

مقرن قيسي

## Master Thesis Megern %286%29.docx

 Midocean University

### Document Details

Submission ID

trn:oid::2945:378733766

Submission Date

Apr 21, 2026, 1:35 PM GMT+2

Download Date

Apr 28, 2026, 2:18 PM GMT+3

File Name

Master Thesis Megern (6).docx

File Size

91.7 KB

20 Pages

3,501 Words

19,846 Characters

## 93% detected as AI

The percentage indicates the combined amount of likely AI-generated text as well as likely AI-generated text that was also likely AI-paraphrased.

**Caution: Review required.**

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

### Detection Groups



**17 AI-generated only 91%**

Likely AI-generated text from a large-language model.



**2 AI-generated text that was AI-paraphrased 2%**

Likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

#### Disclaimer

Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (i.e., our AI models may produce either false positive results or false negative results), so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

### Frequently Asked Questions

#### How should I interpret Turnitin's AI writing percentage and false positives?

The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (\*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

#### What does 'qualifying text' mean?

Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.





# Deep Learning-Based Network Intrusion Detection Using a Hybrid CNN-LSTM Framework

Prepared By

Name: Megern Qaisse

ID No.: 253001866

Submitted in fulfillment of the requirements for the degree Master of  
Science in Artificial Intelligence

At

Collage of Informatics

Midocean University

UAE

Supervisor Name: **Dr. Amjad Qtaish**

Collage of Informatics, Midocean University, Comoros

Email: .....

# Acknowledgements

Special thanks to **Dr. Amjad Qtaish**

# **Abstract**

# **List of Publications**

# Table of Contents

|                                    |             |
|------------------------------------|-------------|
| <b>List of Abbreviations .....</b> | <b>vii</b>  |
| <b>List of Figures.....</b>        | <b>viii</b> |
| <b>List of Tables .....</b>        | <b>ix</b>   |

## **Chapter 1: Introduction and**

|                               |  |
|-------------------------------|--|
| Overview.....                 |  |
| 1.2 Problem Definition .....  |  |
| 1.3 Thesis Objectives .....   |  |
| 1.4 Thesis Contribution ..... |  |
| 1.5 Thesis Organization.....  |  |

## **Chapter 2: Background and Related Work**

|                                                   |  |
|---------------------------------------------------|--|
| 2.1 Introduction .....                            |  |
| 2.2 Intrusion Detection Systems (IDS) .....       |  |
| 2.3 Types of Intrusion Detection Approaches ..... |  |
| 2.4 Machine Learning in Intrusion Detection ..... |  |
| 2.5 Deep Learning for Intrusion Detection .....   |  |
| 2.6 Related Work .....                            |  |
| 2.7 Research Gap .....                            |  |

## **Chapter 3: Proposed Framework and Methodology**

|                        |  |
|------------------------|--|
| 4.1 Introduction ..... |  |
|------------------------|--|

## **Chapter 4: Experimental Results and Discussion .....**

## **Chapter 5: Conclusion and Future Work .....**

|                              |  |
|------------------------------|--|
| 7.1 Important Features ..... |  |
| 7.3 Future Work .....        |  |

References .....

## List of Abbreviations

Acc.: Accuracy  
ANN: Artificial Neural Network  
CNN: Convolutional Neural Network  
LSTM: Long Short-Term Memory  
DL: Deep Learning  
ML: Machine Learning  
IDS: Intrusion Detection System  
DoS: Denial of Service  
ROC: Receiver Operating Characteristic  
AUC: Area Under the Curve  
TP: True Positive  
FP: False Positive  
TN: True Negative  
FN: False Negative

## List of Figures

- 2.1 General Architecture of Intrusion Detection Systems
- 2.2 Types of Network Attacks in IDS
- 2.3 Overview of Deep Learning in Cybersecurity
- 3.1 Proposed Hybrid CNN-LSTM Framework
- 3.2 Data Preprocessing Pipeline
- 4.1 Confusion Matrix of the Proposed Model
- 4.2 ROC Curve of the Proposed Model

