contemporary research that applies ML and DL techniques to three major areas of cybersecurity: intrusion detection, phishing detection, and cheating detection. Across these domains, the literature demonstrates that intelligent models have significantly enhanced the capacity to detect, classify, and mitigate diverse forms of cyber threats with remarkable precision and adaptability.

In the field of intrusion detection, studies have shown that deep neural architectures—such as CNN, LSTM, and hybrid CNN-BiLSTM models—offer substantial improvements in detecting complex and evolving attack patterns. These models achieve high accuracy and low FPR, outperforming traditional algorithms like SVM and RF. Nevertheless, challenges remain in terms of computational complexity, scalability, and the imbalance of datasets, which often limit their effectiveness in real-time network environments.

Regarding phishing detection, the reviewed works reveal that DL methods, particularly those based on CNN, LSTM, and hybrid architectures, have achieved outstanding results, with accuracies exceeding 99% in some cases. The evolution of ensemble and hybrid frameworks, as well as the integration of explainable AI and optimization algorithms, has further advanced detection efficiency. However, dependence on static datasets, the rapid evolution of phishing tactics, and limitations in cross-linguistic and contextual understanding continue to pose obstacles to practical deployment.

The third area, cheating detection, represents an emerging but rapidly developing field within cybersecurity and online education. Research leveraging behavioral data, biometric features, and video analysis has demonstrated that ML and DL can play a pivotal role in ensuring academic integrity during remote examinations. Despite promising results—often exceeding 90% accuracy—these systems face persistent challenges related to privacy concerns, dataset variability, and the adaptability of models to different testing conditions.

Overall, the literature underscores the transformative role of artificial intelligence in cybersecurity. Yet, it also highlights that future efforts must focus on enhancing model generalization, optimizing computational efficiency, ensuring real-time adaptability, and integrating ethical and privacy-preserving frameworks. These directions will be crucial to advancing intelligent, transparent, and reliable cybersecurity systems capable of addressing the continuously evolving landscape of digital threats.

CHAPTER 04

RESEARCH METHODOLOGY

Chapter 04: Research Methodology

1. Introduction

This chapter describes the research methodology of this thesis, which is composed based on four independent DL-based contributions in order to address important security needs in e-learning systems, online examinations, networked systems, and IoT spaces. The proposed approach focuses on hybrid intelligent models that graft the remarkable feature-extracting power from deep neural networks and optimization and attention mechanisms to improve detection accuracy, robustness, and adaptability in complicated cyber habitats.

The first proposed approach is based on the CNN–OOA framework that performs an optimal feature selection and phishing attack classification using CNN combined with the Orca Optimization Algorithm. It is based on the protection of personal data and login credentials by maximizing CNN parameters with OOA for better classification ability and to harden against new challenges arising from phishing variants.

The second contribution introduces the LSTM–SWAP model, which marries LSTM networks and SWAP-based attention for localized anomaly detection for analyzing temporal behavior of students during online exams. Through emulating the sequential interaction and behavior dynamics, this method helps to identify cheating or fraudulent incidents, promoting the integrity of remote assessment systems.

The third contribution is the architecture of CNN–GWO to improve network threat detection, including SQL injection attacks and malware, by using CNN with GWO. The embedment of GWO not only provides efficient hyperparameter optimization, which enhances convergence and accurate threat detection in complex network traffic data.

Finally, the last contribution presents a model called CNN–CAT that combines CNN with Cross Attention Transformer to enhance anomaly detection in IoT networks. The attention mechanism used in the model enhances the model's ability to capture abnormal device behaviors and further guarantees confidentiality and security of learner data in IoT-based educational systems.

Collectively, these four contributions pave the way for holistic and systematic research into hybrid deep learning models to combat varied and emerging cybersecurity threats. This chapter describes the principles, strategies for data processing, model architectures, and evaluation methods of all proposed approaches.

2. Contribution 01: LSTM-SWAP: A Hybrid Deep Learning Model for Cheating Detection [220].

2.1Description

E-learning has revolutionized how education is delivered by providing learners with learning flexibility, access and scaling so that students can learn wherever they are. Such platforms as MOOCs or Open Classrooms have expanded the education market far beyond educational institutions. Nevertheless, this move has introduced issues of academic integrity since online examinations can easily be cheated through plagiarism and impersonation among others. These behaviors can hardly be detected by just ordinary systems that will track and decode the actions of students on real-time basis. Keeping abreast of changing methods used to cheat and the differences in student behavior makes the recognition more difficult and exposes them to false positivity. The issue is also muddled by privacy concerns of objects that are intrusive to monitor. In order to solve these problems, scholars are turning to cyber security tools of behavioral anomaly detection, e.g. IDS, to detect potential cheating based on individual atypical behavioral deviations. The offered LSTM-SWAP model, a hybrid DL system that combines LSTM networks and a Sliding Window Attention Processor (SWAP) is a data-driven dynamic model. By capturing user behavior patterns on a sequence basis, the LSTM model is used, and SWAP mechanism is provided to improve the detection results which involve prioritizing crucial behavioral parts, noise reduction, and increasing precision. The model intends to remain dynamic to be more precise and experience fewer false positives than the technique based on static content. The LSTM-SWAP model, experimented on the Mouse Dynamics for User Authentication Dataset has been effective in identifying abnormality in the process being observed in some examinations with e-learning ensured scalability, better detection ability with related considerations of user privacy upholding. This strategy offers a viable, privacy-sensitive, and scalable method to maintaining academic purity on the web-based courses.

2.2Methodology

2.2.1 Long Short-Term Memory (LSTM)

The LSTM component is engineered to process sequential data, enabling the identification of temporal trends in user activity. It processes mouse movement data, encompassing coordinates, timestamps, and derived properties such as velocity and acceleration. The LSTM utilizes memory units and gating mechanisms to retain significant behavioral patterns over time.

The output from the LSTM's final hidden states is subsequently transmitted to the SWAP module for further processing.

- **Feature Extraction:** The data is subjected to preprocessing, encompassing the normalization of screen coordinates, timestamps, and motion-derived attributes such as velocity and acceleration. Missing data is resolved, and sequences of user interactions are generated.
- **Sequence Modeling:** The LSTM examines interaction sequences, identifying long-term dependencies that reveal patterns of normal or questionable behavior.
- **Passing to SWAP:** The retrieved temporal features are transmitted to the SWAP module for targeted attention-based augmentation.

2.2.2 Sliding Window Attention Processor (SWAP)

The SWAP module uses a localized attention mechanism in order to detect short-term anomalies within mouse interaction sequences. This step is extremely important in distinguishing the difference between normal and irregular behaviors in e-learning exams.

- **Attention-Based Processing**: SWAP identifies important behavioral moments by having the size of each interaction event dimension with a pre-defined fixed size sliding window.
- **Query-Key-Value Attention Mechanism**: The output from LSTM is processed into query, key and value vector which are used to calculate attention scores inside sliding window.
- **Refinement of Behavioral Patterns**: The weighted calculation of attention scores helps in better detecting of movements as such it is critical to be able to assess the variation of they.

2.2.3 LSTM-SWAP Hybrid Model

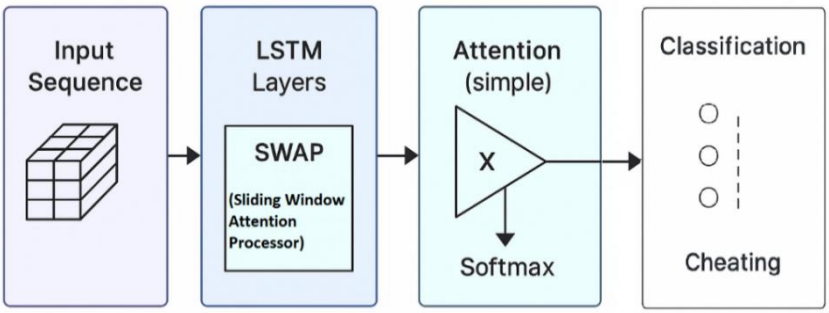
As shown in Figure 5, the LSTM-SWAP model is a synergistic approach that combines the capabilities of the two components that make it to identify cheating behaviors with increased efficiency:

- **Input Processing**: The model takes in augmented and normalized logs of user-interaction.
- **LSTM Encoding**: Captures long-term temporal movement patterns throughout the examination session.

- **SWAP Layer Application**: Highlights short-term irregularities using attention within a fixed window.
- **Feature Fusion & Classification**: The combined outputs of LSTM and SWAP are passed through fully connected layers for final classification.
- **Prediction Output**: A softmax activation function determines the probability of the class, hence, the legitimacy or cheating tendencies of a certain behavior.

This hybrid DL system is also found to be successful in the modelling of user interaction, using the global dependency-capturing propensity of the LSTM and the localized event-detection capacity of the SWAP, hence guaranteeing a high degree of accuracy in the detection of cheating behavior in examinations conducted online.

Figure 5 : Architecture of the LSTM-SWAP Model for Cheating Detection.



Pseudo code: The pseudo-code of LSTM-SWAP:

Algorithm 1 LSTM-SWAP Pseudo-code

Input: Sequence $X = [x_1, x_2, ..., x_T]$, window size w
Initialize LSTM with hidden dimension d_h
$H \leftarrow \text{LSTM}(X)$ ▷ $H = [h_1, h_2, ..., h_T]$
Initialize weight matrices W_Q, W_K, W_V
Compute attention inputs:
    $Q \leftarrow H \times W_Q$
    $K \leftarrow H \times W_K$
    $V \leftarrow H \times W_V$
for $t = 1$ to T **do**
    $start \leftarrow \max(0, t - w)$
    $end \leftarrow \min(T, t + w)$
    $K_{\text{win}} \leftarrow K[start : end]$
    $V_{\text{win}} \leftarrow V[start : end]$
    $score_t \leftarrow \frac{Q_t \times K_{\text{win}}^\top}{\sqrt{d_k}}$
    $weights_t \leftarrow \text{softmax}(score_t)$
    $Z_t \leftarrow weights_t \times V_{\text{win}}$
end for
$Z \leftarrow [Z_1, Z_2, ..., Z_T]$
for $t = 1$ to T **do**
    $F_t \leftarrow \text{concat}(h_t, Z_t)$
    $F'_t \leftarrow \text{ReLU}(F_t \times W_{fc} + b_{fc})$
    $y_t \leftarrow \text{softmax}(F'_t \times W_{out} + b_{out})$
end for
Output: $Y = [y_1, y_2, ..., y_T]$ ▷ Probabilities of cheating
Train: Use cross-entropy loss and Adam optimizer

2.3 Dataset

In During this step, we used our methodology on the open-source Mouse Dynamics for User Authentication Dataset, originally obtained from Kaggle and adapted for the identification of cheating behaviors. The dataset has 7 features and 2,464 records, which include 1,937 labeled training records and 527 labeled testing records. It offers comprehensive mouse movement data, including cursor positions, click events, timestamps, and movement patterns, all of which indicate user participation in online learning environments. This behavioral data is crucial for differentiating between genuine and potentially deceptive user actions during online evaluations. Furthermore, Table 4 delineates the feature kinds extracted from the mouse movement records utilized for user authentication. The dataset comprises session identities, timestamps, event categories, and screen positions. We developed and tuned the LSTM-SWAP model for the specific requirements of our research in cheating detection and classification. These changes aimed to enhance accuracy, minimize false positives, and guarantee consistent performance across various user activities.

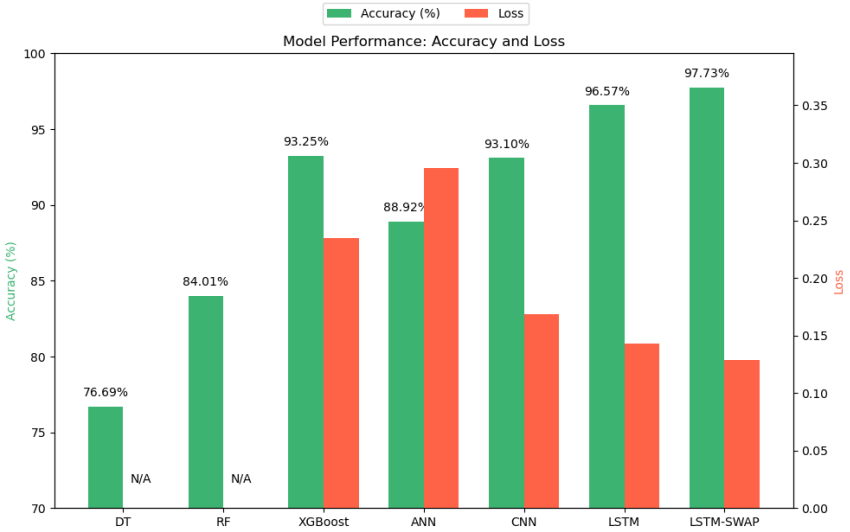
Table 4 Description of features used for Mouse Dynamics dataset

Feature Name	**Description**
uid	Unique identifier for each mouse event (possibly UUID format)
session_id	Unique identifier for the user session
user_id	Identifier for the user who generated the event (anonymized numeric ID)
timestamp	Time when the mouse event occurred (in milliseconds)
event_type	Type of mouse event (likely encoded numerically, e.g., 1=Move, 2=Click)
screen_x	X coordinate of the mouse cursor on the screen at the event
screen_y	Y coordinate of the mouse cursor on the screen at the event

2.4Results and comparison

In this part, we identify the results that are obtained by the LSTM-SWAP model and compare them with those derived by other methodologies. The analysis applies the measures of accuracy and loss to strictly evaluate the performance of Mouse Dynamics in the User Authentication Dataset.

Figure 6 : Accuracy and loss analysis of the evaluated models in cheating detection dataset



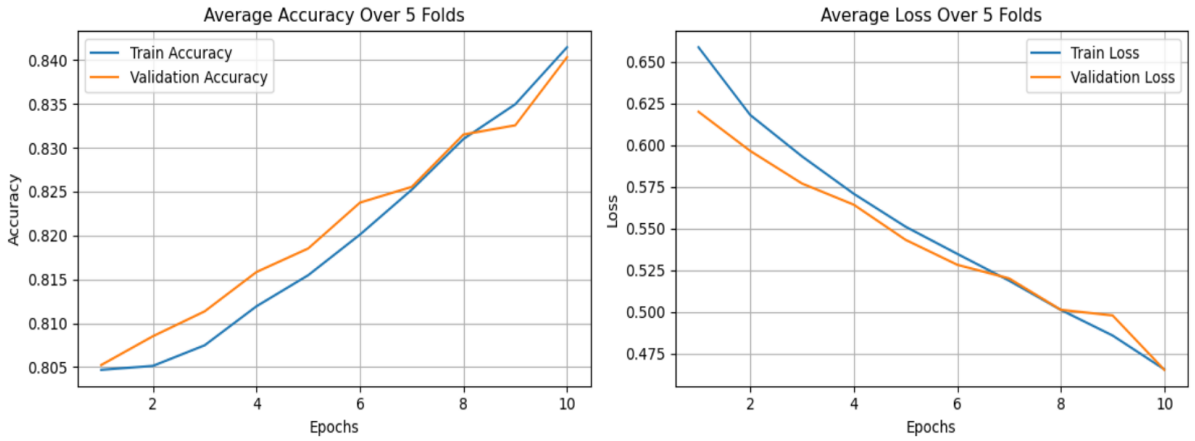
The Mouse Dynamics for User Authentication Dataset was utilized to train multiple ML and DL models, as depicted in Figure 6. The DT achieved an accuracy rate of 76.69% in the first experiment employing ML models, while the RF model raised the accuracy to 84.01%. Subsequently, the accuracy increased to 93.25% in the XGBoost model, with an associated loss rate of 0.2343. The DL models produced a greater accuracy versus loss ratio, where the ANN achieved an accuracy rate of 88.92% with a loss value of 0.2952 and the CNN performed slightly better at 93.10% accuracy and an associated loss of 0.1687. The LSTM model outperformed each previous model, with an accuracy rate of 96.57% for the LSTM and an associated loss of 0.1432. Finally, the proposed LSTM-SWAP model had the highest accuracy of 97.73% and the lowest loss rate of 0.1286, evidencing its ability to capture complicated behavioral trends and superior generalization ability. The results demonstrate the effectiveness of combining sequential learning and attention methods for high levels of reliability and accuracy in user authentication for e-learning platforms.

Table 5 Evaluation of LSTM-SWAP Model Performance Across Different Epoch Counts.

Epochs	**Accuracy(%)**	**Precision**	**Recall**	**F1-Score**	**FPR**	**AUROC**
10	84.07	0.7800	0.8407	0.7887	0,0532	0.8138
50	96.33	0.9629	0.9633	0.9629	0,0107	0.9846
100	97.73	0.9775	0.9773	0.9774	0,0089	0.9885

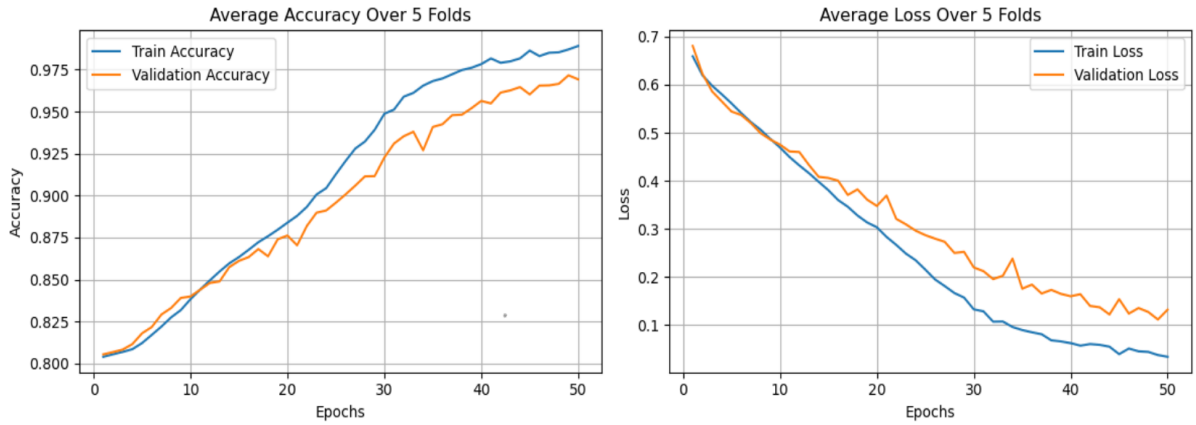
Table 5 presents the performance metrics of the LSTM-SWAP model evaluated at 10, 50, and 100 training epochs. Metrics include accuracy, precision, recall, F1-score, FPR and AUC-ROC, highlighting the model's progressive improvement in classification performance as training epochs increase.

Figure 7 : Training and Validation Curves of the LSTM-SWAP Model Across 5-Fold Cross-Validation over 10 epochs.



During the first 10 epochs, the LSTM-SWAP model showed consistent and steady learning with a training accuracy of 0.8407, a precision of 0.7800, the recall of 0.8407, F1-score of 0.7887, and AUC-ROC of 0.8138. The FPR was also measured as 0.0532, a rather low false positive rate, which is lower, but close to the training accuracy. This trend indicates that it successfully generalizes to unobserved data, and it does not overfit. The results, as illustrated in Figure 7, showed a progressive reduction in training and validation loss, as the validation loss was 0.4652, which is a stable convergence and efficient optimization. The fact that the loss values were declining in tandem also demonstrates that the model was indeed extracting significant trends in the Mouse Dynamics User Authentication Data Set. These findings relate to the strong learning abilities of the LSTM-SWAP model and give a solid basis of increasing performance in the next epochs.

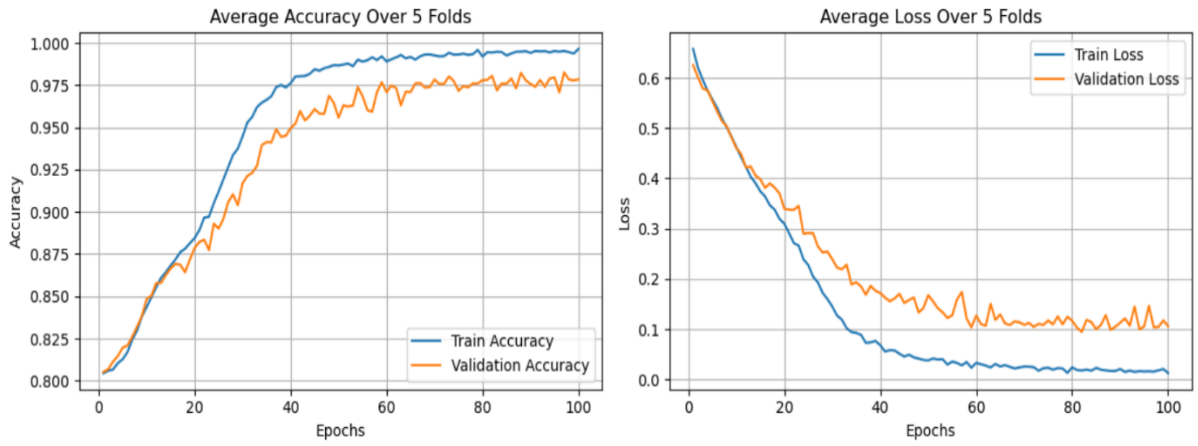
Figure 8 : Training and Validation Curves of the LSTM-SWAP Model Across 5-Fold Cross-Validation over 50 epochs.



The LSTM-SWAP model recorded significant gains in performance as the training continued to 50 epochs with average accuracy, precision, recall, F1-score, and AUC-ROC of

0.9633, 0.9629, 0.9633 and 0.9846 over five folds. The model also retained a low FPR of 0.0107, meaning that there was a low rate of false positive cases which was classified as well as a weak divergence between the training and validation accuracy curves (Figure 8, left), which confirms the good generalization of the model to unseen data and strengthens the robustness of the model. Equally, the training and validation loss curves (Figure 8, right) displayed a congruent and uniform decrease with the ultimate validation loss being 0.1319. This gradual loss decrease testifies to the stability of this model and to its good performance in learning, and nothing in this case indicates any form of overfitting even in the long term. These findings affirm the fact that LSTM-SWAP is appropriate in capturing the behavioral patterns in the Mouse Dynamics for User Authentication Dataset; hence, confirming its appropriateness in authenticate users in a secure and accurate task.

Figure 9 : Training and Validation Curves of the LSTM-SWAP Model Across 5-Fold Cross-Validation Over 100 Epochs.



The LSTM-SWAP model with 100 epochs of training gave nearly optimal results, with an average accuracy of 97.73, a precision of 97.75, recall of 97.73, F1-score of 97.74, and an AUC-ROC of 0.9885 on the 5-fold cross-validation. Very low False Positive Rate (FPR) (0.0089) was also observed in the model and this implies that the model is highly reliable when it comes to differentiating between the real and unauthorized users. Last validation loss was close to a small number of 0.1060 which means good learning and minimal error of generalization. The training and validation accuracy curves coincided at high values interpreting the aptitude of the model to learn complex patterns of user behavior with little overfitting. The steadily falling curves of losses end up being flat at lower values as illustrated in Figure 9, thus reinforcing the stable convergence of the model. The uniformity of such behavior in folds underscores the high generalization and robustness of the LSTM-SWAP model, which is very suitable in user authentication according to the dynamics of the mouse.

Table 6 Evaluation of models using performance metrics on the Mouse Dynamics Dataset

Algorithms/Metrics	Accuracy	Loss
DT	76.69%	N/A
RF	84.01%	N/A
XGBoost	93.25%	0.2343
ANN	88.92%	0.2952
CNN	93.10%	0.1687
LSTM	96.57%	0.1432
LSTM-SWAP	**97.73%**	0.1286

2.5Discussion and Future Directions

To address the research question in this work, several ML and DL models have been tested on the Mouse Dynamics for User Authentication Dataset to determine their ability to identify cheating behaviors in an eLearning setting. As a performance metric, accuracy was employed, and loss values of DL models were also taken into account. The models that are assessed are DT, RF, ANN, CNN, XGBoost, LSTM, and the developed hybrid LSTM-SWAP model.