# List of Tables

- 2.1 Summary of Related Work in Deep Learning-based IDS
- 3.1 NSL-KDD Dataset Description
- 4.1 Performance Comparison Between Proposed Model and Traditional ML Models
- 4.2 Evaluation Metrics Results (Accuracy, Precision, Recall, F1-score)

# Chapter 1 Introduction

## 1.1 Overview

The use of digital systems has increased a lot in recent years. Almost every organization depends on networks in one way or another. Because of that, security is no longer just important. It is something that needs constant attention, and honestly, it is not easy to manage in practice.

At the same time, cyberattacks are happening more frequently. Not only that, but they are also changing in how they behave. Some attacks do not follow a clear or fixed pattern, which makes early detection more difficult than expected.

If we look at real network environments, things are rarely stable. Traffic keeps changing, sometimes very quickly. Because of this, it becomes harder to clearly decide what is normal and what is not. Many monitoring systems try to handle this, but a large number of them still rely on predefined logic or simple detection methods. That is part of the problem.

These limitations create several issues. For example, such systems can usually detect known threats without much difficulty. However, when something new appears, the detection is not always reliable. Another issue is the number of alerts. In some cases, there can be many warnings in a short time, and not all of them represent real threats. This can be confusing for security teams.

Because of these challenges, researchers started to look for better approaches. Artificial intelligence became one of the main directions. From my perspective, this shift makes sense, especially when dealing with complex and changing data. Deep learning, in particular, is useful because it works directly with data instead of depending only on manually defined rules.

In this study, a model that combines CNN and LSTM is used. The idea is quite straightforward. CNN helps in capturing useful patterns, while LSTM focuses on how data evolves over time. Using both together may lead to better results compared to relying on a single model.

## 1.2 Problem Definition

Even with the progress made in cybersecurity, intrusion detection systems still have clear weaknesses. Some of these systems work well only when the attack is already known. In other words, if the pattern exists in advance, detection becomes easier. The problem starts when the attack is new or has been slightly modified.

Another problem shows up here. Some anomaly-based systems try to detect unusual behavior instead of relying on known signatures. This may seem useful at first, but in real situations it often produces too many false alarms. As a result, the system becomes less reliable, since not every alert points to an actual threat.

Machine learning did help, but not in every case. It allowed many IDS models to improve in classification and pattern recognition. Still, not every model can fully handle the nature of network data. Some models are better at capturing feature relationships, while others perform better when dealing with sequential behavior over time. In many situations, one side is handled better than the other.

This matters even more when attacks start getting more complicated. Modern network threats do not always appear in a simple or predictable way. Because of that, there is a need for a model that can learn from the data more effectively and handle both types of patterns, instead of focusing on only one aspect.

Based on this, the main question in this research is:

Can a hybrid deep learning model that combines CNN and LSTM provide more accurate and more reliable intrusion detection results compared to traditional machine learning approaches?

### 1.3 Thesis Objectives

This research is built around a set of connected goals rather than separate steps.

A key part of the work is developing a hybrid model that combines CNN and LSTM for intrusion detection. The reason behind this is to use the strengths of both models together instead of depending on only one approach.

Another important aspect involves working with the dataset. The NSL-KDD dataset will be prepared before training so that it fits the model properly and avoids common issues that may affect the results.

The study also looks at how the model behaves during evaluation. Instead of only reporting results, the aim is to understand performance using measures such as Accuracy, Precision, Recall, and F1-score, since these metrics are commonly used in similar problems.

There is also a need to compare the proposed model with traditional machine learning approaches. This comparison helps in checking whether the hybrid design actually provides better results in practice.

In addition to that, the research pays attention to false positives. In real scenarios, a system that generates too many unnecessary alerts can become difficult to rely on, so reducing them is considered an important goal in this study.