The DT model had the lowest accuracy of 76.69 as shown in Table 6 and this indicates that it is not very effective with complex or sequential trends of data. Even though it is simple and interpretable, DT could not be generalized on this data easily. Enhanced with a RF model, which is a collection of DTs, elevated the accuracy of the model to 84.01% and lessened overfitting and generalization. XGBoost classifier also did a better job with 93.25% accuracy and a loss of 0.2343. This enhancement indicates how XGBoost is better than the conventional ensemble algorithms such as RF in dealing with structured data with GB and enhanced regularization algorithms.

The ANN is one of the models based on the neural network with an accuracy of 88.92% and the loss of 0.2952. ANN was good at capturing non-linear relationships in data, however, due to the inability to model sequential dependency, its overall performance was compromised. CNN also increased accuracy to 93.10% with a lower loss value of 0.1687, which was improved by the fact that it is capable of capturing localized and hierarchical spatial features of the input data. Nonetheless, CNN did not have a mechanism to handle time dependencies thus, it was not optimal with sequential mouse movement patterns.

The LSTM network was ranked much higher than the previous models with an accuracy of 96.57 and a considerably low loss of 0.1432. This was achieved due to the ability of LSTM to

learn long-term dependencies and sequential patterns of the data and hence it is suitable in mouse dynamics-based authentication. Lastly, our SWAP-optimized LSTM-sequence learning model, which combines the process of sequence learning using LSTM with the ability to optimize using SWAP algorithm, had the most accurate result of 97.73, with the lowest possible loss recorded at 0.1286. This finding highlights the benefit of an LSTM + optimization algorithm hybrid algorithm because it facilitates a superior parameter adaptation, resulting in increased learning ability and generalization.

The improvements in the model performance, starting with the conventional ML methods, to the sophisticated hybrid DL models, prove that the LSTM-SWAP model provides the most precise, well-developed, and efficient method of user authentication and cheating detection. Riding on the patterns of sequential data and optimization techniques, the suggested model introduces an important innovation to the security of eLearning, and it proves the significance of DL-based strategies when it comes to behavioral authentication activities.

Although mouse dynamics is a non-invasive and efficient approach to detect the cheating activities in the eLearning settings, the behavioral biometric data raises crucial ethical and privacy concerns. These are the risks that may occur which include misuse of the data, absence of the informed consent and access to sensitive information about behavior without permission. In order to overcome these issues, we are going to make sure that all data gathered in the form of mouse dynamics are thoroughly anonymized and depersonalized. The LSTM-SWAP model is aimed at analyzing the behavioral patterns without associating them with a specific user identity and following such principles of data protection as data minimization and purpose limitation. Moreover, we focus on the significance of open data collection policy and informed user consent before such systems can be implemented in the real-life educational settings. These are necessary to enhance ethical practice and inspire users trust as well as increase the security of online education. Despite the good results of the proposed LSTM-SWAP model, one can draw some shortcomings. The model relies solely on the data of mouse movement, and this is the only limitation because they can be used by the users who use a computer with a mouse. This consequently might not be effective with users on touchscreen such as tablets and smartphones. Moreover, the system concentrates on the behavior of the user at the eLearning site and fails to monitor any form of cheating, which occurs outside the system, like chatting on messaging apps or even using the phone. These issues can be resolved in future work with the help of data of various forms of user interaction and research of the possibilities to identify group or external cheating behaviors.

3. Contribution 02: CNN-OOA-Based Cyber Threat Detection: Protecting E-Learning from Phishing [221].

3.1 Description

The shift to online education has led to the rapid growth of e-learning platforms, such as MOOCs and open classrooms, which offer flexibility, broader access, and self-paced learning, especially during the COVID-19 pandemic. However, these platforms also face significant cybersecurity risks, including data breaches, malware, phishing, and DoS attacks, exposing sensitive student information. Despite the increasing importance of these platforms, research on their security risks remains limited. To address these challenges, ML and DL methods, such as the CNN-OOA framework, have emerged. This hybrid model combines CNN with Orca Optimization Algorithm (OOA) to enhance phishing detection by analyzing the structure and text of phishing emails and websites, achieving impressive accuracy rates of 96.83% and 98.04%, respectively. The CNN-OOA system is designed to protect e-learning platforms from cyberattacks while maintaining user privacy and offering real-time threat detection, making it an effective solution to safeguard sensitive data in online education.

3.2 Methodology

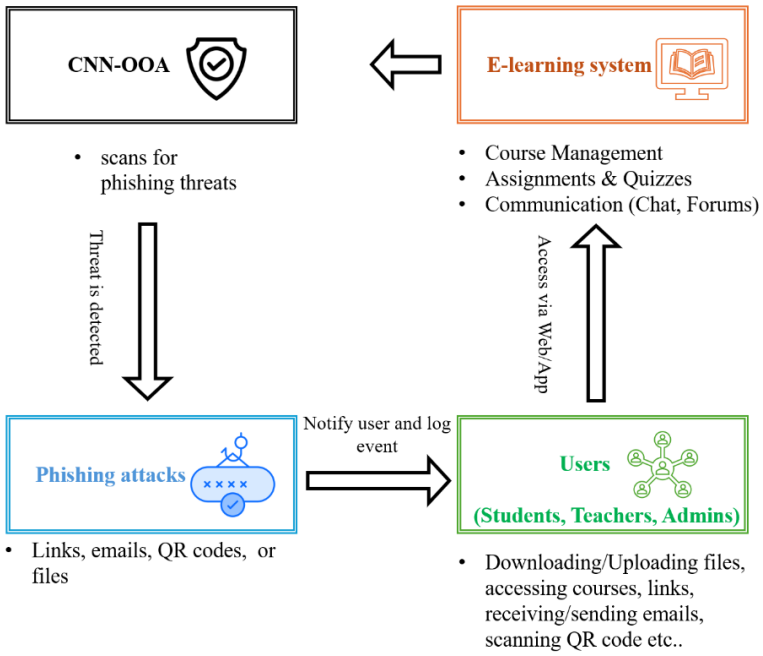
3.2.1 Overview of the Proposed CNN-OOA Framework

The CNN–OOA architecture involves the combination of CNN-based feature extraction power with OOA for auto-tuning of hyperparameters. In this architecture, the convolutional layers learn to extract discriminative patterns from sequential data (e.g., phishing emails and malicious URLs), while pooling layers reduce dimensionality, and fully connected layers carry out the final classification. Inspired by the cooperative hunting behavior of orcas, OOA can effectively traverse hyperparameter space and tune search parameters such as learning rate, dropout ratio, etc. This hybrid approach provides considerable improvement on the phishing detection while decreasing false positives and increasing computation efficiency with respect to both precision and recall. This truly enables CNN–OOA to converge faster and generalize well, which is a good candidate for practical e-learning environments as more phishing attacks are occurring these days.

The system guarantees safe e-learning by detection and reaction in real time to any phishing threat, as shown in Figure 10. User requests (e.g., URLs, emails, QR codes, or uploaded files) are first handled by CNN–OOA to identify possible phishing content.

Detected threats are flagged or blocked, and alerts are generated to users and events stored for further security analysis.

Figure 10 : The architecture of secure E-Learning by detecting and preventing phishing threats in real time.



3.2.2 The convolutional neural network (CNN)

A 1D-CNN is a deep network architecture employed to classify sequential data, such as detection of threats in cybersecurity. It processes input data with a time-based or sequential profile and consists of the following major components:

- **Input layer:** Takes sequential input data in the form of numeric (e.g., feature vectors) or text (e.g., tokenized strings). The input data is preprocessed or reshaped to be compatible with 1D convolutional layers.
- **Convolutional layers (Feature extraction):** Multiple 1D convolution layers obtain local features and patterns from the sequential input. Each convolutional layer applies a series of convolutions to the input, extracting meaningful spatial or temporal relationships. Following each convolution, a ReLU activation function is used to introduce non-linearity and enable the model to learn complex patterns. Padding = 'same' is generally applied to maintain the length of the sequence input, i.e., so that no data on the edges are lost.
- **Pooling layers (Dimensionality reduction):** are employed after some convolutional layers to reduce the feature map's dimension. Pooling retains important information and reduces computation to prevent overfitting.

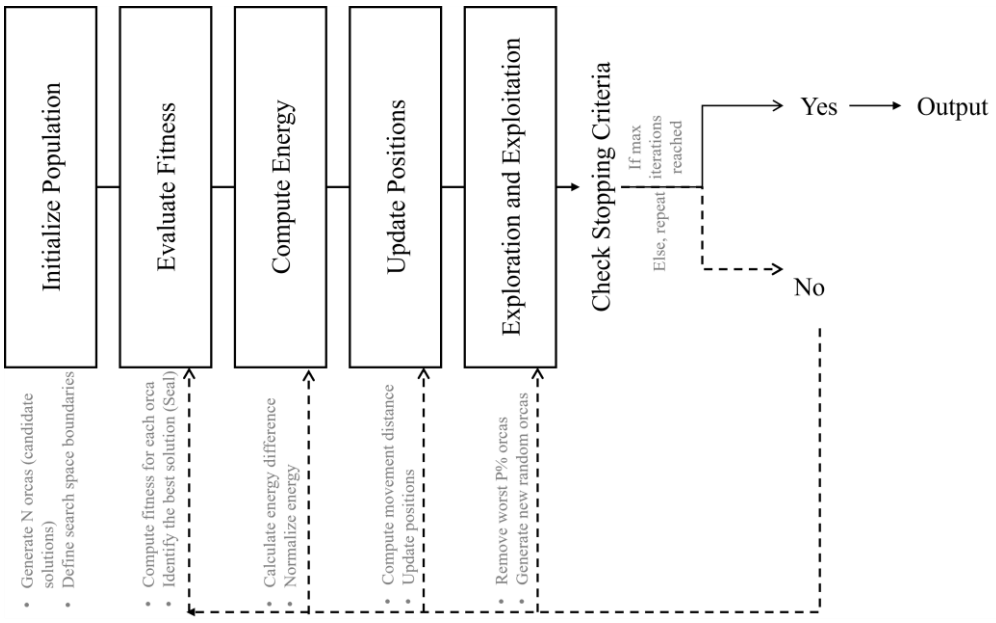
- **Fully connected layers (Classification):** Features are flattened into a 1D vector to convert from spatial or temporal to dense representation.
- **Dense layers** One or more dense (fully connected) layers are employed for processing the flattened features, wherein activation functions are incorporated for the introduction of non-linearity.
- **Dropout layers**: are usually included between dense layers to regularize and avoid overfitting by disabling random neurons during training.
- **Output Layer (Classification Decision):** There is a final dense layer with softmax activation that produces the probability distribution over the target classes. Softmax output supports multi-class classification, which provides probabilities for all classes.

3.2.3 Orca Optimization Algorithm (OOA)

The orcas employ a strategy to clean the ice floe by beating it, which reduces its sheer size and gets the seal closer to them. Hunting thousands of feet from the ice floe, orcas can gain momentum and power, creating surges that diminish the ice floe. Orcas by the ice floe emit shorter signals because they are not blowing as hard. This process is the knapsack solution algorithm from housing zero to hit the seal, as shown in Figure 11.

OOA is a metaheuristic algorithm employed to hunting behavior of orcas (killer whales) for catching seal on ice flake. The OOA is an effective method that mimics the energy-based and random propagation of orcas in a specific search space to find the best solution.

Figure 11 : The flowchart of OOA.



The OOA mechanism (Figure 11):

3.2.3.1 Initialize Population (Candidate Solutions):

The initial population consists of N orcas, each randomly placed in the search space.

- The search space is the set of all possible solutions to an optimization problem. Each orca represents a candidate solution.
- Each orca's position in the search space is a potential solution to the optimization problem:

$$X_i = (x_{i_1}, x_{i_2}, \ldots, x_{i_D}) \quad (1)$$

Where: D is the dimensionality of the search space.

- Each orca's position is initialized randomly as:

$$X_i = X_{min} + (X_{max} - X_{min}) \times rand() \quad (2)$$

Where:

- X_{min} and X_{max} define the lower and upper bounds of the search space.
- $rand()$ is a uniform random number in the range $[0,1]$.
- N is the total number of orcas in the population.

- The radius of the ice floe is initialized as:

$$R = R_0 \quad (3)$$

Where: R_0 is a predefined constant.

3.2.3.2 Fitness function:

The fitness function evaluates how good each orca's position is compared to others.

- The fitness of an orca is given by:

$$F_i = F(X_i) \quad (4)$$

- The best solution found so far is assigned as the seal position:

For minimization problem: $X_s = arg \min_{X_i} F(X_i)$ (5)

For maximization problems: $X_s = arg \max_{X_i} F(X_i)$ (6)

3.2.3.3 Computing the Energy:

Energy represents the ability of an orca to wash the ice floe. The farther an orca is from the seal, the higher its energy.

- The energy of each orca is:

$$e_i = F_i - F_s \quad (7)$$

Where:

- e_i is the energy of orca i.
- F_s is the fitness of the best solution (seal).
- Higher e_i means the orca is farther from the best solution.

- To normalize energy:

$$E_i = \frac{e_i - e_{min}}{e_{max} - e_{min}} \quad (8)$$

Where:

- e_{max} and e_{min} are the maximum and minimum energy values in the population.
- E_i is the normalized energy, ranging between 0 and 1.

3.2.3.4 Update Positions:

Orcas move toward the best solution (seal), with stronger orcas moving farther.

- The movement distance is:

$$d_i = E_i \times R \quad (9)$$

Where:

- d_i is the step size for the orca.
- R is the current radius of the ice floe.
- Higher energy E_i leads to larger movements.

- The new position of an orca is:

$$X_i^{new} = X_s + d_i \times (rand() - 0.5) \quad (10)$$

Where: $rand() - 0.5$ introduces randomness to ensure diverse exploration.

3.2.3.5 Exploitation and Exploration :

- **Exploitation (Reduce Ice Floe Radius):** The ice floe shrinks over time, reducing the search space and allowing the algorithm to focus on the best solutions:

$$R = \alpha \times R \quad (11)$$

Where: α is the shrinkage factor ($0.9 \leq \alpha \leq 1$).

This ensures that orcas converge towards the best solution as the search progresses.

- **Exploration (Avoiding Local Minima):** To prevent the algorithm from getting stuck in local minima, the worst-performing orcas are replaced with new random solutions.
  - A fraction P of the weakest orcas is eliminated and replaced with new ones:

$$X_i^{new} = X_{min} + (X_{max} - X_{min}) \times rand() \quad (12)$$

Where: P is the elimination rate (e.g., 10%-20% of the population).

This step ensures that new orcas bring fresh diversity to the population.

3.2.3.6 Convergence Criteria:

The algorithm stops when a solution is good enough or when maximum iterations are reached.

- Maximum number of iterations reached.

$$t \geq t_{max} \quad (13)$$

- No significant improvement over iterations.

$$| F_{best}^{(t)} - F_{best}^{(t-1)} | < \epsilon \quad (14)$$

Where: ϵ is a small threshold (e.g., 10^{-6}).

3.2.4 Optimize Hyperparameters Using OOA

Hyperparameter optimization is the process of selecting the best set of hyperparameters for a ML model to improve its performance. OOA can be used as an optimization method to find the optimal set of hyperparameters by treating each candidate solution (set of hyperparameters) as an orca in the search space.

3.2.4.1 Hyperparameter Space :

- The set of hyperparameters can be represented as a D-dimensional vector:

$$X = (h_1, h_2, \ldots, h_D) \in \mathbb{R}^D \quad (15)$$

Where:

- D is the number of hyperparameters.
- Each h_j is a hyperparameter that belongs to a predefined domain H_j:

$$h_j \in H_j \quad (16)$$

- The search space H is the Cartesian product of all hyperparameter domains:

$$H = H_1 \times H_2 \times \ldots \times H_D \quad (17)$$

3.2.4.2 Objective Function:

The performance of a model is evaluated using a loss function or performance metric, denoted as:

$$F: H \rightarrow \mathbb{R} \quad (18)$$

Where: $F(X)$ measures the model's error (e.g., validation loss, classification error).

- The goal is to minimize $F(X)$:

$$X^* = arg \min_{x \in H} F(X) \quad (19)$$

- For cases where a metric (e.g., accuracy, F1-score) is maximized, we reformulate it as:

$$X^* = arg\max_{x \in H} G(X) \quad (20)$$

Where: $G(X)$ is a performance metric such as accuracy.

3.2.4.3 Categorization of Hyperparameters:

Hyperparameters can be categorized based on their type:

- **Continuous Hyperparameters:** These take value in a real-valued range:

$$h_j \in [a_j, b_j] \subset \mathbb{R} \quad (21)$$

Example: Learning rate $h_j \in [10^{-5}, 10^{-1}]$

- **Discrete Hyperparameters:** These take values from a finite set:

$$h_j \in \{h_1, h_2, \ldots, h_K\} \quad (22)$$

Example: Batch size $h_j \in \{16, 32, 64, 128\}$.

- **Categorical Hyperparameters:** These are non-numeric choices:

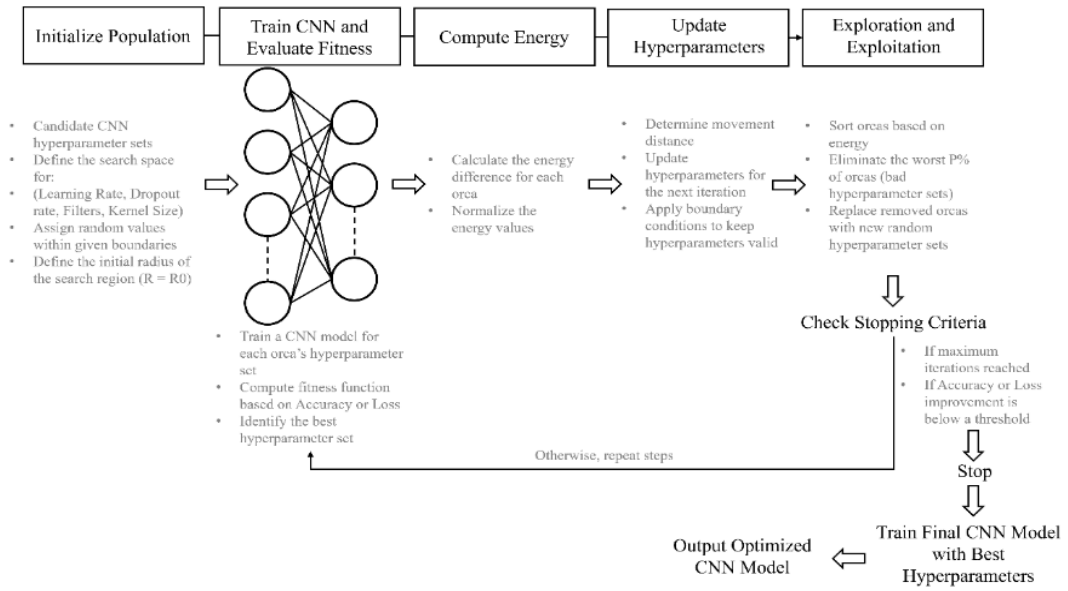
$$h_j \in \{A, B, C, \ldots\} \quad (23)$$

Example: Optimizer $h_j \in \{\text{SGD, Adam, RMSprop}\}$.

3.2.5 CNN-OOA

This framework aims to optimize the hyperparameters of a CNN using the OOA. The process involves iterative updates to improve CNN performance by selecting the best hyperparameter configuration, as shown in Figure 12.

Figure 12 : The architecture of CNN-OOA.



Step1: Initialization: At the start, an initial population of orcas is generated, where each orca represents a candidate CNN hyperparameter set.

- Define the search space for the hyperparameters, including:
  - Learning Rate h_1
  - Dropout Rate h_2
  - Number of Filters h_3
  - Kernel Size h_4
  - Batch Size h_5
- Assign random values to each hyperparameter from their respective domains.
- Define the initial search radius R_0, which determines how far the orcas explore in the search space.

Each orca X_i is represented as:

$$X_i = (h_1, h_2, h_3, h_4, h_5) \quad (24)$$

Where: H is the predefined hyperparameter space.

Step2: Train CNN and Evaluate Fitness

Each orca's hyperparameter set is used to train a CNN model.

- Train a CNN using the given hyperparameters X_i.
- Compute the fitness function $F(X_i)$ based on CNN accuracy (or another performance metric like validation loss).
- Identify the best-performing orca (best hyperparameter set) as the seal X_s.

The fitness function can be defined as:

$$F(X_i) = Accuracy(X_i) \quad (25)$$

or

$$F(X_i) = -Loss(X_i) \quad (26)$$

The goal is to maximize accuracy (or minimize loss) to find the optimal X_i.

Step 3: Compute Energy: Calculate energy differences and normalize energy values to determine movement capability.

Step 4: Update Hyperparameters: Each orca updates its hyperparameter values based on its normalized energy.

Step 5: Exploration and Exploitation: To prevent stagnation in local optima and maintain diversity, the algorithm introduces an exploration mechanism by periodically replacing the weakest solutions.

- Sort orcas based on energy.
- Eliminate the worst $P\%$ of the population (i.e., hyperparameter sets with the lowest fitness values).
- Replace eliminated orcas with newly generated random hyperparameter sets to introduce diversity into the search space.

This ensures a balance between exploitation (fine-tuning the best solutions) and exploration (introducing new potential solutions).

Step 6: Check stopping criteria and train the final CNN model: If stopping criteria are met and the best hyperparameters X^* are found Train a final CNN model using X^*.

3.2.6 Pseudo code

The pseudo-code for CNN hyperparameter optimization using the OOA (CNN-OOA):

Inputs:

- H : Hyperparameter search space
- N : Number of orcas (population size)
- T_max : Maximum number of iterations
- P : Elimination rate for weakest orcas (e.g., 10–20%)
- ε : Convergence threshold (minimum improvement in fitness)
- R0 : Initial search radius

```
Function CNN_OOA (H, N, T_max, P, ε, R0)
    # Step 1: Initialize Population
    population ← InitializeOrcas(H, N)
    R ← R0   # Set initial search radius
    # Step 2: Train CNN and Evaluate Fitness
    For each orca X_i in population:
        model ← TrainCNN(X_i)
        fitness[i] ← ComputeFitness(model)
    # Identify the best orca (seal)
    X_s ← BestSolution(population, fitness)
    # Main Optimization Loop
    While (stopping condition not met) do
        # Step 3: Compute Energy
        For each orca X_i in population:
            e_i ← ComputeEnergy(fitness[i], fitness_s)
            E_i ← NormalizeEnergy(e_i)
        # Step 4: Update Hyperparameters
        For each orca X_i in population:
            d_i ← ComputeDistance(E_i, R)
            X_i ← UpdateHyperparameters(X_i, d_i)
            ApplyBoundaryConstraints(X_i, H)
        # Step 5: Exploration and Exploitation
```

```
23                      Sort population by energy
24                      RemoveWorstOrcas(P, population)
25                      ReplaceOrcas(population, H, P)
26                      # Step 6: Check Stopping Criteria
27                      If (iteration == T_max) OR (|fitness_best_new - fitness_best_old| < ε) then
28                              break
29              # Step 7: Train Final CNN Model
30              X_best ← BestSolution(population, fitness)
31              final_model ← TrainCNN(X_best)
32              Return final_model
33      End Function
```

3.2.7 Dataset

CNN-OOA utilized two datasets. The first is the Phishing Email Detection dataset, downloaded from Kaggle, which contains 18,650 email samples, with 11,323 labeled as safe emails and 7,327 as phishing emails. The dataset includes the email text body and email type, with text lengths varying significantly, averaging 2,753 characters.

The Phishing Email Detection dataset for CNN-OOA is preprocessed by cleaning text, encoding labels, tokenizing, and splitting the data.

- The email text is cleaned by removing HTML tags, URLs, and special characters, converting it to lowercase, and stripping extra spaces.
- The email type labels are then encoded into numerical values, with Safe Email as 1 and Phishing Email as 0.
- To prepare the text for DL, a Keras Tokenizer is used to convert words into sequences, keeping only the 10,000 most frequent words. Each sequence is padded to a maximum length of 300 to ensure uniform input size.
- The dataset is split into training (80%) and testing (20%) sets to train and evaluate the model.

The second is Phishing E-learning Websites dataset, collected during our research, made up of 19,166 URLs for which each represents 9,583 safe e-learning and 9,583 phishing websites. The safe e-learning URLs were manually selected from some genuine learning portals, while the phishing URLs were collected from CyberCrime Tracker, presenting a true and unbiased dataset for phishing detection.

Each URL is labeled as phishing or safe and subsequently passed through feature extraction, yielding numerical features that capture critical URL characteristics, including:

- Length features: URL length and domain length.
- Structural features: Hyphen count, dot count, and subdomain count.

- Security markers: Presence of HTTPS and IP addresses in the URL.
- Special character use: Special character '@', '?', '&', '=', and '/' frequency.
- Query parameters: Count of query parameters in the URL.

After post preprocessing, all the non-numeric features such as the original URL and top-level domain (TLD) are removed, and the model is trained only on numeric features. Numerical encoding (safe = 0, phishing = 1) is done for labels. The dataset is split into training (80%) and testing (20%) sets and normalized using StandardScaler to standardize feature values.

3.3 Results and comparison

This section presents the results obtained by CNN-OOA and compares them with other approaches. The evaluation uses accuracy to analyze the performance of two datasets, Phishing Email Detection dataset and Phishing E-learning Websites dataset.

- **Phishing Emails**

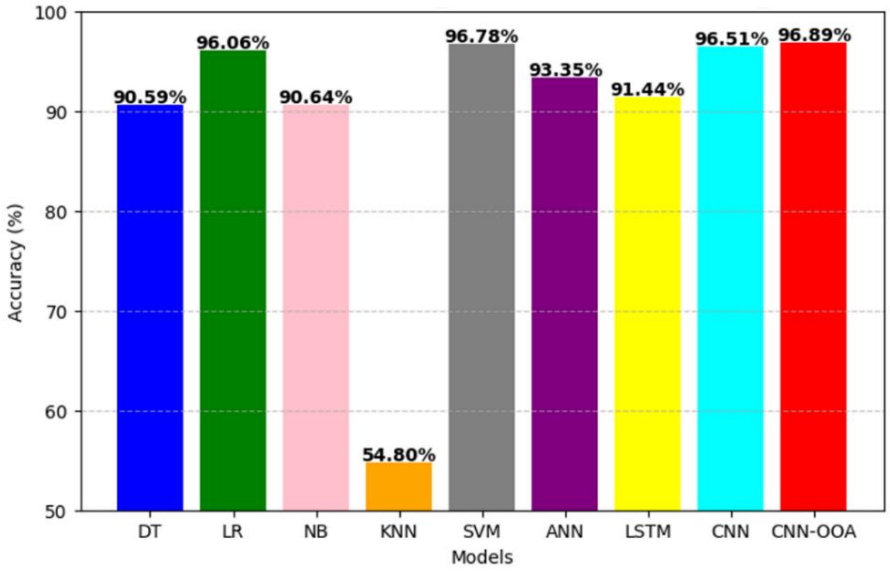
The comparative evaluation of various ML and DL models for phishing emails dataset provides a comprehensive understanding of their effectiveness in handling cybersecurity threats. The models examined (DT, LR, NB, KNN, SVM, ANN, LSTM, CNN, and CNN-OOA) show distinct strengths and weaknesses in terms of accuracy, computational efficiency, and adaptability to phishing data.

Figure 13 presents the results of various ML and DL models that trained on a phishing email detection dataset. Among the ML models, DT achieves an accuracy of 90.59%. While interpretable, this model struggles with capturing complex patterns in text data. LR achieves an accuracy of 96.06%, highlighting the strength of linear models in phishing email classification. The NB model, with an accuracy of 90.64%, demonstrates the effectiveness of probabilistic approaches for this task. KNN, which has the lowest accuracy (54.80%) among the traditional models, illustrates its limitations in handling textual features effectively. Meanwhile, SVM achieves the highest accuracy (96.78%) among traditional models, although it may suffer from computational inefficiencies with large datasets.

For DL methods, ANN achieves an accuracy of 93.35%, performing better than some classical models but falling short of more advanced DL architectures. LSTM reaches 91.44%, demonstrating the capability of sequential models to capture dependencies in text data. CNN achieves an accuracy of 96.51%, showcasing the advantage of convolutional networks in learning local patterns. Finally, CNN-OOA, an optimized version of CNN, achieves the highest

accuracy of 96.89%, demonstrating that the optimization strategy enhances the model's ability to detect phishing emails more effectively.

Figure 13 : The accuracy of the evaluated models in phishing email detection dataset



A 1D-CNN used with OOA for hyperparameter tuning. The OOA generates 15 random sets of hyperparameters to initialize the population and further enhances them over 10 iterations, as shown in Table 7. For each iteration, the CNN is trained for 10 epochs to verify the performance of the hyperparameters.

The OOA evaluates the hyperparameters during CNN training and validation. It updates the population by favoring configurations with higher accuracy. The algorithm then updates the population of hyperparameter by favoring hyperparameter configurations that yield higher accuracy and replacing the worst-performing sets with new random combinations. CNN-OOA performs 10 iterative updates to the hyperparameters and shifts its search space to concentrate on hyperparameter ranges that improve model performance. Following 10 iterative updates, the model achieved 96.89% accuracy with the following hyperparameters: an embedding dimension of 207, a kernel size of 3, 88 filters, a dropout rate of 0.5, a learning rate of 0.0001, and a batch size of 82. The OOA explores and iteratively refines hyperparameters. This iterative process ensures the model converges to a high-performing configuration, demonstrating the effectiveness of the OOA in hyperparameter optimization.

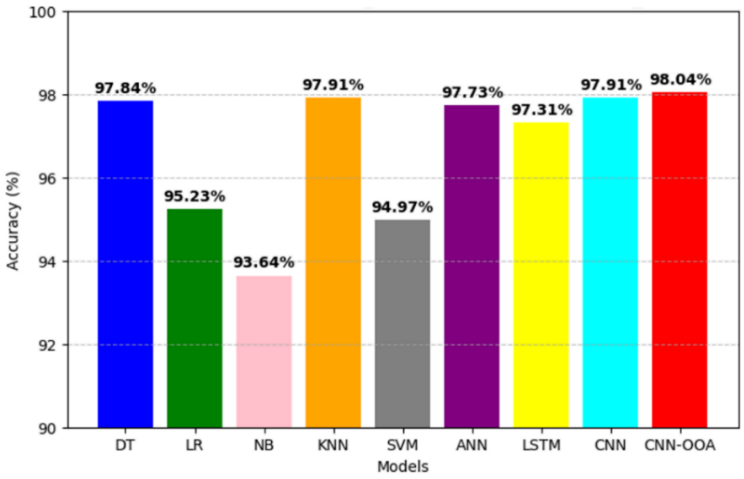
Table 7: Best accuracy over 10 iterations tested on phishing emails dataset

Iteration	Embedding	Kernel	Filters	Dropout	Learning Rate	Batch Size	Best accuracy
Initial Best Solution	207	4	88	0.2309000657213245	0.0006454357165594786	78	96.59%
1	208	3	88	0.5	0.0001	78	96.78%
2	208	3	87	0.5	0.0001	78	96.83%
3	207	3	88	0.5	0.0001	79	96.78%
4	208	3	87	0.5	0.0001	78	96.80%
5	207	3	88	0.5	0.0001	82	**96.89%**
6	198	3	69	0.1	0.0001	101	96.83%
7	207	3	88	0.39848723269013236	0.0001	82	96.86%
8	212	3	91	0.1	0.0001	82	96.72%
9	157	3	70	0.5	0.0001	55	96.70%
10	179	3	120	0.1	0.0001	46	96.62%

- **Phishing Websites**

Various ML and DL models were trained on a phishing website detection data set using numeric features, as shown in *Figure 14*. Among the ML models, DT achieves an accuracy of 97.84%, showing its effectiveness in handling structured numerical data. LR reaches 95.23%, confirming that linear models can provide strong baselines for phishing website classification. The NB model, with an accuracy of 93.64%, performs slightly lower than other traditional models, likely due to its assumption of feature independence. KNN achieves 97.91%, demonstrating its ability to classify phishing websites effectively when numerical distance-based comparisons are meaningful. SVM, with an accuracy of 94.97%, performs well but does not surpass other traditional models in this task.

Figure 14 : The accuracy of the evaluated models in phishing websites dataset.



For DL models, ANN achieves an accuracy of 97.73%, proving to be highly competitive with traditional methods. LSTM reaches 97.31%, showing the effectiveness of sequential modeling even for numerical features. CNN achieves 97.91%, leveraging its capability to detect local patterns in structured numerical data. Lastly, CNN-OOA, an optimized version of CNN, achieves the highest accuracy of 98.04%, demonstrating that optimization techniques further enhance phishing website detection performance.

Following the phishing email detection step, OOA initializes 15 random sets of hyperparameters and conducts 10 iterations, as shown in Table 8, each involving 10 epochs of training, to evaluate performance. The hyperparameters optimized include filters, kernel size, dropout, learning rate, and batch size. OOA trains the CNN and tests its accuracy on the validation set. The population of hyperparameter sets is updated to prioritize those with higher accuracy, swapping out the worst-performing sets for new random combinations. In the 10 iterations, the best accuracy achieved was 98.04%, and the optimal hyperparameters utilized were 95 filters, a kernel size of 2, a dropout rate of 0.1, a learning rate of 0.01, and a batch size of 83. Systematic and iterative OOA hyperparameter exploration guarantees that the model achieves an optimal configuration, resulting in high accuracy for phishing URL detection.

Table 8: Best accuracy over 10 iterations tested on phishing websites dataset

Iteration	Filters	Kernel	Dropout	Learning Rate	Batch Size	Best accuracy
Initial Best Solution	89	2	0.18	0.00143	81	97.94%
1	95	2	0.1	0.01	83	**98.04%**
2	101	4	0.5	0.01	57	97.91%
3	96	2	0.1	0.01	90	97.91%
4	128	2	0.1	0.01	109	97.57%
5	116	4	0.5	0.01	91	97.94%
6	98	2	0.1	0.01	89	97.73%
7	98	4	0.5	0.01	102	97.97%
8	93	4	0.1	0.01	90	97.78%
9	99	2	0.1	0.01	85	97.78%
10	94	4	0.1	0.01	80	97.81%

The optimal hyperparameters differ significantly due to the nature of the datasets, as shown in Table *9*. Phishing emails are aided by an embedding dimension of 207, a kernel size of 3, and a dropout rate of 0.5 due to the complexity and variability of text data. On the other hand, the phishing URL detection model performs best with a kernel size of 2, a dropout rate of 0.10, and a learning rate of 0.01, as these settings are better suited to simpler numerical features and

minimize overfitting. OOA is an effective technique for hyperparameter optimization, as it natively incorporates such differences into the optimization process and uses tailored hyperparameters for each task. The phishing URL detection model shows slightly higher accuracy (98.04%) than the phishing email detection model (96.89%) because numeric features in URLs are much more distinctive than text features in emails.

Table 9: The best hyperparameters according to the highest accuracy.

Metric	Phishing emails	Phishing websites
Input type	Text (Tokenized)	Numeric Features
Preprocessing	Tokenization, Embedding	Normalization, Reshaping
Accuracy	96.89%	98.04%
Learning rate	0.0001	0.01
Kernal size	3	2
Dropout rate	0.5	0.1
Batch size	82	83
Embedding	88	95

The comparison between the standard CNN and the CNN optimized with the OOA algorithm (over 10 epochs) on phishing emails dataset shows significant differences in terms of performance and generalization as shown in **Error! Reference source not found.** and Table 5 presents the performance metrics of the LSTM-SWAP model evaluated at 10, 50, and 100 training epochs. Metrics include accuracy, precision, recall, F1-score, FPR and AUC-ROC, highlighting the model's progressive improvement in classification performance as training epochs increase.

. The standard CNN achieved high training accuracy early on, but its validation loss varies and does not decrease consistently, which may indicate overfitting. CNN-OOA generalizes more, with its training and validation loss decreasing more gradually and consistently. It also has improved validation accuracy, which is closer to its training accuracy than the baseline CNN, indicating overfitting in the baseline CNN. OOA optimizes hyperparameters such as dropout rate, kernel size, and learning rate for improved CNN performance.

Figure 15 : Training history of CNN on phishing emails dataset.

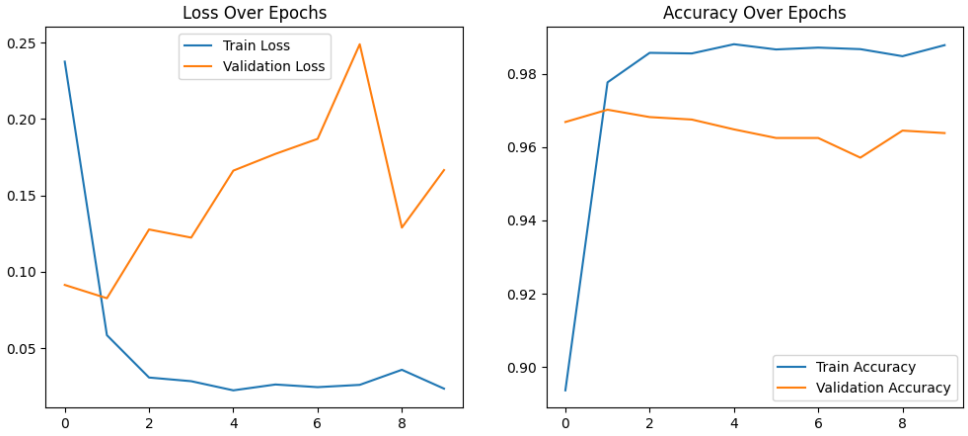


Figure 16 : Training history of CNN-OOA over 10 Iterations on phishing emails dataset.

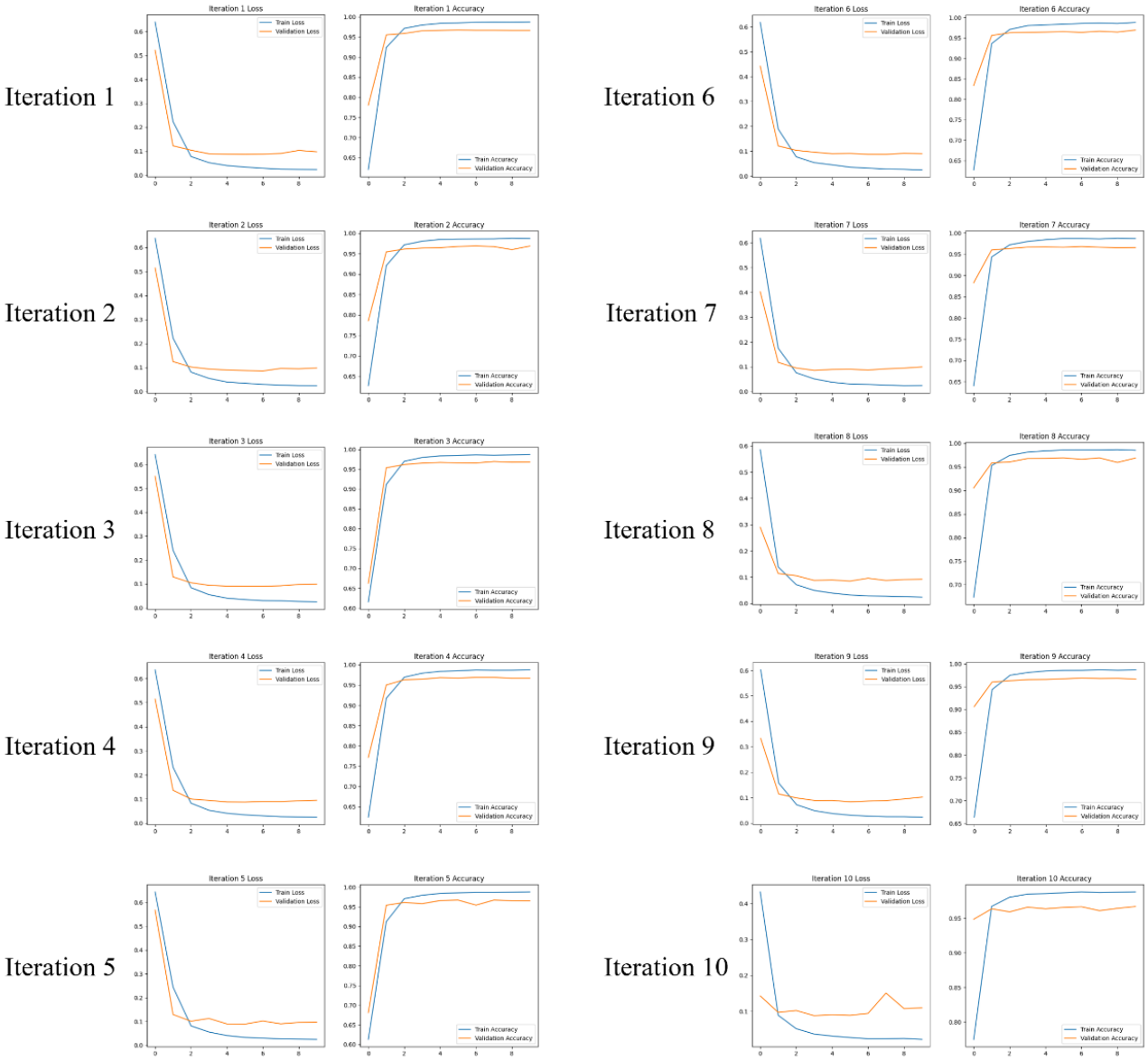


Figure 17 : Training history of CNN on phishing websites dataset.

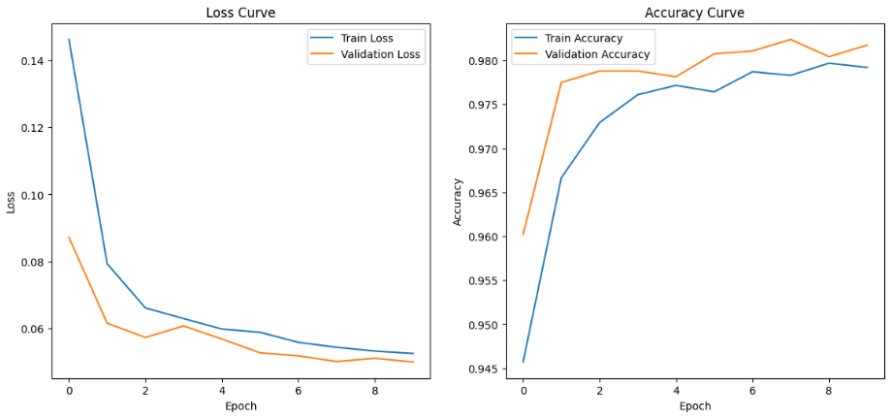
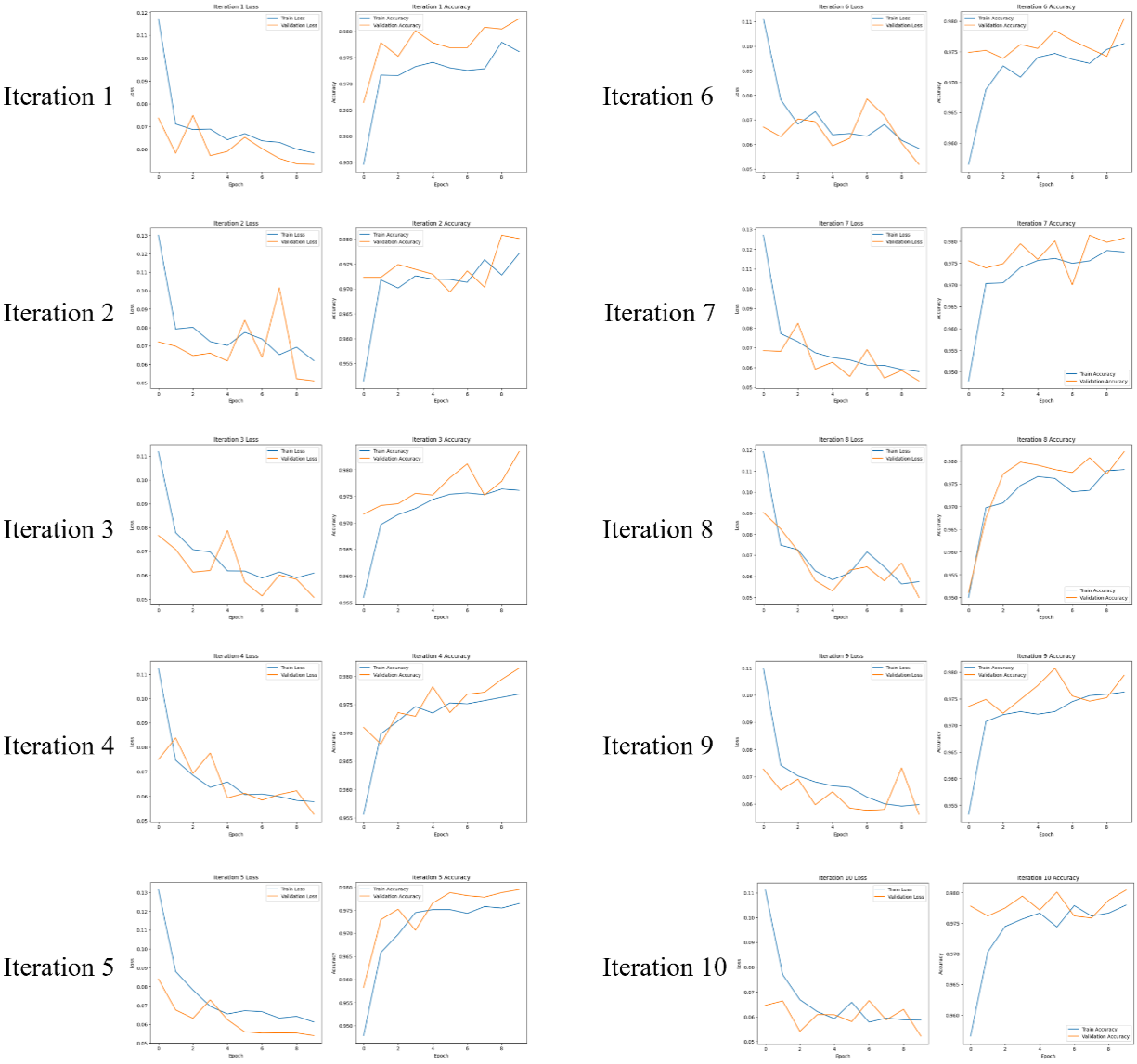


Figure 18 : Training history of CNN-OOA over 10 Iterations on phishing websites dataset.



The variance in training histories of the CNN and CNN-OOA suggests the impact of optimization on model performance. In the first image, the CNN model illustrates a decreasing loss curve, as shown in Figure 17, but the significant gap between training and validation loss indicates potential overfitting, while the training accuracy increases consistently, while validation accuracy stays stable.