### 1.4 Thesis Contribution

- The contribution of this work can be viewed from different angles.
- One important contribution is the use of a hybrid deep learning model that combines CNN and LSTM for intrusion detection. Bringing these two models together allows the system to handle both pattern extraction and temporal behavior within the same framework.

- Another aspect of the contribution is applying this model to a well-known dataset, which is NSL-KDD. Using a benchmark dataset makes it easier to understand the results and compare them with previous studies.
- The research also adds value by including a comparison with traditional machine learning methods. This part is useful because it shows whether the hybrid model offers any real improvement instead of assuming that it does.
- A further point comes from analyzing the results in a practical way. The focus is not only on improving detection accuracy, but also on reducing false alarms. From a real-world perspective, this is important because too many false alerts can reduce the usefulness of any detection system.

## 1.5 Thesis Organization

The structure of this thesis follows the main stages of the research.

Chapter 2 gives the background needed for the study. It explains the basic ideas related to intrusion detection systems and also reviews earlier work in this area. The chapter is useful for showing what has already been done and where the research gap still exists.

Chapter 3 focuses on the proposed framework of this study. It explains how the hybrid CNN-LSTM model is designed and how the methodology is carried out. This includes dataset preparation, model development, and the main implementation steps.

Chapter 4 is dedicated to the results. It presents the performance of the proposed model and discusses the findings in detail. It also includes a comparison with traditional machine learning methods to show how the proposed approach performs.

Chapter 5 closes the thesis. It summarizes the main outcomes of the work and points to possible directions that can be explored in future research.

# Chapter 2 Background and Related Work

## 2.1 Introduction

As digital systems grow, keeping networks secure becomes harder. Most organizations now depend on connected environments for storing data, sharing information, and running daily work. This makes network protection more necessary, but also more difficult.

Cyber threats are part of this picture. Some attacks are easy to notice, but others are not. In many cases, harmful activity looks similar to normal traffic at the beginning. That is why early detection matters.

A common way to deal with this issue is to use intrusion detection systems. These systems watch network activity and try to identify behavior that does not look normal. They are often used to catch actions linked to denial-of-service, probing, or unauthorized access.

Still, older detection methods have clear limits. If the attack is already known, rule-based or signature-based systems may work well enough. But once the attack changes, or appears in a new form, those systems may not respond properly. Anomaly-based methods try to solve this, yet they often raise too many false alarms. In practice, that creates another problem instead of fully solving the first one.

Because of this, researchers started paying more attention to machine learning and deep learning. These approaches can work with large and complex data, and they are better suited for patterns that are difficult to define manually.

This chapter explains the background behind intrusion detection and then moves to earlier studies in the same area. The goal here is to show how the field developed and why there is still room for improvement.

This chapter provides the theoretical background necessary for understanding intrusion detection systems and the role of machine learning and deep learning in this domain. It also reviews related studies that have applied artificial intelligence techniques to improve IDS performance.

## 2.2 Intrusion Detection Systems (IDS)

Intrusion Detection Systems, known as IDS, are used to observe activity within a system or a network and determine whether anything unusual is taking place. Their role is mainly to notice signs of intrusion or actions that may break security rules. They do not block the attack by themselves, but they help indicate that something may be wrong.

In actual operation, an IDS can rely on more than one source of data. It may inspect traffic packets, examine recorded system events, or follow what is happening on a host machine. After that, it tries to judge whether the activity looks normal or whether it should raise concern. If unusual behavior is detected, the system sends an alert so that the issue can be checked by the security team.

IDS is often divided into two broad categories. The first is host-based intrusion detection, or HIDS. This form works on an individual device and focuses on internal activity such as modifications in files, operating system events, or locally stored records.

The second is network-based intrusion detection, usually called NIDS. Instead of focusing on one device, it pays attention to communication moving between systems across the network. Its purpose is to notice traffic patterns that may indicate harmful activity.