Figure 18 highlights the role of the OOA and depicts the training curve of CNN-OOA for 10 iterations. The iterative optimization results in a lower and more stable loss, indicating faster and more stable convergence. The accuracy curves also show a steady increase, with validation accuracy closely tracking training accuracy, reflecting enhanced generalization. In summary, CNN-OOA enhances model performance through dynamic hyperparameter selection during

training, preventing overfitting, and producing more consistent results across different training runs.

Table 10: The performance of different models on phishing email and website datasets

Model	Phishing Emails (Accuracy)	Phishing Websites (Accuracy)
DT	90.59%	97.84%
LR	96.06%	95.23%
NB	90.64%	93.64%
KNN	54.80%	97.91%
SVM	96.78%	94.97%
ANN	93.35%	97.73%
LSTM	91.44%	97.31%
CNN	96.51%	97.91%
CNN-OOA	96.89%	98.04%

3.4 Discussion

The findings demonstrate CNN-OOA's resilience in phishing detection, showing superior performance compared to traditional ML and DL approaches. A comparison of phishing email and phishing website datasets offers insights into the effects of different architectures and optimization strategies on model accuracy and generalization.

The CNN-OOA model outperforms all standard classifiers, as shown in Table 10, including DT, LR, NB, KNN, and SVM, in phishing email and website classification across all experiments. Among the traditional models, SVM achieves the highest accuracy (96.78%) for phishing email classification, while KNN achieves the highest accuracy (97.91%) for phishing website classification. While these models suffer from disadvantages such as data distribution sensitivity (NB) and computational inefficiencies (SVM, KNN), they are less suitable for large-scale phishing detection.

DL methods, including ANN, LSTM, and CNN, are better than the traditional classifiers since they can learn complex feature representations. CNN achieves an accuracy of 96.51% for phishing email detection and 97.91% for phishing website classification, outperforming both ANN and LSTM. CNN's performance is enhanced by the OOA, reaching 96.89% accuracy for phishing emails and 98.04% for phishing websites. The increase in accuracy highlights the role of hyperparameter tuning in DL models. Parameters such as learning rate, kernel size, dropout rate, and batch size are essential for improving model generalization and stability.

The application of OOA in training CNN produces more robust and stable results across the two datasets studied. Compared to typical CNN training, where hyperparameter selection is typically unstructured and inefficient, OOA optimizes across numerous diverse hyperparameter settings and continually refines them based on validation performance. The findings indicate that:

- For phishing emails, embedding dimension of 207, kernel size of 3, dropout rate of 0.5, and learning rate of 0.0001 provided the highest accuracy.
- For phishing websites, kernel size of 2, dropout rate of 0.1, and learning rate of 0.01 were the optimal configurations.

The differences between the best hyperparameters reveal that text-based phishing emails necessitate more intricate feature extraction, characterized by a higher embedding dimension and dropout rate. Meanwhile, phishing websites with structured numerical features achieve optimal performance with a lower dropout rate and a smaller kernel size. This underscores the necessity of dataset-specific hyperparameter tuning, a task that OOA effectively handles.

The training history comparisons between standard CNN and CNN-OOA reveal key differences in generalization. Standard CNN exhibits a larger gap between training and validation loss, indicating overfitting, particularly on the phishing email dataset. In contrast, CNN-OOA demonstrates improved convergence, with validation accuracy closely tracking training accuracy, suggesting that OOA prevents overfitting by optimizing dropout rates and batch sizes effectively. The iterative nature of OOA ensures that the search space is gradually refined, avoiding configurations that lead to poor generalization. The findings of this study have significant practical implications for cybersecurity, particularly in phishing detection systems deployed in real-world environments. By leveraging CNN-OOA, organizations can achieve higher detection accuracy, reducing false positives and false negatives in phishing email and website classification. The automated hyperparameter tuning process minimizes the need for manual experimentation, making it more feasible for deployment in dynamic threat landscapes where phishing techniques constantly evolve. The integration of OOA with CNN significantly enhances phishing detection accuracy, outperforming traditional ML and DL models. The results validate the effectiveness of hyperparameter optimization in improving model generalization and preventing overfitting. Future work can extend this approach to other cybersecurity domains, ensuring more robust and adaptable phishing detection frameworks.

4. Contribution 03: A hybrid deep learning method employing both of Convolutional Neural Network and Grey Wolf Optimization for intrusion detection attacks

4.1 Description

The rapid expansion of digital communication and online transactions has led to an increase in cyber threats, with phishing attacks becoming one of the most common forms of cybercrime. These attacks involve tricking individuals into disclosing personal information by impersonating trusted entities, making them increasingly difficult for both individuals and cybersecurity experts to detect. Traditional defense mechanisms, such as heuristic-based and signature-based intrusion detection systems, have proven ineffective against sophisticated, evolving cyberattacks like network intrusions, DoS attacks, and exploitation of vulnerabilities. These systems rely on rigid, predefined rules, which quickly become obsolete due to the continuous innovation of cybercriminals. In response, there has been a shift towards more advanced technologies, particularly DL, which uses multi-layered neural networks to automatically learn and extract complex data patterns without the need for manual feature extraction. DL models are highly effective at detecting subtle deviations and complex patterns that traditional methods miss, and the accessibility of DL through platforms like MOOCs and OpenClassrooms has accelerated its adoption in cybersecurity. By constructing dynamic and adaptable network intrusion detection systems, DL offers the potential for continuous improvement and greater defense against evolving threats. These models are capable of detecting signs of malicious activity that are too complex for human detection, significantly reducing the success rates of cyberattacks. This work proposes a novel model that combines CNNs with the Grey Wolf Optimization (GWO) algorithm, optimizing the CNN's hyperparameters for enhanced precision in intrusion detection. The combination of CNN and GWO provides an effective tool for detecting and classifying various types of network intrusions, offering a robust solution to counter the growing and ever-changing landscape of cyber threats.

4.2 Methodology

4.2.1 Preprocessing phase

The preprocessing stage of the DL method involves preparing the dataset for introduction into the detection approach. The procedures we implemented in this stage were as follows:

- **Loading the Dataset:** the first step in preprocessing is to load the NSL-KDD dataset. which is split into the training set (KDDTrain+.txt) and the test set (KDDTest+.txt).
- **Assigning Column Names:** the NSL-KDD data set consists of 41 variables and one label of being a normal connection or an attack. It is imperative to give these columns meaningful descriptions that make their readability easy and prompt the analysis.
- **Handling Categorical Features:** The dataset includes protocol type, service and flag. To enable DL models to process these features, they must be turned into numeric data. For this, one-hot encoding is a frequently used method.
- **Label Encoding:** Label Encoding: the column ranging by label is used to show a relation to whether a network connection is either normal or an attack. To do binary classification tasks, it is helpful to put these labels into two categories, 'normal' connections being coded as 0 and attack connections coded as 1.
- **Feature scaling:** is an essential one because it makes sure that all features are equally important in the model. In the absence of scaling, the features that have bigger ranges may overrule the features that have smaller ranges and hence the model performance is not optimal. The common one is standardization, under which every feature is scaled to possess a mean of 0 and a standard deviation of 1.
- **Handling Missing Values:** Missing data may also be a problem since it might interfere with the training of the model which would result in inaccurately-predicted values. Missing values should be checked and treated by replacing them with suitable values (the mean or the median) or eliminating the corresponding row or column.
- **Splitting the Dataset:** Dataset is split to the training set and the testing set using an 80-20 “train_test_split” split. This requires to know how well performance on unseen data.

4.2.2 The CNN approach

The CNN model develops a 1D-CNN and optimizes it using a fitness function that learns to sets of key hyperparameters two convolutional filter sizes, dropout rate, and learning rate. An input sequence is fed to the model through two convolutional-max pools where features are extracted, the result is then Flatten to have a dense layer in each sequential one including the last one that belongs to a different activation like a Softmax activation which gives a class probability.

4.2.3 The GWO algorithm overview

The GWO is a population-based optimization algorithm inspired by the leadership hierarchy and hunting behavior of grey wolves in nature. It was introduced by Seyedali Mirjalili in 2014 and has since been widely used for solving various optimization problem.

4.2.3.1 Key Components of GWO:

The GWO algorithm mimics the social behavior of grey wolves, especially how they hunt and organize themselves in the wild. It uses the following key concepts:

- **Hierarchy of Wolves:**

  - **Alpha Wolf (α):** This is the best solution found so far. It guides the rest of the wolves and is responsible for leading the hunting process.
  - **Beta Wolves (β):** These wolves support the alpha and are second in the hierarchy. They provide assistance to the alpha in the decision-making process.
  - **Delta Wolves (δ):** These are lower-ranked wolves that follow the alpha and beta wolves. They contribute to the group's strategy.
  - **Omega Wolves (ω):** These are the least influential wolves. They follow all other wolves and are often seen as "followers."

- **Hunting Mechanism (Exploration and Exploitation):**

- The wolves search for food by exploring their surroundings and then refining their search to find the best solution (exploitation).
- **Exploration:** When a wolf is exploring, it moves randomly, trying to search for new solutions in the search space.
- **Exploitation:** The wolves exploit the known best solutions and move closer to them to refine and improve their findings.

- **Algorithm Steps:**

- **Initialization:**

- The initial population of wolves is randomly generated.
- Each wolf represents a possible solution to the optimization problem.

- **Update Positions:**

The positions of the wolves are updated based on the positions of the alpha, beta, and delta wolves. The mathematical model for updating the position of a wolf is given by:

$$Xi = X\alpha - A \cdot D\alpha \text{ (for the alpha wolf)}$$

$$Xi = X\beta - A \cdot D\beta \text{ (for the beta wolf)}$$

$$Xi = X\delta - A \cdot D\delta \text{ (for the delta wolf)}$$

Where:

- Xi is the current position of the wolf.
- $X\alpha, X\beta, X\delta$ are the positions of the alpha, beta, and delta wolves, respectively.
- A and D are vectors that help control the exploration and exploitation balance.

A is the coefficient that determines the distance between wolves, and D is the distance vector from the current position to the leader's position.

- **Convergence:**

The algorithm keeps updating the positions of the wolves until a stopping criterion is met, such as a maximum number of iterations or an acceptable error threshold.

5.2.3 Implementation and evaluation

5.2.3.1 Tools used

- Operating system: Windows 10 (64-bit)
- Environment: jupyter notebook in anaconda 3
- Programming language: Python 3.11
- The frameworks of ML: Torch.

5.2.3.2 Pseudo code:

The pseudo-code of CNN-GWO

```
Function CNN_GWO(H, N, T_max, lb, ub, SearchAgents_no)
        # Step 1: Initialize Population
        population ← InitializeGreyWolves(SearchAgents_no, H)
        # Step 2: Define Alpha, Beta, and Delta positions
        Alpha_pos ← Zero_Vector(H)  # Best position found so far
```

```
        Beta_pos ← Zero_Vector(H)   # Second-best position:
        Delta_pos ← Zero_Vector(H)  # Third-best position
        Alpha_score ← +∞            # Best fitness score
        Beta_score ← +∞             # Second-best fitness score
        Delta_score ← +∞            # Third-best fitness score
        # Step 3: Main Optimization Loop
        While (stopping condition not met) do
            # Step 4: Evaluate Fitness for each wolf in the population
            For each wolf X_i in population:
                fitness[i] ← fitness_function(X_i)  # Evaluate fitness
                If fitness[i] < Alpha_score then
                    Alpha_score ← fitness[i]
                    Alpha_pos ← X_i
                Else If fitness[i] < Beta_score then
                    Beta_score ← fitness[i]
                    Beta_pos ← X_i
                Else If fitness[i] < Delta_score then
                    Delta_score ← fitness[i]
                    Delta_pos ← X_i
            # Step 5: Update Positions Based on GWO Algorithm
            a ← 2 - Iteration * (2 / T_max)  # Decrease a over iterations
            For each wolf X_i in population:
                For each dimension j in X_i:
                    r1, r2 ← Random_Values()
                    A1 ← 2 * a * r1 - a
                    C1 ← 2 * r2
                    D_alpha ← Abs(C1 * Alpha_pos[j] - X_i[j])
                    X1 ← Alpha_pos[j] - A1 * D_alpha
                    # Update using Beta and Delta positions similarly
                    # Apply boundary constraints
                    X_i[j] ← Clip(X_i[j], lb, ub)
            # Step 6: Explore and Exploit New Solutions
            Sort population by fitness
            RemoveWorstWolves(P, population)
            ReplaceWolves(population, H, P)
            # Step 7: Check Stopping Criteria
            If (iteration == T_max) OR (|fitness_best_new - fitness_best_old| < ε) then
                break
            # Step 8: Train Final CNN Model with Best Hyperparameters
            X_best ← BestSolution(population, fitness)
            final_model ← TrainCNN(X_best)
            Return final_model
End Function
```

6 Dataset

We implemented the open-source NSL-KDD dataset, an improved version of the renowned KDD Cup99 dataset [222], during the process of execution process. Various malicious network techniques, involving DOS, R2L, U2R, and probe assaults, are included in this dataset. Furthermore, in order to better meet the requirements of the studies conducted as part of the experiment, we have improved and used the code for CNN and GWO.

7 Results and comparison

This chapter discusses the results of CNN-GWO and compared with other techniques. In this evaluation, we will use accuracy to assess the effectiveness of the models used with the NSL-KDD dataset. the results of the comparative evaluation of various ML and DL models with the NSL-KDD dataset provide useful understanding of the effectiveness of the models, with respect to coverage of the entire dataset, accuracy of results and computational efficiency of the models with intrusions. The models include in the consideration were: DBN, RNN-GRU, RNN-LSTM, CNN, RNN, GRU-GWO-FS, LSTM-GWO-FS, DE method, PSO method, FPA method and CNN and GWO had differences and differences in model abilities with respect to accuracy, computational efficiency and their flexibility with the intrusion data set.

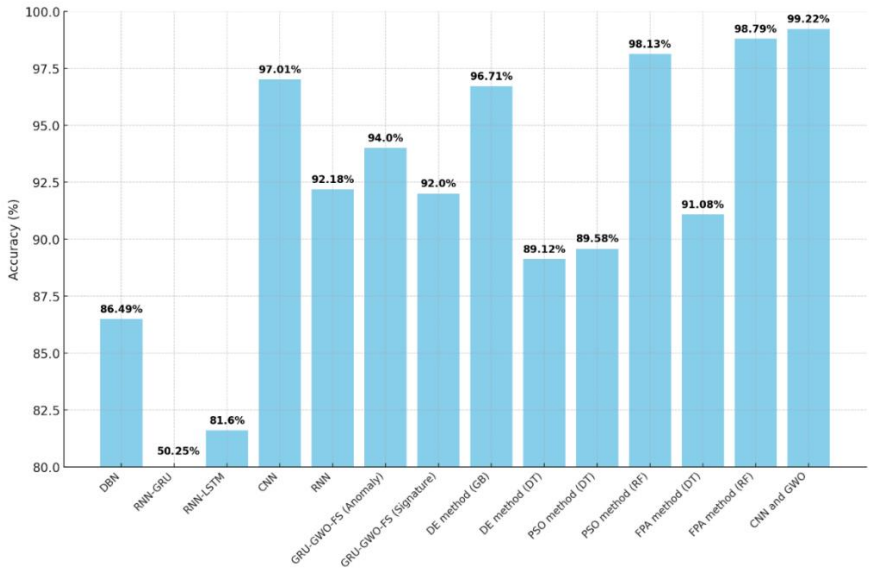
Figure 19 shows the results for the different ML and DL model with the NSL-KDD dataset. For example, in their investigation on the DBN model, the authors have been able to achieve an average accuracy of 86.49%, which indicates the challenges of conventional DL methods to identify complex patterns in a variety of network traffic data because of probability distributions. RNN-GRU and RNN-LSTM by comparison are able to achieve accuracies of 50.25% and 82.60%, suggesting that RNN based models do have some benefits, but may struggle with understanding non-linear interactions without more robust optimizations.

The CNN model outperforms the others much higher at 97.01%. This indicates that convolutional networks have a great advantage in terms of utilizing the spatial aspect of the dataset to extract features. The GRU-GWO-FS (Anomaly) model achieves 94% and GRU-GWO-FS (Signature) and 92% which effectively demonstrates the combination of GWO with GRU as being a successful approach to feature selection and classification.

For the optimization-based models, DE approach gains 96.71% (GB) and 89.12% (DT), while the PSO approach gets 98.13% (RF) and 89.58% (DT) The introduction of evolutionary algorithms provides improvements to traditional ML approaches.

Finally, the CNN-GWO method has the highest accuracy at 99.22% when compared to all of these models. This demonstrates the solid combination of Convolutional Neural Networks and Grey Wolf Optimization for purpose of feature extraction and classification being the best solution for network intrusion detection in this research.

Figure 19: The accuracy of the evaluated models in NSL-KDD dataset



The CNN-GWO approach integrates GWO to optimize the hyperparameters, thereby improving the effectiveness of the model in terms of network intrusion detection. In Table 11, we illustrate that the optimization process begins with unknown hyperparameters that are randomized including filter sizes, dropout rate and learning rate which are adjusted at each sequence of 10 iterations. At every iteration, the CNN model was trained for 10 epochs and the GWO algorithm improved the hyperparameter population based on validation accuracy to select hyperparameter configurations that performed better than others.

After 10 iterations, the model attains an impressive accuracy of 99.22% with the optimal hyperparameters: 32 filters for the first layer, 64 filters for the second layer, a dropout rate of 0.36, and a learning rate of 0.00598. This iterative method guarantees that the CNN model attains a high-performing configuration, illustrating the effectiveness of GWO in optimizing hyperparameters for enhanced model performance.

Table 11: Best accuracy over 10 iterations on NSL-KDD dataset

Iteration	Filters1	Filters2	Dropout	Learning Rate	Val Accuracy	Val Loss
1	43	94	0.41	0.00379	0.9898	0.0327
2	51	91	0.20	0.00047	0.9784	0.0624
3	48	85	0.37	0.00172	0.9868	0.0409
4	33	113	0.42	0.00702	0.9919	0.0276
5	51	64	0.30	0.00579	0.9906	0.0318
6	32	114	0.26	0.00882	0.9917	0.0283
7	42	64	0.25	0.00152	0.9881	0.0408
8	32	104	0.41	0.00010	0.9682	0.1429
9	32	101	0.32	0.00071	0.9863	0.0513
10	32	64	0.36	0.00598	**0.9922**	0.0309

The training history of the CNN-GWO model for a total of 10 iterations on the NSL-KDD dataset is presented in Figure 20. Each iteration consists of two plots, Model Accuracy and Model Loss. These plots show the model performance during training in relation to both training and validation datasets.

The model accuracy plots illustrate how accuracy progresses for the training and validation datasets over epochs. The model shows improving accuracy very quickly, with further significant increases in validation accuracy as it learned the data in the initial epochs. The model loss plots show the decreasing loss for the training and validation datasets as the number of epochs increases. The loss decreased quickly for the training dataset during the initial rounds of training and then continued to decrease mostly consistently in subsequent rounds. The repeating patterns in all iterations indicate that the CNN-GWO model was effectively learning and reducing and optimizing parameters over the course of the 10 iterations.

The model validation accuracy demonstrates consistent increases in each iteration, which highlights the ability of the model to generalize well to unseen data. The results in the 10th iteration indicates an acceptable level of accuracy on the dataset with minimal loss, showing the model effectively converged and optimized parameters using GWO hyperparameter tuning.

This repeated enhancement underscores the efficacy of GWO in hyperparameter tuning, markedly augmenting model accuracy and diminishing loss during training.

Figure 20 : Training history of CNN-GWO over 10 Iterations on NSLKDD dataset

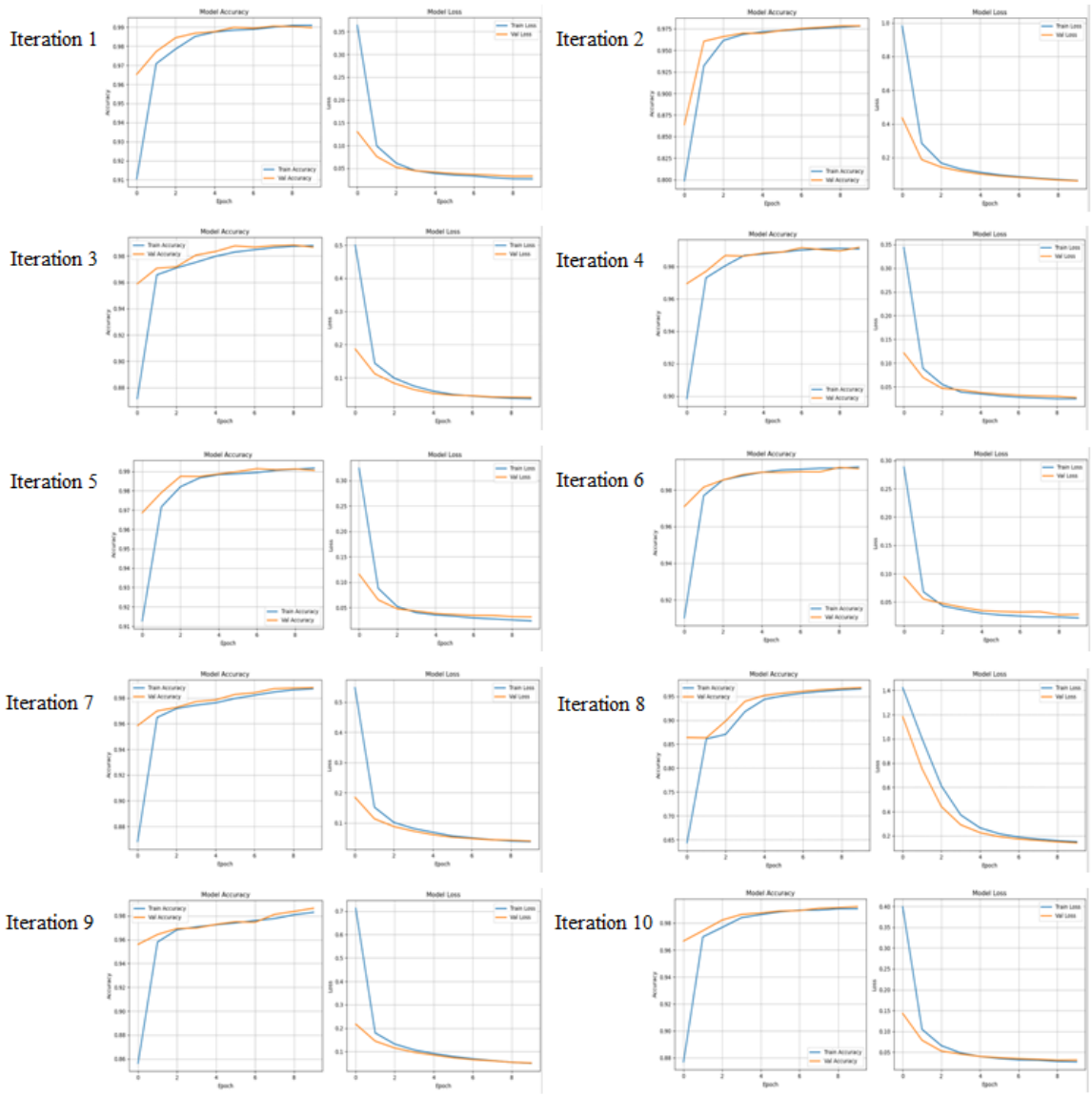


Table 12 Comparison with other methods on the NSL-KDD dataset

(Athors, year)	**Algorithm**	**Result (Accuracy)**
Tian et al, 2020 [171]	DBN	86.49
Al-Emadi et al., 2020 [174]	RNN- GRU	50.25%
	RNN-LSTM	81.60%
	CNN	97.01%
Almiani et al, 2020 [172]	RNN	92.18%
Elsaid et al., 2024 [223]	GRU-GWO-FS	94% (Anomaly Detection)
	LSTM-GWO-FS	92% (Signature Detection)
Emirmahmutoğlu et Atay, 2025 [224]	DE method	96.71% (GB), 89.12% (DT)
	PSO method	89.58% (DT), 98.13% (RF)
	FPA method	91.08% (DT), 98.79% (RF)
Ours	**CNN and GWO**	**99.22%**

4.3 Discussion

NSL-KDD dataset is a widely used dataset in network intrusion detection studies, offering a daunting task with its dynamic and heterogeneous attack classes. A number of ML and DL methods have been tried on this dataset with variable results depending on the model used. Traditional models like SVM, RF, KNN, and DT perform well but typically do not perform as well as newer approaches like CNN or RNN.

For example, Tian et al., 2020 proposed a DBN intrusion detection with 86.49% accuracy. While effective, DBNs are often limited in high-dimensional space and also require complex, layer-by-layer training, which may undermine their effectiveness. Conversely, Al-Emadi et al., 2020 made a comparison of the performance of several models, where CNN achieved a much higher accuracy of 97.01%, revealing the potential of CNNs in spatial feature extraction with no manual feature engineering. However, traditional models such as RNN-LSTM and RNN-GRU obtained only 81.60% and 50.25% accuracy, respectively, due to the drawbacks of these models in sequential data processing with high computational costs.

Besides, Elsaid et al., 2024 put forward the GRU-GWO-FS model that achieved 94% accuracy for anomaly detection and 92% for signature detection. Despite being promising, the LSTM-GWO-FS model did not surpass recent DL approaches in detection accuracy. Comparatively, Emirmahmutoğlu and Atay, 2025 explored feature selection-based ML algorithms with Differential Evolution (DE), PSO, and Flower Pollination Algorithm (FPA) as feature selectors and achieved 96.71% with GB, and 98.13% with RF. The FPA method also demonstrated a respectable performance with 98.79% accuracy with RF.

But ours, developed from a combination of CNN and GWO, outperforms all the current methods with 99.22% accuracy. The discovery shows that not only can the hybrid CNN-GWO model increase accuracy, but it also eliminates the requirement for manual feature extraction, a common drawback in traditional ML methods. Unlike the LSTM-GWO or GRU-GWO-FS methods with increased training time and computational requirements, the CNN-GWO model has improved performance without compromising computational efficiency. The ability of the CNN to automatically learn hierarchical features combined with GWO's optimization capabilities enables the CNN-GWO to outperform not just traditional models but also other hybrid models in detection accuracy and computational efficiency.

In short, our CNN-GWO model sets a new state of the art in intrusion detection in the NSL-KDD dataset, surpassing the current state-of-the-art methods such as CNN by Al-Emadi et al. and DE and PSO-based methods. The CNN-GWO model performs better in terms of accuracy

with a good tradeoff between computational overhead and detection accuracy. Future work needs to investigate the applicability of such a strategy on other datasets like UNSW-NB15 and CSE-CIC-IDS2018, and its feasibility for real-time implementation in cybersecurity networks.

5. Contribution 04: Convolutional Neural Network and Cross-Attention Transformer for Robust Intrusion Detection in IoT Systems

5.1 Description

As IoT devices such as smart whiteboards, interactive projectors, and wearable technologies become increasingly prevalent in educational environments, securing these devices from cyber threats has become a critical concern. This research introduces a novel hybrid model that combines the strengths of CNNs and CAT to detect both known and unknown threats in real time. CNNs are employed for efficient feature extraction, while CATs are used to identify intricate correlations in the data, providing a more comprehensive and accurate detection system than traditional security techniques. This hybrid approach addresses the limitations of conventional methods, which often fail to detect sophisticated attacks such as data theft, MITM attacks, phishing, and intrusion.

Additionally, this contribution enhances the academic understanding of IoT security in the context of e-learning systems by proposing a tailored security framework. This framework is specifically designed to protect the integrity of IoT-based educational systems and safeguard sensitive data from malicious activities. By focusing on the unique challenges posed by IoT environments, this research offers an effective solution for securing the growing number of interconnected devices used in educational settings. Another significant contribution is the methodology for classifying different types of attacks on IoT networks, which aids in the identification and mitigation of various security threats. This classification framework enhances the ability to distinguish between normal and malicious behavior, providing valuable tools for educators and administrators to maintain the security of their systems.

Furthermore, the proposed system incorporates real-time threat detection, a crucial feature for responding to the dynamic nature of cyber-attacks. The hybrid DL model allows for the continuous monitoring of IoT networks, ensuring that threats are identified and mitigated as they arise. The research also lays the groundwork for future developments in the field by suggesting potential areas for further investigation, such as the integration of additional ML models or the exploration of emerging security technologies.

5.2 Methodology

5.2.1 Preprocessing phase

In the preprocessing phase, raw IoT network traffic data is prepared for the CNN-CAT model through several steps:

- **Data Cleaning:** Irrelevant or missing data is handled by removing columns with excessive missing values and replacing placeholder values (e.g., "-") with NaN.
- **Feature Encoding:** Categorical features are converted to numerical values using Label Encoding, and in some cases, One-Hot Encoding.
- **Feature Scaling:** The data is standardized using the StandardScaler to ensure all features have the same scale, preventing bias during training.
- **Resampling (SMOTE):** To address class imbalance (e.g., normal vs. malicious traffic), SMOTE is used to generate synthetic samples of the minority class, balancing the dataset.
- **Data Splitting:** The data is split into training and test sets using Stratified K-Fold Cross-Validation, ensuring each fold maintains the class distribution.
- **Tensor Conversion:** The processed data is converted into PyTorch tensors, making it compatible for training with the CNN-CAT model.

These steps ensure that the data is clean, balanced, and ready for training, allowing the model to effectively detect malicious IoT traffic.

5.2.2 The CNN model

The CNN approach involves using convolutional layers to extract local patterns from the input, followed by max-pooling to reduce dimensionality while preserving important features. The extracted features are then passed through fully connected layers to make predictions. ReLU activation adds non-linearity to the model, and dropout is used as a regularization technique to prevent overfitting. This approach effectively captures complex patterns, making CNNs suitable for various tasks such as classification and detection in time-series or sequential data.

5.2.3 Cross-Attention Transformer (CAT)

To improve learning on sequential or structured data, the CAT method combines the ability of transformers with cross-attention to improve learning. In CAT, the attention algorithm is applied to learn connections between various elements of the input sequence by calculating attention scores among queries, keys and values. The model would be able to capture

dependencies among data assuming there is one dependency or several dependencies in a series. This model would allow this dependency to be captured no matter how far apart it was in the sequence. The CAT algorithm operates through an embedding space and self-attention helper of the input data and processes the result using a sequence of encoder-decoder networks to obtain finer learnt representations. It allows this model to become interested in important data features and thus enhances its capability to classify or identify complicated tendencies.

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Where:

- Q is the Query matrix.
- K is the Key matrix.
- V is the Value matrix.
- d_k is the dimension of the key vectors (used for scaling).
- The SoftMax function normalizes the values.

The attention mechanism computes a weighted sum of the values V using the attention scores obtained from the query-key interaction. This operation helps the model to focus on different parts of the input sequence.

5.2.4 Implementation and evaluation

Pseudo code: The pseudo code for the CNN-CAT model

CNN-CAT Pseudo-code

```
Function CNN_CrossAttentionTransformer(input_dim, num_heads, num_classes, embed_dim, dropout):
    # Step 1: Initialize CNN layers, Transformer, and output layer
    conv1 ← Conv1D(1, 16, kernel_size=3)
    pool ← MaxPool1D(kernel_size=2)
    fc1 ← Linear(input_dim * 16, embed_dim)
    transformer ← Transformer(d_model=embed_dim, num_heads=num_heads, dropout=dropout)
    fc2 ← Linear(embed_dim, num_classes)
# Step 2: Define forward pass
Function forward(x):
x ← apply_conv1d(x)              #Apply 1D convolution
x ← apply_maxpool(x)             #Apply max pooling
x ← flatten(x)                   #Flatten for fully connected layer
x ← fc1(x)                       #Pass through the fully connected layer
```

```
x ← reshape_for_transformer(x)      #Add sequence dimension for Transformer
output ← transformer(x, x)          # Cross-Attention (source = memory)
x ← mean_pooling(output)            # Apply global average pooling
return fc2(x)                       # Final classification layer
# Training loop
Function train_model(model, train_loader, val_loader, epochs):
For epoch in range(epochs):
    For inputs, labels in train_loader:
outputs ← model(inputs)  # Forward pass
loss ← CrossEntropyLoss(outputs, labels)  # Compute loss
optimizer.step()  # Backpropagate and update model parameters
For inputs, labels in val_loader:
val_loss ← CrossEntropyLoss(model(inputs), labels)  # Validation step
return train_loss, val_loss
# Cross-validation loop
Function cross_validate(model, X, y):
For fold in KFold(X, y):
train_loader, test_loader ← prepare_loaders(fold)  # Prepare data loaders
train_model(model, train_loader, test_loader)  # Train model
accuracy ← evaluate_model(model, test_loader)  # Evaluate accuracy
return accuracy
Function main():
model ← CNN_CrossAttentionTransformer()
cross_validate(model, X, y)  # Perform cross-validation
```

5.2.5 Dataset

In this stage, we applied our implementation on the CTU-IoT-Malware-Capture-1-1 dataset, which has been produced at the Stratosphere Laboratory in the ApseMat Project. Relevant to the organization is this initiative, which is based at the Czech Technical University (CTU) in Prague and focuses on the organized gathering of malware captures in the IoT. The data set will consist of network flows categorized as benign and malicious, thus becoming highly invaluable in the study of malware. The labels are created based on eavesdropped malware and can be used to identify various types of attack modes, such as DDoS attacks, botnets (e.g., Mirai, Okiru) and communication with command-and-control (C&C) servers. Furthermore, the labeling system also assists in detecting suspicious activities, including the exploit attempts, downloading of files, periodical checking of the devices already infected, and, in general, it offers an abundant source of detection of IoT-based cyber risks. Then we optimized the CNN-CAT model to detect malicious IoT network traffic. The convolutional neural network discovers local features on the raw tabular data as a result of which attack signatures and anomalies are detected within the network traffic. CAT obtains long-range interactions and focuses on the most relevant elements through an attention

mechanism, which promotes the discriminating capabilities of the model to detect normal and malicious traffic. The combination of the CNN feature-extraction system and the CAT attention system can help the model achieve better accuracy in the fine-grained security of identifying advanced IoT threats.

5.3 Results and comparison

This section discusses the results of the CNN-CAT model and compares it with other techniques as shown in Figure 21. In this evaluation, accuracy is used to assess the effectiveness of the models on the CTU-IoT-Malware-Capture-1-1 dataset. The comparative analysis of various ML and DL models provides valuable insights into their performance in terms of dataset coverage, accuracy, and computational efficiency for intrusion detection. The models considered include SVM, NB, LSTM, CNN, and CNN-CAT, each demonstrating distinct differences in terms of accuracy, computational efficiency, and flexibility.

Figure 21 Comparison of Model Accuracy for Intrusion Detection on the CTU-IoT-Malware-Capture-1-1 dataset.

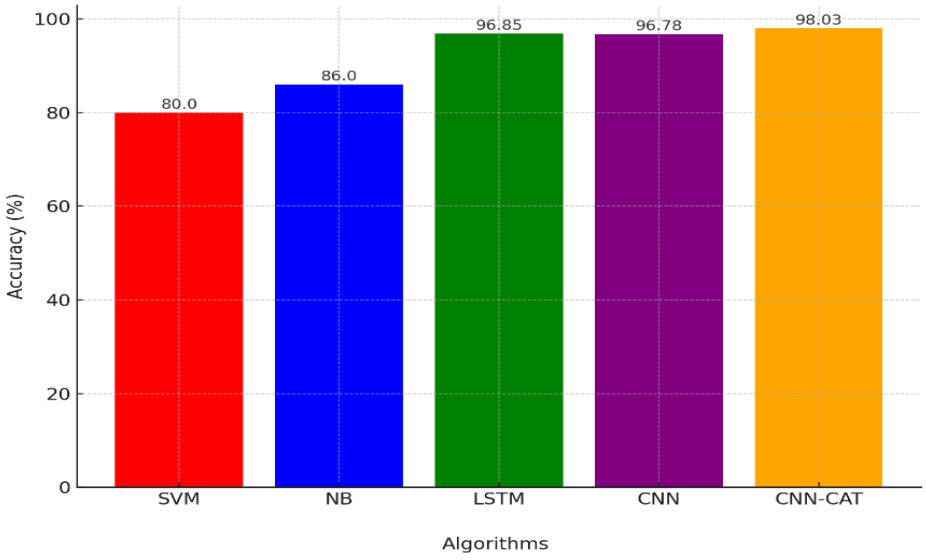


Figure 22: Loss and Accuracy Curves of CNN-CAT for fold 1 on CTU-IoT-Malware-Capture-1-1 dataset.

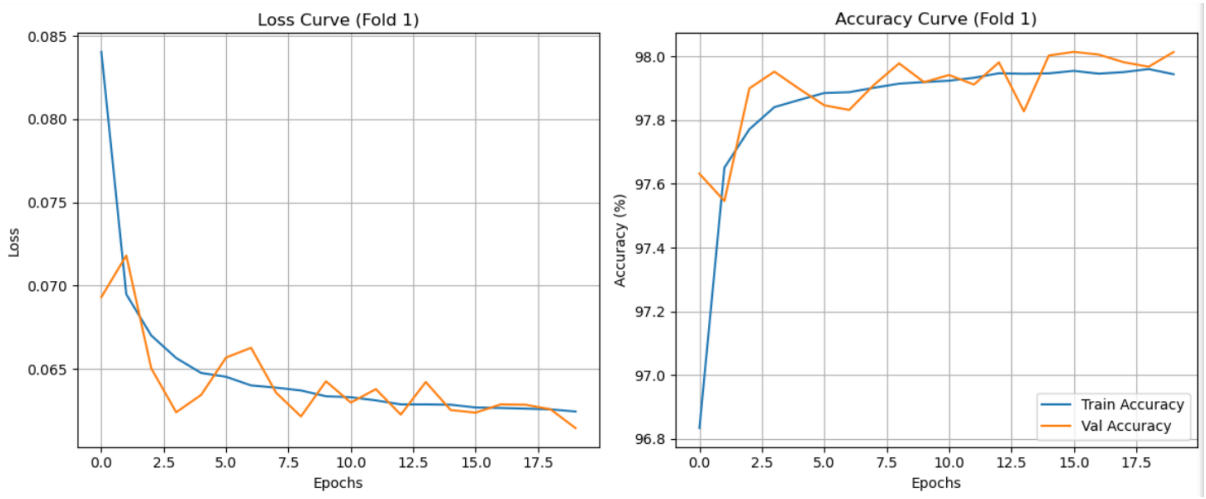


Figure 22 provides the performance of Fold 1 where the starting training accuracy is 96.83% and the launching validation accuracy is 97.63%. The training and validation losses are in a negative increasing order towards the value of 0.0624 and 0.0614, respectively, after the twentieth epoch. The highest accuracy of 97.94% is achieved on the training set and that of 98.01% on the validation set which shows that there is a harmonious balance between training and the validation set. The model confusion table indicates that there is no significant misclassification and the precision, recall and F1-score of both classes 0 and 1 are 0.98.

Figure 23 Loss and Accuracy Curves of CNN-CAT for fold 2 on CTU-IoT-Malware-Capture-1-1 dataset.

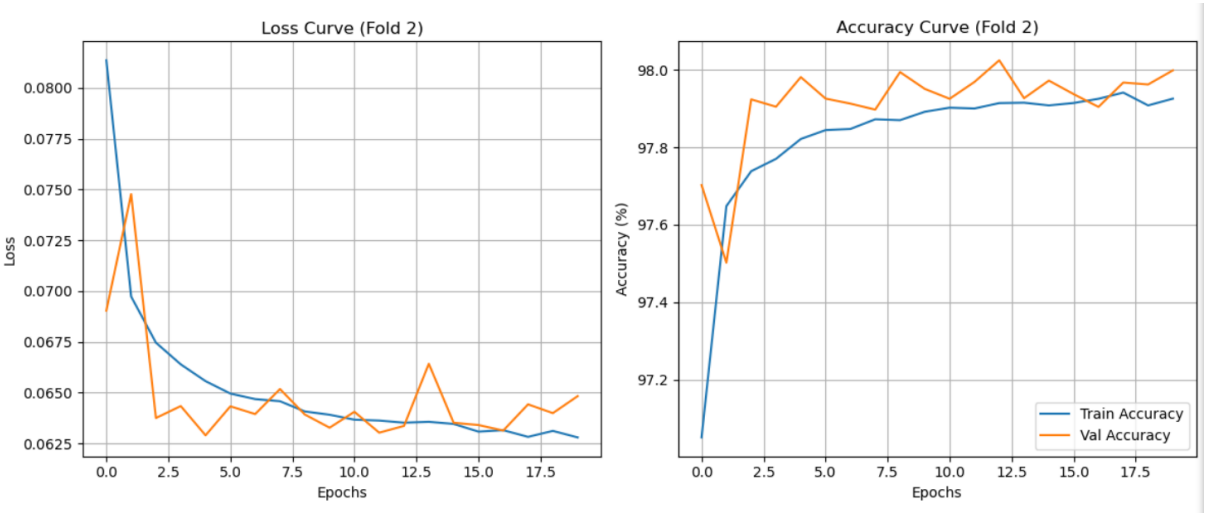


Figure 23 shows the performance measures of Fold 2, and have a similar trend as Fold 1, starting with a training accuracy object of 97.05% and a validation accuracy object of 97.70% in the first epoch. Both training and validation costs have monotonic decreasing curves along the epochs but the validation cost is slightly oscillating in between and reaches its highest

value as of the second epoch of 0.0748 and shows a smooth decline to is steady until the 20th epoch. By the termination epoch, predictive accuracy levels off to 97.93%, 98.00% in the training and validation sets respectively. According to the observation with Fold 1, the confusion matrix shows a fairly balanced repartition in the two classes, which both precision and recall values have an approximate value of 0.98.

Figure 24 Loss and Accuracy Curves of CNN-CAT for fold 3 on CTU-IoT-Malware-Capture-1-1 dataset.

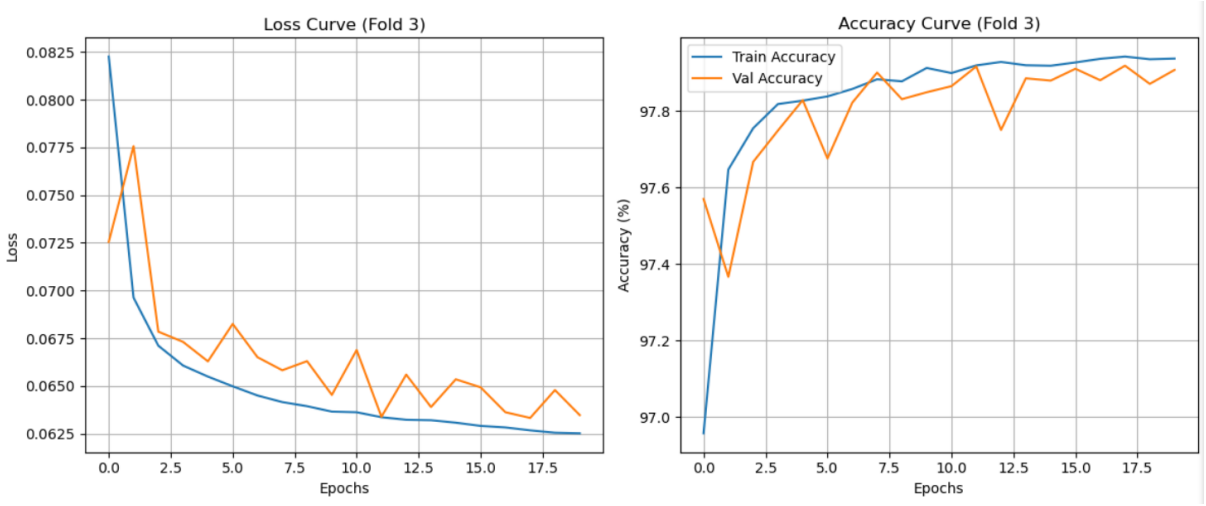


Figure 24 illustrates the performance of Fold 3 which starts with a 96.96% training accuracy and 97.57% validation accuracy. It follows a similar pattern to the previous folds, where both losses decrease gradually, with validation loss fluctuating slightly. By the 20th epoch, training accuracy reaches 97.94%, while validation accuracy hovers around 97.91%. The final results show a slight dip in performance compared to the other folds, achieving a 0.9791 accuracy, but still maintaining excellent performance. The model's confusion matrix confirms minimal misclassifications, with both classes again achieving 0.98 precision, recall, and F1-score.

Figure 25 Loss and Accuracy Curves of CNN-CAT for fold 4 on CTU-IoT-Malware-Capture-1-1 dataset.

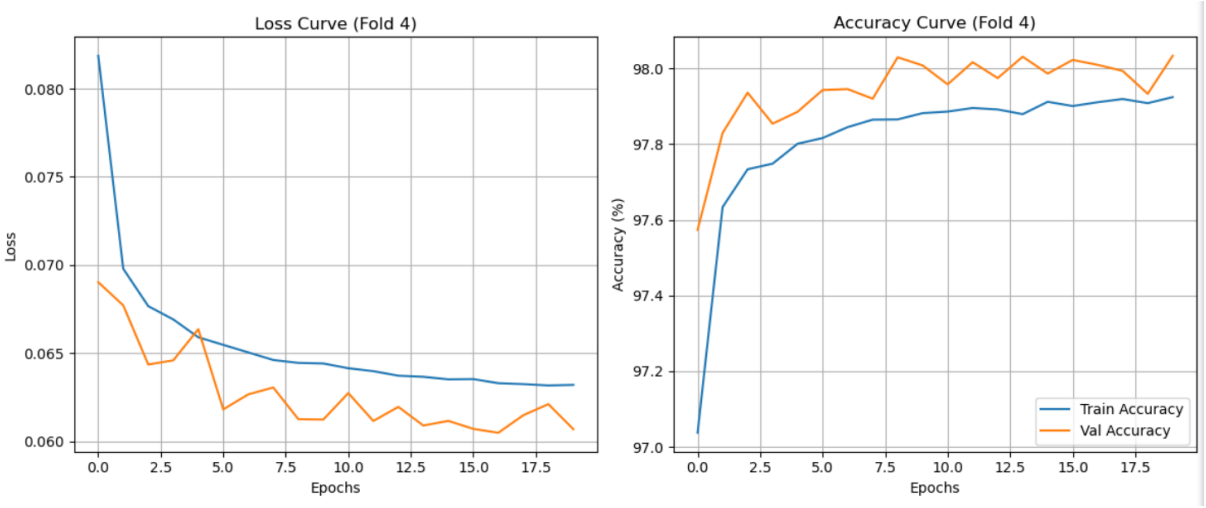


Figure 25 demonstrates the results of Fold 4, which achieves its best with the highest validation accuracy at 98.03% in the 9th epoch, and a final accuracy of 0.9803, the highest among all folds. Starting with 97.04% training accuracy and 97.57% validation accuracy, the model shows a steady decrease in loss. The training accuracy reaches 97.92%, and validation accuracy peaks at 98.03%. The confusion matrix for Fold 4 mirrors the other folds, showing a 0.98 precision, recall, and F1-score for both classes. This fold reflects the best performance, both in terms of accuracy and overall stability.

Figure 26 Loss and Accuracy Curves of CNN-CAT for fold 5 on CTU-IoT-Malware-Capture-1-1 dataset.