Both forms are useful, but each one looks at the problem from a different angle. HIDS pays closer attention to what happens inside the host itself, while NIDS is more interested in traffic behavior across the network. In practice, the choice depends on the environment and on where the likely risk is expected.

## 2.3 Types of Intrusion Detection Approaches

In general, intrusion detection is usually discussed through two main approaches. These are signature-based detection and anomaly-based detection. Each one works in a different way, and each one has its own strengths and weaknesses.

The first approach is signature-based detection. This method depends on patterns that are already known in advance. In simple terms, the system compares current network activity with stored attack patterns. If a similar pattern appears, the activity is treated as an intrusion. This method can be effective when the attack has already been seen before, but it becomes less useful when the threat is new or has been changed.

The second approach is anomaly-based detection. Here, the system tries to learn what normal behavior looks like and then checks for anything that moves away from that pattern. If the difference is large enough, the activity may be treated as suspicious. This approach is useful because it can notice attacks that were not previously defined. At the same time, it often creates another difficulty, which is the large number of false alarms.

As research in artificial intelligence developed, newer methods started to appear in this area. Machine learning and deep learning have been used to improve intrusion detection, especially in cases where normal and abnormal behavior are difficult to separate using traditional techniques alone.

## 2.4 Machine Learning in Intrusion Detection

Machine learning has been used in intrusion detection for a long time as an alternative to fixed rule-based methods. The main advantage here is that the model can learn patterns from data instead of depending only on predefined conditions.

In IDS applications, machine learning helps in separating normal traffic from suspicious activity by learning from labeled examples. This makes it useful in cases where writing exact rules for every possible attack is not practical.

Several algorithms have been used for this purpose. Common examples include Decision Trees, Support Vector Machines, Random Forest, and K-Nearest Neighbors. These methods are usually trained on datasets that contain both normal records and attack records so that the model can learn the difference between them.

Even though machine learning improved intrusion detection in many studies, it still has some limits. One problem is the amount of manual work often needed before training, especially during feature preparation. Another issue is that some traditional models do not perform well when the data becomes large, complex, or highly dimensional.

Because of these limits, researchers started looking for methods that can deal with more complicated patterns without relying so heavily on manual feature design.

## 2.5 Deep Learning for Intrusion Detection

Deep learning became important in intrusion detection because it can work with complex data in a more flexible way. Instead of depending heavily on handcrafted features, deep models can learn useful representations directly from the input data.

This is one of the reasons why deep learning attracted attention in cybersecurity research. Network traffic often contains patterns that are not easy to capture using simpler methods, especially when the attack behavior changes over time or appears in a less obvious form.

Different deep learning models have been applied in IDS research. Artificial Neural Networks were among the earlier choices. Later, models such as CNN and RNN became more common. CNN is useful when the goal is to identify patterns inside structured data, while recurrent models are more suitable when sequential behavior matters.

LSTM, which is a type of recurrent network, is especially useful when the model needs to follow how activity changes over time. This is important in intrusion detection because some attacks are better understood when the sequence of events is considered, not only individual records.

More recent work has moved toward combining deep learning models instead of using one model alone. A hybrid structure such as CNN-LSTM can be useful because one part of the model focuses on feature extraction, while the other deals with temporal behavior. This combination may improve detection performance and make the model more reliable in difficult cases.

## 2.6 Related Work

Research on intrusion detection has moved through several directions, especially with the growing use of machine learning and deep learning. Earlier work did not focus only on building models, but also on improving the datasets used for evaluation. A well-known example is the NSL-KDD dataset introduced by Tavallae et al. (2009), which was presented

as an improved alternative to KDD'99 after addressing problems such as redundancy and class imbalance. Later, Moustafa and Slay (2015) introduced UNSW-NB15, which provided more modern attack scenarios and became another important dataset in IDS research.