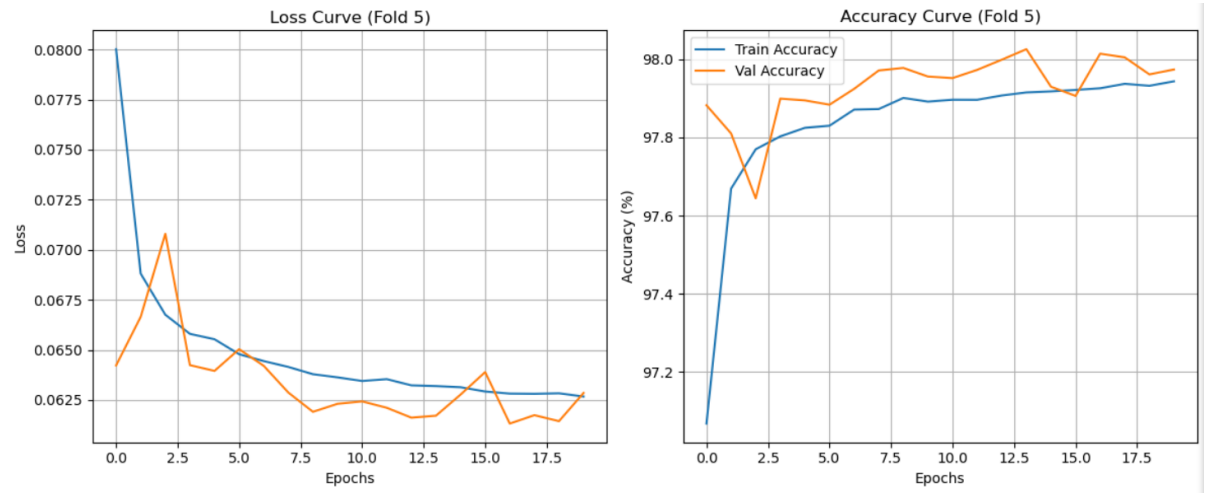


Figure 26 illustrates the results for Fold 5, which begins with 97.07% training accuracy and 97.88% validation accuracy. Over the 20 epochs, training accuracy improves to 97.94%, while validation accuracy settles around 97.97%. There is a slight variation in the validation loss, but it remains stable by the end of the training. The model achieves an accuracy of 0.9797 in

Fold 5, slightly lower than the other folds, but still within an acceptable range. Like the other folds, precision, recall, and F1-score for both classes are consistently 0.98, ensuring reliable performance on the validation data.

Table 13 Final validation accuracy and loss for each fold after training.

FOLD	ACCURACY	LOSS
Fold 1	98.01 %	0.0614
Fold 2	98.00 %	0.0648
Fold 3	97.91 %	0.0635
Fold 4	98.03 %	0.0605
Fold 5	97.97 %	0.0629

Table 13 presents the final validation accuracy and validation loss for each fold after the 20th epoch. Fold 4 leads in both accuracy (98.03%) and loss (0.0605), indicating it was the most optimal fold during training.

Table 14 Comparison with other methods on the CTU-IoT-Malware-Capture-1-1 dataset.

Algorithm	**Result (Accuracy)**
SVM	80%
NB	86%
LSTM	96.85%
CNN	96.78%
CNN-CAT	**98.03%**

5.4 Discussion

The results presented in Table 14 demonstrate the performance of various ML and DL models on the CTU-IoT-Malware-Capture-1-1 dataset for intrusion detection in IoT networks. Each algorithm has shown varying levels of accuracy, offering valuable insights into their effectiveness in identifying malicious activities within smart e-learning systems.

The SVM model achieved an accuracy of 80%. While SVM is a well-established algorithm for classification tasks, its performance in this context is limited, especially when dealing with complex and large datasets like CTU-IoT-Malware-Capture-1-1. This fact might explain why the model is relatively less accurate compared to other methods because it did not promote the ability to embrace complex patterns of data. A slightly better output has been obtained with the NB classifier with a score of 86 percentage accuracy. NB, even with its simplicity in nature, and on the assumption of independence of features, has shown excellent performance

in a wide variety of application activities that display felt properties of probabilities. However, like SVM, its efficiency is limited in the case where using it on the datasets that exhibit highly correlated attributes including complex ones, which is common in IoT networks data. LSTM model significantly improved the classification rate and was at 96.85. LSTM, in particular, is good at characterizing a temporal dependency, thus making it, in particular, effective with sequential data, such as patterns of network traffic. The specialized importance of the temporal dependencies in the detection of intrusions in the IoT network traffic is highlighted by the observed result, and the LSTM is particularly adept at acquiring such patterns. The CNN model was 96.78% accurate. Though CNNs have mainly been used to process images, they have been shown to be useful to process features in tabular data. CNNs linked to the performance of LSTM by obtaining local patterns and hierarchies, which results in the generalization of CNNs in various fields beyond the traditional image recognition.

The CNN-CAT hybrid model, which was in a comparative analysis, reached the highest level of 98.03% accuracy mark. The CNN-CAT model shows a high ability to extract local and global dependencies in the data by introducing CNN feature extraction and the high-level attention mechanisms implemented in the Cross Attention Transformer. The hybrid structure makes it easy to control selective attention on important attributes and interactions and produces better predictive accuracy. These effective gains compared to other models highlight the effectiveness of the synthesis of convolutional and transformer-based models for solving complex intrusion detection problems in IoT networks.

Overall, the CNN-CAT model is the best performing and is quite useful in capturing more detailed dynamics in the data, and also it represents the most effective solution to be used in identifying malicious operations within IoT networks.

5.5 Conclusion

This chapter has systematically presented the research methodology that was employed in order to attain the objectives of this thesis in terms of the development and evaluation of four hybrid deep learning models: CNN–OOA, LSTM–SWAP, CNN–GWO, and CNN–CAT. Each one of the models has been thought out and made to meet certain cybersecurity problems, including identifying phishing attacks, detecting online exam cheating, network attacks, and IoT device anomaly detection.

The approach started with data preprocessing and feature engineering for each domain to make sure that input data, e.g., sequential actions of users or network traffic samples, and device behavior were normalized, complete, and appropriate for learning. Each hybrid model was then designed to utilize the strengths of deep learning and advanced optimization or attention/memorization. For instance, in CNN–OOA and CNN–GWO, convolutional networks are utilized for feature extraction, and biologically inspired optimization techniques are used to adaptively adjust hyperparameters, which accelerates convergence rate, enhances generalization capability, and improves detection accuracy. On the other hand, LSTM–SWAP integrates temporal sequence modeling along with attention-based refinement to better capture long-term and bursty behavioral dependencies, which is beneficial for detecting subtle cheating behaviors. CNN–CAT combines CNN with cross-attention transformers to enhance the detection capabilities of abnormal behaviors in IoT-related systems, preserving private user data.

Integrating the global pattern learning with localized anomaly detection and automatic optimization, the resulting approach can attain both high accuracy and robustness for different security tasks with computational efficiency. It’s an organized workflow too, from preparing data to training the model, tuning the hyperparameters, and evaluating.

To conclude, this chapter presents a generic and systematic methodology to exploit hybrid deep learning architectures for cybersecurity. It chooses to provide solving of complex protecting threats, system integrity, and security performance improvement of e-learning, online testing systems, and networked and IoT environments with the deep neural networks combined exposure optimization method and attention mechanism alternative functions.

CONCLUSION AND PERSPECTIVES

Conclusion and perspectives

The increase in the use of digital communication and online transactions due to their exponential rise also led to a rise in cyber threats that have increased in sophistication as well as in complexity. Cybercriminals constantly advance their efforts; therefore, the traditional security approaches can hardly keep pace. This has also necessitated the need to come up with highly developed and dynamic systems with the capacity of either detecting or preventing cyber threats. The paper includes three notable contributions to cybersecurity, that is, the LSTM-SWAP model used to detect cheating, CNN-ORCA model used to detect phishing attacks, and the CNN-GWO model used to detect intrusion. Such models combine state of the art technologies, including DL and optimization algorithms, to deliver optimal answers to solutions to some of the most important cybersecurity issues. This conclusion underlines how these solutions affect the process of resolving the current issues in the cybersecurity sphere.

The LSTM-SWAP model presents a newer sense of detecting cheating especially in online education and assessment. Manual proctoring schemes under traditional approaches tend to be inefficient and is based on rule-based systems, which is resource-intensive. LSTM-SWAP is a model that considers the timing of other students in answering the questions and the method of selecting answers, where not responding promptly or selecting answers, which are typical of a student cheating in examinations. SWAP component aids the refinement of predictions and the learning process with adjustment such that even the slightest instances of cheating can be identified. This model resolves the dilemma of distinguishing between honest mistakes and deliberate cheating which is a serious problem on mass online exams. The unusualities that may pass unnoticed to the eye can be identified through the analysis of the temporal dependencies of the behavior conducted by the LSTM-SWAP. Such an approach is more accurate and provides fewer false positives because of the consistency in detecting activities suggesting cheating. Consequently, it offers a solid platform on which online learning institutions can track and ensure integrity in academics.

One of the most widespread types of cybercrimes became phishing attacks when the person is fooled to disclose some sensitive information. The CNN-ORCA algorithm addresses this problem by fusing the strength of the CNN with the OOA. CNNs are the most appropriate mechanisms in spotting sophisticated patterns in data and when used in phishing detection, it performs excellently in identifying malicious actions in emails, websites, and links. The CNN-ORCA model is effective because the CNN has been optimized to adapt to new

phishing strategies thanks to the tool of OOA to use in the parameterization of CNN. Constant development in phishing models can be viewed as one of the major issues of phishing detection since cybercriminals refine their techniques to improve on ways, they can present their fraudulent activities as genuine. Historical signature-based systems do not always identify new forms of phishing. CNN-ORCA does not have this limitation since it enables the model to learn automatically using new data and find new examples of phishing that otherwise would have been missed. The optimization process of the model also strengthens its rate of accuracy in picking up phishing attacks, which makes it apt despite the appearance of new methods. This will go a long way in improving detection of phishing since it is an improvement that is both scalable and indispensable to address the increasing cybersecurity threat of phishing attacks.

Another key in cybersecurity is intrusion detection since organizations are required to observe their networks to detect any suspicious activities on their network. Traditional intrusion detection systems are usually rule based or signature based and thus may be weak in detecting emerging or advanced attacks. The CNN-GWO model can help solve this problem, since it homogenizes CNNs with the GWO algorithm to enhance the effectiveness and versatility of intrusion detection. The data that the network traffic represents are then converted into their features using CNNs, which are then used to isolate the anomalies and any possible intrusions. The hyperparameters of the CNN are the input in the GWO algorithm and this narrows down its performance in real-time identification. The potential size and complexity of traffic in a network is one of the major issues in intrusion detection because it is not easy to detect abnormal behavior with the conventional systems. CNN-GWO outdoes this by automatically identifying and responding to complex patterns and abnormalities in network traffic and thus minimizing the dependence on the predetermined rules and signatures. The optimization procedure will make the model generalized to new forms of attacks relative to the changes in cyber threats. A better submission will make the intrusion detection systems more effective in the collective sense and gives the organizations an opportunity to have a scalable solution to protect themselves against various events of cyberattack, including DoS and more sophisticated persistent threats.

All these models can relate to the crucial challenges in the sphere of cybersecurity because DL is combined with optimization techniques, which are more adaptive and flexible than their conventional analogues. LSTM-SWAP cheating detection model helps to monitor online exams and assessments accurately because it identifies low-level behavior patterns. CNN-

ORCA model is an effective solution to identify phishing attacks as it can alter to changes in phishing techniques. Finally, CNN-GWO model enhances intrusion detection, which increases the efficiency of monitoring systems in networks.

Such contributions have an enormous benefit since they provide more adaptable, accurate, and scalable responses to some of the most significant cybersecurity issues. They have models which through the use of DL and optimization algorithms offer systems that can evolve and learn continuously following all the changing cyber threats so as to remain effective. These advanced techniques give a guaranteed system that can be used to remain ahead of the cybercriminals and also offers better security to individuals, organizations and institutions as they are more developed and complicated in nature. This incorporation of the technologies is a bright development regarding the continuous struggle to protect the digital environment and consequently defend any sensitive information against cyberattacks of all kinds

BIBLIOGRAPHY

Bibliography

[1] I. Boutabia, A. Benmachiche, A. A. Betouil, and C. Chemam, “A Survey in the Use of Deep Learning Techniques in The Open Classroom Approach,” in *2024 6th International Conference on Pattern Analysis and Intelligent Systems (PAIS)*, IEEE, 2024, pp. 1–7. Accessed: Aug. 18, 2024.

[2] I. Boutabia, A. Benmachiche, A. A. Betouil, M. Boutassetta, and M. Derdour, “A Survey on AI Applications in the Open Classroom Approach,” in *2025 International Conference on Networking and Advanced Systems (ICNAS)*, Oct. 2025, pp. 1–7. doi: 10.1109/ICNAS68168.2025.11298090.

[3] I. Boutabia, A. Benmachiche, A. Bennour, A. A. Betouil, M. Derdour, and F. Ghabban, “Hybrid CNN-ViT Model for Student Engagement Detection in Open Classroom Environments,” *SN COMPUT. SCI.*, vol. 6, no. 6, p. 684, Jul. 2025, doi: 10.1007/s42979-025-04228-2.

[4] I. Boutabia, A. Benmachiche, A. A. Betouil, C. Chemam, and K. Rais, “Advanced Text Prediction System Integrated Within the Search Engine for the Open Classroom Approach Based on Particle Swarm Optimization and Long Short-Term Memory Models,” *Arab J Sci Eng*, vol. 51, no. 9, pp. 12653–12675, May 2026, doi: 10.1007/s13369-026-11247-5.

[5] S. O. Boufaida, A. Benmachiche, M. Maatallah, and C. Chemam, “Hybrid Secure Routing in Mobile Ad-hoc Networks (MANETSs),” Jan. 09, 2026, *arXiv*: arXiv:2602.13204. doi: 10.48550/arXiv.2602.13204.

[6] S. O. Boufaida, A. Benmachiche, and M. Maatallah, “Real-Time Image Processing Algorithms for Embedded Systems,” Jan. 09, 2026, *arXiv*: arXiv:2601.06243. doi: 10.48550/arXiv.2601.06243.

[7] M. Boutassetta, A. Makhlouf, N. Messaoudi, A. Benmachiche, I. Boutabia, and M. Derdour, “Cyberattack Detection in Smart Cities Using AI: A literature review,” in *2025 International Conference on Networking and Advanced Systems (ICNAS)*, Oct. 2025, pp. 1–9. doi: 10.1109/ICNAS68168.2025.11298103.

[8] S. Wheeler, “e-Learning and Digital Learning,” in *Encyclopedia of the Sciences of Learning*, N. M. Seel, Ed., Boston, MA: Springer US, 2012, pp. 1109–1111. doi: 10.1007/978-1-4419-1428-6_431.

[9] Z. Bezovski and S. Poorani, “The evolution of e-learning and new trends,” in *Information and Knowledge Management*, Iiste, 2016, pp. 50–57. Accessed: Aug. 09, 2025.

[10] J. Seale, *E-learning and disability in higher education: Accessibility research and practice*. Routledge, 2013. Accessed: Aug. 09, 2025.

[11] U. O. Matthew, J. S. Kazaure, and N. U. Okafor, “Contemporary development in E-Learning education, cloud computing technology & internet of things.,” *EAI Endorsed Trans. Cloud Syst.*, vol. 7, no. 20, p. e3, 2021.

[12] M. W. Allen, *Michael Allen’s guide to e-learning: Building interactive, fun, and effective learning programs for any company*. John Wiley & Sons, 2016. Accessed: Aug. 09, 2025.

[13] A. Nagy, “The Impact of E-Learning,” in *E-Content*, P. A. Bruck, Z. Karssen, A. Buchholz, and A. Zerfass, Eds., Berlin/Heidelberg: Springer-Verlag, 2005, pp. 79–96. doi: 10.1007/3-540-26387-X_4.

[14] D. R. Garrison, *E-learning in the 21st century: A community of inquiry framework for research and practice*. Routledge, 2016. Accessed: Aug. 09, 2025.

[15] L. Anthonysamy, A. C. Koo, and S. H. Hew, "Self-regulated learning strategies in higher education: Fostering digital literacy for sustainable lifelong learning," *Educ Inf Technol*, vol. 25, no. 4, pp. 2393–2414, Jul. 2020, doi: 10.1007/s10639-020-10201-8.

[16] V. Bhardwaj, S. Zhang, Y. Q. Tan, and V. Pandey, "Redefining learning: student-centered strategies for academic and personal growth," in *Frontiers in Education*, Frontiers Media SA, 2025, p. 1518602. Accessed: Aug. 09, 2025.

[17] M. Liu and D. Yu, "Towards intelligent E-learning systems," *Educ Inf Technol*, Dec. 2022, doi: 10.1007/s10639-022-11479-6.

[18] M. T. Hora, J. Bouwma-Gearhart, and H. J. Park, "Data driven decision-making in the era of accountability: Fostering faculty data cultures for learning," *The Review of Higher Education*, vol. 40, no. 3, pp. 391–426, 2017.

[19] S. Haniya and S. Roberts-Lieb, "Differentiated learning: diversity dimensions of e-learning," in *E-learning ecologies*, Routledge, 2017, pp. 183–206. Accessed: Aug. 09, 2025.

[20] G. Sara, B. Tahar, and B. Abdelmadjid, "Recommendation approach of ed ucational resources in a fre-quent subnet," *المجلة الجزائرية للعلوم*, vol. 3, no. 6, pp. 51–55, 2019.

[21] P. N. Teixeira and J. C. Shin, Eds., "International Research Interconnectivity," in *The International Encyclopedia of Higher Education Systems and Institutions*, Dordrecht: Springer Netherlands, 2020, pp. 1859–1859. doi: 10.1007/978-94-017-8905-9_300421.

[22] M. A. Goyal, "E-LEARNING PLATFORMS AND FUTURE OF EDUCATION," 2023, Accessed: Aug. 09, 2025. [Online].

[23] B. Fernández-Manjón, J. M. Sánchez-Pérez, J. A. Gómez-Pulido, M. A. Vega-Rodríguez, and J. Bravo-Rodríguez, *Computers and education: E-learning, from theory to practice*. Springer Science & Business Media, 2007. Accessed: Aug. 09, 2025.

[24] C. M. Fletcher-Flinn and B. Gravatt, "The Efficacy of Computer Assisted Instruction (CAI): A Meta-Analysis," *Journal of Educational Computing Research*, vol. 12, no. 3, pp. 219–241, Apr. 1995, doi: 10.2190/51D4-F6L3-JQHU-9M31.

[25] B. Cope and M. Kalantzis, "A little history of e-learning: finding new ways to learn in the PLATO computer education system, 1959–1976," *History of Education*, vol. 52, no. 6, pp. 905–936, Nov. 2023, doi: 10.1080/0046760X.2022.2141353.

[26] R. A. Khan and H. Qudrat-Ullah, *Adoption of LMS in Higher Educational Institutions of the Middle East*. in Advances in Science, Technology & Innovation. Cham: Springer International Publishing, 2021. doi: 10.1007/978-3-030-50112-9.

[27] A. Crawley, *Supporting online students: A practical guide to planning, implementing, and evaluating services*. John Wiley & Sons, 2012. Accessed: Aug. 09, 2025.

[28] P. Nicholson, “A History of E-Learning: Echoes of the pioneers,” in *Computers and Education*, B. Fernández-Manjón, J. M. Sánchez-Pérez, J. A. Gómez-Pulido, M. A. Vega-Rodríguez, and J. Bravo-Rodríguez, Eds., Dordrecht: Springer Netherlands, 2007, pp. 1–11. doi: 10.1007/978-1-4020-4914-9_1.

[29] O. Malysheva, E. Tokareva, L. Orchakova, and Y. Smirnova, “The effect of online learning in modern history education,” *Heliyon*, vol. 8, no. 7, 2022, Accessed: Aug. 09, 2025.

[30] A. A. Amador, *Adoption and Diffusion of a Learning Management System as an Instructional Tool: A Community College Case Study*. Teachers College, Columbia University, 2025. Accessed: Aug. 09, 2025.

[31] M. M. Mashroofa, A. Haleem, N. Nawaz, and M. A. Saldeen, “E-learning adoption for sustainable higher education,” *Heliyon*, vol. 9, no. 6, 2023, Accessed: Aug. 09, 2025.

[32] M. Anderson, *Creating a Student Support Course in Casper College’s Moodle Learning Management System*. University of Wyoming, 2021. Accessed: Aug. 09, 2025.

[33] I. King and W.-I. Lee, *A Decade of MOOCs and Beyond: Platforms, Policies, Pedagogy, Technology, and Ecosystems with an Emphasis on Greater China*. Cham: Springer International Publishing, 2023. doi: 10.1007/978-3-031-15241-2.

[34] L. J. Waks, “Massive Open Online Courses and the Future of Higher Education,” in *Contemporary Technologies in Education*, O. O. Adesope and A. G. Rud, Eds., Cham: Springer International Publishing, 2019, pp. 183–213. doi: 10.1007/978-3-319-89680-9_10.

[35] J. Knox, *Posthumanism and the massive open online course: Contaminating the subject of global education*. Routledge, 2016. Accessed: Aug. 09, 2025.

[36] A. Benmachiche, A. Sahia, S. O. Boufaida, K. Rais, M. Derdour, and F. Maazouzi, “Enhancing learning recommendations in mooc search engines through named entity recognition,” *Educ Inf Technol*, vol. 30, no. 9, pp. 13041–13071, Jun. 2025, doi: 10.1007/s10639-024-13308-4.

[37] A. M. Khalid and O. Owusu-Boateng, *Navigating through technology in Modern Education*. Bentham Science Publishers, 2024. Accessed: Aug. 09, 2025.

[38] C. N. Quinn, *Designing mLearning: Tapping into the mobile revolution for organizational performance*. John Wiley & Sons, 2011. Accessed: Aug. 09, 2025.

[39] Kinshuk, N.-S. Chen, I.-L. Cheng, and S. W. Chew, “Evolution Is not enough: Revolutionizing Current Learning Environments to Smart Learning Environments,” *Int J Artif Intell Educ*, vol. 26, no. 2, pp. 561–581, Jun. 2016, doi: 10.1007/s40593-016-0108-x.

[40] W. Wu and A. Plakhtii, “E-learning based on cloud computing,” *International Journal of Emerging Technologies in Learning (IJET)*, vol. 16, no. 10, pp. 4–17, 2021.

[41] L. Chen, P. Chen, and Z. Lin, “Artificial intelligence in education: A review,” *IEEE access*, vol. 8, pp. 75264–75278, 2020.

[42] P. A. Galagan, “The e-learning revolution,” *Training & Development*, vol. 54, no. 12, pp. 24–24, 2000.

[43] R. C. Clark and R. E. Mayer, *E-learning and the science of instruction: Proven guidelines for consumers and designers of multimedia learning*. john Wiley & sons, 2023. Accessed: Aug. 09, 2025.

[44] S. Hrastinski, "Asynchronous and synchronous e-learning," *Educause quarterly*, vol. 31, no. 4, pp. 51–55, 2008.

[45] K. Hyder, A. Kwinn, R. Miazga, and M. Murray, "Synchronous e-learning," *The eLearning Guild*, 2007, Accessed: Aug. 09, 2025.

[46] F. Amiti, "Synchronous and asynchronous E-learning," *European Journal of Open Education and E-Learning Studies*, vol. 5, no. 2, 2020, Accessed: Aug. 09, 2025.

[47] A. Bozkurt, "A retro perspective on blended/hybrid learning: Systematic review, mapping and visualization of the scholarly landscape," *Journal of Interactive Media in Education*, vol. 2022, no. 1, 2022, Accessed: Aug. 09, 2025.

[48] M. Sarrab, L. Elgamel, and H. Aldabbas, "Mobile learning (m-learning) and educational environments," *International journal of distributed and parallel systems*, vol. 3, no. 4, p. 31, 2012.

[49] S. O. Boufaida, A. Benmachiche, M. Maatallah, and C. Chemam, "Hybrid Multi-Factor Authentication (MFA) Using Biometrics and Behavioral Analysis," *Available at SSRN 6295298*, 2026, Accessed: Jul. 04, 2026.

[50] S. O. Boufaida, A. Benmachiche, M. Derdour, M. Maatallah, M. S. Kahil, and M. C. Ghanem, "TSA-GRU: A Novel Hybrid Deep Learning Module for Learner Behavior Analytics in MOOCs," *Future Internet*, vol. 17, no. 8, p. 355, 2025.

[51] S. O. Boufaida, A. Benmachiche, M. Maatallah, and C. Chemam, "An Extensive Examination of Varied Approaches in E-Learning and MOOC Research: A Thorough Overview," in *2024 6th International Conference on Pattern Analysis and Intelligent Systems (PAIS)*, IEEE, 2024, pp. 1–8. Accessed: Aug. 18, 2024.

[52] S. O. Boufaida, A. Benmachiche, A. Bennour, M. Maatallah, M. Derdour, and F. Ghabban, "Enhancing MOOC Course Classification with Convolutional Neural Networks via Lion Algorithm-Based Hyperparameter Tuning," *SN COMPUT. SCI.*, vol. 6, no. 6, p. 707, Jul. 2025, doi: 10.1007/s42979-025-04179-8.

[53] S. Ennouamani and Z. Mahani, "An overview of adaptive e-learning systems," in *2017 eighth international conference on intelligent computing and information systems (ICICIS)*, IEEE, 2017, pp. 342–347. Accessed: Aug. 09, 2025.

[54] W. Rebhi, N. Ben Yahia, and N. Bellamine, "Lifelong and multirelational community detection to support social and collaborative e-learning," *Comp Applic In Engineering*, vol. 30, no. 5, pp. 1321–1337, Sep. 2022, doi: 10.1002/cae.22522.

[55] W. K. Monib, A. Qazi, R. A. Apong, and M. M. Mahmud, "Investigating learners' perceptions of microlearning: Factors influencing learning outcomes," *IEEE access*, 2024, Accessed: Oct. 07, 2025.

[56] V. M. Bradley, "Learning Management System (LMS) use with online instruction.," *International Journal of Technology in Education*, vol. 4, no. 1, pp. 68–92, 2021.

[57] L. Caprara and C. Caprara, “Effects of virtual learning environments: A scoping review of literature,” *Educ Inf Technol*, vol. 27, no. 3, pp. 3683–3722, Apr. 2022, doi: 10.1007/s10639-021-10768-w.

[58] A. Labrie, T. Mok, A. Tang, M. Lui, L. Oehlberg, and L. Poretski, “Toward Video-Conferencing Tools for Hands-On Activities in Online Teaching,” *Proc. ACM Hum.-Comput. Interact.*, vol. 6, no. GROUP, pp. 1–22, Jan. 2022, doi: 10.1145/3492829.

[59] Y.-L. Jeng, T.-T. Wu, Y.-M. Huang, Q. Tan, and S. J. Yang, “The add-on impact of mobile applications in learning strategies: A review study,” *Journal of Educational Technology & Society*, vol. 13, no. 3, pp. 3–11, 2010.

[60] M. E. Eltahir and F. M. E. Babiker, “The influence of artificial intelligence tools on student performance in e-learning environments: Case study,” *Electronic Journal of E-Learning*, vol. 22, no. 9, pp. 91–110, 2024.

[61] M. A. H. Masud and X. Huang, “A novel approach for adopting cloud-based e-learning system,” in *2012 IEEE/ACIS 11th International Conference on Computer and Information Science*, IEEE, 2012, pp. 37–42. Accessed: Aug. 09, 2025.

[62] A. Razzaque and A. Hamdan, “Internet of Things for Learning Styles and Learning Outcomes Improve e-Learning: A Review of Literature,” in *Proceedings of the International Conference on Artificial Intelligence and Computer Vision (AICV2020)*, vol. 1153, A.-E. Hassanien, A. T. Azar, T. Gaber, D. Oliva, and F. M. Tolba, Eds., in Advances in Intelligent Systems and Computing, vol. 1153. , Cham: Springer International Publishing, 2020, pp. 783–791. doi: 10.1007/978-3-030-44289-7_73.

[63] H. M. N. Iqbal, R. Parra-Saldivar, R. Zavala-Yoe, and R. A. Ramirez-Mendoza, “Smart educational tools and learning management systems: supportive framework,” *Int J Interact Des Manuf*, vol. 14, no. 4, pp. 1179–1193, Dec. 2020, doi: 10.1007/s12008-020-00695-4.

[64] B. K. Sedraoui, A. Benmachiche, and A. Makhlouf, “Developing and Evaluating Lightweight Cryptographic Algorithms for Secure Embedded Systems in IoT Devices,” Jan. 06, 2026, *arXiv*: arXiv:2601.02981. doi: 10.48550/arXiv.2601.02981.

[65] A. Bouamrane, M. Derdour, A. Bennour, A. Benmachiche, and M. Gasmi, “Machine Learning for Medical Image Analysis,” in *AI for Medical Image Analysis*, N. Ben Aoun, S. Ahmad, and M. Hammad, Eds., Cham: Springer Nature Switzerland, 2026, pp. 97–125. doi: 10.1007/978-3-032-02963-8_4.

[66] A.-A. Betouil, A. Benmachiche, K. Rais, A. Sahki, and I. Soualmia, “Comparative Analysis of Attention Mechanisms in Pix2Pix for Multimodal MRI Fusion,” *Journal of Electronics, Electromedical Engineering, and Medical Informatics*, vol. 8, no. 3, pp. 948–965, 2026.

[67] M. Maatallah, A. Benmachiche, K. Rais, and S. Touam, “Intelligent Fusion of Multi-Modal Medical Imaging: A Comprehensive Review of Methods, Challenges, and Clinical Integration,” *Journal of Electronics, Electromedical Engineering, and Medical Informatics*, vol. 8, no. 3, pp. 897–934, 2026.

[68] A. Benmachiche, B. Tahar, L. M. Tayeb, and Z. Asma, “A dynamic navigation for autonomous mobiles robots,” *Intelligent Decision Technologies*, vol. 10, no. 1, pp. 81–91, 2016.

[69] A. Benmachiche, A. A. Betouil, I. Boutabia, A. Nouari, K. Boumahni, and H. Bouzata, “A Fuzzy Navigation Approach Using the Intelligent Lights Algorithm for an Autonomous Mobile Robot,” in *12th International Conference on Information Systems and Advanced Technologies “ICISAT 2022,”* vol. 624, M. R. Laouar, V. E. Balas, B. Lejdel, S. Eom, and M. A. Boudia, Eds., in Lecture Notes in Networks and Systems, vol. 624. , Cham: Springer International Publishing, 2023, pp. 112–121. doi: 10.1007/978-3-031-25344-7_11.

[70] A. Makhlouf, A. Benmachiche, and I. Boutabia, “Enhanced Autonomous Mobile Robot Navigation Using a Hybrid BFO/PSO Algorithm for Dynamic Obstacle Avoidance,” *Informatica*, vol. 48, no. 17, 2024, Accessed: Sep. 27, 2025.

[71] A. Benmachiche, M. Derdour, M. S. Kahil, M. C. Ghanem, and M. Deriche, “Adaptive hybrid PSO–APF algorithm for advanced path planning in next-generation autonomous robots,” *Sensors*, vol. 25, no. 18, p. 5742, 2025.

[72] E. Spaho, B. Çiço, and I. Shabani, “IoT Integration Approaches into Personalized Online Learning: Systematic Review,” *Computers*, vol. 14, no. 2, p. 63, 2025.

[73] J. M. Nyaga, “IoT-enhanced adaptive learning environments: personalized online education for the digital age,” *African Journal of Computing and Information Systems (AJCIS)*, vol. 7, no. X, pp. 1–14, 2023.

[74] D. Abbas, A. Benmachiche, M. Derdour, and B. K. Sedraoui, “Toward Trustworthy IoT Systems: Integrating Blockchain and Artificial Intelligence,” *Available at SSRN 6460959*, 2025, Accessed: Jul. 04, 2026.

[75] J. Q. Kadhim, I. A. Aljazaery, and H. T. S. ALRikabi, “Enhancement of online education in engineering college based on mobile wireless communication networks and IoT,” *International Journal of Emerging Technologies in Learning (Online)*, vol. 18, no. 1, p. 176, 2023.

[76] A. Benmachiche, A. Makhlouf, and T. Bouhadada, “Evolutionary learning of HMM with Gaussian mixture densities for Automatic speech recognition,” in *Proceedings of the 9th International Conference on Information Systems and Technologies*, Cairo Egypt: ACM, Mar. 2019, pp. 1–6. doi: 10.1145/3361570.3361591.

[77] M. Amina, B. Tahar, B. Abdelmadjid, and M. Majda, “Apprentissage évolutionnaire d’une chaîne de Markov cachée pour la reconnaissance automatique de la parole audiovisuelle,” *المجلة الجزائرية للعلوم*, vol. 1, no. 2, pp. 65–73, 2018.

[78] A. Benmachiche and A. Makhlouf, “Optimization of hidden Markov model with Gaussian mixture densities for Arabic speech recognition,” *WSEAS Transactions on Signal Processing*, vol. 15, pp. 85–94, 2019.

[79] A. Benmachiche, A. Makhlouf, and T. Bouhadada, “Optimization learning of hidden Markov model using the bacterial foraging optimization algorithm for speech recognition,” *International Journal of Knowledge-Based and Intelligent Engineering Systems*, vol. 24, no. 3, pp. 171–181, 2020.

[80] I. Boutabia, A. Benmachiche, A. A. Betouil, C. Chemam, and M. Boutassetta, “Learning-Based Dynamic Obstacle Avoidance: A Reinforcement Learning Framework for Autonomous Mobile Robots,” *Available at SSRN 6424078*, 2026, Accessed: Jul. 08, 2026.

[81] B. Abdelmdjid, B. Tahar, M. Amina, Z. Meriem, and M. Majda, “Cooperative Approach navigation for autonomous Mobile Robots (Ap-plication on a building Site),” *المجلة الجزائرية للعلوم*, vol. 1, no. 2, pp. 41–45, 2018.

[82] A. Mellouk and A. Benmachiche, “A survey on Navigation Systems in Dynamic Environments,” in *Proceedings of the 10th International Conference on Information Systems and Technologies*, Lecce Italy: ACM, Jun. 2020, pp. 1–7. doi: 10.1145/3447568.3448527.

[83] A. Benmachiche, B. Hadjar, I. Boutabia, A. A. Betouil, M. Maatallah, and A. Makhlouf, “Development of a biometric authentication platform using voice recognition,” in *2022 4th International Conference on Pattern Analysis and Intelligent Systems (PAIS)*, IEEE, 2022, pp. 1–7. Accessed: Mar. 29, 2024.

[84] I. Boutabia, A. Benmachiche, A. A. Betouil, C. Chemam, and M. Boutassetta, “Learning-Based Dynamic Obstacle Avoidance: A Reinforcement Learning Framework for Autonomous Mobile Robots,” *Available at SSRN 6424078*, 2026, Accessed: Jul. 08, 2026.

[85] H. Eljak *et al.*, “E-learning-based cloud computing environment: A systematic review, challenges, and opportunities,” *IEEE Access*, vol. 12, pp. 7329–7355, 2023.

[86] A. Benmachiche, K. Rais, and H. Slimi, “Adaptive Real-Time Scheduling Algorithms for Embedded Systems,” Jan. 06, 2026, *arXiv*: arXiv:2512.21364. doi: 10.48550/arXiv.2512.21364.

[87] I. Boutabia, A. Benmachiche, A. A. Betouil, C. Chemam, and M. Boutassetta, “Analyzing Accuracy-Latency Trade-offs in Deep Learning Models for Embedded Vision Systems,” *Available at SSRN 6435019*, 2026, Accessed: Jul. 08, 2026.

[88] S. Rogers, *Online Pedagogy: The Pedagogical Variation Model for Asynchronous Online Teaching*. University of South Wales (United Kingdom), 2013. Accessed: Aug. 09, 2025.

[89] K. Jirasatjanukul, N. Pakprod, P. Dokkulab, A. Changkwanyeun, T. Jantakoon, and R. Laoha, “Creation of Educational Innovations through Cloud-Based Constructivism and Connectivism Learning for Undergraduates.,” *Higher Education Studies*, vol. 13, no. 4, pp. 119–127, 2023.

[90] K. Dikilitas and I. N. Fructuoso, “Conceptual framework for flexible learning design: The Context of flipped classroom,” *Conceptual framework for flexible learning design: The Context of flipped classroom. https://doi. org/10.31265/usps*, vol. 267, 2023, Accessed: Aug. 09, 2025.

[91] I. K. Suartama, A. H. Simamora, K. Susiani, K. Suranata, M. Yunus, and G. D. Tisna MS, “Designing Gamification for Case and Project-Based Online Learning: A Study in Higher Education.,” *Journal of Education and E-Learning Research*, vol. 10, no. 2, pp. 86–98, 2023.

[92] M. Umar and I. Ko, “E-learning: Direct effect of student learning effectiveness and engagement through project-based learning, team cohesion, and flipped learning during the COVID-19 pandemic,” *Sustainability*, vol. 14, no. 3, p. 1724, 2022.

[93] L. Weis, “The Future of E-Learning: Designing Sustainable Education _,” *Sustainable development in a modern knowledge society*, vol. 5, 2021, Accessed: Aug. 09, 2025.

[94] M. Majda, S. Hassina, B. Abdelmadjid, and M. Amina, “Un système de recommandation hybride basé flou adapté à l’apprentissage amélioré par la technologie,” *المجلة الجزائرية للعلوم*, vol. 1, no. 2, pp. 57–64, 2018.

[95] S. Gasmi, T. Bouhadada, and A. Benmachiche, “Survey on Recommendation Systems,” in *Proceedings of the 10th International Conference on Information Systems and Technologies*, Lecce Italy: ACM, Jun. 2020, pp. 1–7. doi: 10.1145/3447568.3448518.

[96] N. Guragain, “E-learning benefits and applications,” 2016, Accessed: Aug. 09, 2025.

[97] M. Hillier, “Bridging the digital divide with off-line e-learning,” in *Expanding Horizons in Open and Distance Learning*, Routledge, 2020, pp. 110–121. Accessed: Aug. 09, 2025.

[98] S. Kim and C. Choi, “Institutional and managerial dimensions of digital government development in the Republic of Korea,” *Bringing Government into the 21st Century*, vol. 13, 2016, Accessed: Aug. 09, 2025.

[99] A. Nawaz and M. Z. Khan, “Issues of technical support for e-learning systems in higher education institutions,” *International Journal of Modern Education and Computer Science*, vol. 4, no. 2, p. 38, 2012.

[100] S. Alam and A. Hameed, “Teaching concerns in higher education: Impact of Covid-19 in pedagogy,” *Journal of Education Culture and Society*, vol. 14, no. 1, pp. 318–332, 2023.

[101] S. Y. Almossa and S. M. Alzahrani, “Lessons on maintaining assessment integrity during COVID-19,” *Int J Educ Integr*, vol. 18, no. 1, p. 19, Dec. 2022, doi: 10.1007/s40979-022-00112-1.

[102] M. Hubalovska, “Implementation of E-learning at Primary School Education”.

[103] S. Ahmad, A. S. Mohd Noor, A. A. Alwan, Y. Gulzar, W. Z. Khan, and F. A. Reegu, “eLearning acceptance and adoption challenges in higher education,” *Sustainability*, vol. 15, no. 7, p. 6190, 2023.

[104] K. Shahzad, S. A. Khan, Y. Javed, and A. Iqbal, “E-learning for continuing professional development of university librarians: a systematic review,” *Sustainability*, vol. 15, no. 1, p. 849, 2023.

[105] B. S. Oumaima, B. Abdemadjid, M. Majda, R. Mounira, and M. Derdour, “Examining Intelligent Tutoring Systems and Their AI Underpinnings in the Design of the Future of Learning,” in *2025 International Conference on Networking and Advanced Systems (ICNAS)*, Oct. 2025, pp. 1–8. doi: 10.1109/ICNAS68168.2025.11298109.

[106] A. A. Marawar and A. D. B. Chaudhari, *The future of learning: Innovations In education*. Academic Guru Publishing House, 2024. Accessed: Aug. 10, 2025.

[107] I. Soualmia, S. Maalem, A. Benmachiche, K. Rais, and M. Derdour, “Comparative Survey of AI-Driven Credit Card Fraud Detection: Machine Learning, Deep Learning and Hybrid Systems,” in *2025 International Conference on Networking and Advanced Systems (ICNAS)*, Oct. 2025, pp. 1–9. doi: 10.1109/ICNAS68168.2025.11298125.

[108] D. Kumar, “How emerging technologies are transforming education and research: trends, opportunities, and challenges,” *Infinite Horizons: Exploring the Unknown*, pp. 89–117, 2023.

[109] E. Dritsas and M. Trigka, “Methodological and technological advancements in E-learning,” *Information*, vol. 16, no. 1, p. 56, 2025.

[110] M. Celestin, M. Vasuki, A. D. Kumar, and S. Sujatha, “The Future of Education,” *DK International Research Foundation*, 2024, Accessed: Aug. 10, 2025.

[111] N. Yadav, “The impact of digital learning on education,” *International Journal of Multidisciplinary Research in Arts, Science and Technology*, vol. 2, no. 1, pp. 24–34, 2024.

[112] R. Mounira, M. Majda, B. Abdelmadjid, B. S. Oumaima, and M. Derdour, “A Comprehensive Survey on Blockchain, Federated Learning, and AI for Securing the Internet of Medical Things,” in *2025 International Conference on Networking and Advanced Systems (ICNAS)*, Oct. 2025, pp. 1–8. doi: 10.1109/ICNAS68168.2025.11298047.

[113] D. Abbas, A. Benmachiche, M. Derdour, and B. K. Sedraoui, “Privacy and Security in Decentralized Cyber-Physical Systems: A Survey,” in *2025 International Conference on Networking and Advanced Systems (ICNAS)*, Oct. 2025, pp. 1–10. doi: 10.1109/ICNAS68168.2025.11297996.

[114] Z. Sadiqzade and H. Alisoy, “Cybersecurity and Online Education–Risks and Solutions,” *Luminis Applied Science and Engineering*, vol. 2, no. 1, pp. 4–12, 2025.

[115] S. A.-L. Akacha and A. I. Awad, “Enhancing security and sustainability of e-learning software systems: A comprehensive vulnerability analysis and recommendations for stakeholders,” *Sustainability*, vol. 15, no. 19, p. 14132, 2023.

[116] F. Shersad and S. Salam, “Managing risks of E-learning during COVID-19,” *International Journal of Innovation and Research in Educational Sciences*, vol. 7, no. 4, pp. 2349–5219, 2020.

[117] H. J. Hejase, H. F. Fayyad-Kazan, A. J. Hejase, and I. A. Moukadem, “Cyber security amid COVID-19,” *Computer and Information Science*, vol. 14, no. 2, pp. 1–10, 2021.

[118] B. K. Sedraoui, A. Benmachiche, A. Makhlouf, and C. Chemam, “Exploring the Integration of Differential Privacy in Cybersecurity Analytics: Balancing Data Utility and Privacy in Threat Intelligence,” Jan. 01, 2026, *arXiv*: arXiv:2601.00385. doi: 10.48550/arXiv.2601.00385.

[119] M. Alier, M. J. Casañ Guerrero, D. Amo, C. Severance, and D. Fonseca, “Privacy and E-Learning: A Pending Task,” *Sustainability*, vol. 13, no. 16, p. 9206, Aug. 2021, doi: 10.3390/su13169206.

[120] B. K. Sedraoui, A. Benmachiche, A. Makhlouf, and C. Chemam, “Applications of Secure Multi-Party Computation in Financial Services,” Jan. 01, 2026, *arXiv*: arXiv:2601.00334. doi: 10.48550/arXiv.2601.00334.

[121] H. J. Sienkiewicz, *The art of cyber conflict*. Dog Ear Publishing, 2017. Accessed: Aug. 10, 2025.

[122] V. Bhavsar, A. Kadlak, and S. Sharma, “Study on phishing attacks,” *International Journal of Computer Applications*, vol. 182, no. 33, pp. 27–29, 2018.

[123] M. Baykara and Z. Z. Gürel, “Detection of phishing attacks,” in *2018 6th International Symposium on Digital Forensic and Security (ISDFS)*, IEEE, 2018, pp. 1–5. Accessed: Aug. 10, 2025.

[124] M. Redjimi, A. Makhlouf, M. Maatallah, A. Benmachiche, and S. O. Boufaida, “Enhancing Malware Detection Robustness through Hybrid Adversarial Machine Learning,” *Available at SSRN 6722779*, 2025, Accessed: Jul. 08, 2026.

[125] J. Chandy, “Review on malware, types, and its analysis,” *IJRASET*, vol. 10, pp. 386–390, 2022.

[126] A. Bonguet and M. Bellaiche, “A survey of denial-of-service and distributed denial of service attacks and defenses in cloud computing,” *Future Internet*, vol. 9, no. 3, p. 43, 2017.

[127] B. Bhushan, G. Sahoo, and A. K. Rai, “Man-in-the-middle attack in wireless and computer networking—A review,” in *2017 3rd International Conference on Advances in Computing, Communication & Automation (ICACCA)(Fall)*, IEEE, 2017, pp. 1–6. Accessed: Aug. 10, 2025.

[128] M. A. Lawal, A. B. M. Sultan, and A. O. Shakiru, “Systematic literature review on SQL injection attack,” *International Journal of Soft Computing*, vol. 11, no. 1, pp. 26–35, 2016.

[129] R. Ciancioso, D. Budhwa, and T. Hayajneh, “A framework for zero day exploit detection and containment,” in *2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech)*, IEEE, 2017, pp. 663–668. Accessed: Aug. 10, 2025.

[130] S. Gupta and B. B. Gupta, “Cross-Site Scripting (XSS) attacks and defense mechanisms: classification and state-of-the-art,” *Int J Syst Assur Eng Manag*, vol. 8, no. S1, pp. 512–530, Jan. 2017, doi: 10.1007/s13198-015-0376-0.

[131] A. Alshamrani, S. Myneni, A. Chowdhary, and D. Huang, “A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1851–1877, 2019.

[132] S. Subangan and V. Senthooran, “Secure authentication mechanism for resistance to password attacks,” in *2019 19th International Conference on Advances in ICT for Emerging Regions (ICTer)*, IEEE, 2019, pp. 1–7. Accessed: Aug. 10, 2025.

[133] C. Beaman, A. Barkworth, T. D. Akande, S. Hakak, and M. K. Khan, “Ransomware: Recent advances, analysis, challenges and future research directions,” *Computers & security*, vol. 111, p. 102490, 2021.

[134] L. M. Brumă, “Security vulnerabilities in cloud-based e-learning,” in *Conference proceedings of» eLearning and Software for Education «(eLSE)*, Carol I National Defence University Publishing House, 2020, pp. 190–197. Accessed: Aug. 10, 2025.

[135] M. Bhatia and J. K. Maitra, “E-learning platforms security issues and vulnerability analysis,” in *2018 International Conference on Computational and Characterization Techniques in Engineering & Sciences (CCTES)*, IEEE, 2018, pp. 276–285. Accessed: Aug. 10, 2025.