As deep learning started receiving more attention, researchers explored different model types for intrusion detection. Some studies focused on general deep neural networks. For example, Kim et al. (2016) reported that deep neural networks could improve classification results when compared with more traditional machine learning methods, although their approach still depended heavily on feature engineering. In a similar direction, Alrawashdeh and Purdy (2016) used deep belief networks and showed that deep architectures could be useful in recognizing more complex attack patterns.

Another group of studies focused on sequential learning. Yin et al. (2017) used LSTM to capture temporal patterns in network traffic and showed that this model performed better than a number of conventional machine learning methods, especially when the attack behavior had a sequential nature. Li et al. (2020) later followed a related direction by using bidirectional LSTM, aiming to improve the detection of more complex threats.

At the same time, some researchers concentrated more on feature extraction. Zhang et al. (2019) applied CNN to intrusion detection and showed that convolution-based models were effective in learning useful spatial patterns from the data. Vinayakumar et al. (2019) also evaluated deep learning approaches such as CNN and DNN on larger intrusion detection datasets and reported that these models generally performed better than many traditional classifiers when the data became large and complex.

There were also attempts to combine deep learning with other methods. Shone et al. (2018), for instance, used a deep autoencoder together with Random Forest and achieved strong results while reducing the dimensionality of the data. Verma and Ranga (2020) explored ensemble learning combined with deep learning and reported improvements in classification performance. In IoT-related environments, Sohi et al. (2021) found that deep models could also provide useful results when detecting more advanced threats.

A more recent direction in the literature is the use of hybrid models. Khan et al. (2020) combined CNN and LSTM in one framework and showed that joining the two models can improve detection by capturing both feature-level and temporal information. This same idea appears in the broader discussion presented by Ferrag et al. (2020), who reviewed deep learning-based IDS studies and highlighted hybrid architectures as one of the promising directions in the field. Bedi et al. (2021) also supported the move toward more advanced deep learning frameworks for anomaly detection, especially when dealing with unknown attacks.

Looking across these studies, a general pattern can be seen. Single-model approaches such as DNN, CNN, or LSTM can perform well, but each one tends to focus on one side of the problem more than the other. This is why hybrid approaches, especially those that combine CNN and LSTM, have gained interest in recent work. They offer a way to learn both structural patterns and temporal behavior within the same model, which is directly relevant to the direction of the present study.

## 2.7 Research Gap

Even though intrusion detection research has developed a lot, there are still clear gaps that have not been fully addressed.

One of the main issues is that many traditional machine learning models still depend too much on manual feature preparation. This creates limitations, especially when the network data becomes more complex or when the attack pattern is not easy to describe in advance.

Deep learning improved this situation in many studies, but the problem is not completely solved. Some models focus more on extracting feature patterns, while others are better at handling changes over time. When only one side is emphasized, the model may miss part of the full behavior found in network traffic.

Another difficulty appears in false alarms. A number of existing methods may achieve good detection results, yet they still produce alerts that do not correspond to real attacks. In practical environments, this reduces the usefulness of the detection system and makes analysis more difficult for security teams.

Because of these points, there is still room for a model that can deal with both feature relationships and temporal behavior within the same framework. This is where the current research is positioned.

For that reason, this study proposes a hybrid CNN-LSTM model. The goal is to combine spatial and temporal learning in a way that may improve intrusion detection performance and reduce some of the limitations seen in earlier approaches.

## **Chapter 4 Proposed Approach (Methodology)**

## **Chapter 5 :Experimental Results and Discussion**

## Chapter 6 Conclusion and Future Work

## References

Tavallae et al., 2009 – NSL-KDD Dataset

Yin et al., 2017 – LSTM-based IDS

Shone et al., 2018 – Deep Autoencoder IDS

Vinayakumar et al., 2019 – Deep Learning IDS

Khan et al., 2020 – CNN-LSTM IDS

Ferrag et al., 2020 – Survey on DL IDS

Moustafa & Slay, 2015 – UNSW-NB15 Dataset