[136] A. Suokas, “Privileged Accounts Protection with Multi-factor Authentication,” 2023, Accessed: Aug. 10, 2025.

[137] D. D. Kumar, J. D. Mukharzee, C. V. D. Reddy, and S. M. Rajagopal, “Safe and secure communication using SSL/TLS,” in *2024 International Conference on Emerging Smart Computing and Informatics (ESCI)*, IEEE, 2024, pp. 1–6. Accessed: Aug. 10, 2025.

[138] R. O. Jagannath and A. K. Jain, “Browser-in-the-middle attacks: A comprehensive analysis and countermeasures,” *Security and Privacy*, vol. 7, no. 5, p. e410, Sep. 2024, doi: 10.1002/spy2.410.

[139] S. Kirange, D. Sawai, and I. M. Director, “A comparative study of e-learning platforms and associated online activities,” *The Online Journal of Distance Education and e-Learning*, vol. 9, no. 2, pp. 194–199, 2021.

[140] C. Alario-Hoyos *et al.*, “A study of learning-by-doing in MOOCs through the integration of third-party external tools: comparison of synchronous and asynchronous running modes,” 2018, Accessed: Aug. 11, 2025.

[141] M. C. Sánchez, J. M. C. de Gea, J. L. Fernández-Alemán, J. Garcerán, and A. Toval, “Software vulnerabilities overview: A descriptive study,” *Tsinghua Science and Technology*, vol. 25, no. 2, pp. 270–280, 2019.

[142] P. Kamal, “State of the art survey on session hijacking,” *Global Journal of Computer Science and Technology*, vol. 16, no. 1, pp. 39–49, 2016.

[143] V. Jain, D. R. Sahu, and D. S. Tomar, “Session hijacking: threat analysis and countermeasures,” in *Int. Conf. on Futuristic Trends in Computational Analysis and Knowledge Management*, 2015. Accessed: Aug. 11, 2025.

[144] N. Soms, M. S. Oswalt, and K. P. Santhosh, “A case study on cloud security controls,” *International Journal of Health Sciences*, vol. 6, no. s1, pp. 11374–11380, 2022.

[145] R. Umar, I. Riadi, and M. I. A. Elfatiha, “Security Analysis of Web-Based Academic Information System using OWASP Framework,” *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, 2024, Accessed: Aug. 11, 2025.

[146] Ö. Aslan, S. S. Aktuğ, M. Ozkan-Okay, A. A. Yilmaz, and E. Akin, “A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions,” *Electronics*, vol. 12, no. 6, p. 1333, 2023.

[147] G. B. Liluashvili, “Cyber risk mitigation in higher education,” *Law & World*, vol. 17, p. 15, 2021.

[148] M. A. Haque *et al.*, “Cybersecurity in Universities: An Evaluation Model,” *SN COMPUT. SCI.*, vol. 4, no. 5, p. 569, Jul. 2023, doi: 10.1007/s42979-023-01984-x.

[149] S. Onofri and D. Onofri, *Attacking and Exploiting Modern Web Applications: Discover the mindset, techniques, and tools to perform modern web attacks and exploitation*. Packt Publishing Ltd, 2023. Accessed: Aug. 11, 2025.

[150] M. Barbu, A. Zamfiroiu, I. A. Marinescu, and D. Iordache, “Cybersecurity in e-learning during the COVID-19 Pandemic,” in *5th International Open and Distance Learning Conference Proceedings Book*, 2022, p. 165. Accessed: Aug. 11, 2025.

[151] N. A. Karim and A. H. Ali, “E-learning virtual meeting applications: A comparative study from a cybersecurity perspective,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 24, no. 2, pp. 1121–1129, 2021.

[152] B. Kaur, K. Sood, and S. Grima, “A systematic review on forensic accounting and its contribution towards fraud detection and prevention,” *Journal of Financial Regulation and Compliance*, vol. 31, no. 1, pp. 60–95, 2023.

[153] B. K. Sedraoui, A. Benmachiche, A. Makhlouf, and C. Chemam, “Intrusion Detection with deep learning: A literature review,” in *2024 6th International Conference on Pattern Analysis and Intelligent Systems (PAIS)*, IEEE, 2024, pp. 1–8. Accessed: Aug. 18, 2024.

[154] A. Benmachiche, K. Rais, and H. Slimi, “Real-Time Machine Learning for Embedded Anomaly Detection,” Dec. 22, 2025, *arXiv*: arXiv:2512.19383. doi: 10.48550/arXiv.2512.19383.

[155] M. Boutassetta, A. Makhlouf, N. Messaoudi, A. Benmachiche, and I. Boutabia, “Hybrid IDS Using Signature-Based and Anomaly-Based Detection,” Jan. 17, 2026, *arXiv*: arXiv:2601.11998. doi: 10.48550/arXiv.2601.11998.

[156] O. H. Abdulganiyu, T. Ait Tchakoucht, and Y. K. Saheed, “A systematic literature review for network intrusion detection system (IDS),” *Int. J. Inf. Secur.*, vol. 22, no. 5, pp. 1125–1162, Oct. 2023, doi: 10.1007/s10207-023-00682-2.

[157] S. Abbas, W. Naser, and A. Kadhim, “Subject review: Intrusion detection system (IDS) and intrusion prevention system (IPS),” *Global Journal of Engineering and Technology Advances*, vol. 2, no. 14, pp. 155–158, 2023.

[158] J. M. Kizza, “Firewalls,” in *Guide to Computer Network Security*, in Texts in Computer Science. , Cham: Springer International Publishing, 2024, pp. 265–294. doi: 10.1007/978-3-031-47549-8_12.

[159] A. Kamruzzaman, S. Ismat, J. C. Brickley, A. Liu, and K. Thakur, “A comprehensive review of endpoint security: Threats and defenses,” in *2022 International Conference on Cyber Warfare and Security (ICCWS)*, IEEE, 2022, pp. 1–7. Accessed: Aug. 11, 2025.

[160] S. Berrios, D. Leiva, B. Olivares, H. Allende-Cid, and P. Hermosilla, “Systematic Review: Malware Detection and Classification in Cybersecurity,” *Applied Sciences*, vol. 15, no. 14, p. 7747, 2025.

[161] M. Ahsan, K. E. Nygard, R. Gomes, M. M. Chowdhury, N. Rifat, and J. F. Connolly, “Cybersecurity threats and their mitigation approaches using Machine Learning—A Review,” *Journal of Cybersecurity and Privacy*, vol. 2, no. 3, pp. 527–555, 2022.

[162] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, “Machine learning and deep learning approaches for cybersecurity: A review,” *IEEE Access*, vol. 10, pp. 19572–19585, 2022.

[163] M. Pawlicki, R. Kozik, and M. Choraś, “A survey on neural networks for (cyber-) security and (cyber-) security of neural networks,” *Neurocomputing*, vol. 500, pp. 1075–1087, 2022.

[164] F. Rehman, F. Mushtaq, and H. Zaman, “A Host-based Intrusion Detection: Using Signature-based and AI-driven Anomaly Detection for Enhanced Cybersecurity,” in *2024 4th International Conference on Digital Futures and Transformative Technologies (ICoDT2)*, IEEE, 2024, pp. 1–7. Accessed: Aug. 11, 2025.

[165] T. Arjunan, “Detecting anomalies and intrusions in unstructured cybersecurity data using natural language processing,” *International Journal for Research in Applied Science and Engineering Technology*, vol. 12, no. 9, pp. 10–22214, 2024.

[166] A. Bouamrane, M. Derdour, M. S. Kahil, A. Benmachiche, M. Deriche, and M. Nasor, “A Novel Sequential Mobility-Based Federated Learning Framework with Agentic Explainable AI for Histopathology-Based Lung Diagnosis,” in *2026 IEEE 23rd International Multi-Conference on Systems, Signals & Devices (SSD)*, IEEE, 2026, pp. 855–864. Accessed: Jul. 08, 2026. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/11558924/

[167] K. Kumar and A. Al-Besher, “IoT enabled e-learning system for higher education,” *Measurement: Sensors*, vol. 24, p. 100480, 2022.

[168] S. Haque *et al.*, “Assessing the Impact of IoT Enabled E-Learning System for Higher Education,” *SN COMPUT. SCI.*, vol. 4, no. 5, p. 459, Jun. 2023, doi: 10.1007/s42979-023-01860-8.

[169] B. K. Sedraoui, A. Benmachiche, A. Makhlouf, D. Abbas, and M. Derdour, “Cybersecurity in E-Learning: A Literature Review on Phishing Detection Using ML and DL Techniques,” in *2025 International Conference on Networking and Advanced Systems (ICNAS)*, Oct. 2025, pp. 1–10. doi: 10.1109/ICNAS68168.2025.11298114.

[170] D. Jing and H.-B. Chen, “SVM based network intrusion detection for the UNSW-NB15 dataset,” in *2019 IEEE 13th international conference on ASIC (ASICON)*, IEEE, 2019, pp. 1–4. Accessed: Oct. 05, 2024.

[171] Q. Tian, D. Han, K.-C. Li, X. Liu, L. Duan, and A. Castiglione, “An intrusion detection approach based on improved deep belief network,” *Applied Intelligence*, vol. 50, Oct. 2020, doi: 10.1007/s10489-020-01694-4.

[172] M. Almiani, A. AbuGhazleh, A. Al-Rahayfeh, S. Atiewi, and A. Razaque, “Deep recurrent neural network for IoT intrusion detection system,” *Simulation Modelling Practice and Theory*, vol. 101, p. 102031, May 2020, doi: 10.1016/j.simpat.2019.102031.

[173] L. Ashiku and C. Dagli, “Network intrusion detection system using deep learning,” *Procedia Computer Science*, vol. 185, pp. 239–247, 2021.

[174] S. Al-Emadi, A. Al-Mohannadi, and F. Al-Senaid, “Using Deep Learning Techniques for Network Intrusion Detection,” in *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*, Feb. 2020, pp. 171–176. doi: 10.1109/ICIoT48696.2020.9089524.

[175] J. O. Mebawondu, O. D. Alowolodu, J. O. Mebawondu, and A. O. Adetunmbi, “Network intrusion detection system using supervised learning paradigm,” *Scientific African*, vol. 9, p. e00497, 2020.

[176] V. Dutta, M. Choraś, R. Kozik, and M. Pawlicki, “Hybrid model for improving the classification effectiveness of network intrusion detection,” in *13th International Conference on Computational Intelligence in Security for Information Systems (CISIS 2020) 12*, Springer, 2021, pp. 405–414. Accessed: Oct. 08, 2024. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-57805-3_38

[177] R. B. Said, Z. Sabir, and I. Askerzade, “CNN-BiLSTM: A Hybrid Deep Learning Approach for Network Intrusion Detection System in Software Defined Networking with Hybrid Feature Selection.,” *IEEE Access*, 2023, Accessed: Oct. 07, 2024.

[178] M. Alsharaiah, M. Abualhaj, L. Baniata, A. Al-saaidah, Q. Kharma, and M. Al-Zyoud, “An innovative network intrusion detection system (NIDS): Hierarchical deep learning model based on Unsw-Nb15 dataset,” *International Journal of Data and Network Science*, vol. 8, no. 2, pp. 709–722, 2024.

[179] N. Q. Do, A. Selamat, O. Krejcar, T. Yokoi, and H. Fujita, “Phishing Webpage Classification via Deep Learning-Based Algorithms: An Empirical Study,” *Applied Sciences*, vol. 11, no. 19, p. 9210, Oct. 2021, doi: 10.3390/app11199210.

[180] A. Mughaid, S. AlZu’bi, A. Hnaif, S. Taamneh, A. Alnajjar, and E. A. Elsoud, “An intelligent cyber security phishing detection system using deep learning techniques,” *Cluster Comput*, vol. 25, no. 6, pp. 3819–3828, Dec. 2022, doi: 10.1007/s10586-022-03604-4.

[181] Z. Alshingiti, R. Alaqel, J. Al-Muhtadi, Q. E. U. Haq, K. Saleem, and M. H. Faheem, “A Deep Learning-Based Phishing Detection System Using CNN, LSTM, and LSTM-CNN,” *Electronics*, vol. 12, no. 1, p. 232, Jan. 2023, doi: 10.3390/electronics12010232.

[182] M. A. Adebowale, K. T. Lwin, and M. A. Hossain, “Intelligent phishing detection scheme using deep learning algorithms,” *JEIM*, vol. 36, no. 3, pp. 747–766, Apr. 2023, doi: 10.1108/JEIM-01-2020-0036.

[183] E. A. Aldakheel, M. Zakariah, G. A. Gashgari, F. A. Almarshad, and A. I. A. Alzahrani, “A Deep Learning-Based Innovative Technique for Phishing Detection in Modern Security with Uniform Resource Locators,” 2023.

[184] A. Karim, M. Shahroz, K. Mustofa, S. B. Belhaouari, and S. R. K. Joga, “Phishing detection system through hybrid machine learning based on URL,” *IEEE Access*, vol. 11, pp. 36805–36822, 2023.

[185] A. Bezerra *et al.*, “A case study on phishing detection with a machine learning net,” *Int J Data Sci Anal*, Jun. 2024, doi: 10.1007/s41060-024-00579-w.

[186] S. S. Shafin, “An explainable feature selection framework for web phishing detection with machine learning,” *Data Science and Management*, 2024, Accessed: Jul. 07, 2025.

[187] H. Fares *et al.*, “Machine learning approach for email phishing detection,” *Procedia Computer Science*, vol. 251, pp. 746–751, 2024.

[188] S. Kavya and D. Sumathi, “Staying ahead of phishers: a review of recent advances and emerging methodologies in phishing detection,” *Artif Intell Rev*, vol. 58, no. 2, p. 50, Dec. 2024, doi: 10.1007/s10462-024-11055-z.

[189] B. B. Gupta, A. Gaurav, R. W. Attar, V. Arya, A. Alhomoud, and K. T. Chui, “Optimized Phishing Detection with Recurrent Neural Network and Whale Optimizer Algorithm.,” *Computers, Materials & Continua*, vol. 80, no. 3, 2024, Accessed: Jul. 07, 2025.

[190] K. Barik, S. Misra, and R. Mohan, “Web-based phishing URL detection model using deep learning optimization techniques,” *Int J Data Sci Anal*, pp. 1–23, Feb. 2025, doi: 10.1007/s41060-025-00728-9.

[191] R. S. Rao, C. Kondaiah, A. R. Pais, and B. Lee, “A hybrid super learner ensemble for phishing detection on mobile devices,” *Sci Rep*, vol. 15, no. 1, Art. no. 1, May 2025, doi: 10.1038/s41598-025-02009-8.

[192] S. A. A. A. Alsaidi *et al.*, “HawkPhish-DNN cybersecurity model: adaptive hybrid optimization and deep learning for enhanced multi-objective phishing URL detection,” *Int. j. inf. tecnol.*, Jun. 2025, doi: 10.1007/s41870-025-02597-8.

[193] F. S. Garki, N. Francisca, and F. Binta, “PREDICTION OF PHISHING ATTACKS IN ODL USING MACHINE LEARNING TECHNIQUES”.

[194] J. Yoo and Y. Cho, “ICSA: Intelligent chatbot security assistant using Text-CNN and multi-phase real-time defense against SNS phishing attacks,” *Expert Systems with Applications*, vol. 207, p. 117893, 2022.

[195] R. Alazaidah *et al.*, “Website phishing detection using machine learning techniques,” *Journal of Statistics Applications & Probability*, vol. 13, no. 1, pp. 119–129, 2024, doi: 10.18576/jsap/130108.

[196] O. K. Sahingoz, E. BUBE, and E. Kugu, “Dephides: Deep learning based phishing detection system,” *IEEE Access*, vol. 12, pp. 8052–8070, 2024, doi: 10.1109/ACCESS.2024.3352629.

[197] F. Trad and A. Chehab, “Prompt engineering or fine-tuning? a case study on phishing detection with large language models,” *Machine Learning and Knowledge Extraction*, vol. 6, no. 1, pp. 367–384, 2024, doi: 10.3390/make6010018.

[198] S. R. Borra *et al.*, “Oecnet: Optimal feature selection-based email classification network using unsupervised learning with deep cnn model,” *e-Prime-Advances in Electrical Engineering, Electronics and Energy*, vol. 7, p. 100415, 2024, doi: 10.1016/j.prime.2023.100415.

[199] R. Jayaraj, A. Pushpalatha, K. Sangeetha, T. Kamaleshwar, S. U. Shree, and D. Damodaran, “Intrusion detection based on phishing detection with machine learning,” *Measurement: Sensors*, vol. 31, p. 101003, 2024, doi: 10.1016/j.measen.2023.101003.

[200] S. Al-Ahmadi, A. Alotaibi, and O. Alsaleh, “PDGAN: Phishing detection with generative adversarial networks,” *Ieee Access*, vol. 10, pp. 42459–42468, 2022, doi: 10.1109/ACCESS.2022.3168235.

[201] A. A. Orunsolu, A. S. Sodiya, and A. T. Akinwale, “A predictive model for phishing detection,” *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 2, pp. 232–247, 2022, doi: 10.1016/j.jksuci.2019.12.005.

[202] M. K. M. Boussougou, P. Hamandawana, and D.-J. Park, “Enhancing Voice Phishing Detection Using Multilingual Back-Translation and SMOTE: An Empirical Study,” *IEEE Access*, 2025, doi: 10.1109/ACCESS.2025.3545250.

[203] P. C. R. Chinta, C. S. Moore, L. M. Karaka, M. Sakuru, V. Bodepudi, and S. R. Maka, “Building an Intelligent Phishing Email Detection System Using Machine Learning and Feature Engineering,” *European Journal of Applied Science, Engineering and Technology*, vol. 3, no. 2, pp. 41–54, 2025, doi: 10.59324/ejaset.2025.3(2).04.

[204] G. S. Nayak, B. Muniyal, and M. C. Belavagi, “Enhancing Phishing Detection: A Machine Learning Approach with Feature Selection and Deep Learning Models,” *IEEE Access*, 2025, doi: 10.1109/ACCESS.2025.3543738.

[205] A. O. Taofeek, “Development of a Novel Approach to Phishing Detection Using Machine Learning,” *ATBU Journal of Science, Technology and Education*, vol. 12, no. 2, pp. 336–351, 2024.

[206] H. A. Hassan Hosny, A. A. Ibrahim, M. M. Elmesalawy, and A. M. Abd El-Haleem, “An intelligent approach for fair assessment of online laboratory examinations in laboratory learning systems based on Student’s mouse interaction behavior,” *Applied Sciences*, vol. 12, no. 22, p. 11416, 2022.

[207] N. Siddiqui, R. Dave, M. Vanamala, and N. Seliya, “Machine and deep learning applications to mouse dynamics for continuous user authentication,” *Machine Learning and Knowledge Extraction*, vol. 4, no. 2, pp. 502–518, 2022.

[208] M. Valızadeh, “CHEATING IN ONLINE LEARNING PROGRAMS: LEARNERS’PERCEPTIONS AND SOLUTIONS,” *Turkish Online Journal of Distance Education*, vol. 23, no. 1, pp. 195–209, 2022.

[209] T. Zhou and H. Jiao, “Data augmentation in machine learning for cheating detection in large-scale assessment: An illustration with the blending ensemble learning algorithm,” *Psychological Test and Assessment Modeling*, vol. 64, no. 4, pp. 425–444, 2022.

[210] A. M. Duhaim, S. O. Al-mamory, and M. S. Mahdi, “Cheating Detection in Online Exams during Covid-19 Pandemic Using Data Mining Techniques,” *WEB*, vol. 19, no. 1, pp. 341–366, Jan. 2022, doi: 10.14704/WEB/V19I1/WEB19026.

[211] W. Alsabhan, “Student cheating detection in higher education by implementing machine learning and LSTM techniques,” *Sensors*, vol. 23, no. 8, p. 4149, 2023.

[212] N. El Rhezzali, I. Hilal, and M. Hnida, “Cheating Detection in Online Exams,” in *Digital Technologies and Applications*, vol. 668, S. Motahhir and B. Bossoufi, Eds., in Lecture Notes in Networks and Systems, vol. 668. , Cham: Springer Nature Switzerland, 2023, pp. 431–440. doi: 10.1007/978-3-031-29857-8_44.

[213] P. M. Moreno-Marcos, J. Barredo, P. J. Muñoz-Merino, and C. Delgado Kloos, “Statoodle: A Learning Analytics Tool to Analyze Moodle Students’ Actions and Prevent Cheating,” in *Responsive and*

Sustainable Educational Futures, vol. 14200, O. Viberg, I. Jivet, P. J. Muñoz-Merino, M. Perifanou, and T. Papathoma, Eds., in Lecture Notes in Computer Science, vol. 14200. , Cham: Springer Nature Switzerland, 2023, pp. 736–741. doi: 10.1007/978-3-031-42682-7_70.

[214] T. Volarić, G. Martinović, and H. Ljubić, "Detecting academic fraud at online tests during COVID-19 using machine learning-based methods," in *International Conference on Digital Transformation in Education and Artificial Intelligence Application*, Springer, 2023, pp. 144–158. Accessed: Mar. 07, 2025.

[215] S. Z. Ong, T. Connie, and M. K. O. Goh, "Cheating Detection for Online Examination Using Clustering Based Approach," *JOIV : Int. J. Inform. Visualization*, vol. 7, no. 3–2, p. 2075, Nov. 2023, doi: 10.30630/joiv.7.3-2.2327.

[216] N. Tran, M. Nguyen, T. Le, T. Huynh, T. Nguyen, and T. Nguyen, "Exploring the potential of skeleton and machine learning in classroom cheating detection," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 32, no. 3, pp. 1533–1544, 2023.

[217] L. Zhao *et al.*, "Cheating among elementary school children: A machine learning approach," *Child Development*, vol. 94, no. 4, pp. 922–940, Jul. 2023, doi: 10.1111/cdev.13910.

[218] S. Gopane, R. Kotecha, J. Obhan, and R. K. Pandey, "Cheat Detection in Online Examinations Using Artificial Intelligence," *ASEAN Engineering Journal*, vol. 14, no. 1, pp. 121–128, 2024.

[219] G. Abeba, A. Genetu, S. Getenet, D. Bitew, and F. Alemu, "Identification of Classroom Sheet Exam Cheating Trials using a Deep Learning Approach," *International Journal of Computer Applications*, vol. 975, p. 8887.

[220] B. K. Sedraoui, A. Benmachiche, A. Bennour, A. Makhlouf, M. Derdour, and F. Ghabban, "LSTM-SWAP: A Hybrid Deep Learning Model for Cheating Detection," *SN COMPUT. SCI.*, vol. 6, no. 7, p. 798, Sep. 2025, doi: 10.1007/s42979-025-04334-1.

[221] B. K. Sedraoui, A. Benmachiche, A. Makhlouf, K. Rais, and C. Chemam, "CNN-OOA-Based Cyber Threat Detection: Protecting E-Learning from Phishing," *Arabian Journal for Science and Engineering*, 2026, doi: 10.1007/s13369-026-11122-3.

[222] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE symposium on computational intelligence for security and defense applications*, Ieee, 2009, pp. 1–6. Accessed: Aug. 17, 2024.

[223] S. A. Elsaid, E. Shehab, A. M. Mattar, A. T. Azar, and I. A. Hameed, "Hybrid intrusion detection models based on GWO optimized deep learning," *Discov Appl Sci*, vol. 6, no. 10, p. 531, Oct. 2024, doi: 10.1007/s42452-024-06209-1.

[224] E. Emirmahmutoğlu and Y. Atay, "A feature selection-driven machine learning framework for anomaly-based intrusion detection systems," *Peer-to-Peer Netw. Appl.*, vol. 18, no. 3, p. 161, May 2025, doi: 10.1007/s12083-025-01947-4.